Deep Learning Theory and Applications

CNNs

Yale

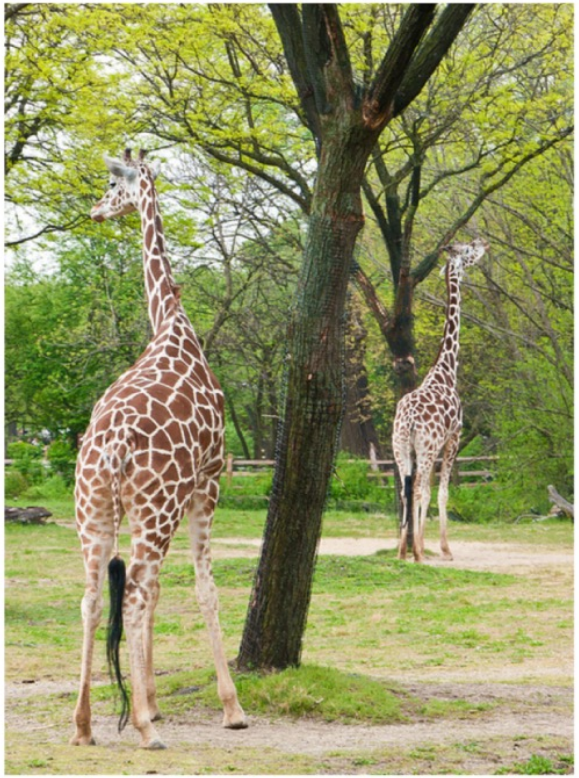
CPSC/AMTH 452/552

CBB 663

Taking An Image as an Input

a soccer player is kicking a soccer ball

a street sign on a pole in front of a building

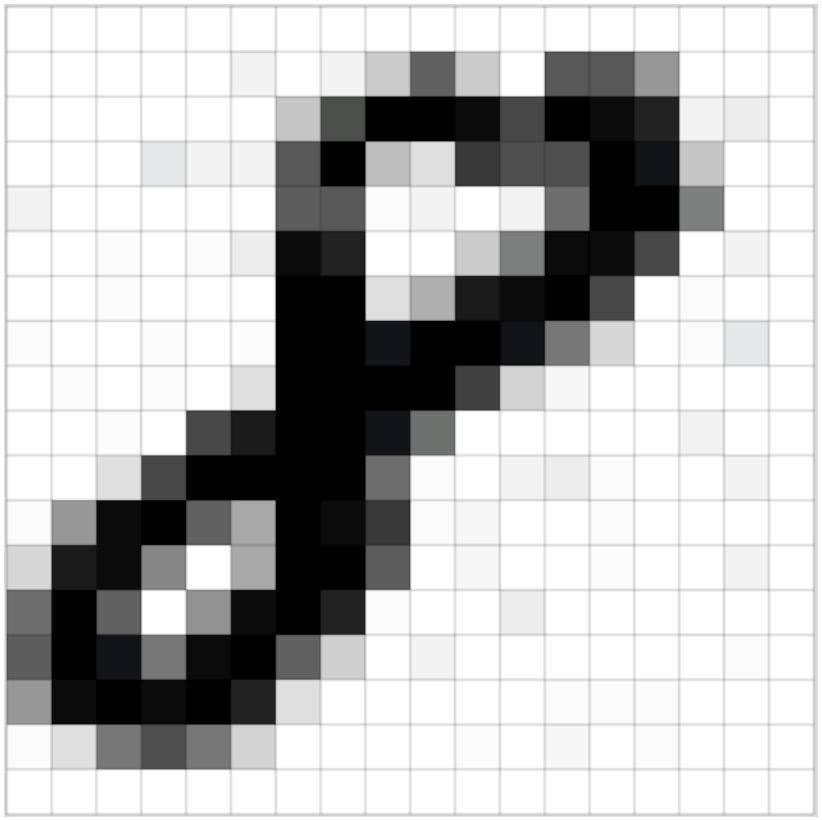


a couple of giraffe standing next to each other

Nnets have been very good at classifying real world images

First architecture was LeNet formulated by Yann Lecun in 1988

Images are a series of Pixel Values



Grayscale images:
0=Black
255 = White

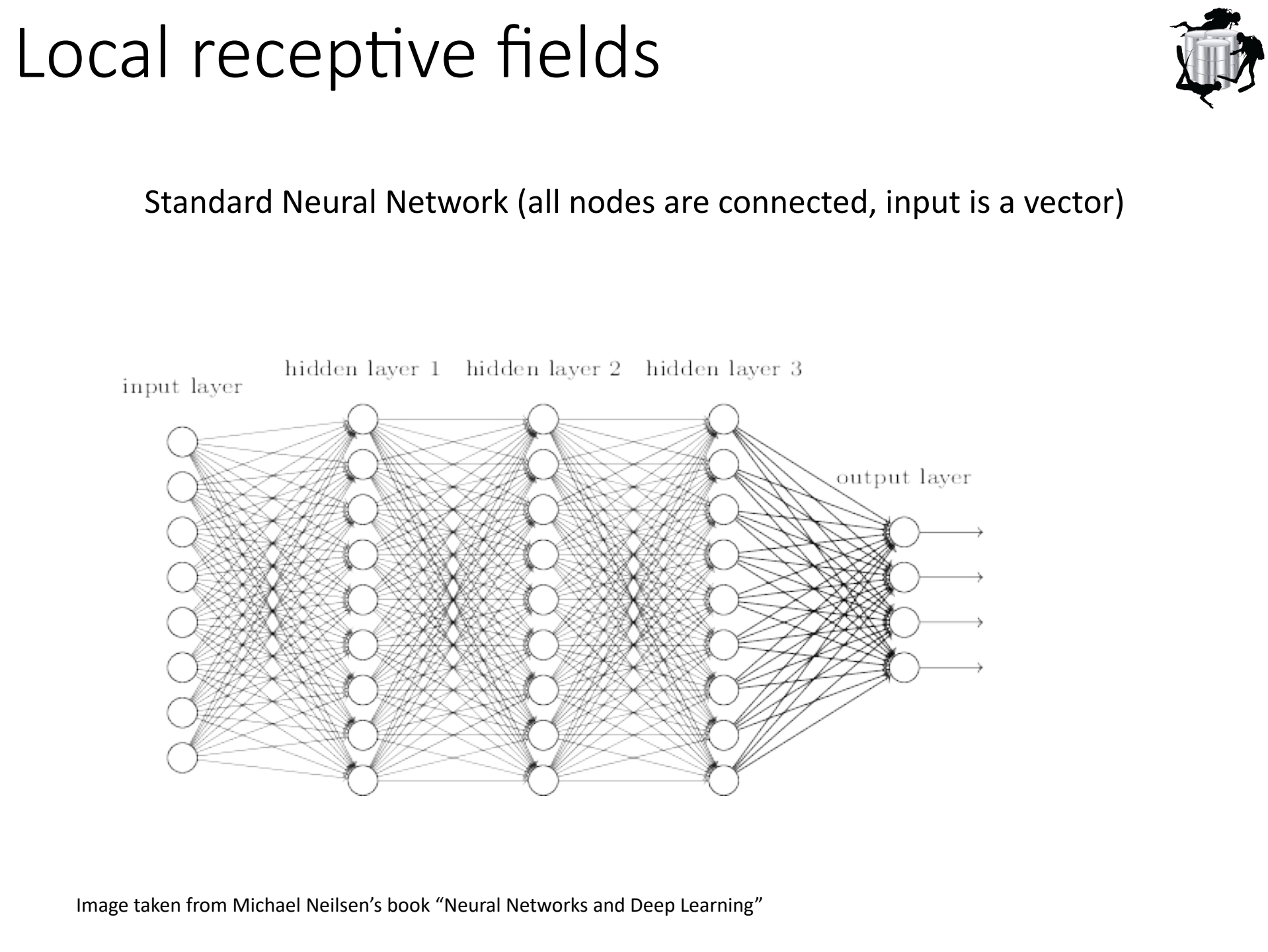
each pixel is a feature

Spatial locality structure

CNNs use three basic ideas

1. Local receptive fields — a face is a localized pixels in an image
2. Shared weights — use same weight matrix to scan an image
3. Pooling — may not always want, will lose abstract info

Local receptive fields



Standard Neural Network (all nodes are connected, input is a vector)

Image taken from Michael Neilsen's book "Neural Networks and Deep Learning"

Local receptive fields

Make connections in small, localized regions of the input image

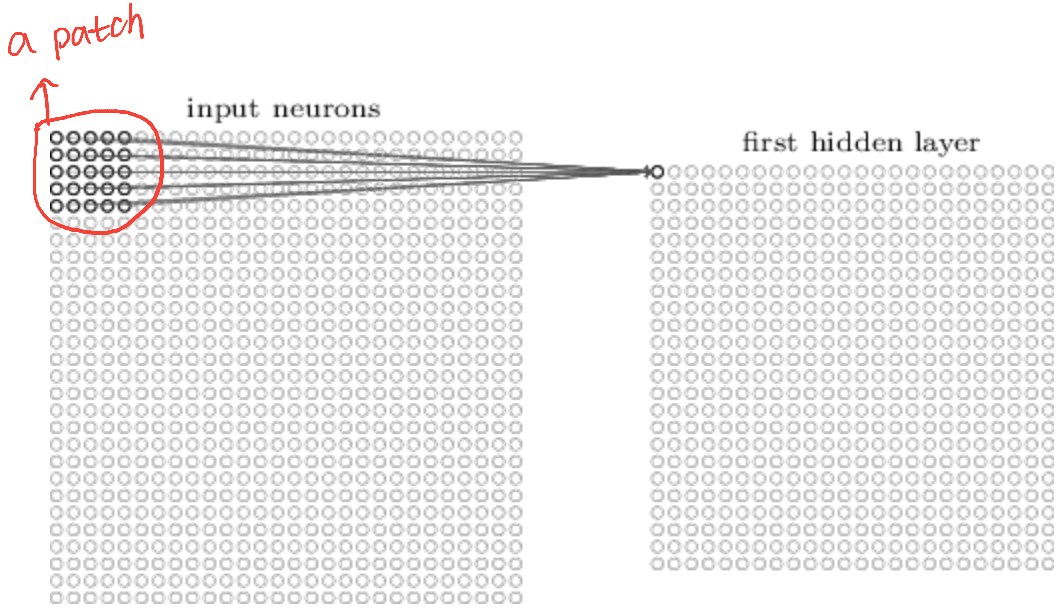


Image taken from Michael Neilsen's book "Neural Networks and Deep Learning"

Local receptive fields

Slide the local receptive field over by one (or more) pixel and repeat

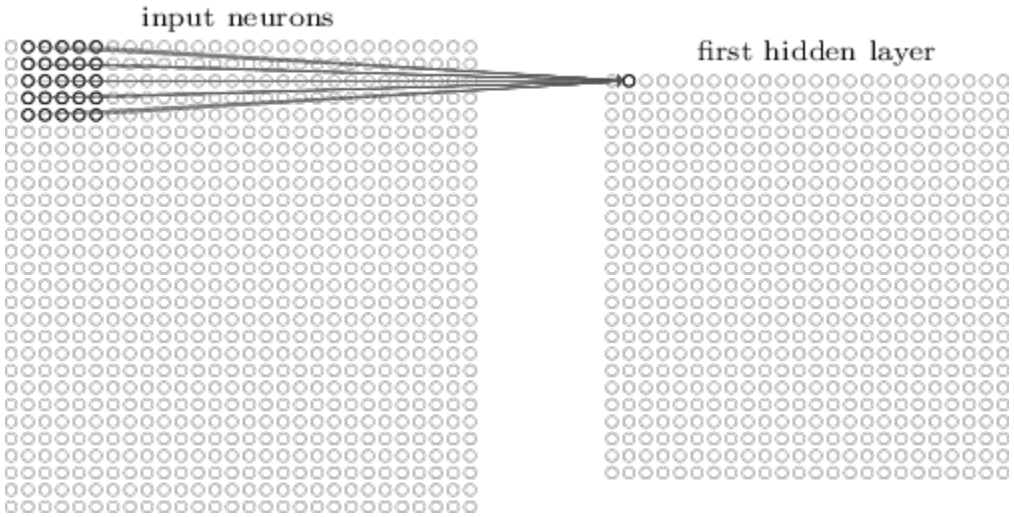


Image taken from Michael Neilsen's book "Neural Networks and Deep Learning"

The convolution operation

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

Image

1	0	1
0	1	0
1	0	1

Filter/
Feature detector

1. Pointwise multiply
2. Add results
3. Translate filter (move a stride)

$1_{\times 1}$	$1_{\times 0}$	$1_{\times 1}$	0	0
$0_{\times 0}$	$1_{\times 1}$	$1_{\times 0}$	1	0
$0_{\times 1}$	$0_{\times 0}$	$1_{\times 1}$	1	1
0	0	1	1	0
0	1	1	0	0

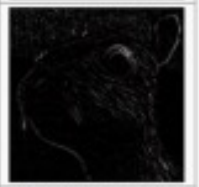
Image

4		

Convolved
Feature (feature map)

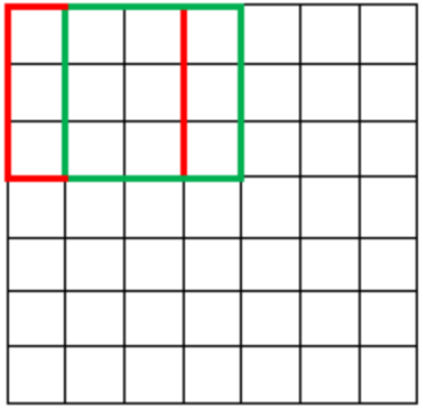
Filters

Original Image

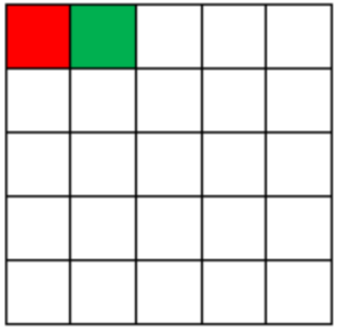
Operation	Filter	Convolved Image
Identity	$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	
Edge detection	$\begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ -1 & 0 & 1 \end{bmatrix}$	
	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	
	$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$	
Sharpen	$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$	
Box blur (normalized)	$\frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$	
Gaussian blur (approximation)	$\frac{1}{16}\begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$	

Stride

7 x 7 Input Volume

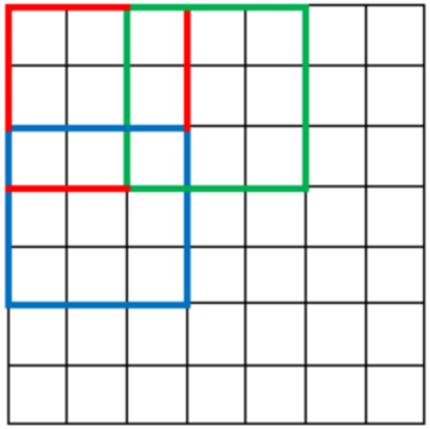


5 x 5 Output Volume

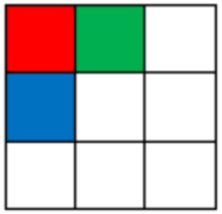


Stride 1

7 x 7 Input Volume



3 x 3 Output Volume



Stride 2

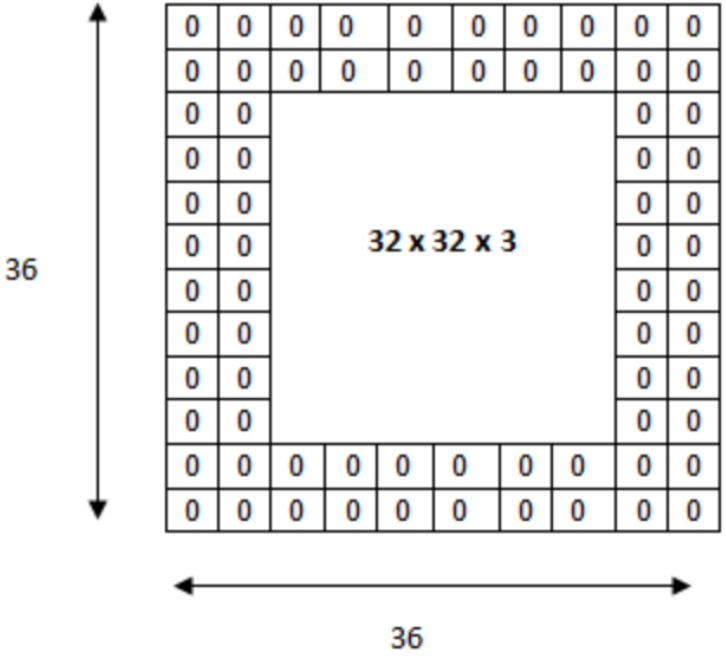
Depth and Zero Padding

and stride are all hyperparams

- Depth

convolved images are still images

- Padding



Shared weights and biases

A single layer in a CNN includes multiple feature maps

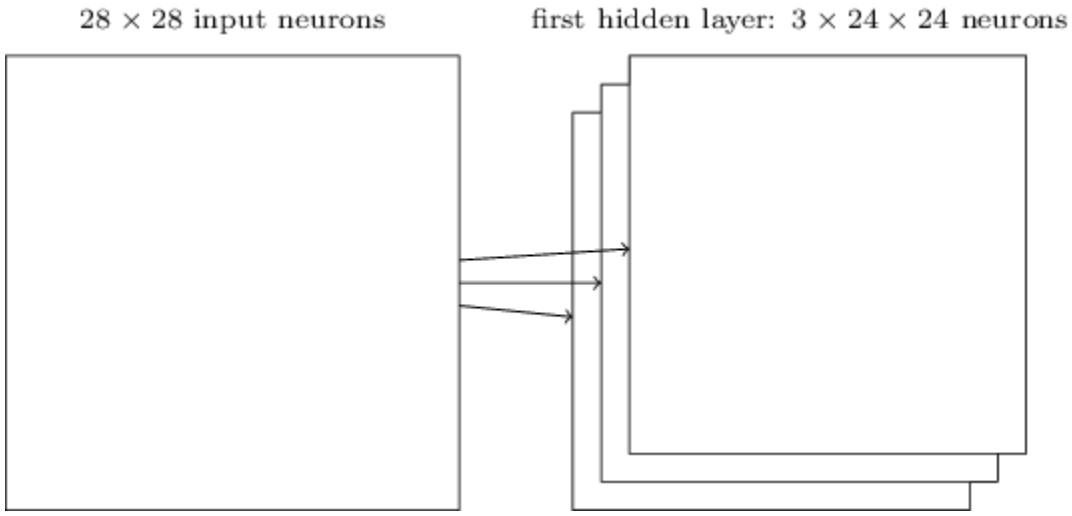


Image taken from Michael Neilsen's book "Neural Networks and Deep Learning"

Shared weights and biases

- Same weights and biases are used within the hidden layer
- E.g., output of the j, kth hidden neuron is

$$\sigma\left(b + \sum_{l=0}^{4}\sum_{m=0}^{4} w_{l,m} a_{j+l,k+m}\right)$$

  - A convolutional operator
- All neurons in the first hidden layer detect the same feature (helps with translations)

Shared weights and biases

Why use shared weights?

- Greatly reduces the number of parameters (faster training)
- MNIST Example
  - CNN layer with 20 feature maps: 520 parameters
  - Fully connected neural net with 30 hidden neurons: 23,550 parameters

Shared weights and biases

Some example features when trained on MNIST

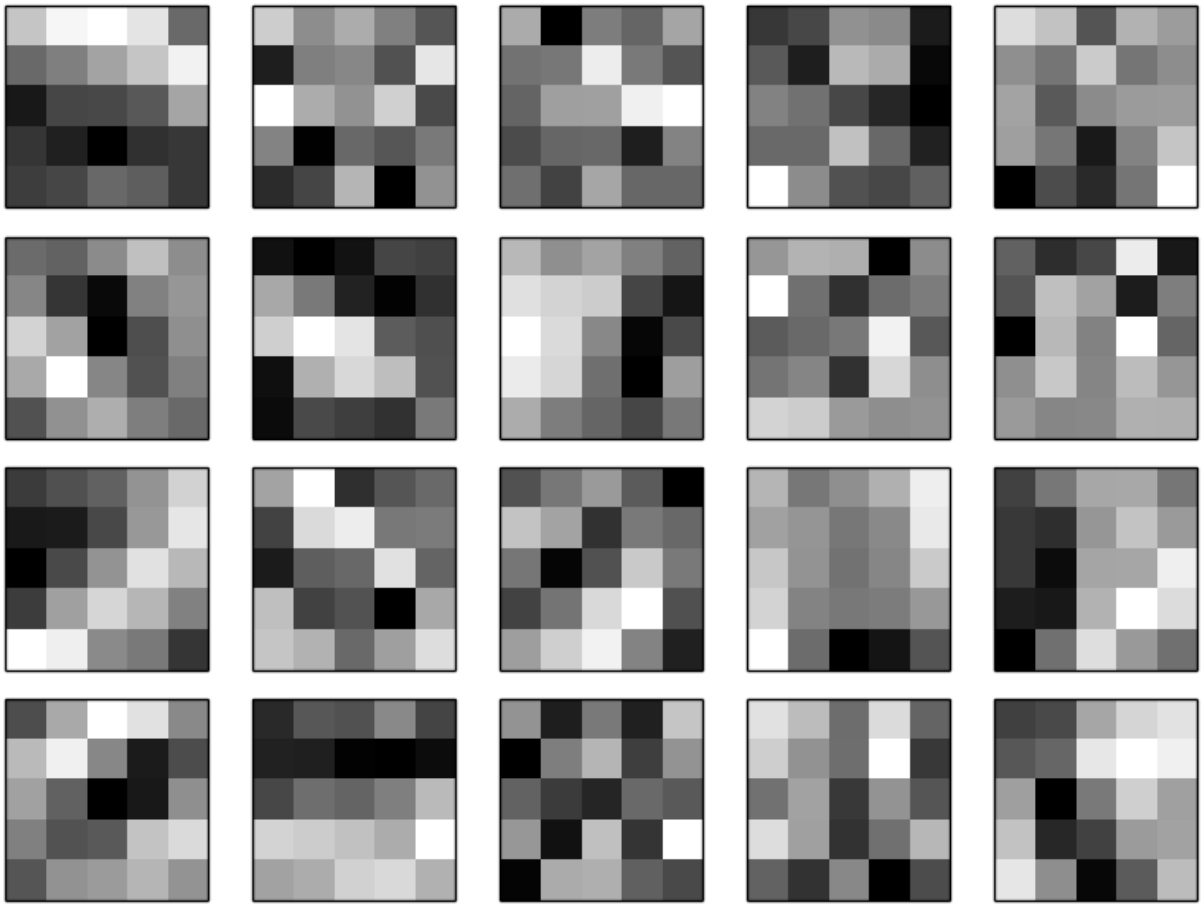


Image taken from Michael Neilsen's book "Neural Networks and Deep Learning"

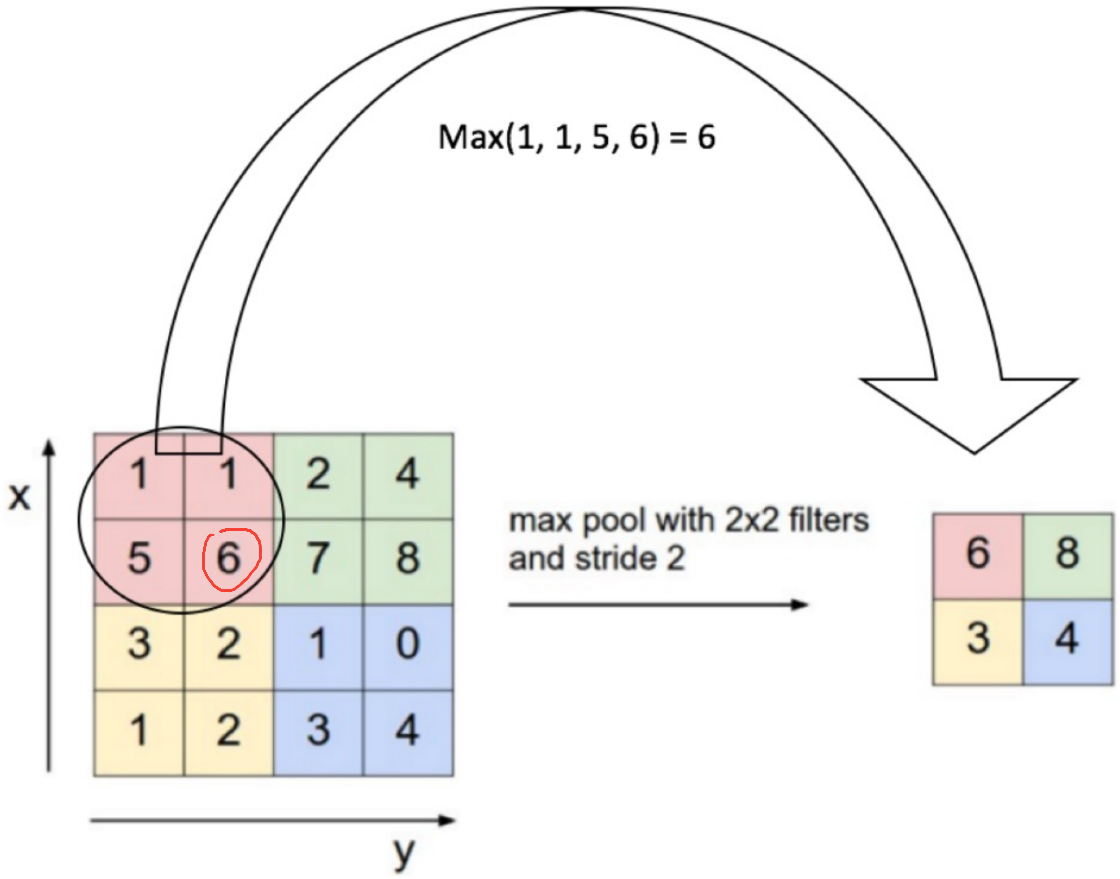
Pooling layers

- Usually used immediately after convolutional layers
- Used to simplify the output from the convolutional layer
  - E.g., max-pooling outputs the maximum activation in a region

Pooling layers

- Intuition: the exact location of a feature isn't as important as its rough location
  - Helps prevent overfitting
- Reduces the number of parameters needed in later layers
- L_2 pooling is also common (L_2 norm)

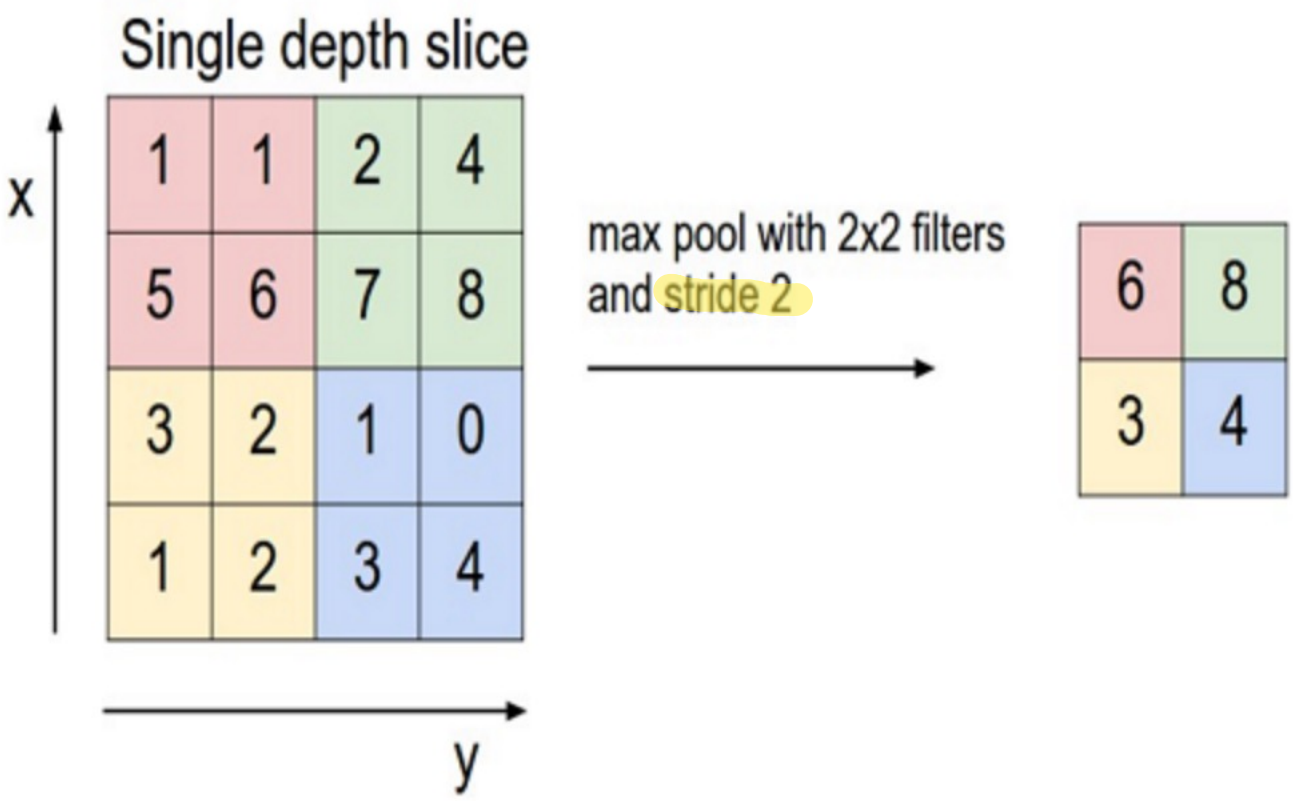
Pooling



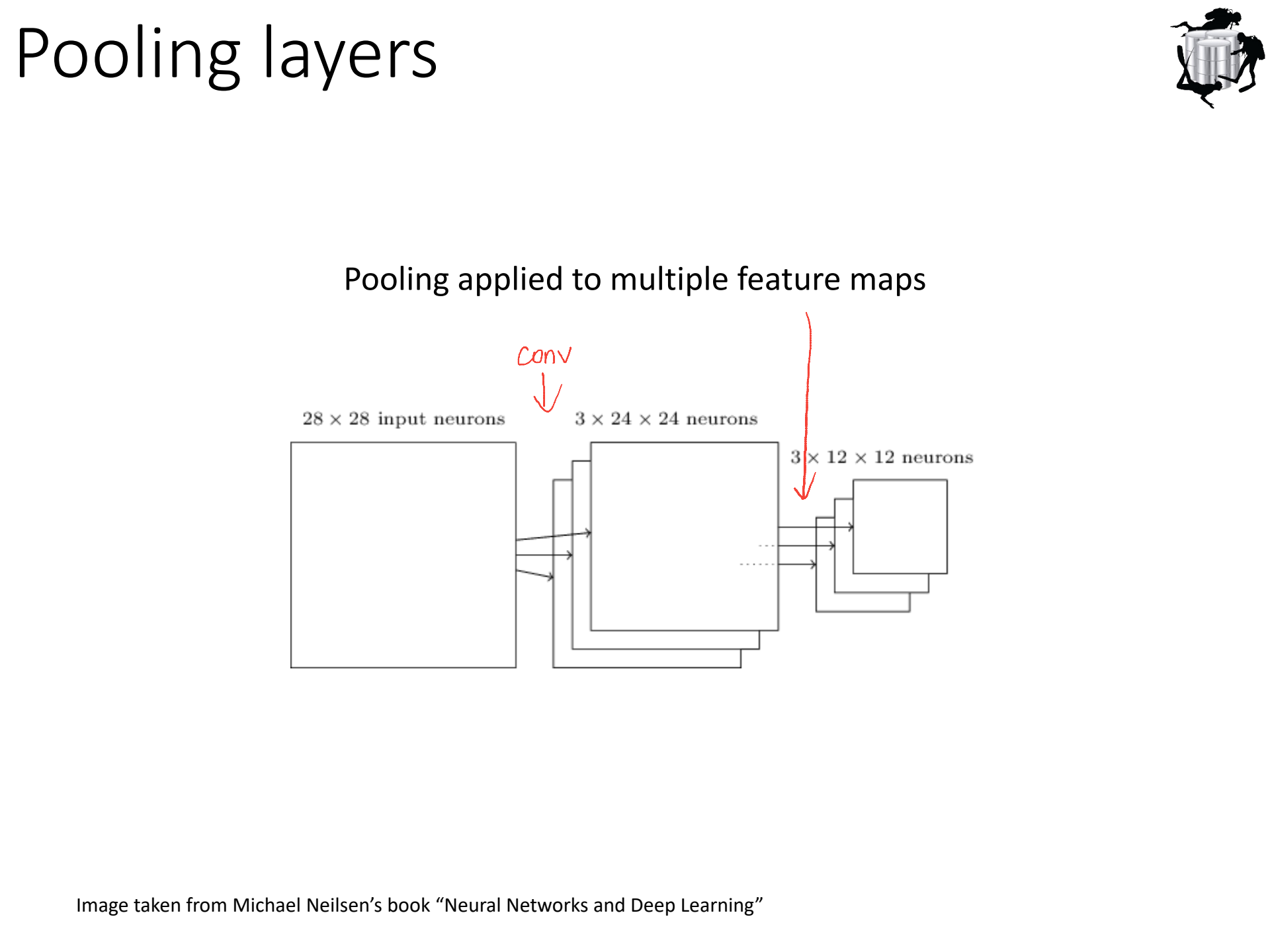
Downsampling the results by pooling values by averaging or max

Max pool is used more commonly

Pooling w/ stride 2



Pooling layers

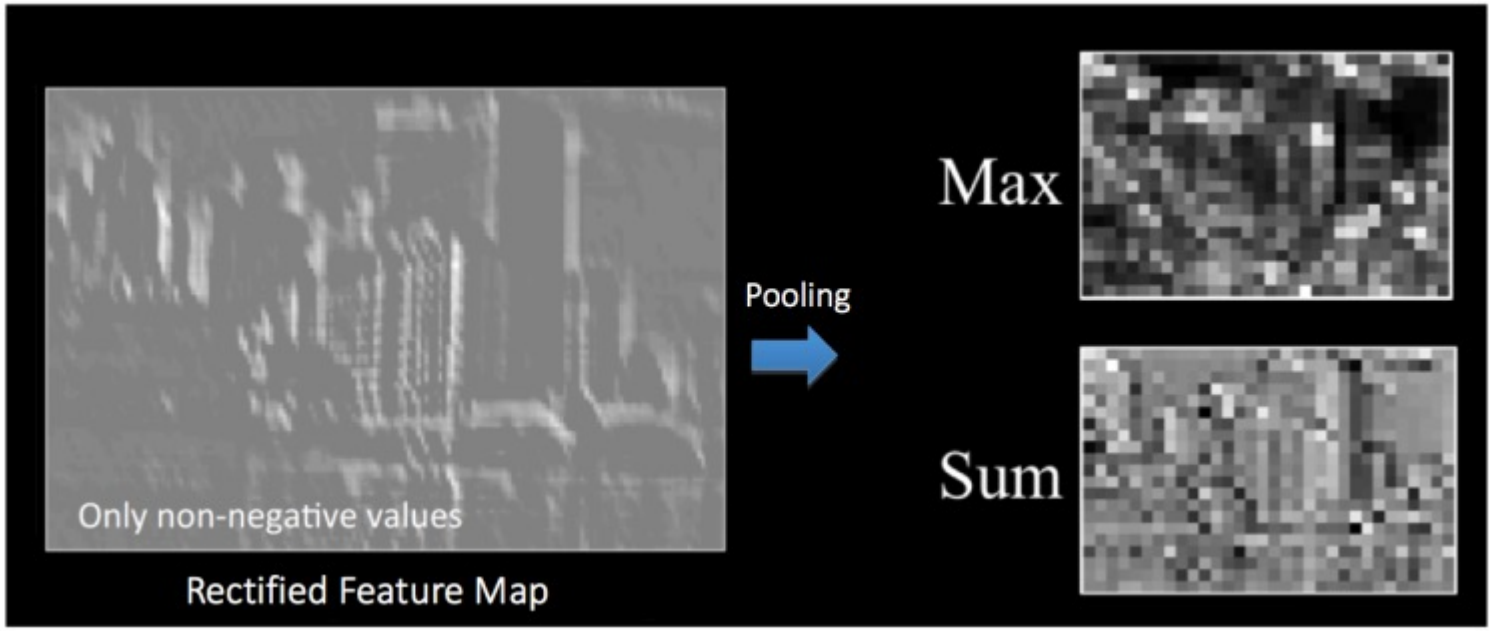


Pooling applied to multiple feature maps

Image taken from Michael Neilsen's book "Neural Networks and Deep Learning"

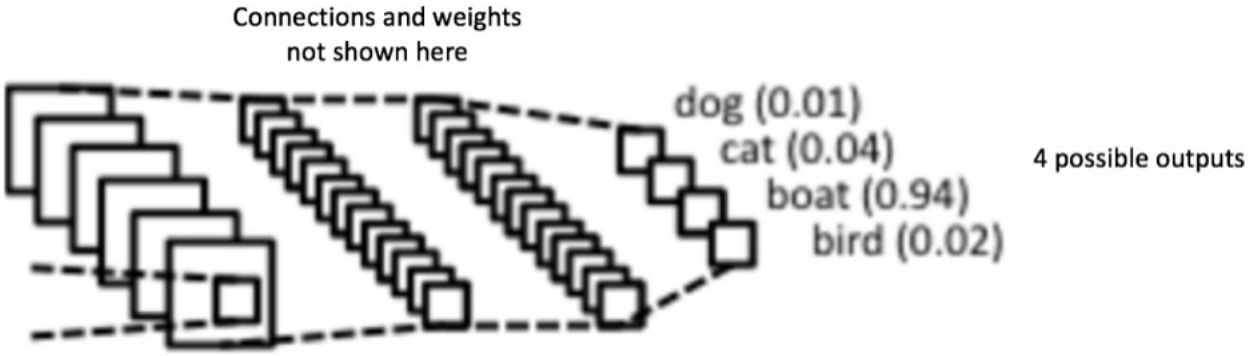
Other Effects of Pooling

- Reducing spatial size of the image
- Making CNNs invariant to distortions and noise
- Reduces parameters
- Scale invariance



Fully connected layer to combine

- Convolutional layers detected features
- Pooling layers reduced complexity
- Now we have a set of feature maps
- Can combine them nonlinearly to classify



A full CNN

Slight modifications required for back propagation

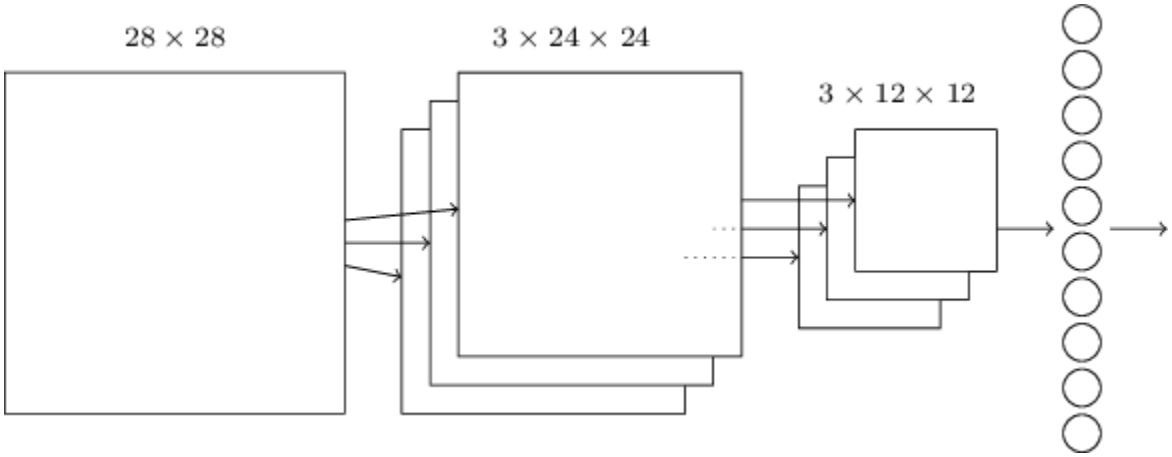
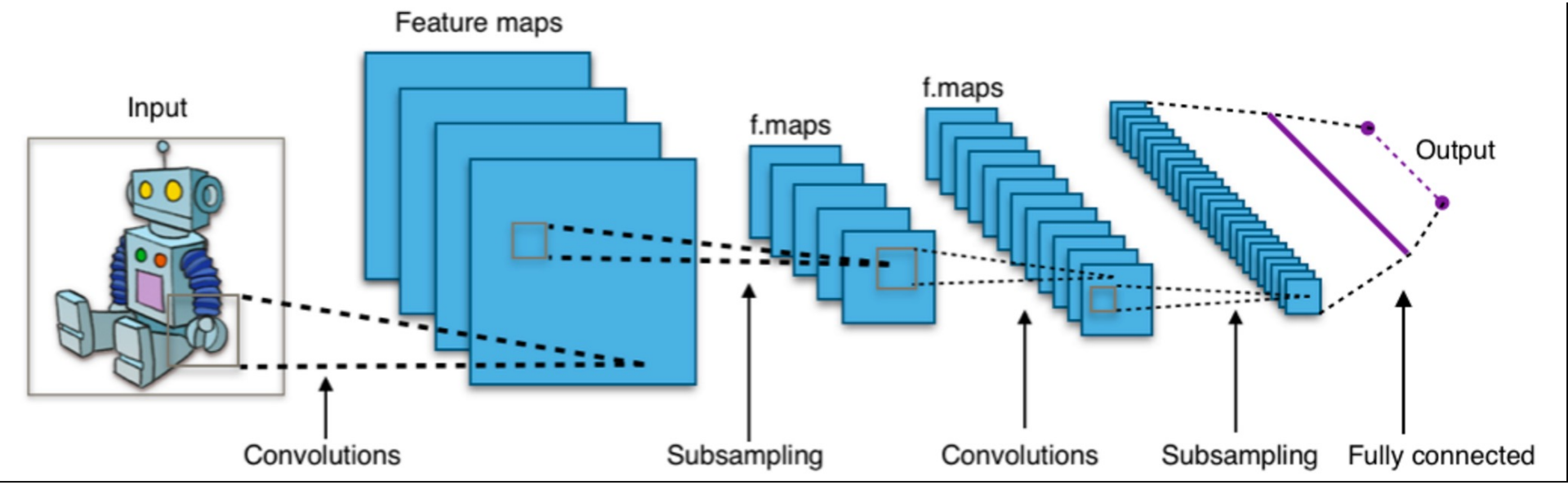


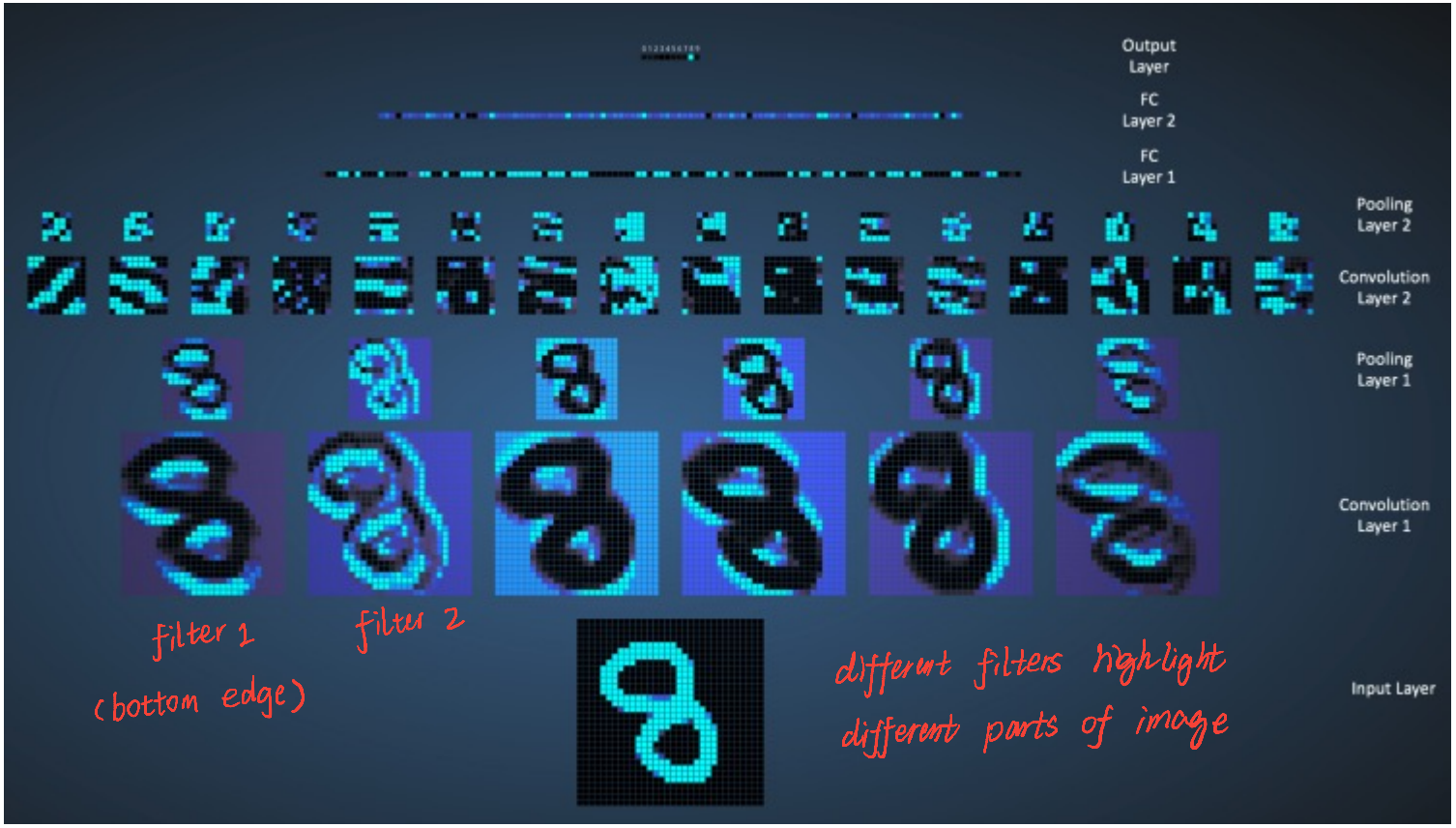
Image taken from Michael Neilsen's book "Neural Networks and Deep Learning"

CNN Architecture



Each convolutional unit has the same weight.

Visualizing a CNN



Learn more abstract features

feature map

Other Modifications: Inception

- GoogleNet 2014: Introduced the Inception module that reduced parameters from 60 M to 4M (order of magnitude)
- ResNet 2015: Residual network

VGG

1x1 conv sum over 1 pixel of different channels

(a depthwise conv)

Inception Network

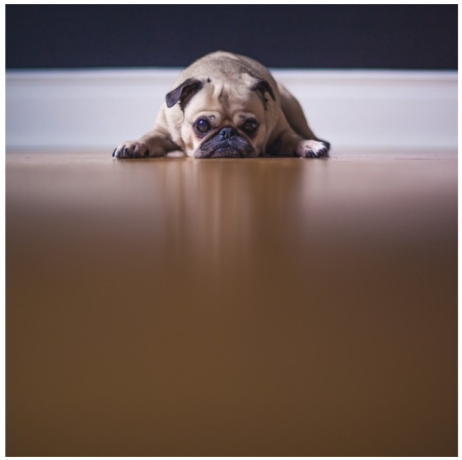
- Prior to Inception CNNs just tried stacking convolutions deeper and deeper to get better performance

Szegedy, Christian, et al. "Going deeper with convolutions." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015.

Resolution differences

- Inception network realized that different features can come at different resolutions, which may need different sized convolutional filters

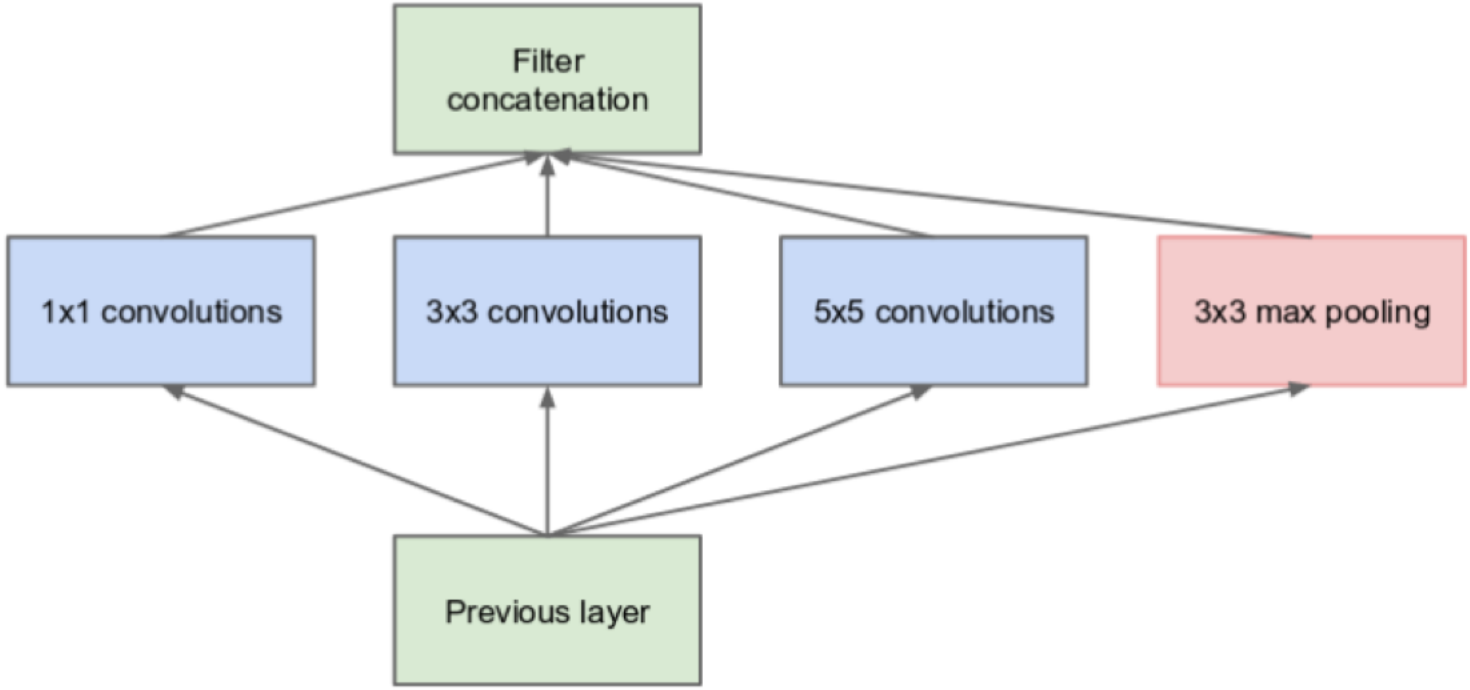
Detet features at different resolutions using 1 NN



a zoom in dog

a zoom out dog

Basic Inception Module

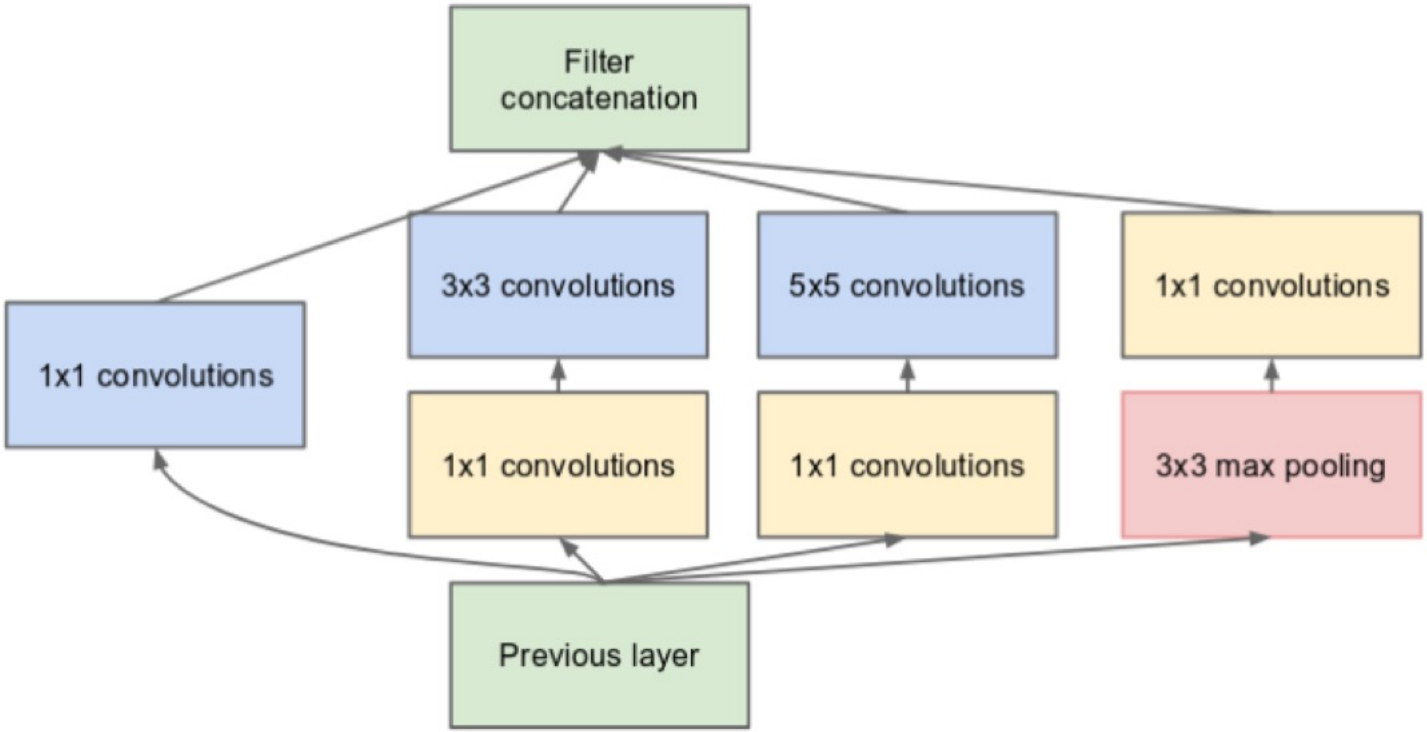


filters

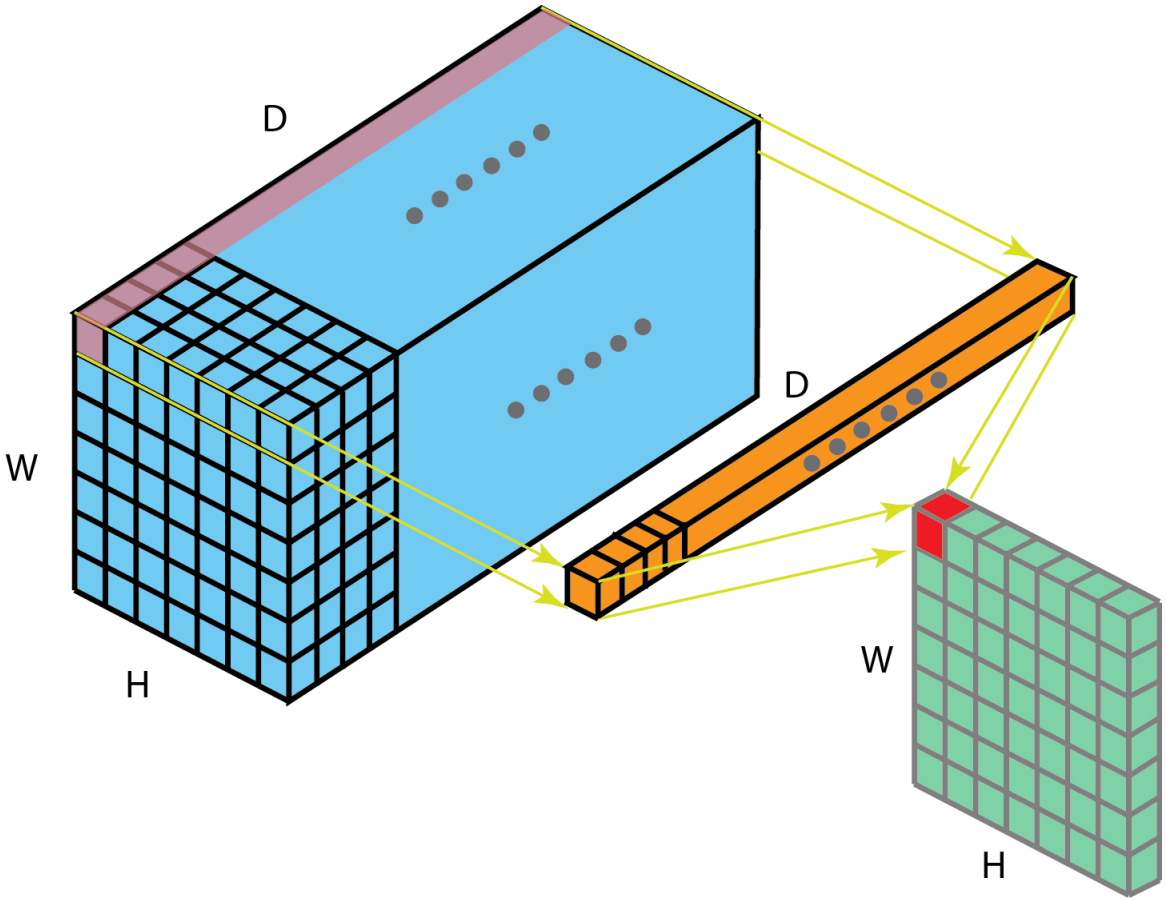
Different resolutions of filters are the same level

More convolutions, more expensive

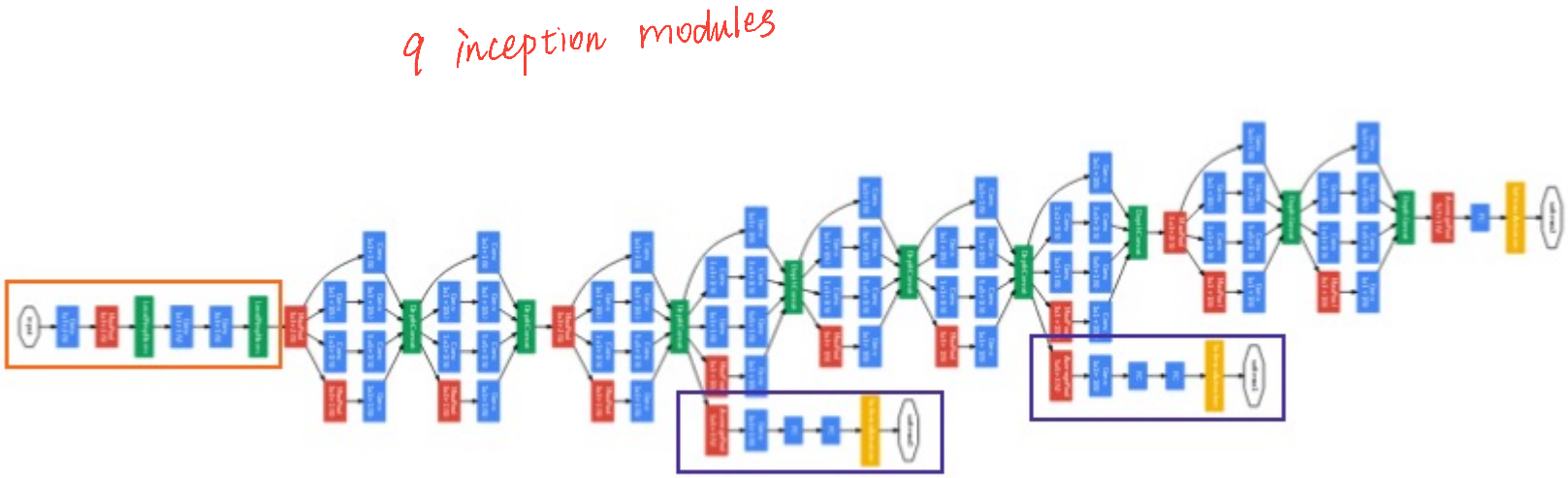
- Instead try to limit the number of input channels of a layer by doing 1x1 convolutions



1x1 Convolutions for Dimension Reduction



GoogleNet Architecture



ResNets

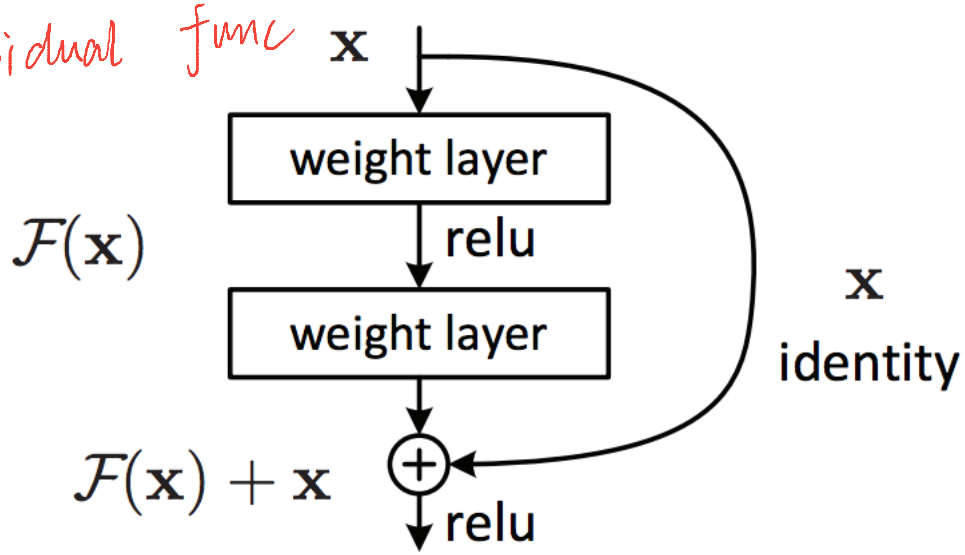
- Inception network performed very well
- However, making it even deeper did not improve performance
- This motivated a new type of block
  - One that you can stack more and more of and get better performance with each additional block
- Gradient flow through the network needs to be improved
- Idea: use “skip connections”

He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

ResNets

- The normal nonlinearity is learned and then *added* to the input
- Instead of the activation being ***a = F(x)*** it is ***a = F(x) + x***

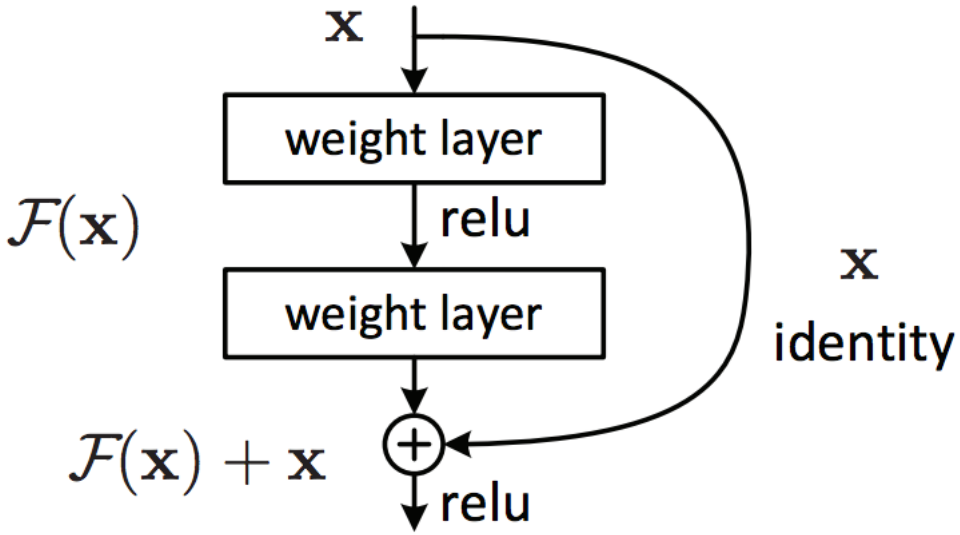
learn a residual func



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

ResNets

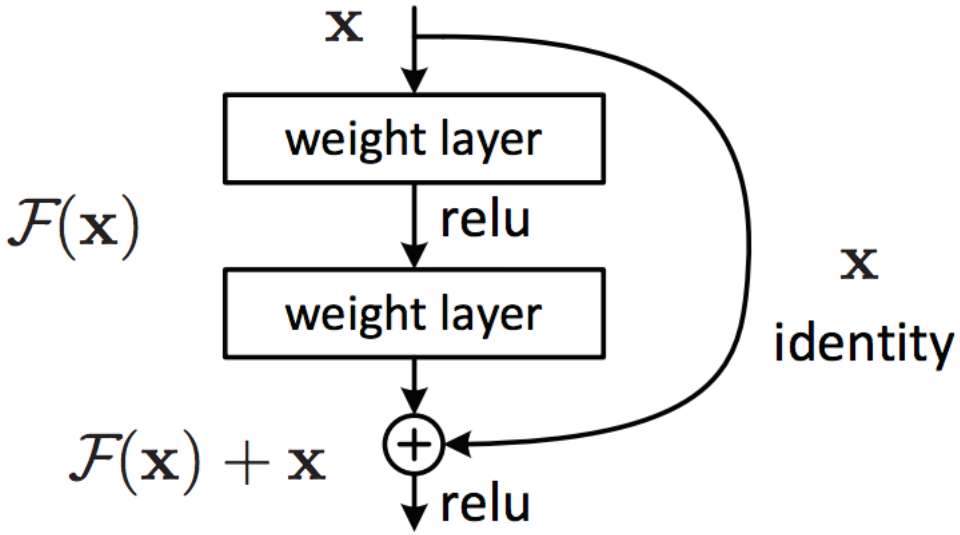
- How does this address vanishing gradients?



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

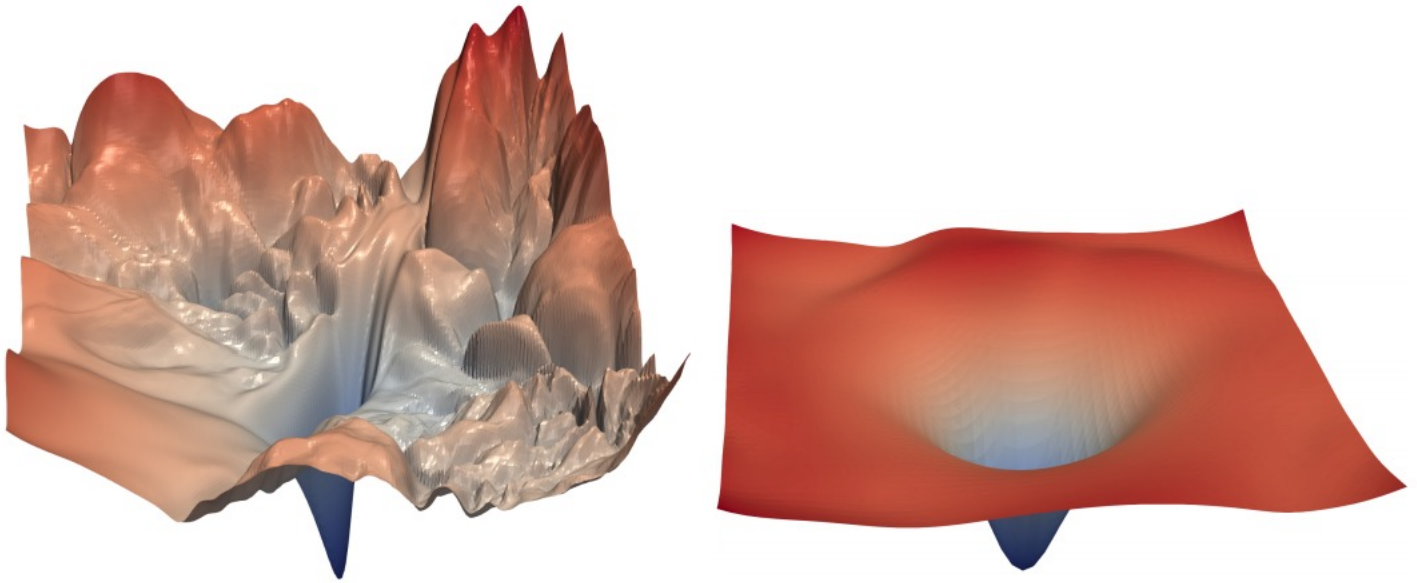
ResNets

- How does this address vanishing gradients?
- No matter what the derivative of ***F*** is, the derivative going through the identity branch is constant



He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Effect of ResNET on Loss Landscape



(a) without skip connections

(b) with skip connections

[From Li et al. 2017]

ResNets

- This allows for training **much** deeper networks
- But is each layer still as expressive as a regular layer? How does the network perform?

- Various architectures of ResNets at just 34 layers already outperform Inception (GoogLeNet)
- But now we can train even up to **152** layers!
- More layers give better performance

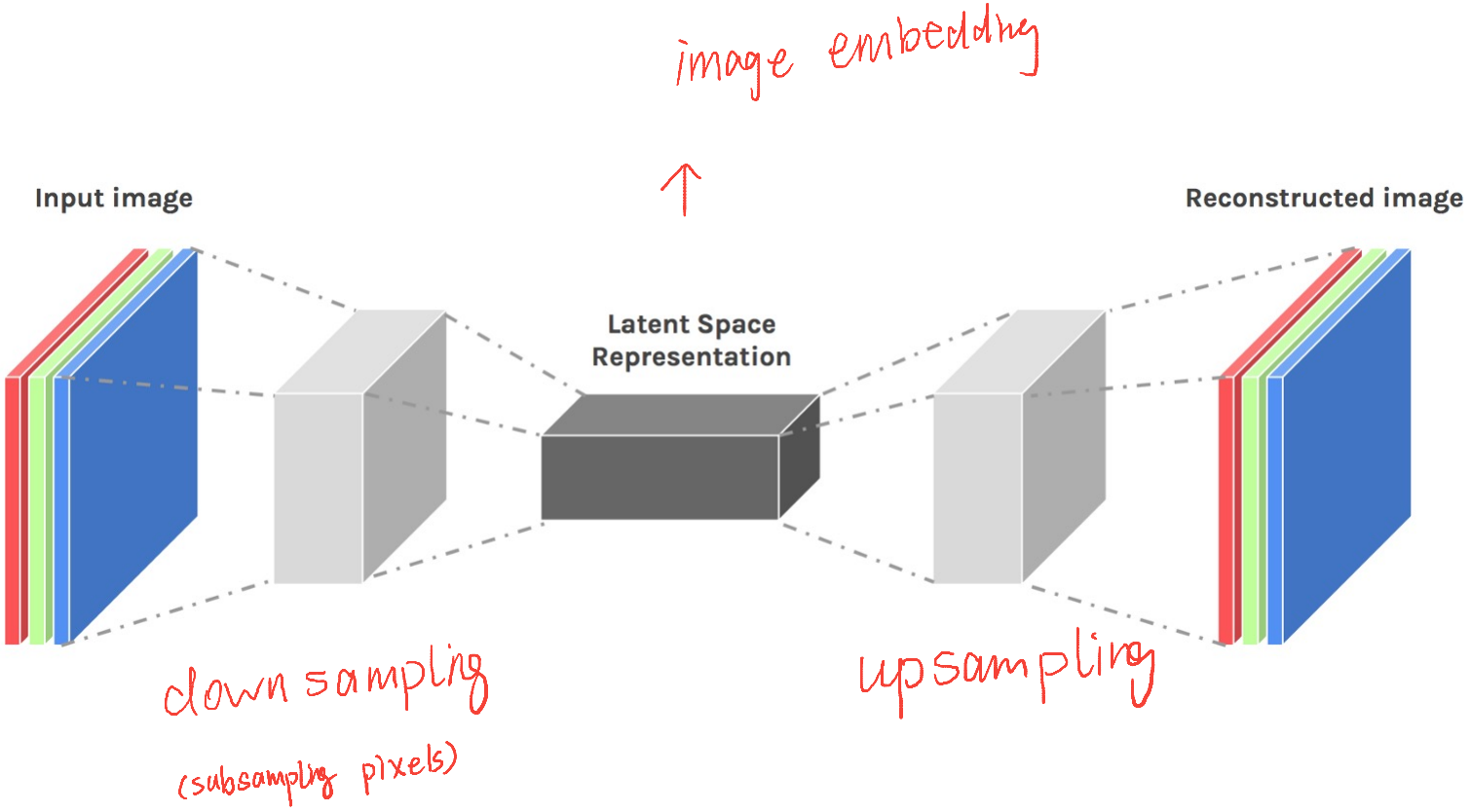
model	top-1 err.	top-5 err.
VGG-16 [41]	28.07	9.33
GoogLeNet [44]	-	9.15
PReLU-net [13]	24.27	7.38
plain-34	28.54	10.02
ResNet-34 A	25.03	7.76
ResNet-34 B	24.52	7.46
ResNet-34 C	24.19	7.40
ResNet-50	22.85	6.71
ResNet-101	21.75	6.05
ResNet-152	**21.43**	**5.71**

Table 3. Error rates (%, **10-crop** testing) on ImageNet validation. VGG-16 is based on our test. ResNet-50/101/152 are of option B that only uses projections for increasing dimensions.

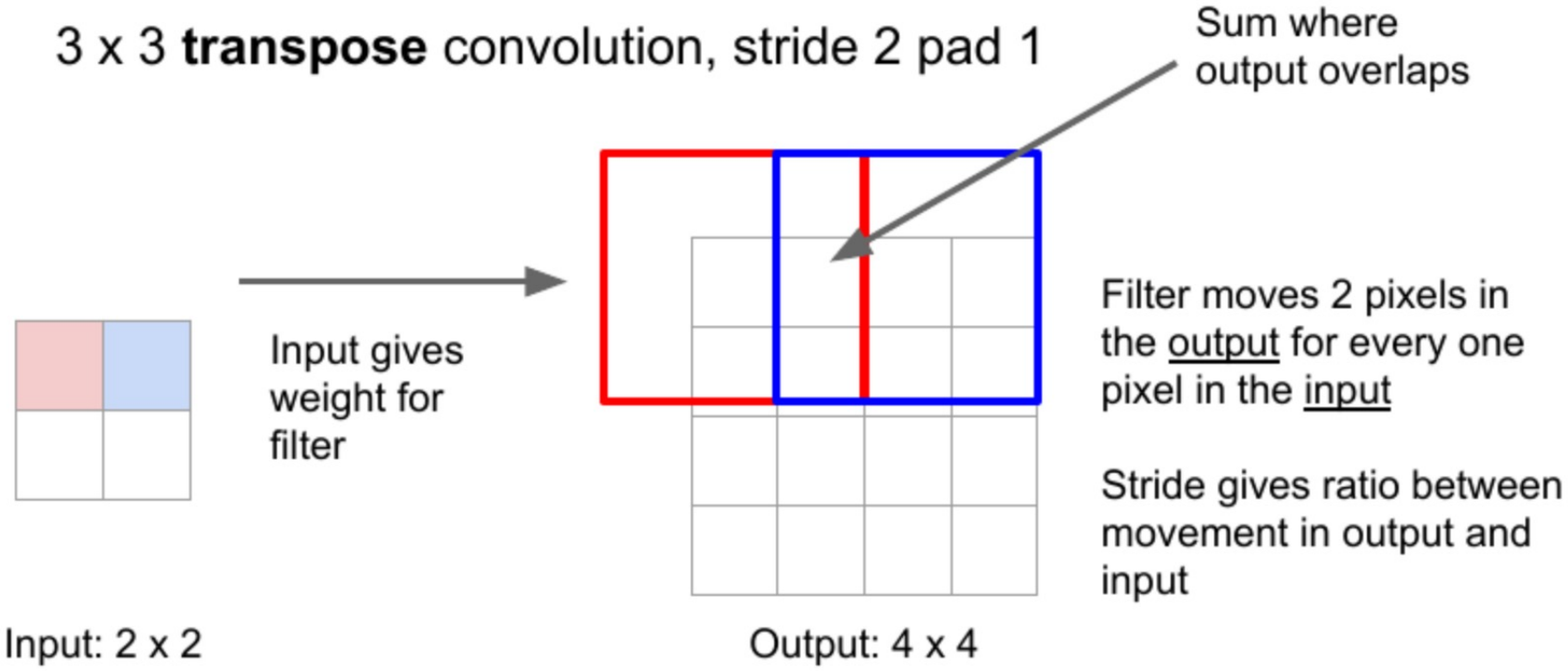
He, Kaiming, et al. "Deep residual learning for image recognition." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Convolutional Autoencoders

bowtie shape

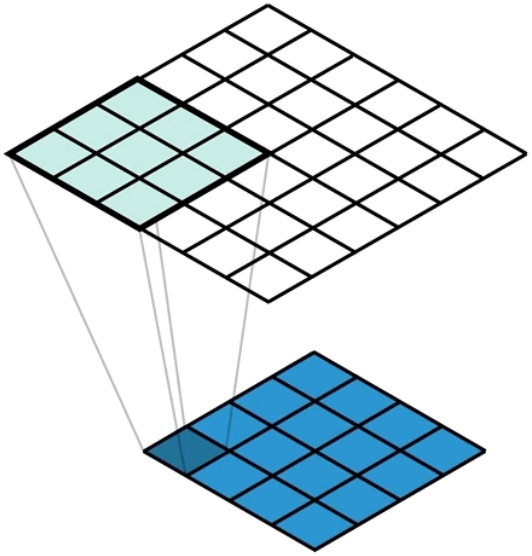


Decoding: Transposing Convolutions



Weight gives amount of input value being distributed to each of the output cells

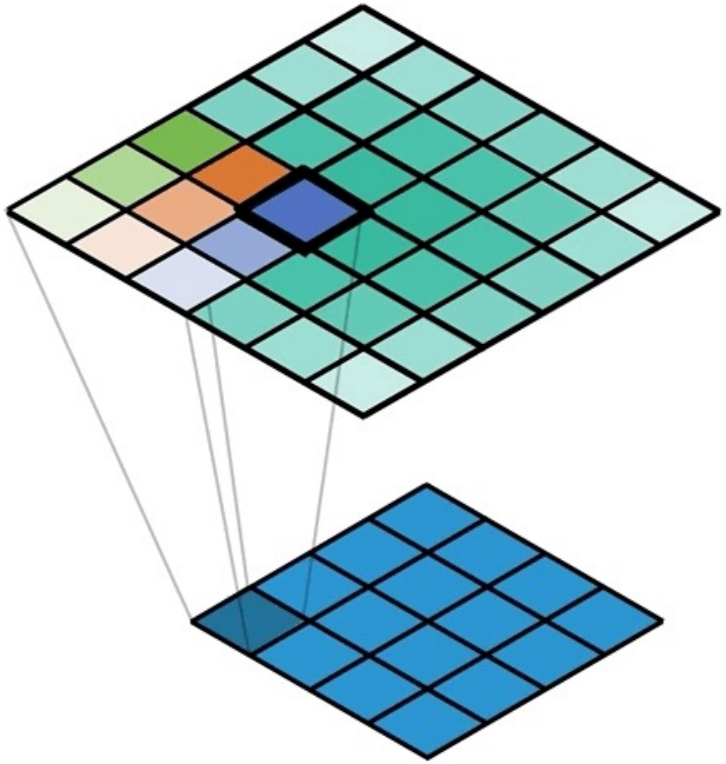
Decoding: Transposed Convolutions



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
1																						
2			Input							Kernel						Output						
3																						
4																1	2	3	3	2	1	
5			1	1	1	1				1	1	1				2	4	6	6	4	2	
6			1	1	1	1				1	1	1				3	6	9	9	6	3	
7			1	1	1	1				1	1	1				3	6	9	9	6	3	
8			1	1	1	1										2	4	6	6	4	2	
9																1	2	3	3	2	1	
10																						

From the output cell perspective

weighted interpolation of input pixels

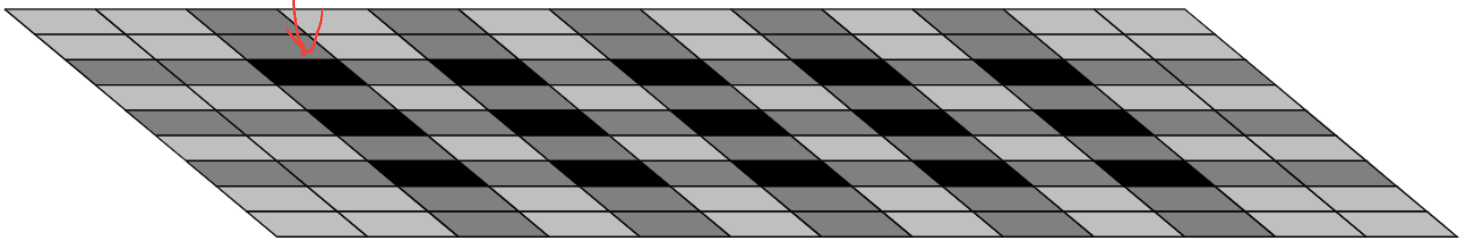


Checkerboard Artifacts

issue

transpose will lead to ↗

more overlap



Upsampling

Nearest Neighbor

1	2
3	4

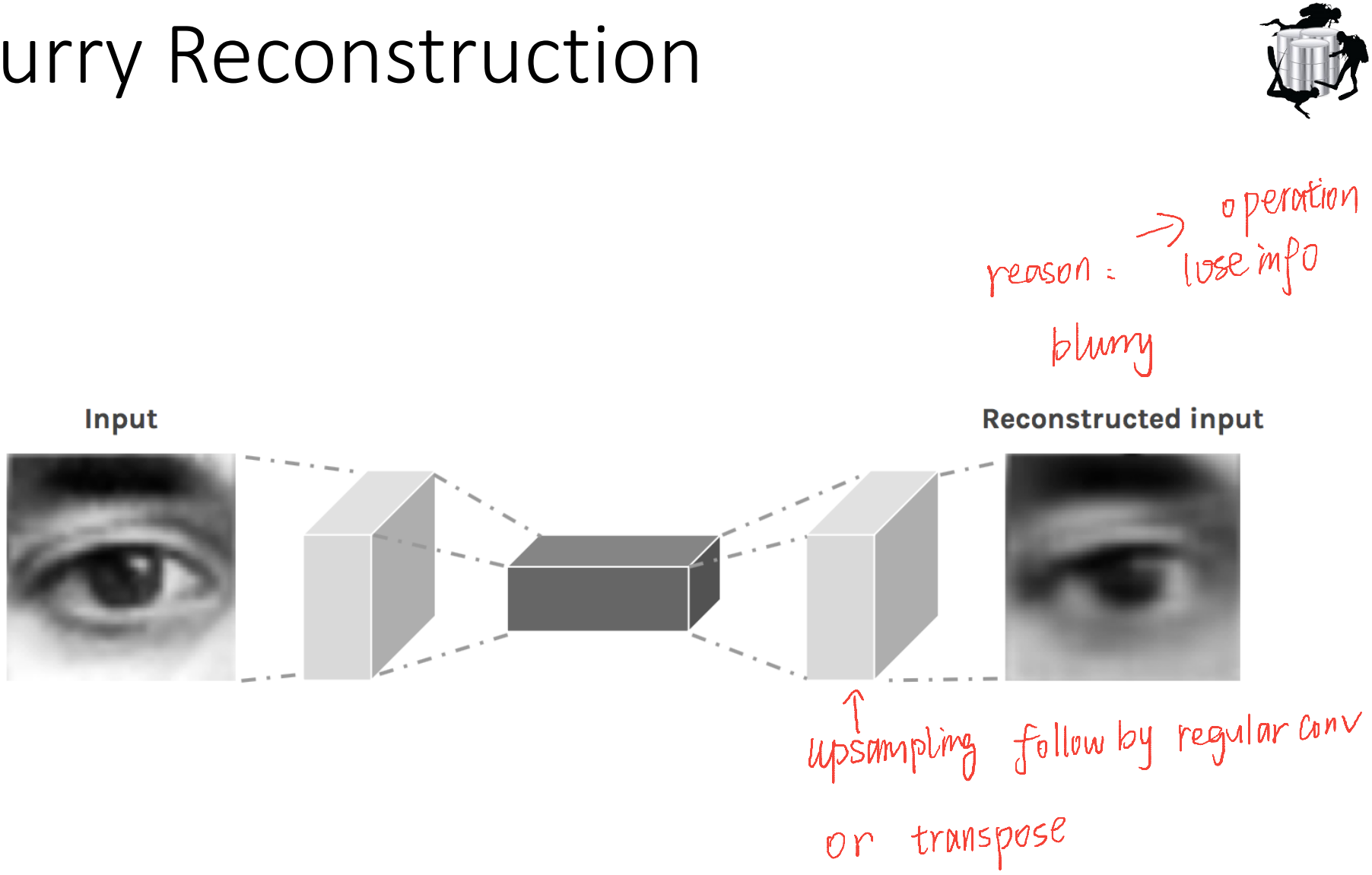
→

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

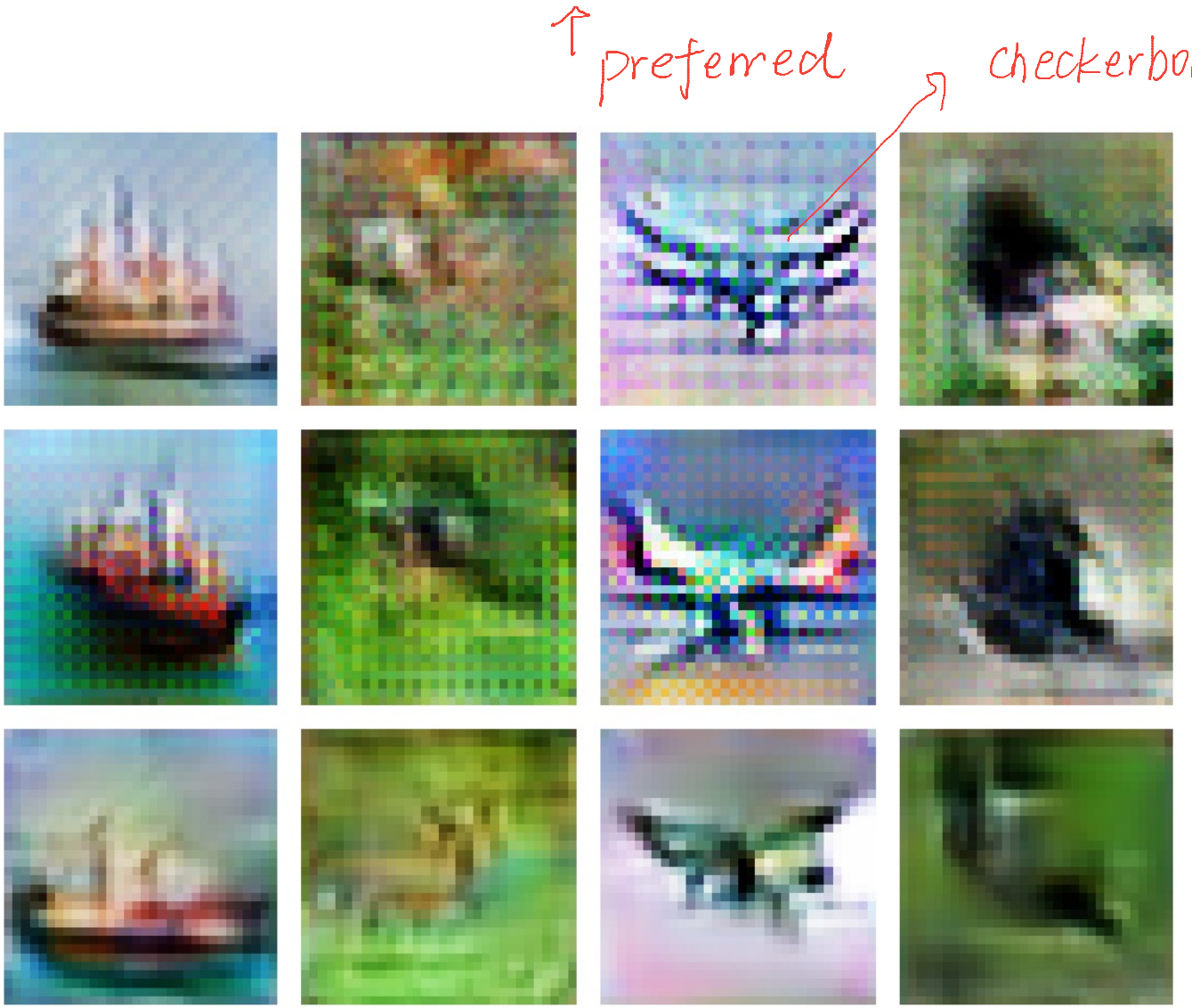
Input: 2 x 2

Output: 4 x 4

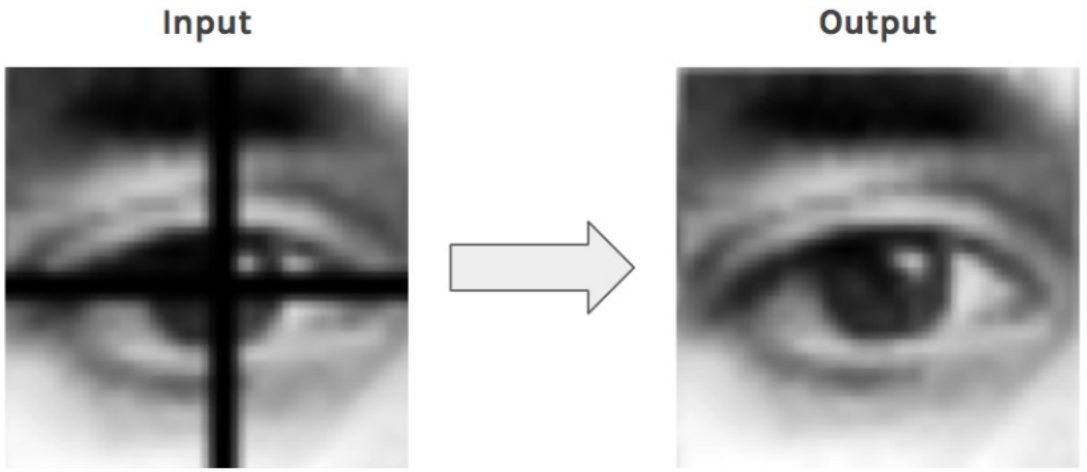
Blurry Reconstruction



Transpose vs Upsampling



Uses: Denoising Images



The CAE is trained to remove the crosshair 十字架

Image Colorization 图象着色

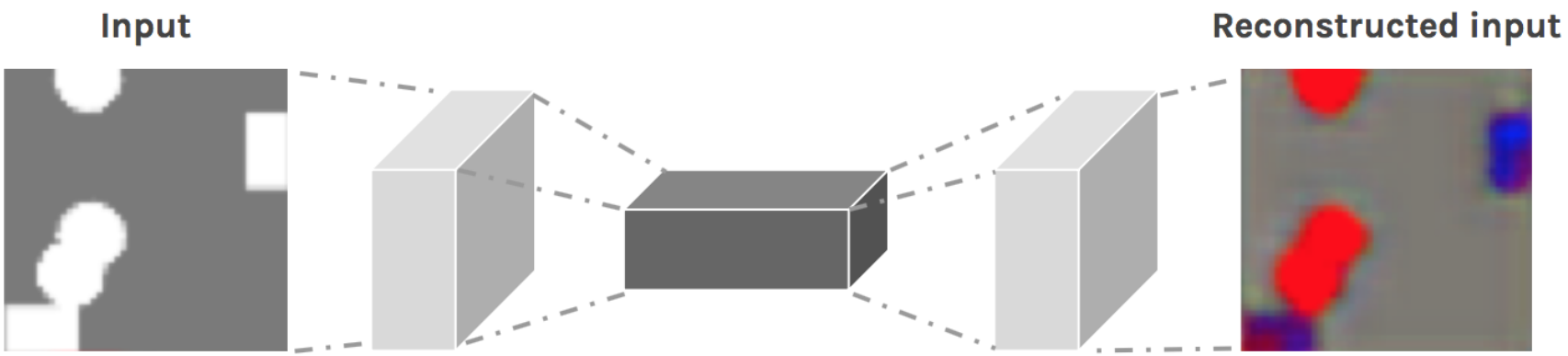
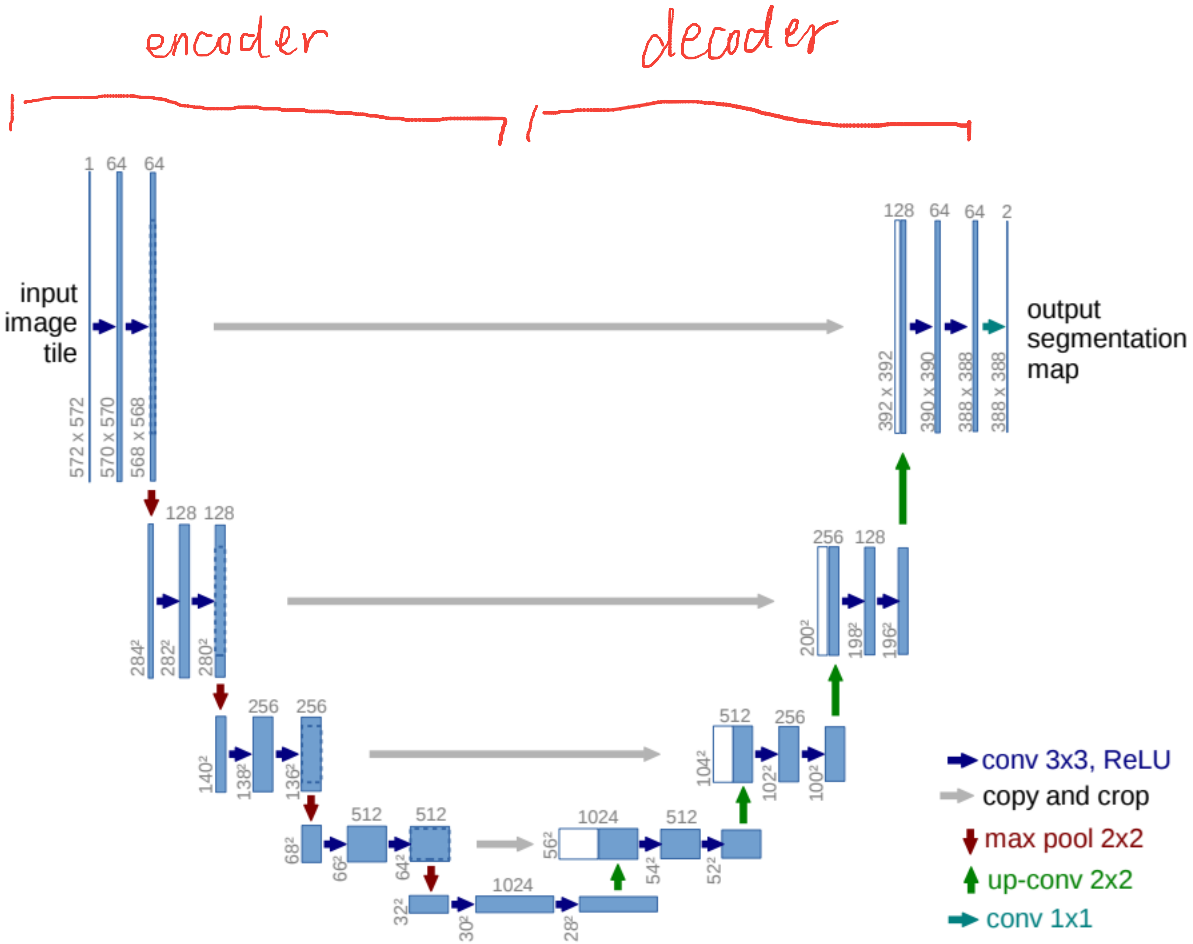


Image Segmentation: Unet

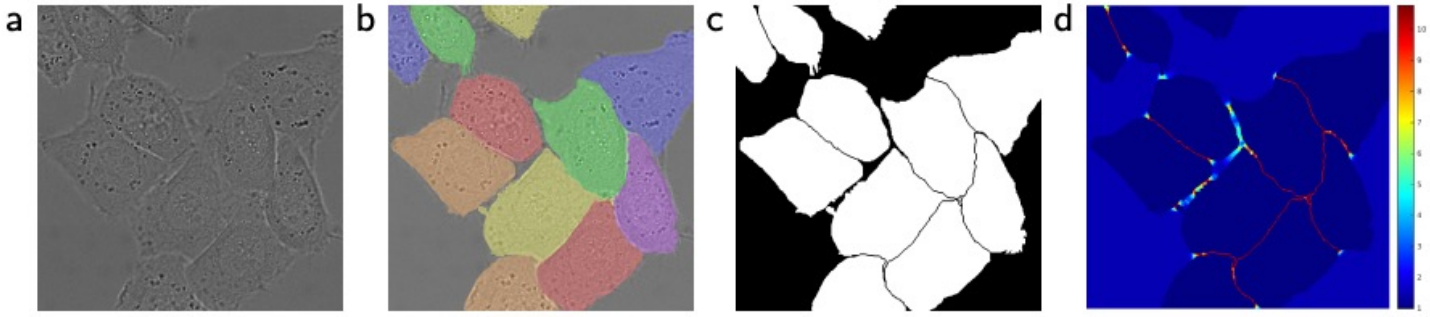


Decoder path: pooling layers replaced by upsampling layers along with skip connection

Segmentation of cells

supervised training

annotated image



Raw

Ground Truth

Unet output

Training Loss

biomedical image

MSE at pixel level

Suggested Readings

- Chapter 6 Nielsen
- Szegedy et al. Going deeper with convolutions 2014 GoogleNet
- He et al. Deep Residual Learning for Image Recognition 2015
- Ronneberger et al. Unet Architecture for Biomedical Image Segmentation Unet
- http://cs231n.github.io/convolutional-networks/
- https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
- https://towardsdatascience.com/a-simple-guide-to-the-versions-of-the-inception-network-7fc52b863202