

Deep learning

episode 2

Computer vision applications



Image recognition



“Dog”

Image recognition



“Gray wall”

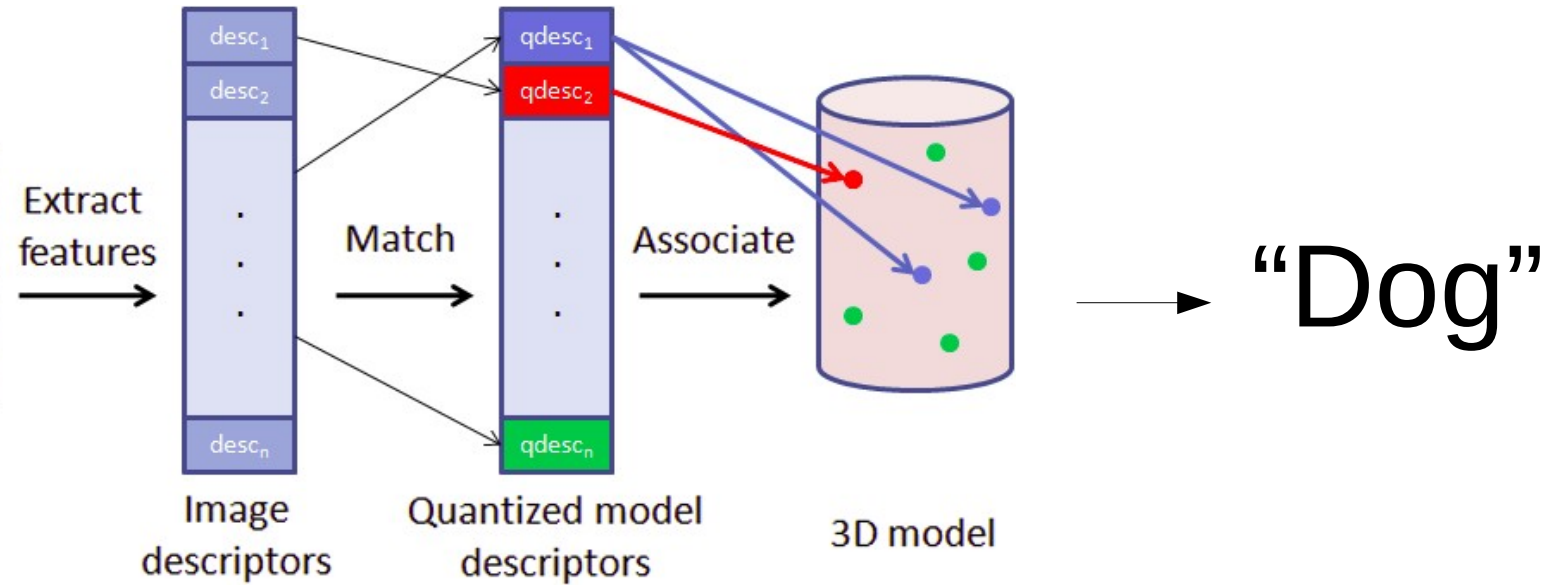
“Dog tongue”

“Dog”

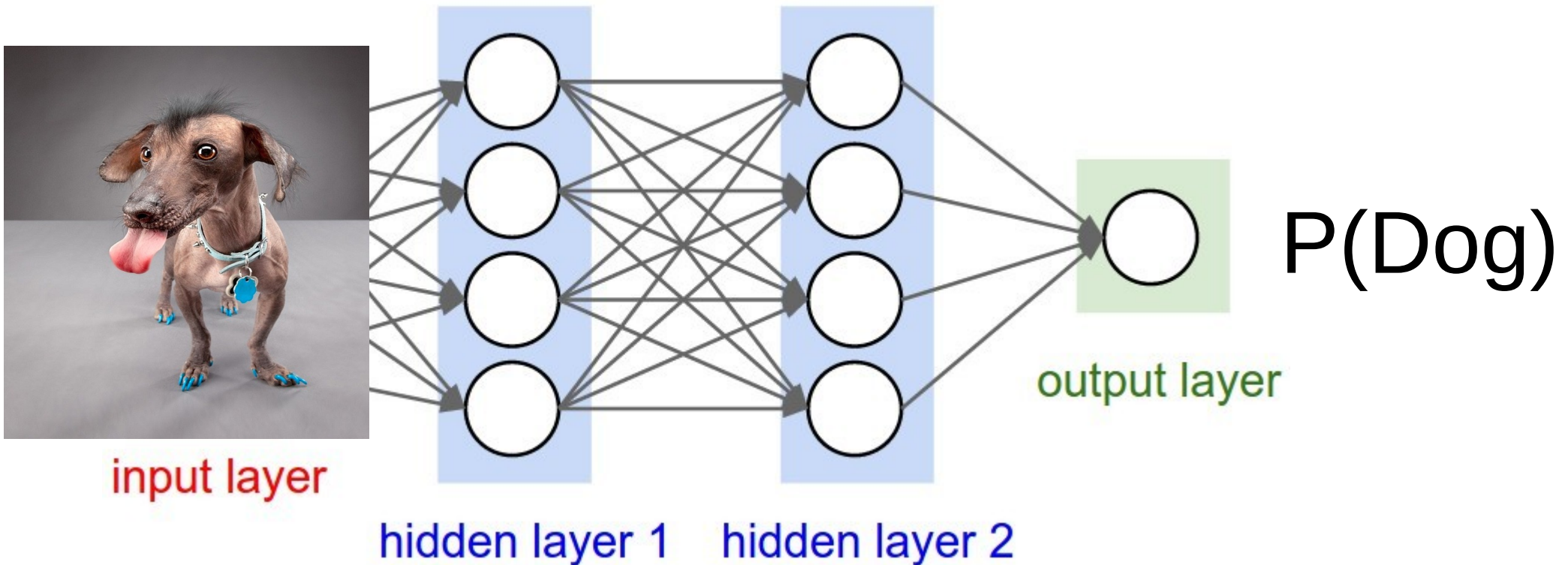
<a particular kind
of dog>

“Animal sadism”

Classical approach



NN approach



What features could NN learn this way?

Problem

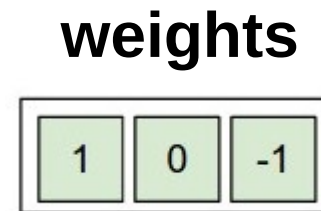
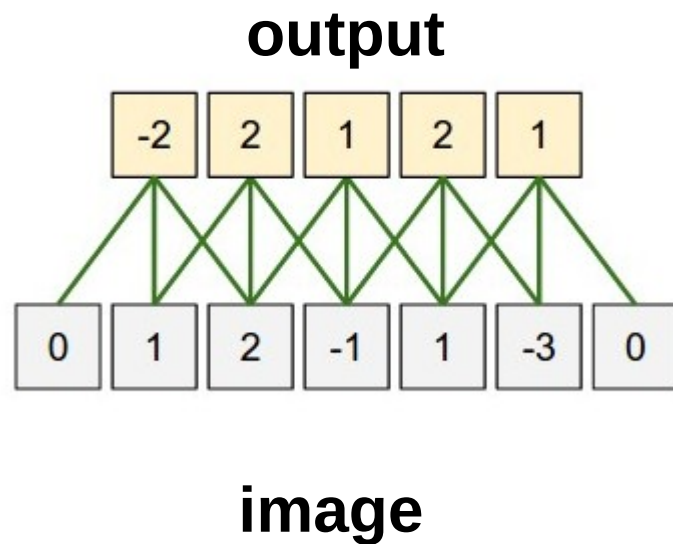
Should we require, say, “Dog ear” feature

- Linear combination can only select dog ear at a one (or a few) positions.
- Need to learn independent features for each position
- Next layer needs to react on “dog ear 0,0 or dog ear 0,1 or ... or dog ear 255,255”
- Introduce **a lot** of parameters and risk overfitting.

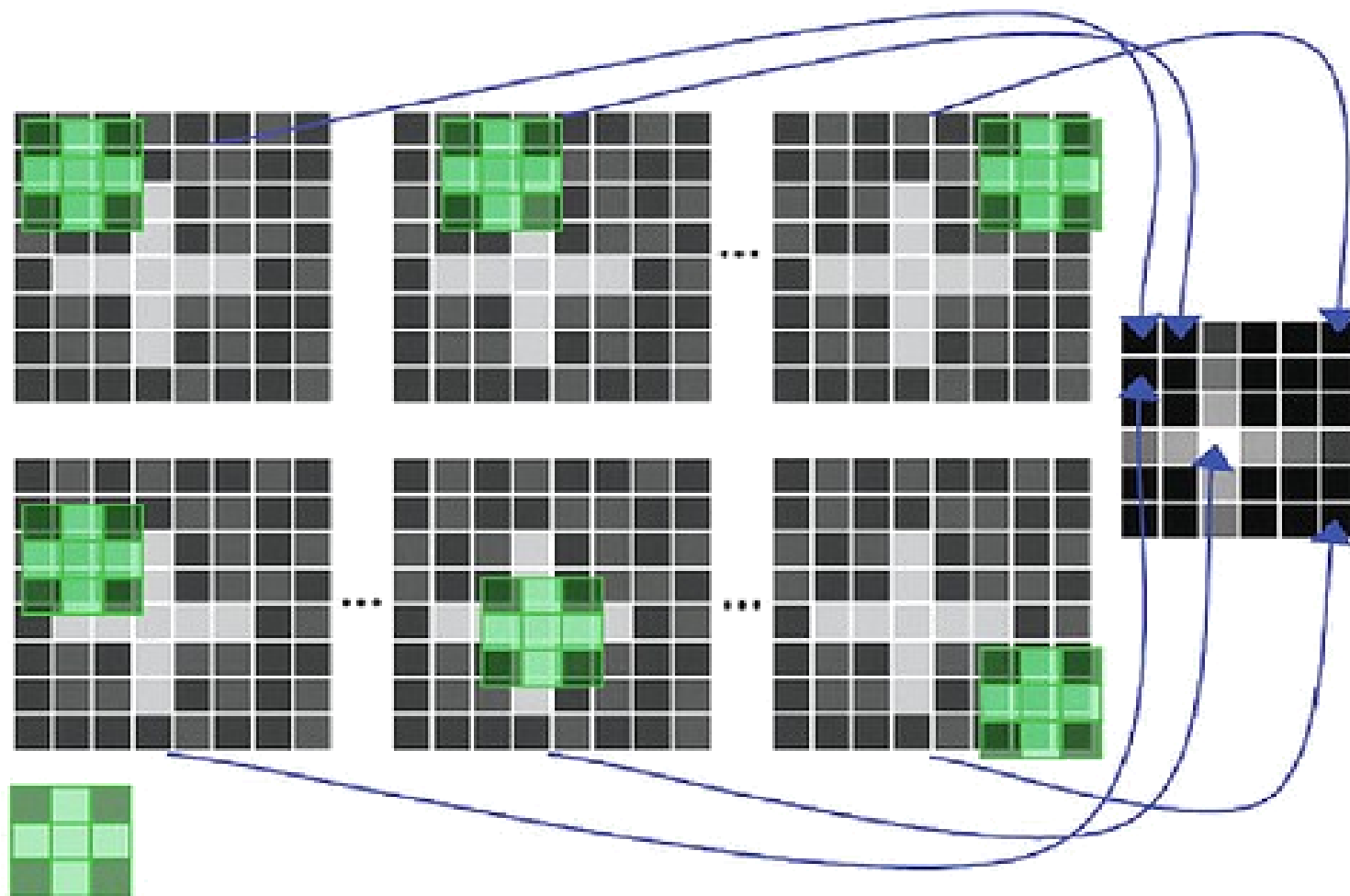
Idea: force all these “dog ear” features to use **exactly same weights**, shifting weight matrix each time.

Convolution

- Apply same weights to all patches



Convolution



apply same filter to all patches

Convolution

5x5

1 _{x1}	1 _{x0}	1 _{x1}	0	0
0 _{x0}	1 _{x1}	1 _{x0}	1	0
0 _{x1}	0 _{x0}	1 _{x1}	1	1
0	0	1	1	0
0	1	1	0	0

Image

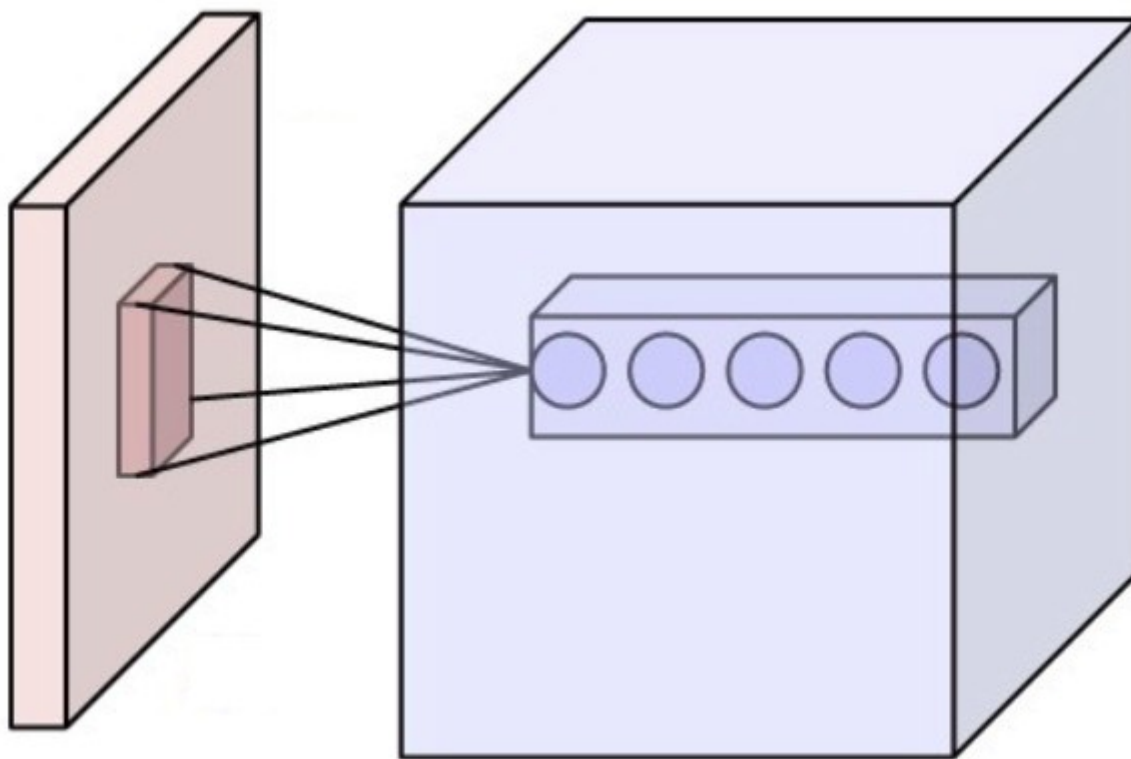
3x3 (5-3+1)

4		

Convolved
Feature

Intuition: how cat-like is this square?

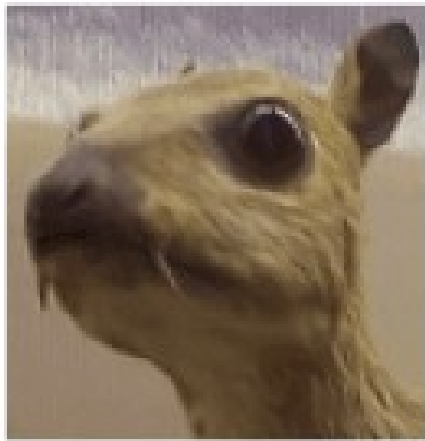
Convolution



Intuition: how cat-like is this square?

Convolution

Input image



Convolution
Kernel

$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Feature map

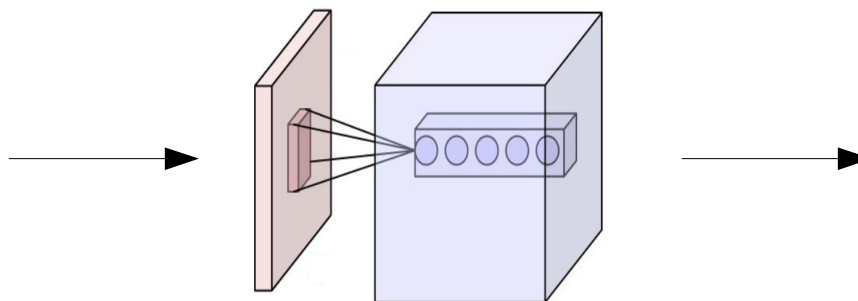


Intuition: how **edge-like** is this square?

Convolution



Image : 3 (RGB) x 100 px x 100 px

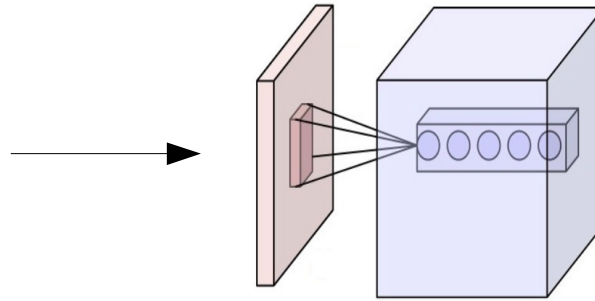


Filters: 100x(3x5x5)

Convolution



Image : 3 (RGB) x 100 px x 100 px

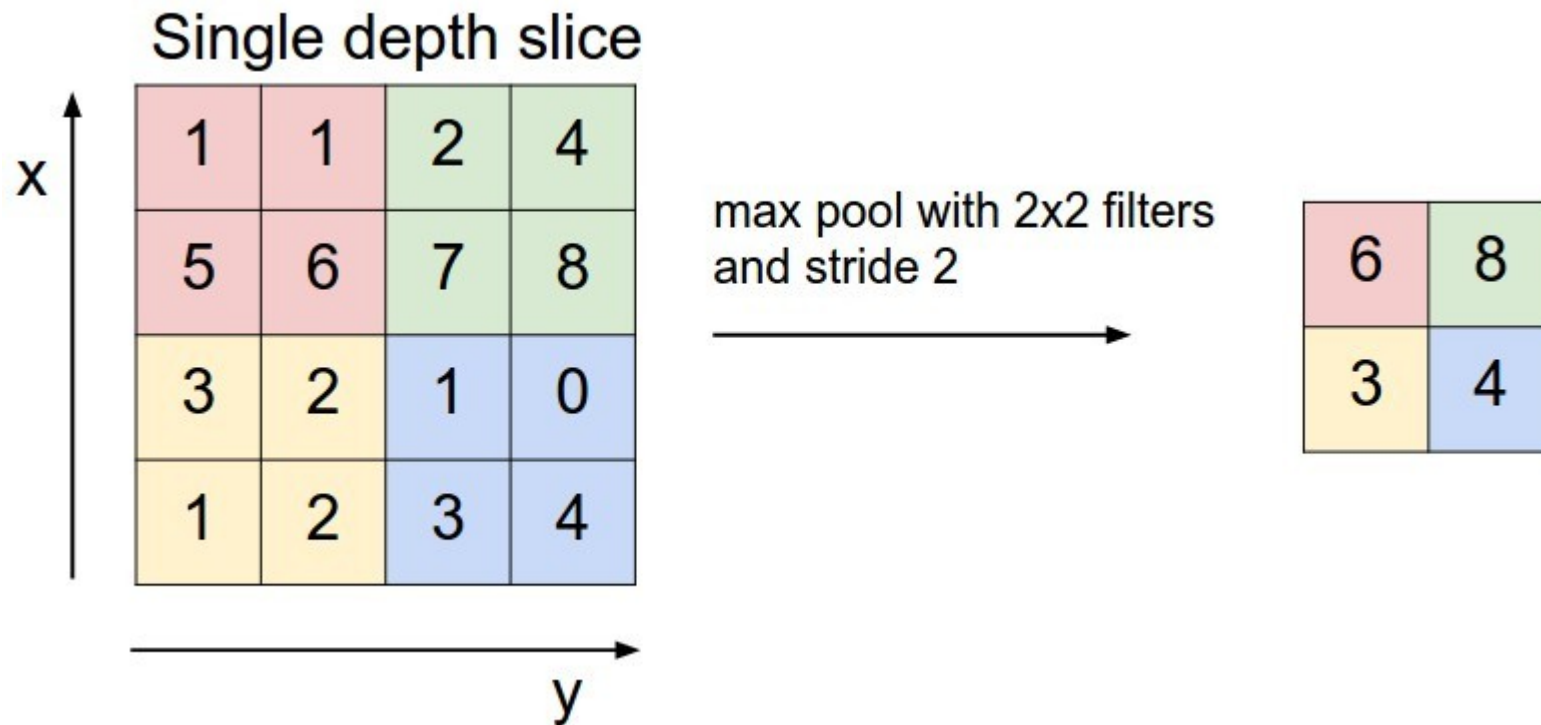


Filters: 100x(3x5x5)

100x96x96
~10⁶

Somewhat too many!

Pooling



Intuition: What is the max cat-likelihood over this area?

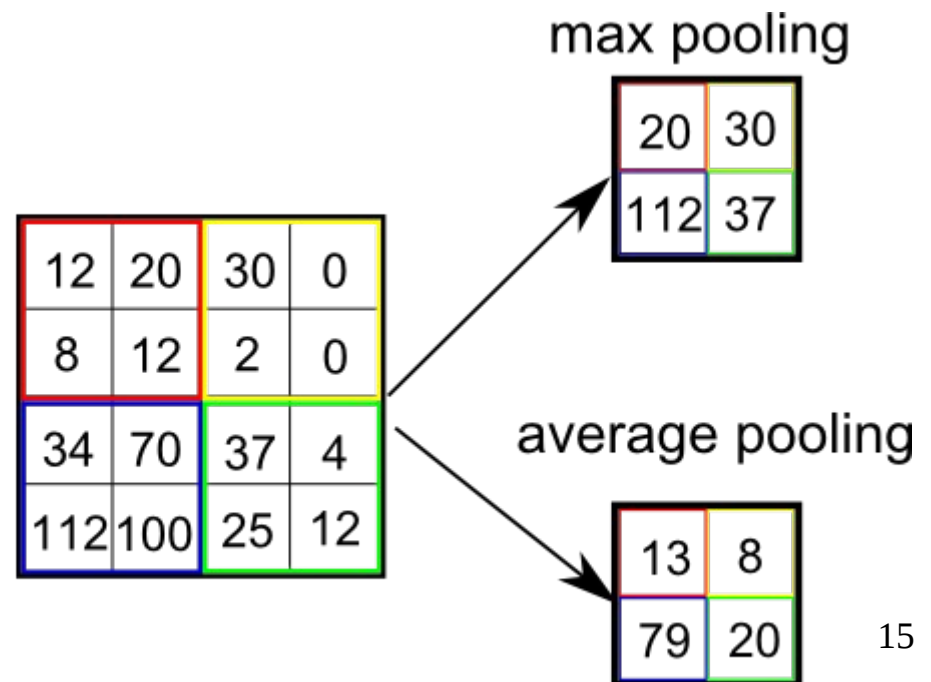
Pooling

Motivation:

- Reduce layer size by a factor
- Make NN less sensitive to small image shifts

Popular types:

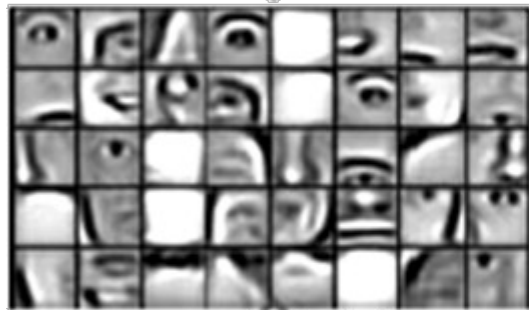
- Max
- Mean(average)



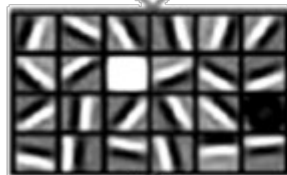


Discrete Choices

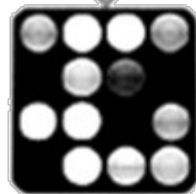
⋮



Layer 2 Features

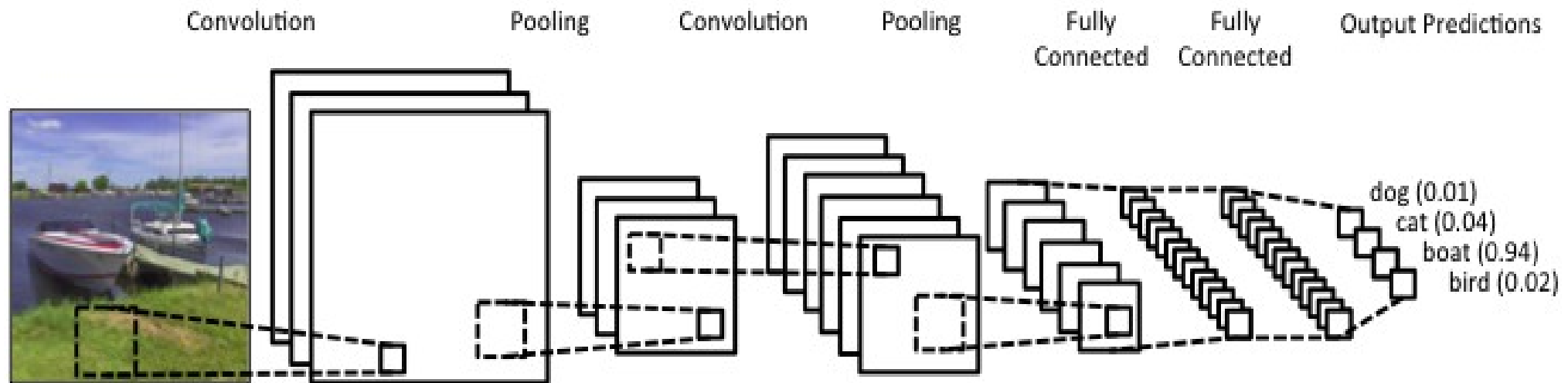


Layer 1 Features

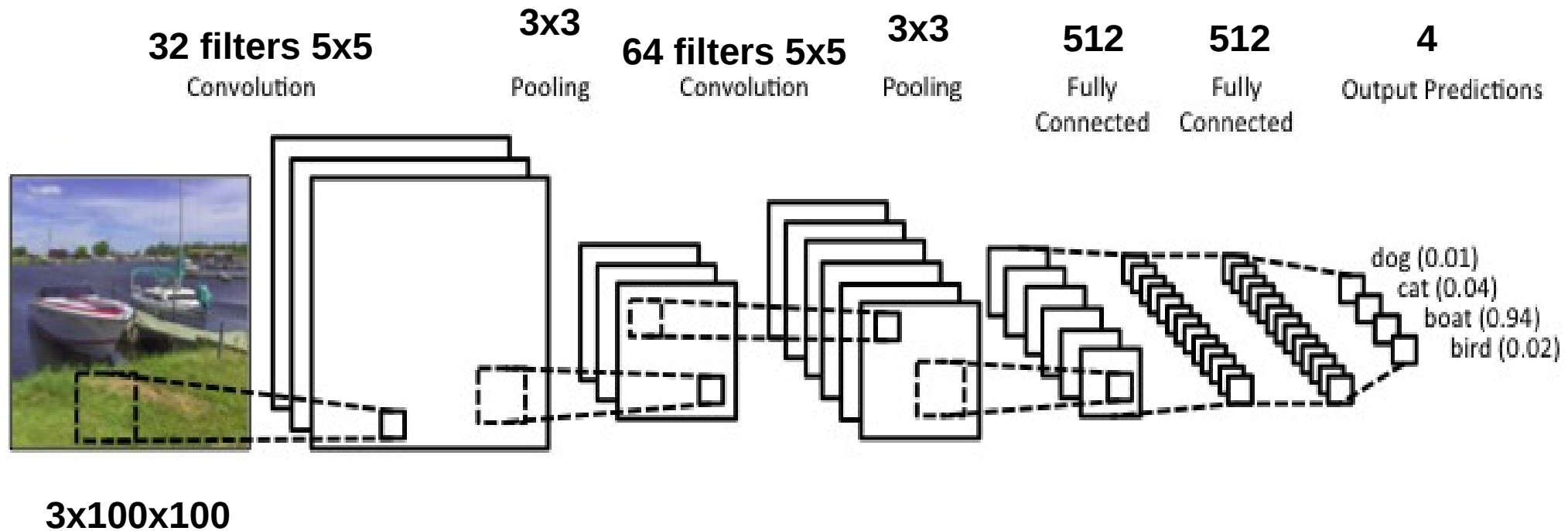


Original Data

Convolutional NNs



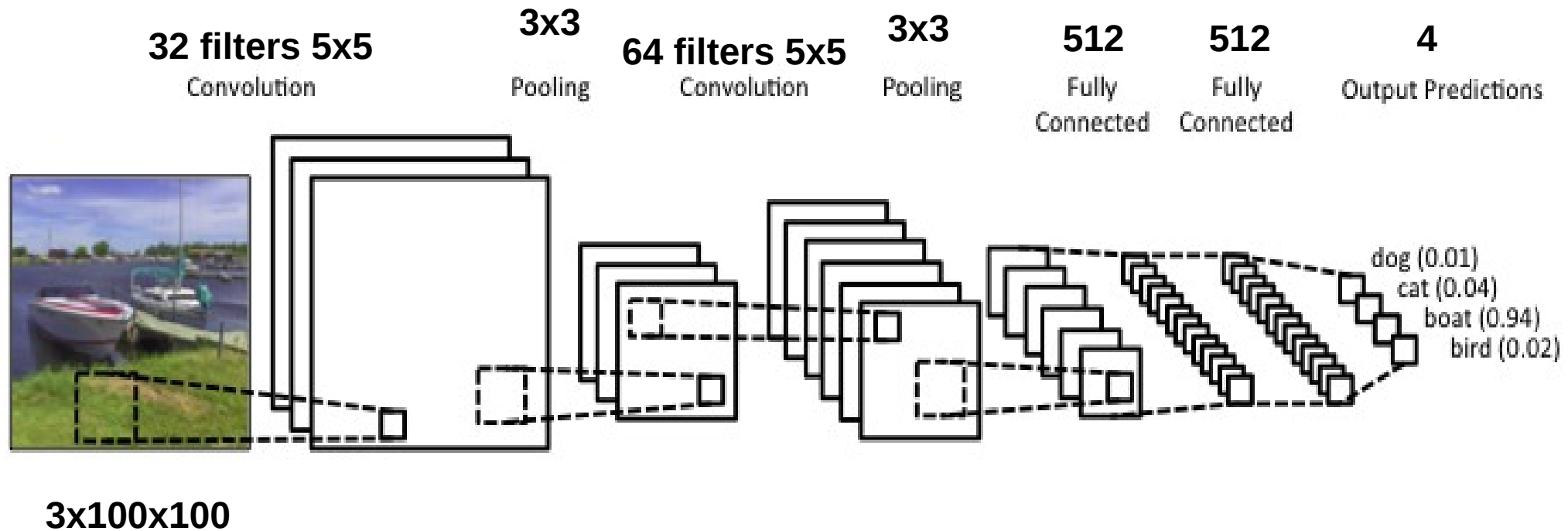
Convolutional NNs



Quiz:

1) What is the blob size **after second pooling**

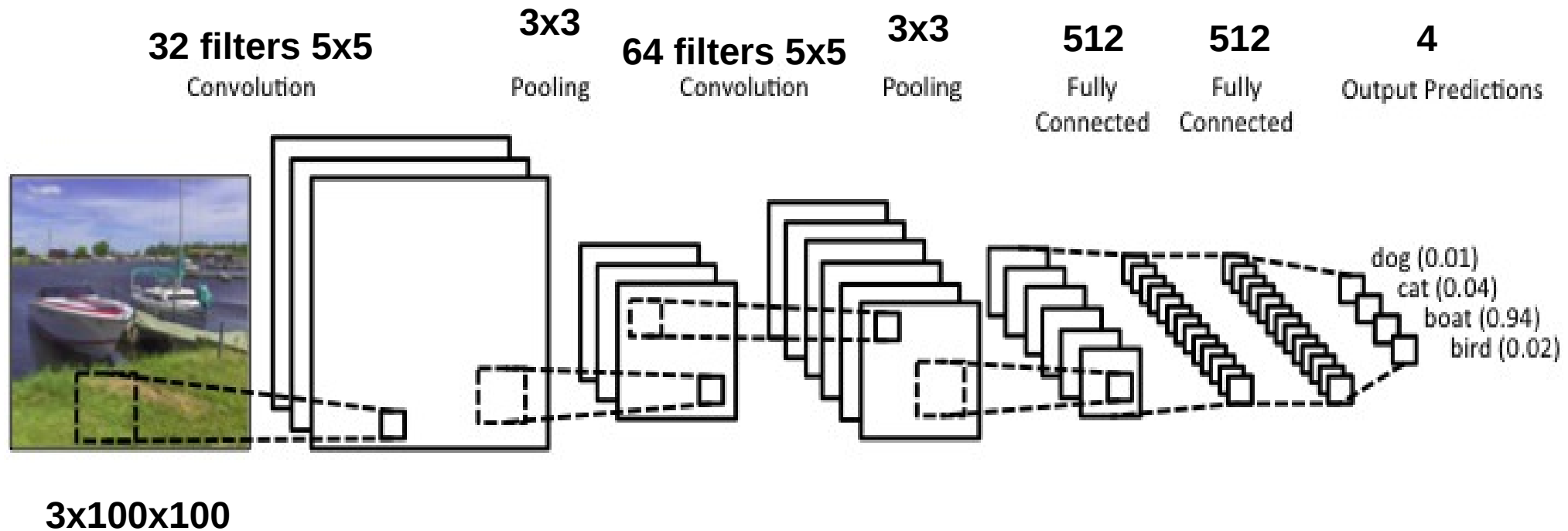
Convolutional NNs



Quiz:

2) How many image pixels does **one cell** after **second convolution** depend on?

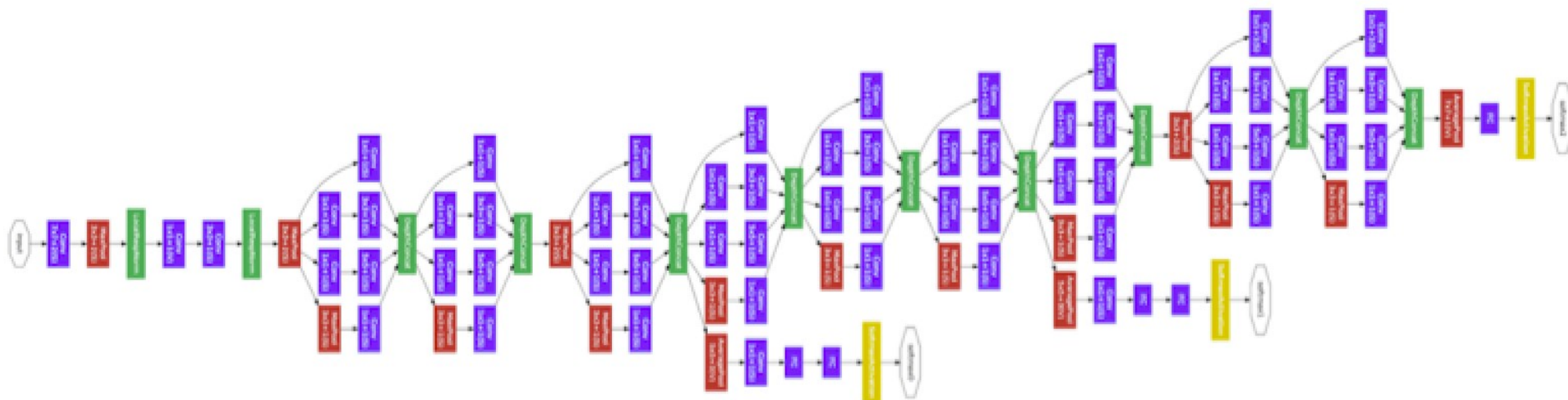
Convolutional NNs



Quiz:

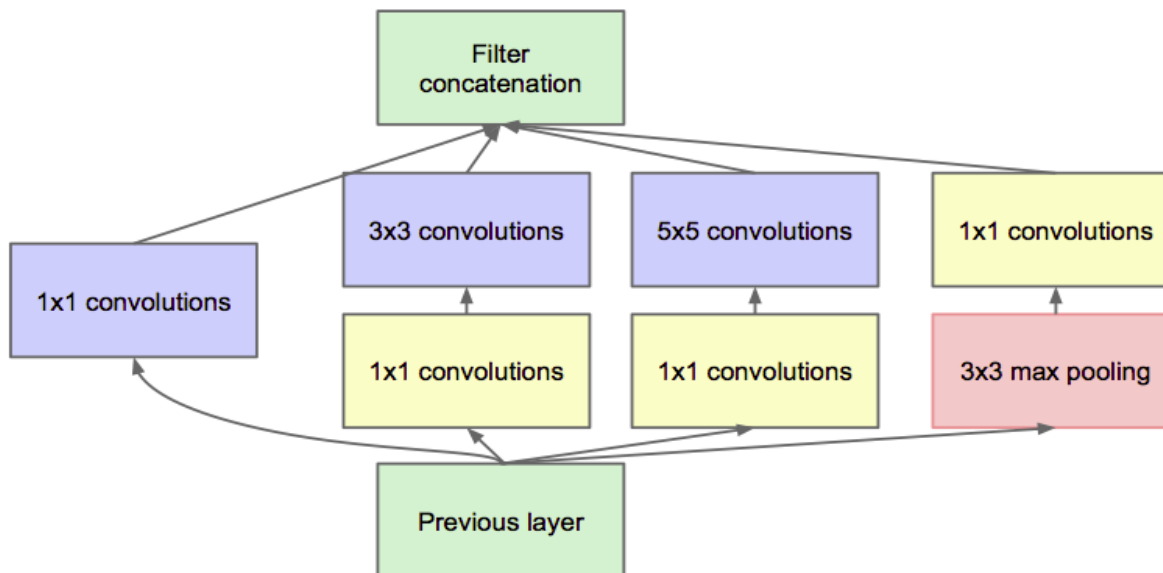
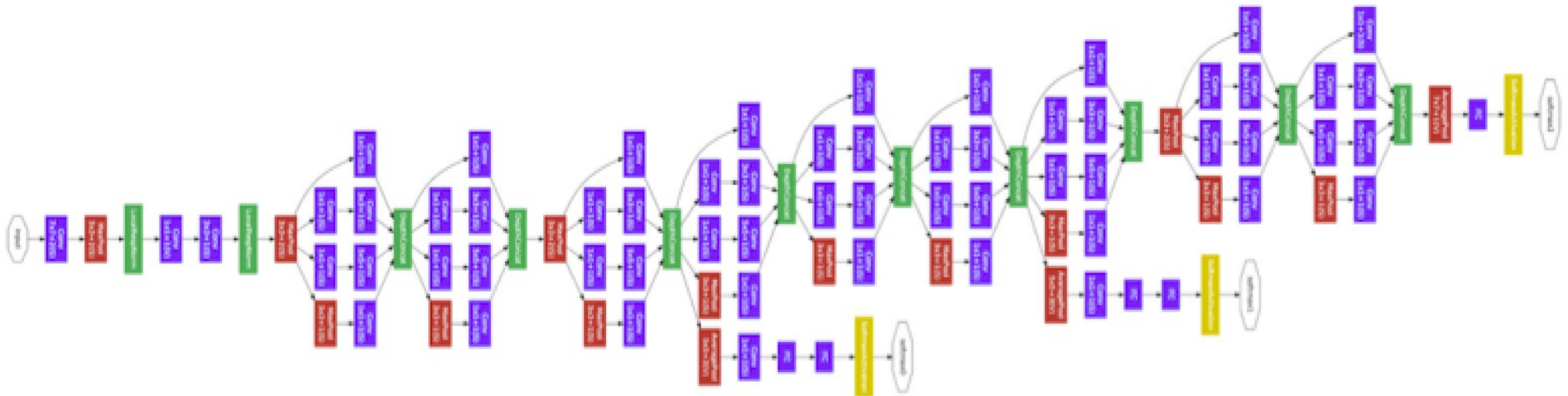
- 3) Which layer is hardest to compute?
- 4) Which layer has most independent parameters?

Inception-GoogLeNet



It is not a moon. It is a space station (c)

Inception-GoogLeNet

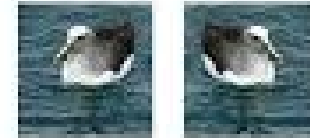


Convolution
Pooling
Softmax
Other

Data augmentation

- Idea: we can get N times more data by tweaking images.
- If you rotate a cat 15 degrees, it's still a cat.
- Rotate, crop, zoom, flip horizontally, add noise, etc.
- Sound data: add background noises

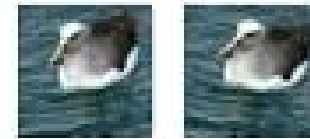
Horizontal Flip



Crop



Rotate



Batch normalization

Problem:

- Consider a neuron in any layer beyond first
- At each iteration we tune it's weights towards better loss function
- But we also tune it's inputs. Some of them become larger, some – smaller
- Now the neuron needs to be re-tuned for it's new inputs

Batch normalization

TL;DR:

- It's usually a good idea to normalize linear model inputs
 - Every data science lecturer, ever

Batch normalization

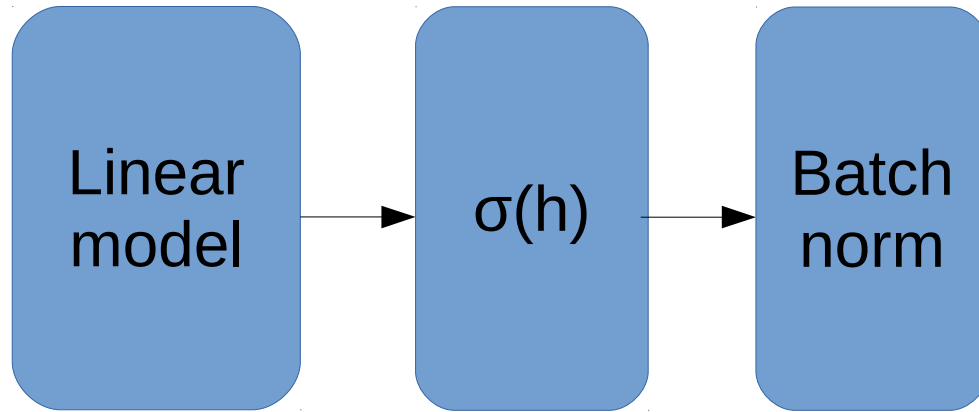
Idea:

- We normalize activation of a hidden layer
(zero mean unit variance)

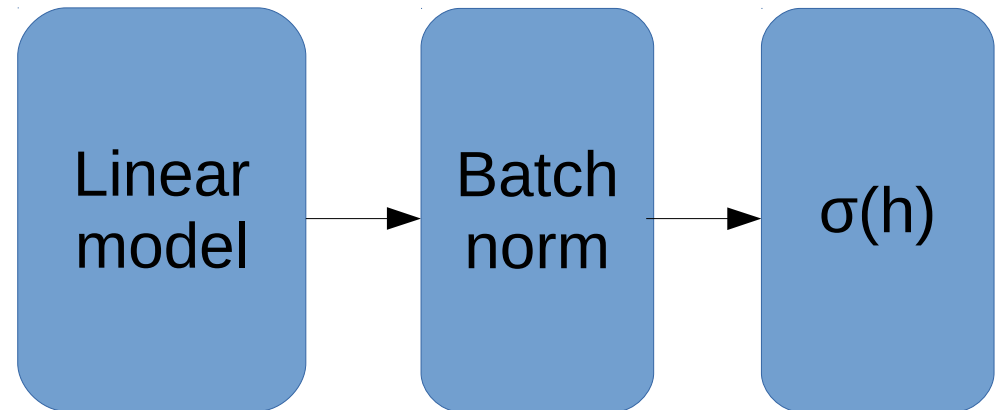
$$\hat{x}^k = \frac{x^k - E[x^k]}{\sqrt{Var[x^k]}}$$

- Now all neurons have ~same output range

Batch normalization



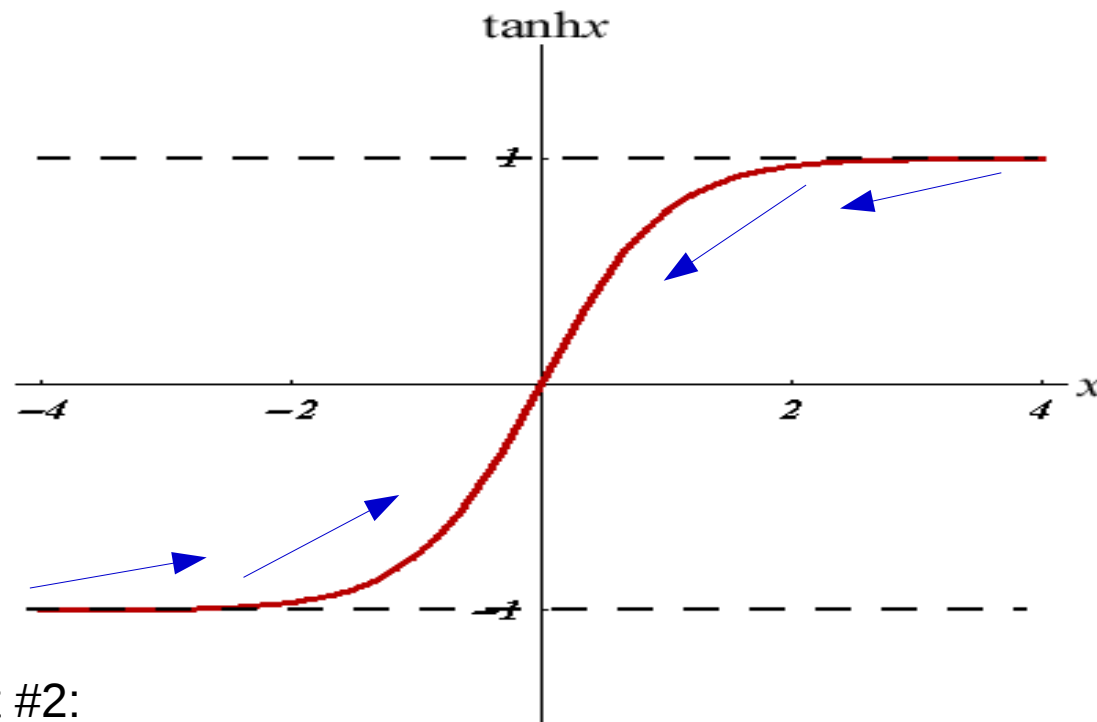
VS



Batch normalization

Good side effect #1:

- Vanishing gradient less a problem for sigmoid-like nonlinearities



Good side effect #2:

- We no longer need to train bias (+b term in $Wx+b$)

Other CV applications

to be covered in 3 weeks

Bounding box regression

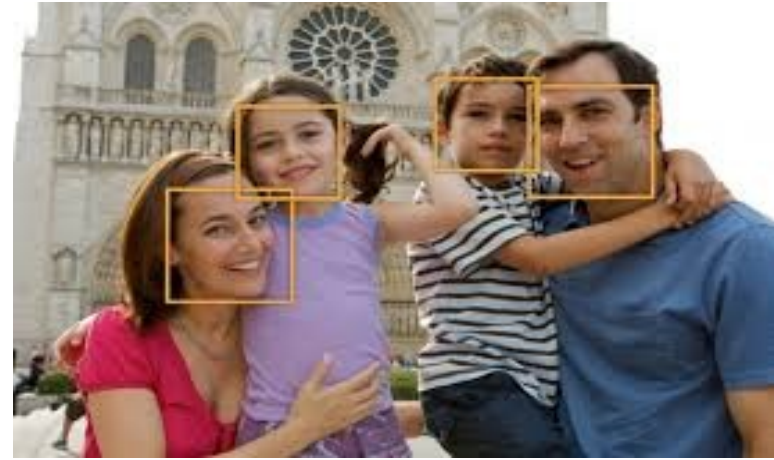
Predict object bounding box

(x_0, y_0, w, h)

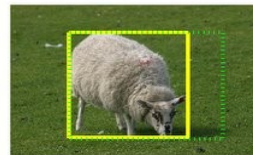
or several bounding boxes for multiple objects.

Applications examples:

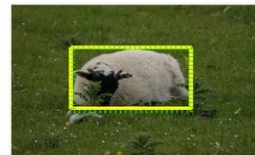
- Face detection @ cameras
- Surveillance cameras
- Self-driving cars



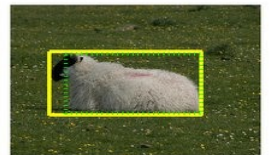
IM:"005194" Conf=0.835223



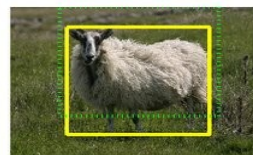
IM:"003538" Conf=0.829488



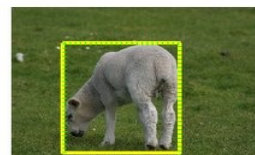
IM:"002810" Conf=0.801748



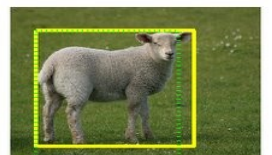
IM:"004522" Conf=0.799045



IM:"001064" Conf=0.797061



IM:"000819" Conf=0.794456



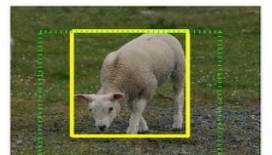
IM:"002306" Conf=0.789123



IM:"001956" Conf=0.788438



IM:"004285" Conf=0.782058



Segmentation

Predict class for each pixel
(fully-convolutional networks)

Applications examples:

- Moar surveillance
- Brain scan labeling
- Map labeling

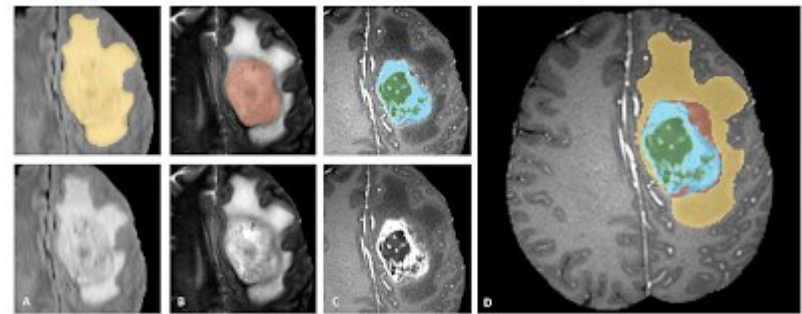
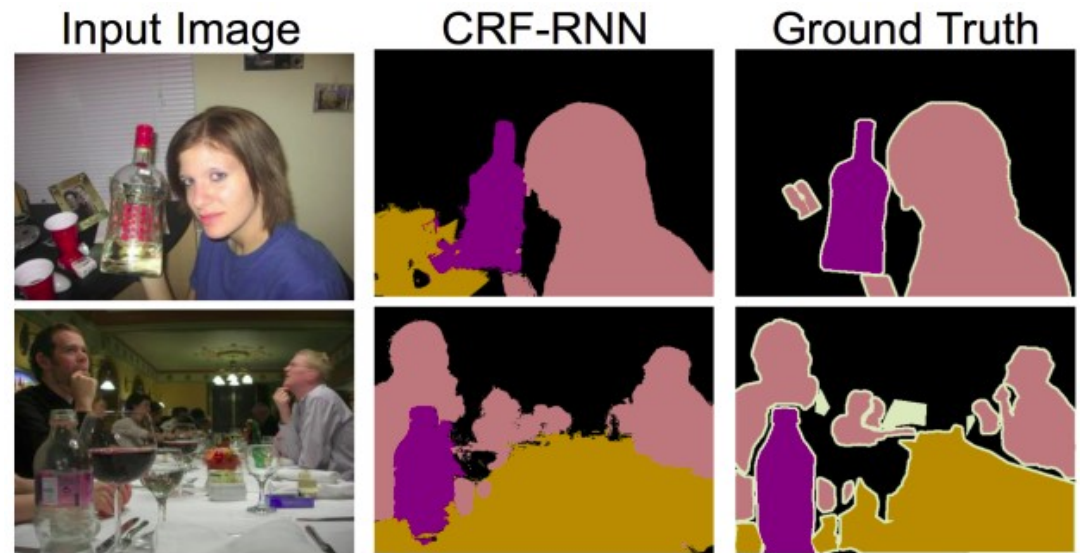
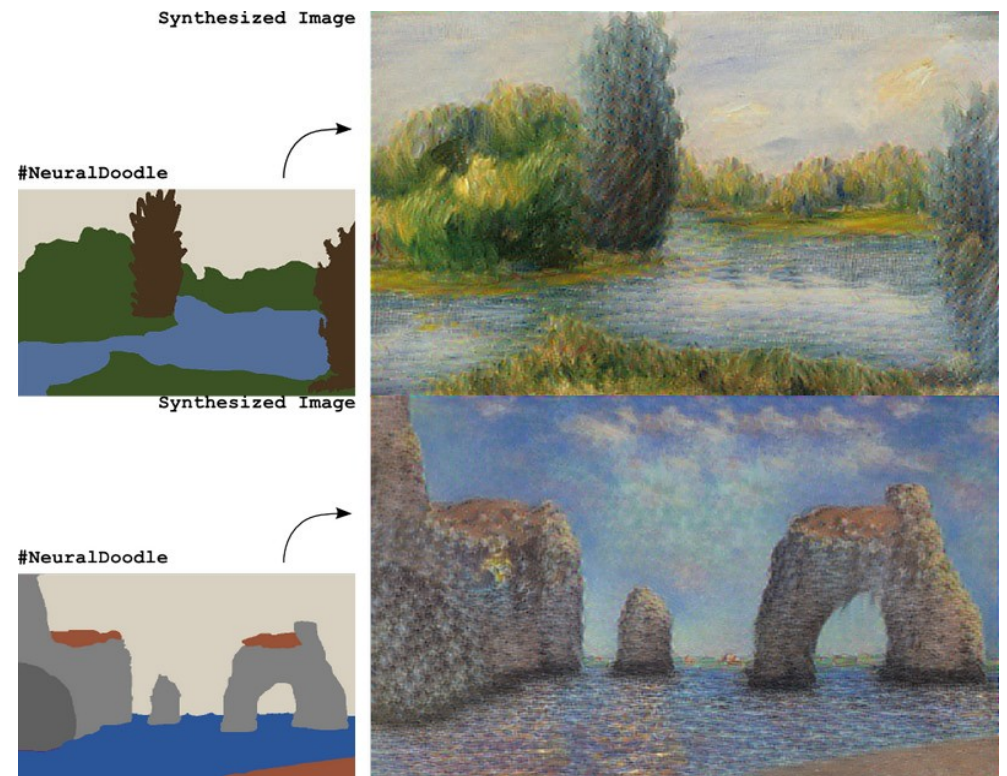


Image generation/transformation

- **Generation:** Given a set of reference images, learn to generate new images, resembling those you were given.
- **Transformation:** Given a set of reference images, learn to convert other images into ones resembling the reference set.



Neural Doodle
(D. Ulyanov et al.)

Image tagging
Image captioning
Image retrieval
Image encoding
Image morphing
Image encoding
Image upscaling
Object tracking on
video
Video processing
Video interpolation

Fine-tuning
Adversarial Networks
Variational Autoencoders
Knowledge transfer
Domain adaptation
Online learning
Explaining predictions
Soft targets
Scene reconstruction
3D object retrieval
Classifier optimization

Nuff

Let's train some CNNs!

