UCF Center for Research in Computer Vision

CAP5415
Computer Vision

Yogesh S Rawat

yogesh@ucf.edu

HEC-241

CENTER FOR RESEARCH IN COMPUTER VISION

Questions?

Introduction to Convolutional Neural Networks

Lecture 6

Agenda

- Overview
- Basics
- Fundamental operation
- Practical considerations
- Case study

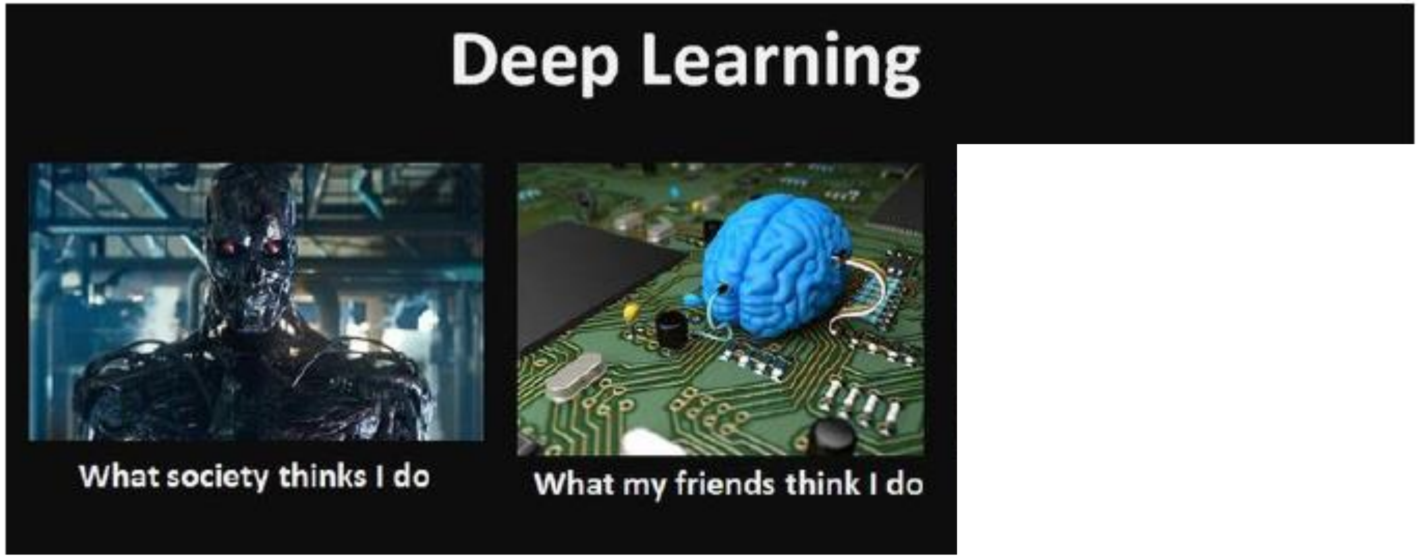
Introduction to Convolutional Neural Networks

Lecture 6

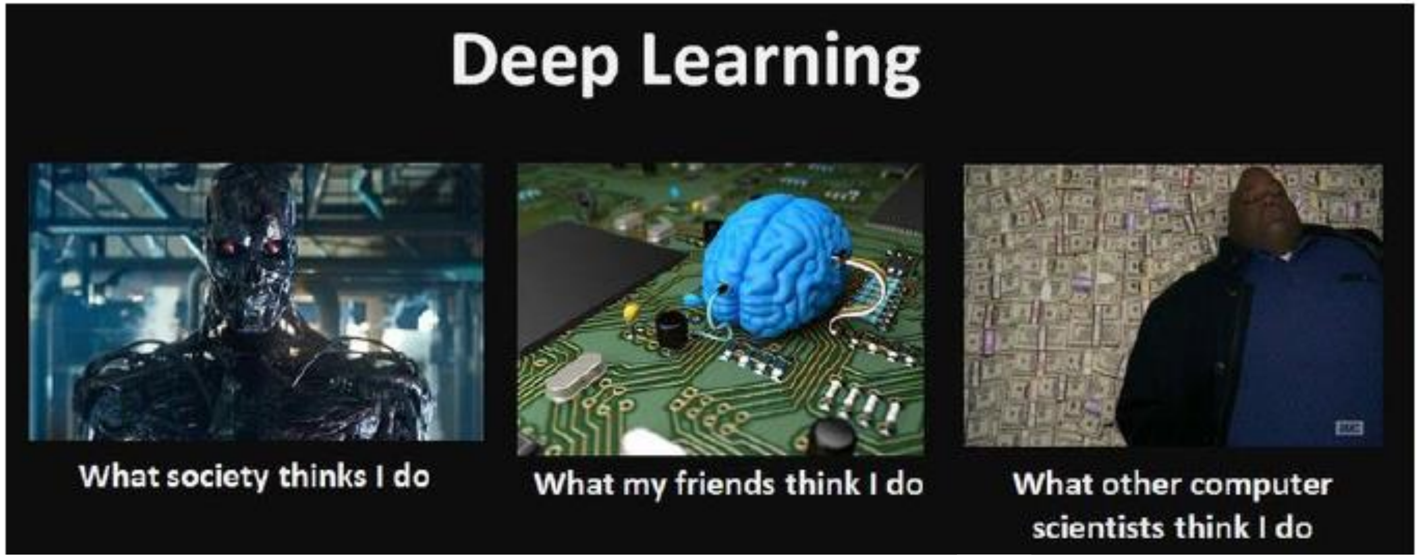
Overview

An interesting quote to cheer you up…

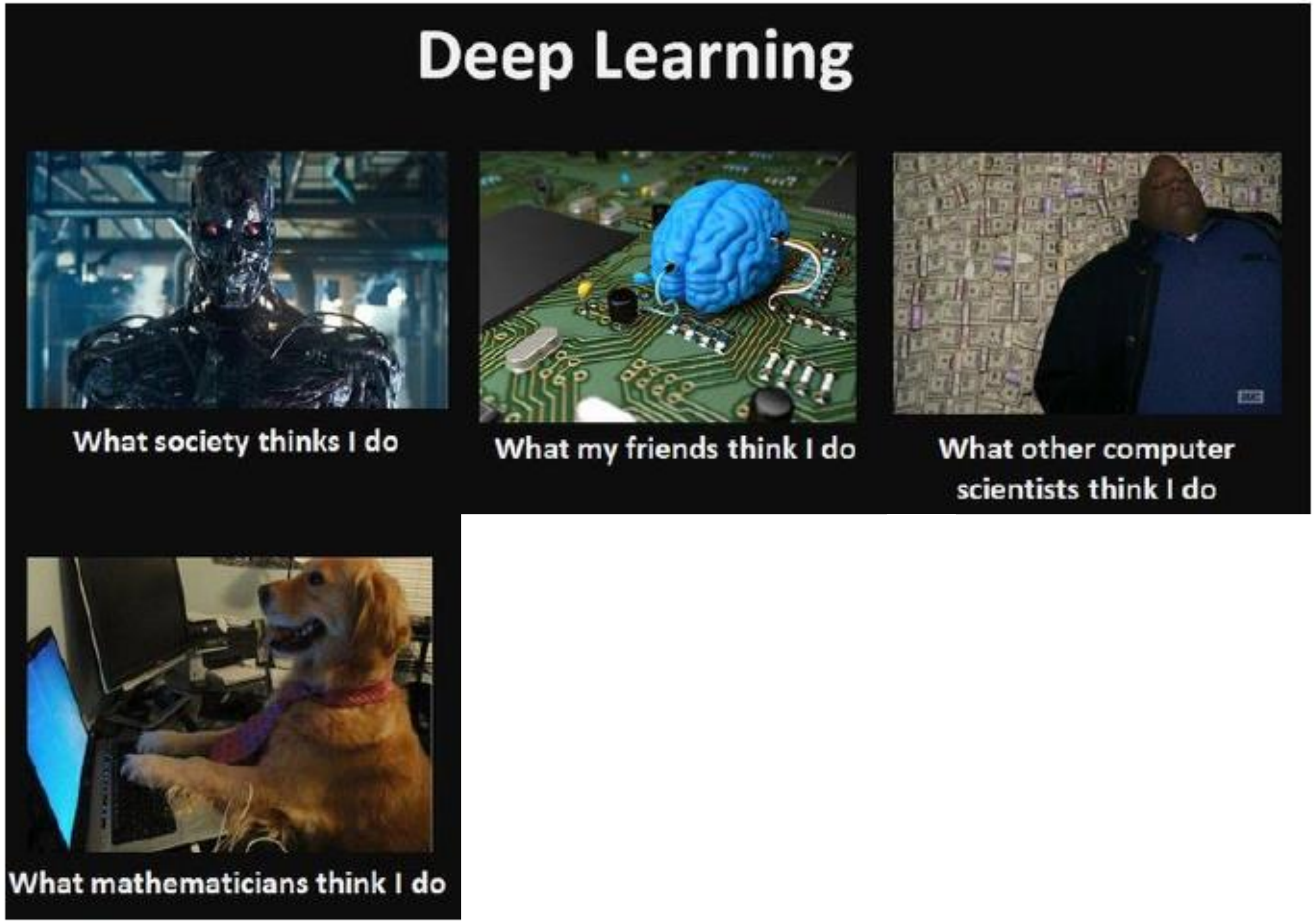
An interesting quote to cheer you up…



An interesting quote to cheer you up…



An interesting quote to cheer you up…



An interesting quote to cheer you up…

An interesting quote to cheer you up…

Generated image won art prize

Jason Allen’s A.I.-generated work, “Théâtre D’opéra Spatial,” took first place in the digital category at the Colorado State Fair.Credit...via Jason Allen

CNN – example: depth estimation

UCF Center for Research in Computer Vision

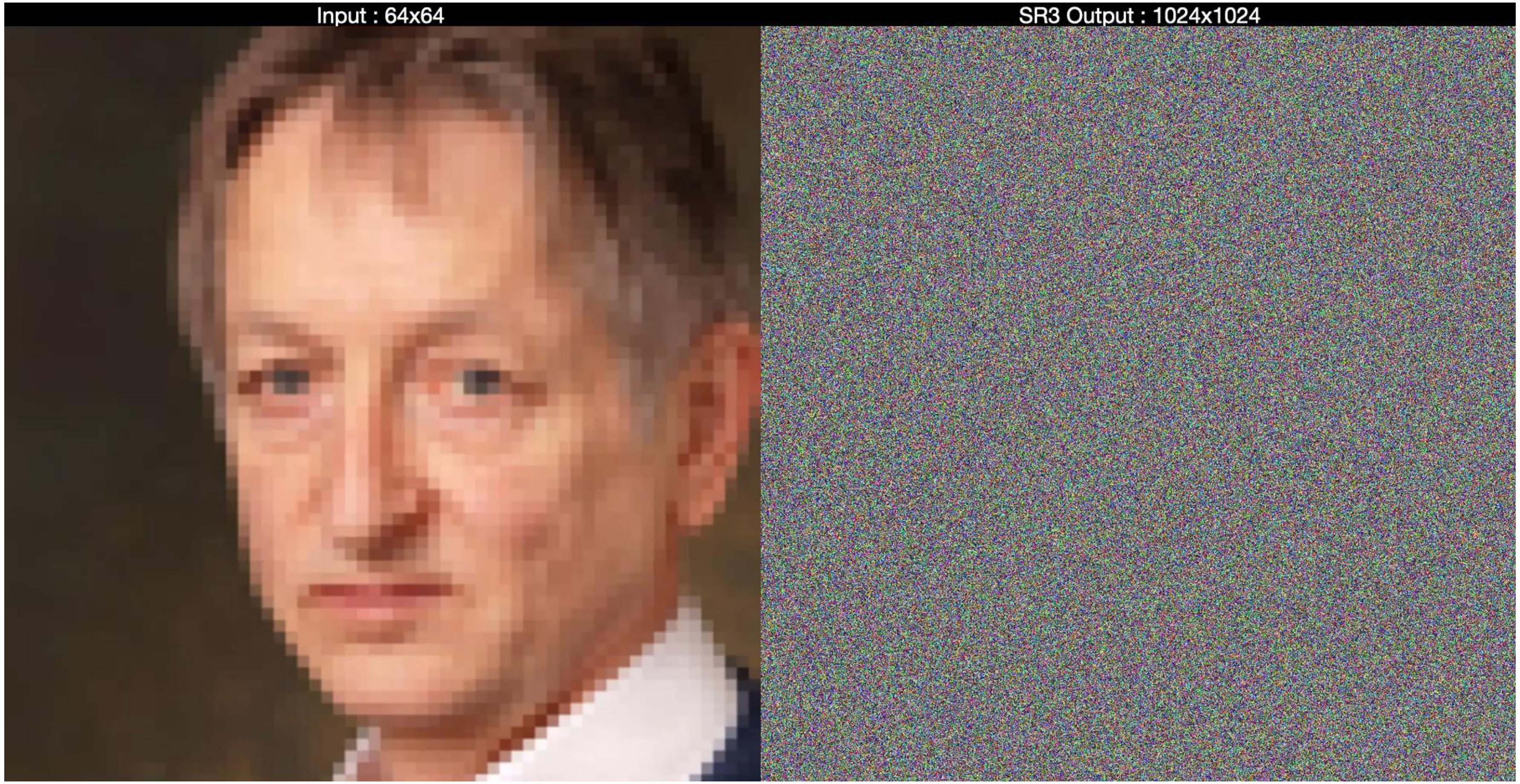
CNN – example: depth estimation

Li, Zhengqi, et al. "Learning the depths of moving people by watching frozen people." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

CNN – example: depth estimation

Li, Zhengqi, et al. "Learning the depths of moving people by watching frozen people." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019.

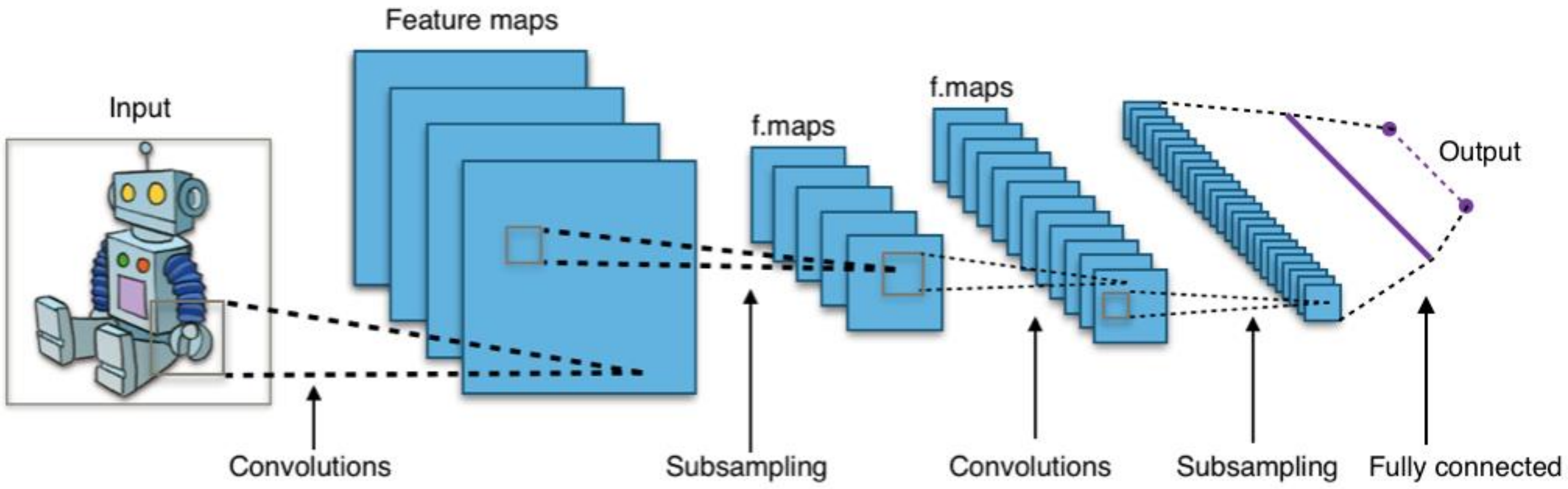
Super-resolution



https://ai.googleblog.com/2021/07/high-fidelity-image-generation-using.html?m=1

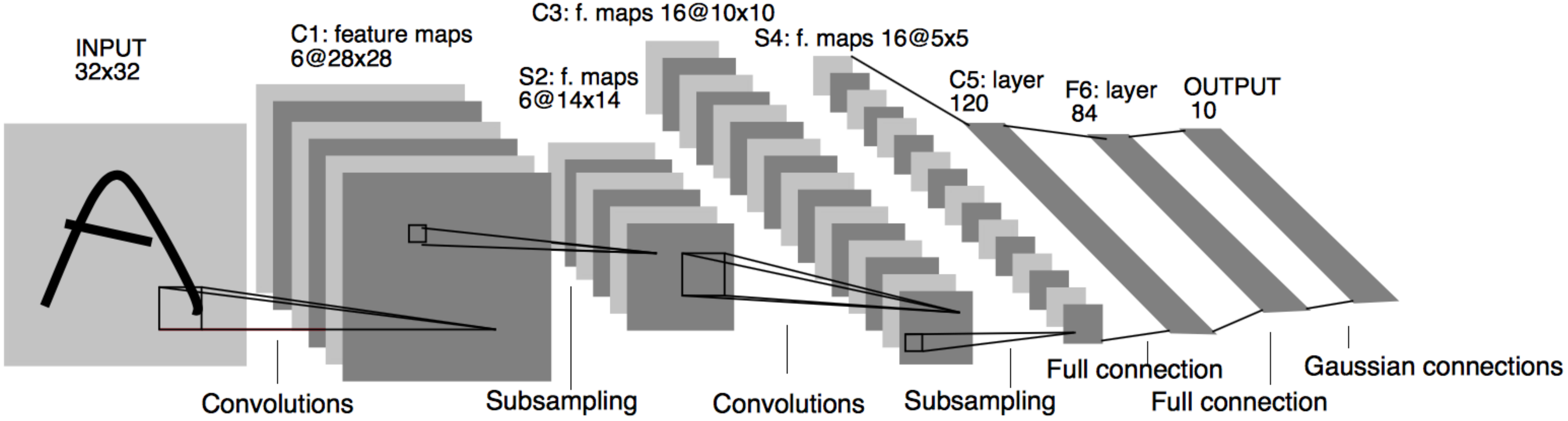
Convolutional Neural Network (CNN)

- A class of Neural Networks
  - Takes image as input (mostly)
  - Make predictions about the input image



History

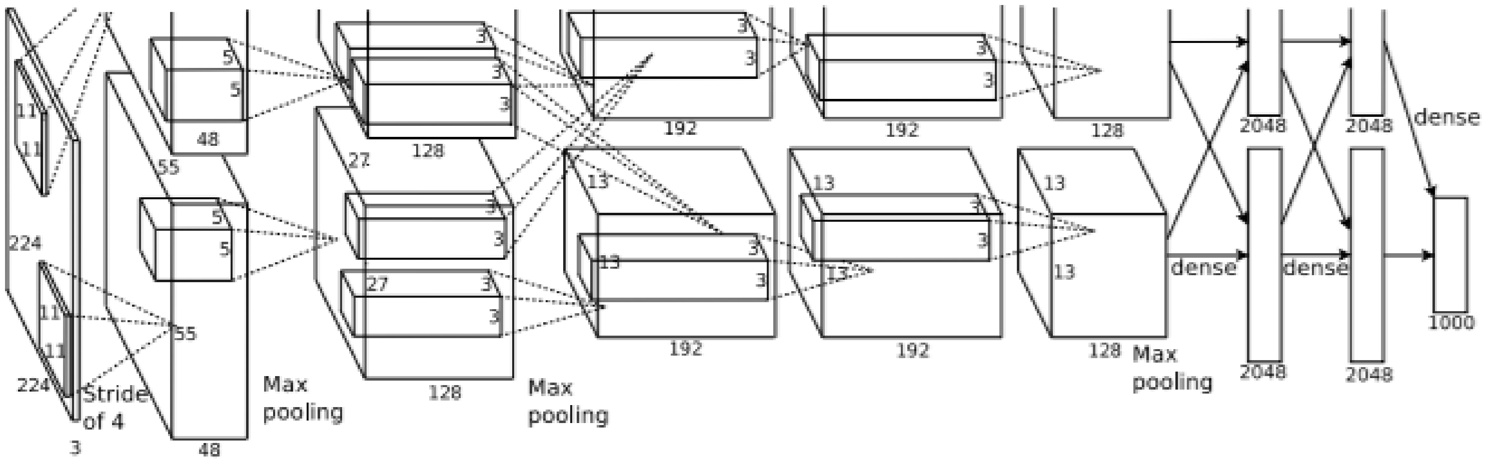
- The LeNet architecture (1990s)



Gradient-based learning applied to document recognition
LeCun Y, Bottou L, Bengio Y, Haffner P. Proceedings of the IEEE. 1998

First Strong Results

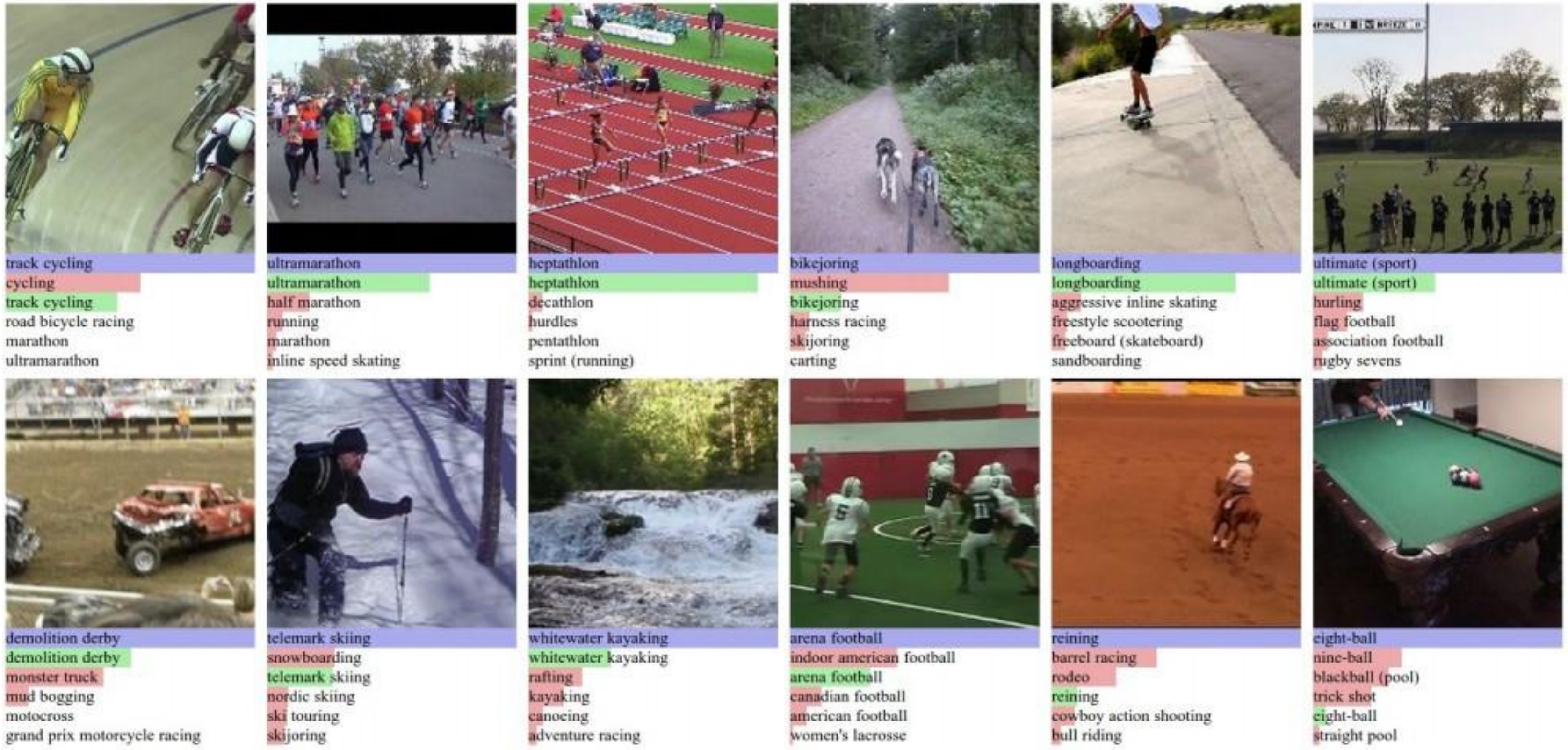
- AlexNet 2012
  - Winner of ImageNet Large-Scale Visual Recognition Challenge (ILSVRC 2012)
  - Error rate – 15.4% (the next best entry was at 26.2%)



Imagenet classification with deep convolutional neural networks
Alex Krizhevsky, Ilya Sutskever, Geoffrey E Hinton, 2012

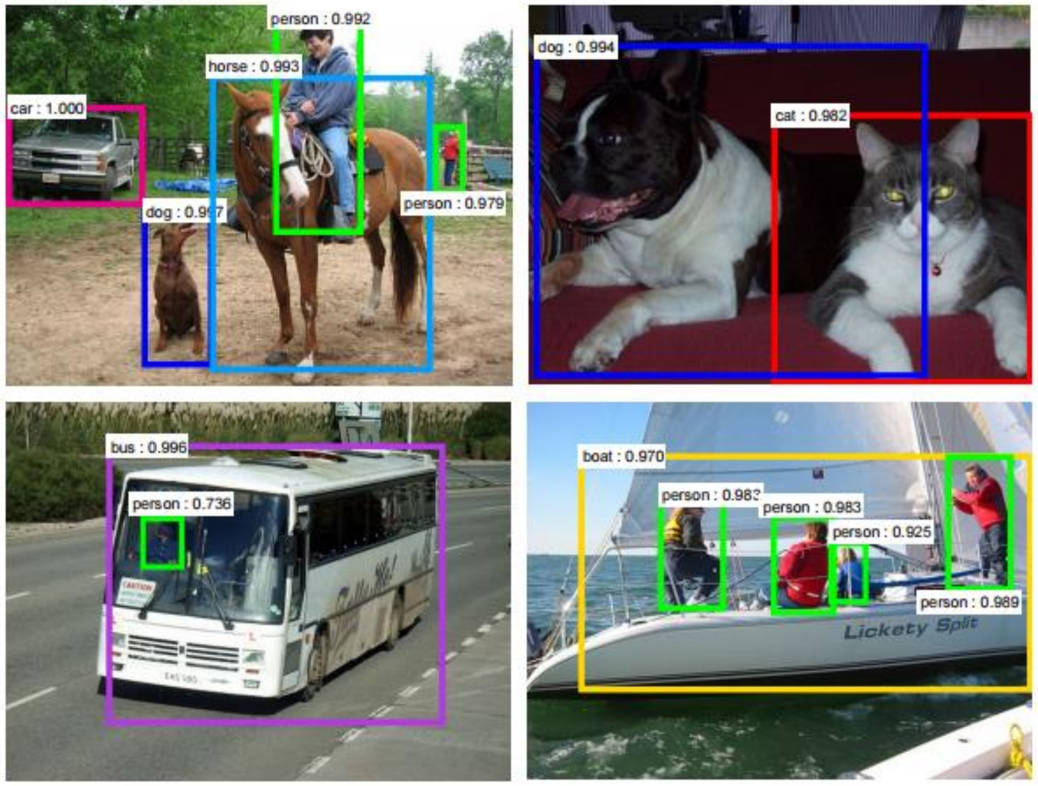
Today: CNNs are everywhere

Classification



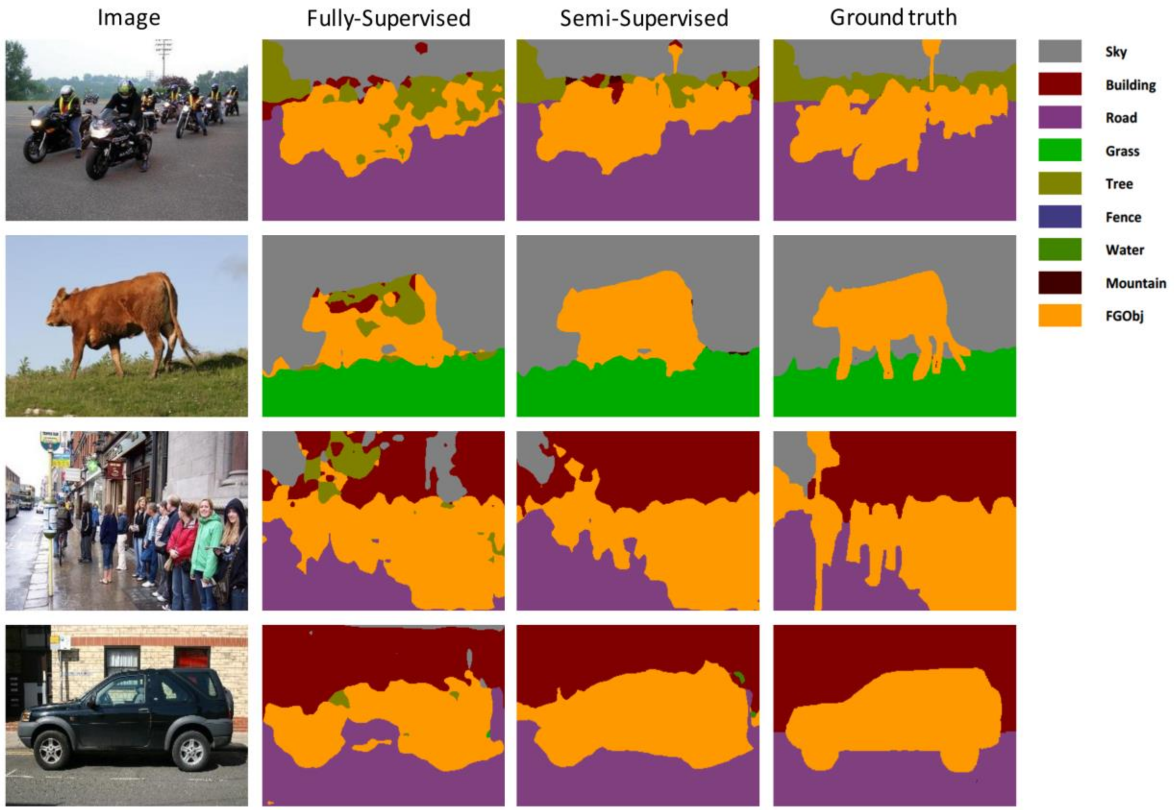
Today: CNNs are everywhere

Object detection



Faster R-CNN: Ren, He, Girshick, Sun 2015

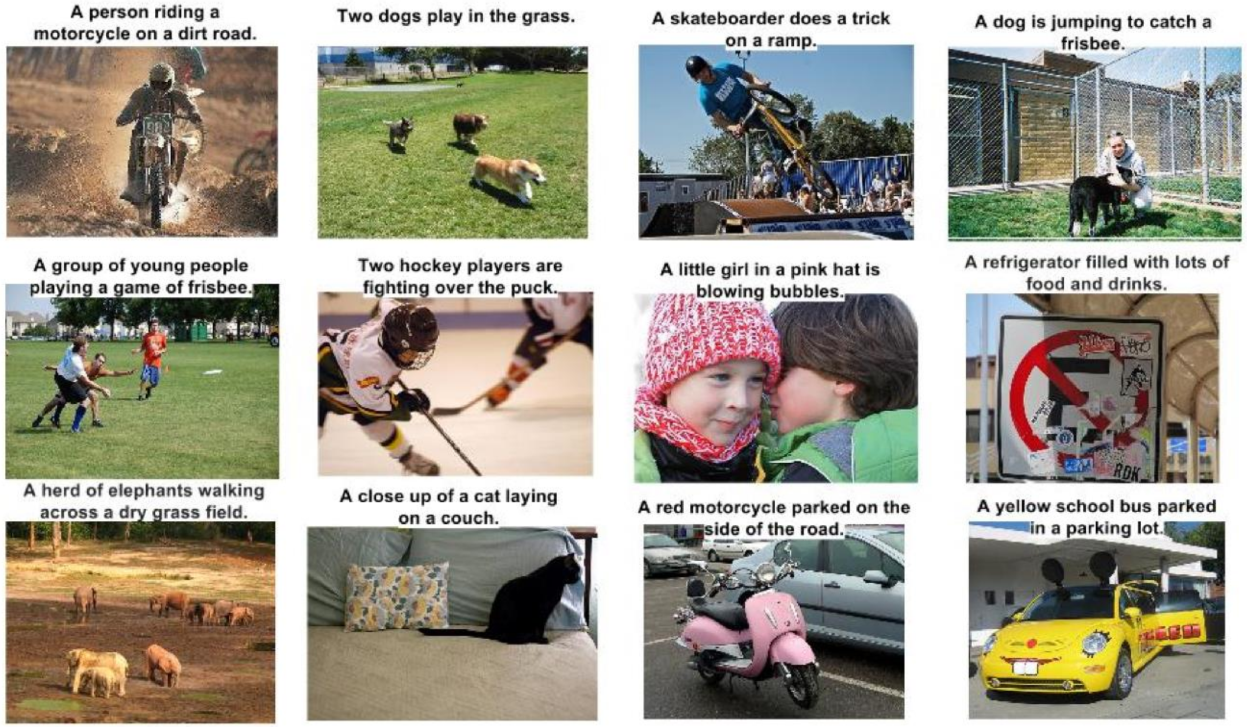
Semantic Segmentation



Semantic Segmentation Using GAN, Nasim, Concetto, and Mubarak, 2017.

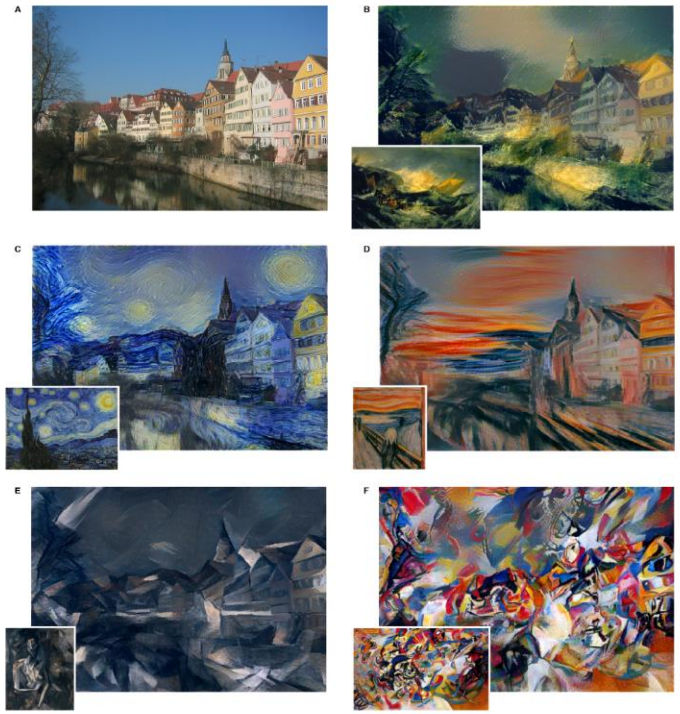
Today: CNNs are everywhere

Image captioning



"Show and tell: A neural image caption generator." Vinyals, Oriol, et al. CVPR 2015.

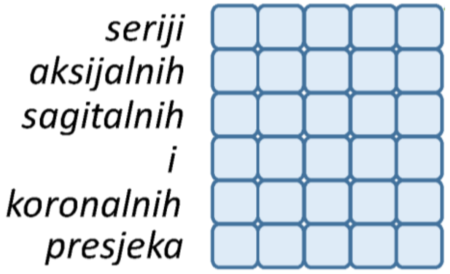
Style transfer

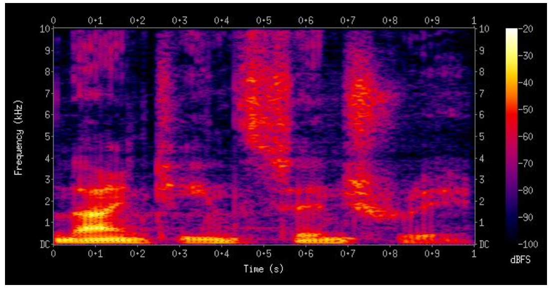


A Neural Algorithm of Artistic Style L. Gatys et al. 2015).

CNN – Not just images

- Natural Language Processing (NLP)
  - Text classification
  - Word to vector
- Audio Research
  - Speech recognition
  - Can be represented as spectrograms
- Converting data to a matrix (2-D) format
  - 1D convolution – Audio, EEG, etc.
  - 3D convolution - Videos





Questions?

Introduction to Convolutional Neural Networks

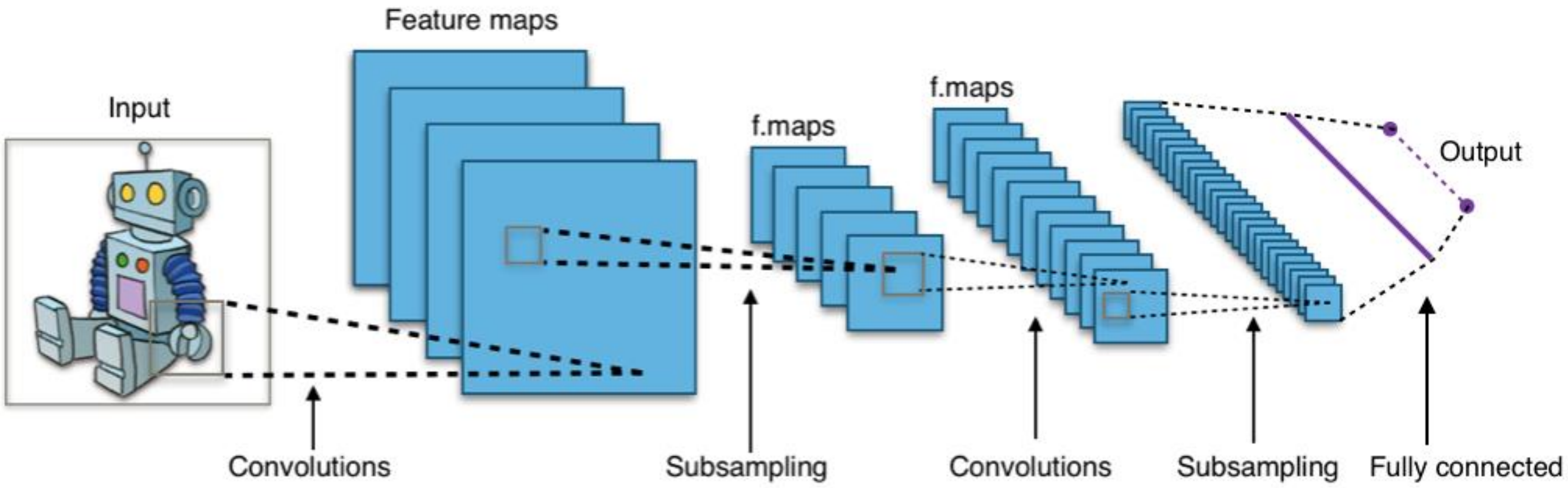
Lecture 6

Basics

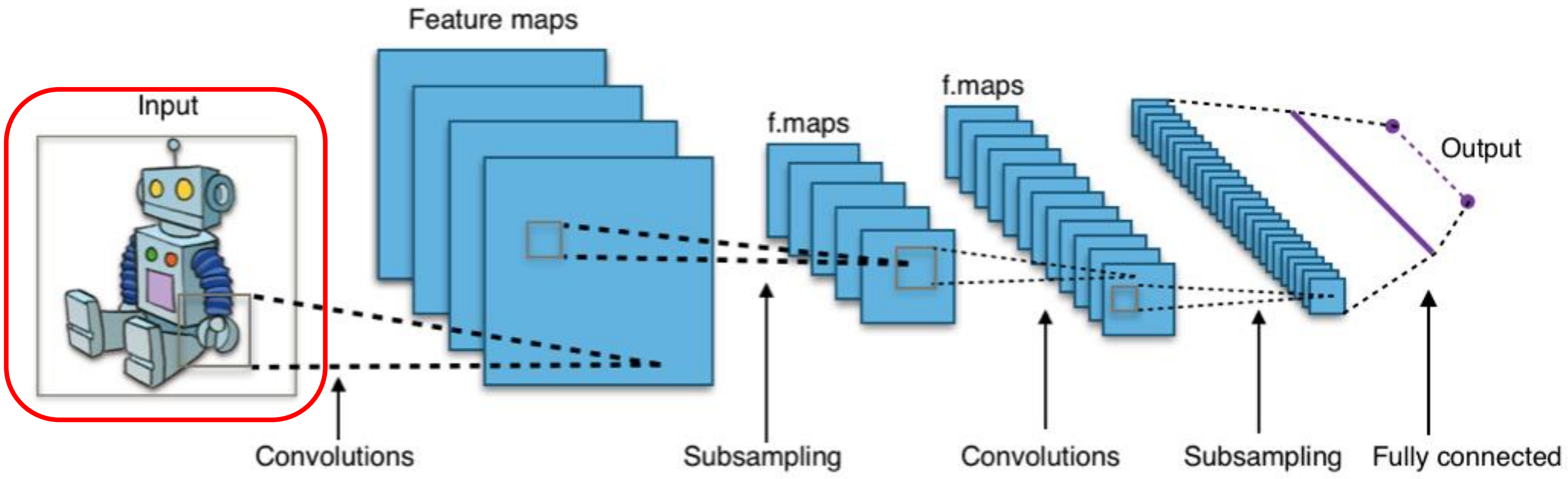
Background

What we already know!

General CNN architecture

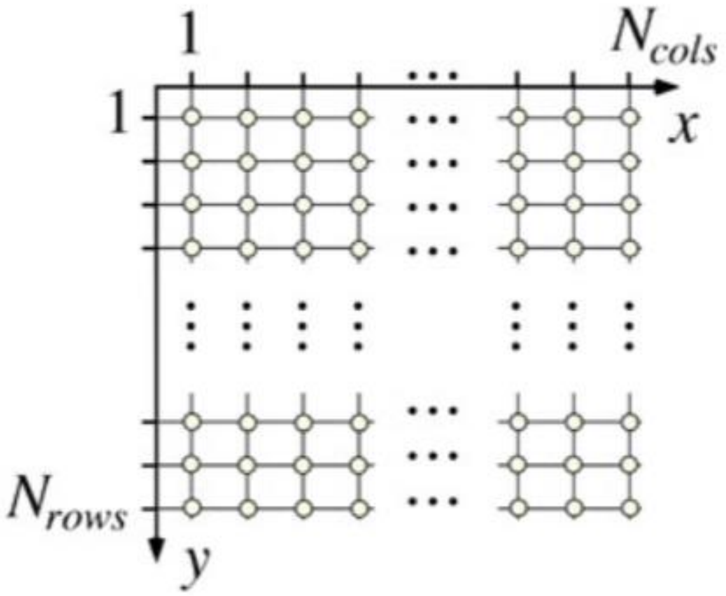


General CNN architecture



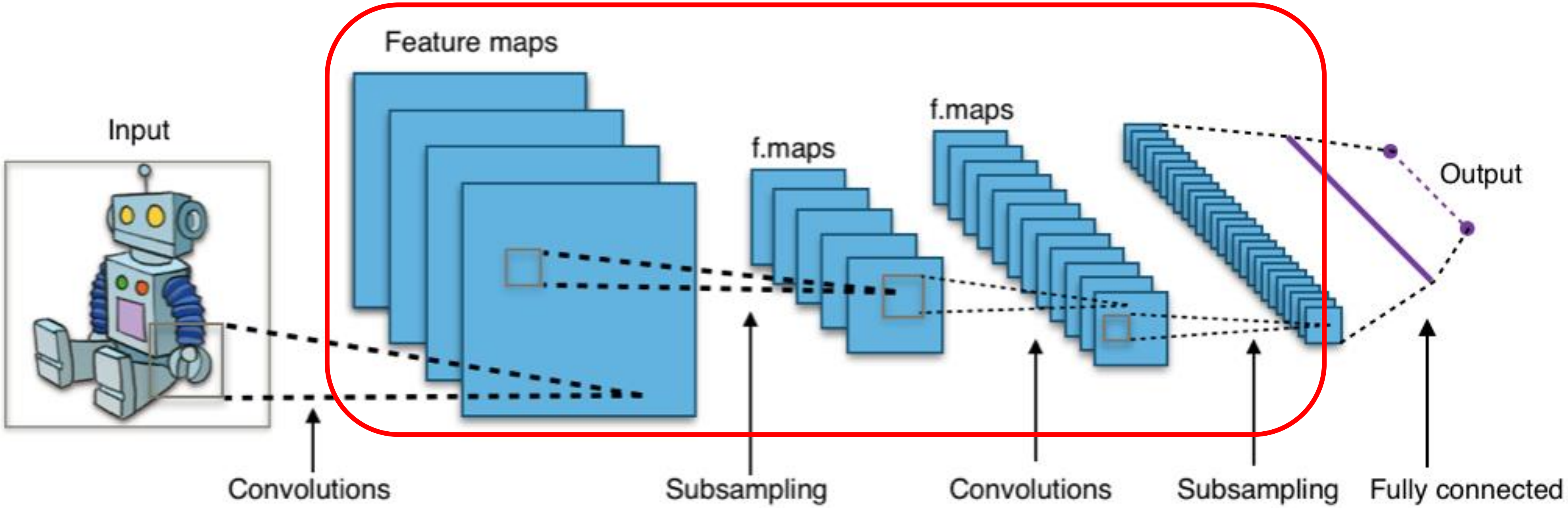
What is a (digital) Image? - recap

- Definition: A digital image is defined by *integrating* and *sampling* continuous (analog) data in a spatial domain [Klette, 2014].



Left hand coordinate system

General CNN architecture



Filtering - recap

- Image filtering: compute function of local neighborhood at each position

h=output f=filter I=image

$$h[m,n]=\sum_{k,l} f[k,l]\, I[m+k,n+l]$$

2d coords=k,l 2d coords=m,n

$$[\quad]\qquad [\;]\qquad [\quad]$$

Filtering - recap

- Output is linear combination of the neighborhood pixels

1	3	0
2	10	2
4	1	1

Image

$\otimes$

1	0	-1
1	0.1	-1
1	0	-1

Kernel

$=$

	5	

Filter Output

Correlation (linear relationship) - recap

$$f \otimes h = \sum_{k} \sum_{l} f(k,l)\,h(k,l)$$

$$f = \text{Image}$$

$$h = \text{Kernel}$$

$$f \qquad\qquad\qquad h$$

$$\begin{array}{|c|c|c|}\hline f_1 & f_2 & f_3 \\ \hline f_4 & f_5 & f_6 \\ \hline f_7 & f_8 & f_9 \\ \hline \end{array} \otimes \begin{array}{|c|c|c|}\hline h_1 & h_2 & h_3 \\ \hline h_4 & h_5 & h_6 \\ \hline h_7 & h_8 & h_9 \\ \hline \end{array} \rightarrow$$

$$\begin{aligned} f \otimes h = {} & f_1 h_1 + f_2 h_2 + f_3 h_3 \\ & + f_4 h_4 + f_5 h_5 + f_6 h_6 \\ & + f_7 h_7 + f_8 h_8 + f_9 h_9 \end{aligned}$$

Convolution – recap

$$f*h=\sum_k\sum_l f(k,l)\,h(-k,-l)$$

f = Image

h = Kernel

h

h_1	h_2	h_3
h_4	h_5	h_6
h_7	h_8	h_9

$X-flip$

h_7	h_8	h_9
h_4	h_5	h_6
h_1	h_2	h_3

$Y-flip$

f

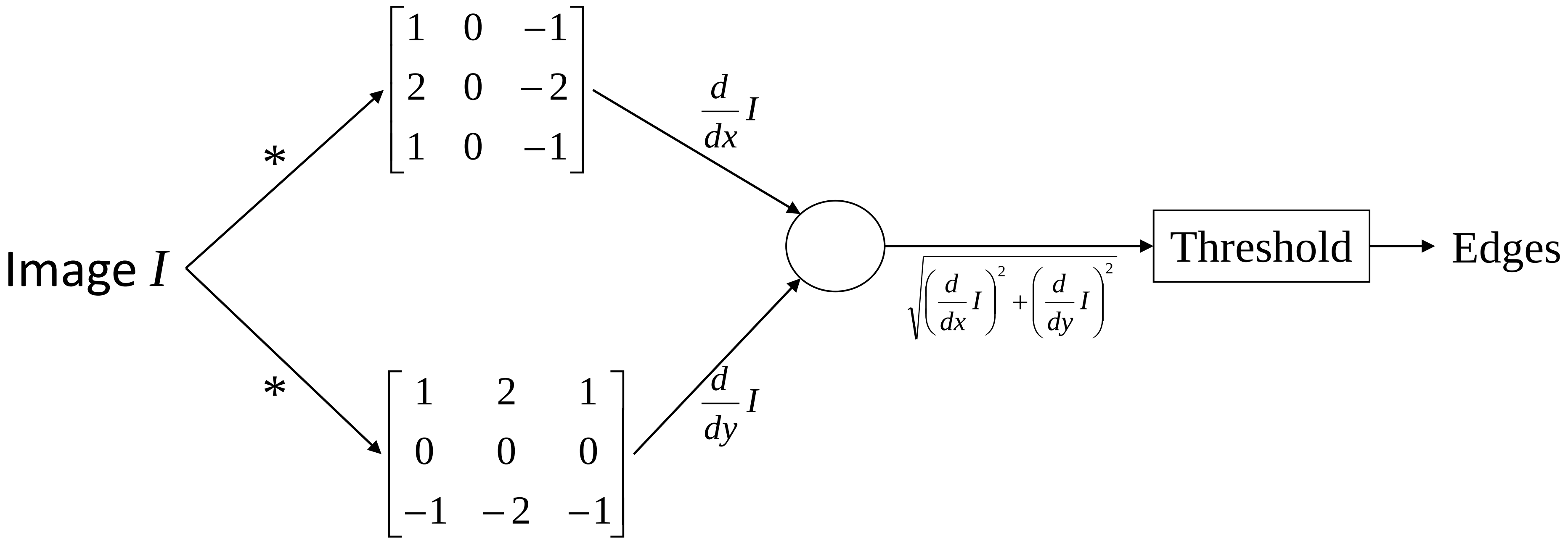
f_1	f_2	f_3
f_4	f_5	f_6
f_7	f_8	f_9

$*$

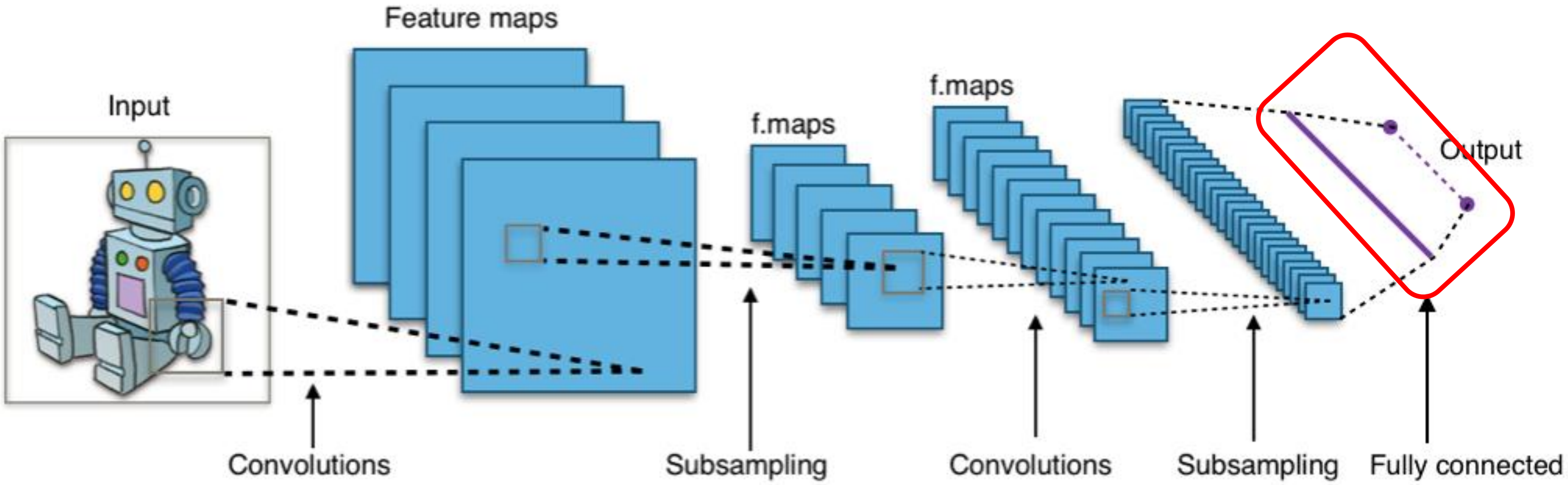
h_9	h_8	h_7
h_6	h_5	h_4
h_3	h_2	h_1

$$\begin{aligned} f*h = {} & f_1h_9+f_2h_8+f_3h_7 \\ & + f_4h_6+f_5h_5+f_6h_4 \\ & + f_7h_3+f_8h_2+f_9h_1 \end{aligned}$$

Sobel Edge Detector

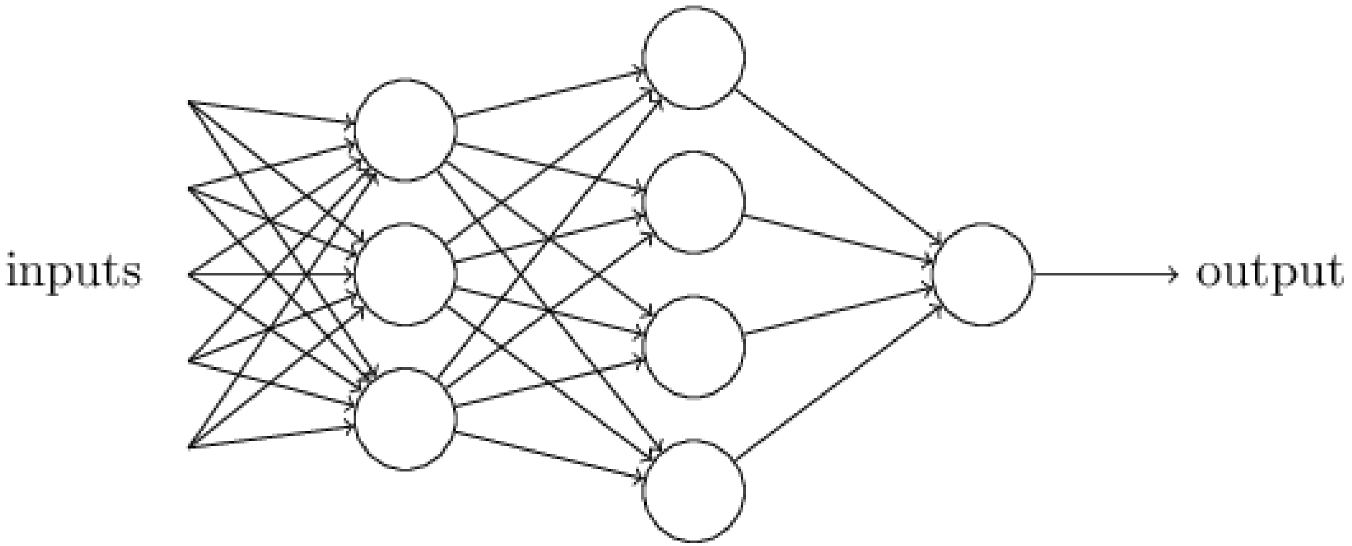


General CNN architecture



Multi-layer perceptron (MLP) – recap

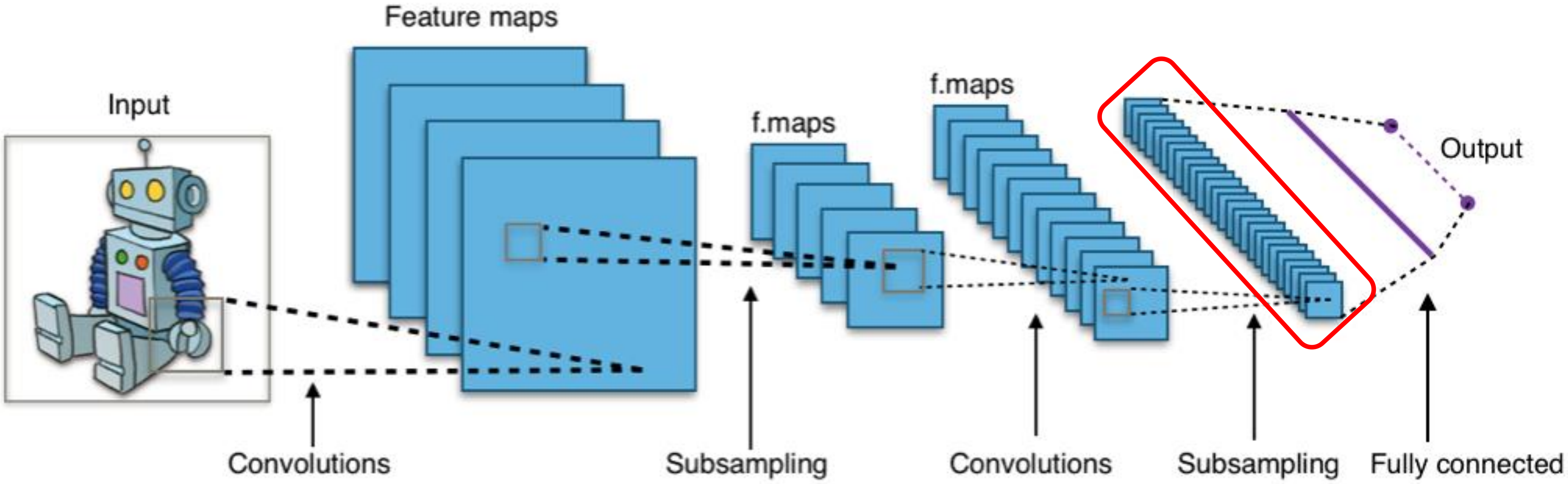
- …is a ‘*fully connected*’ neural network with non-linear activation functions.



- ‘*Feed-forward*’ neural network

Nielson

General CNN architecture



Learning phases

Training

Images

Labels

Image Features → Training → Trained classifier

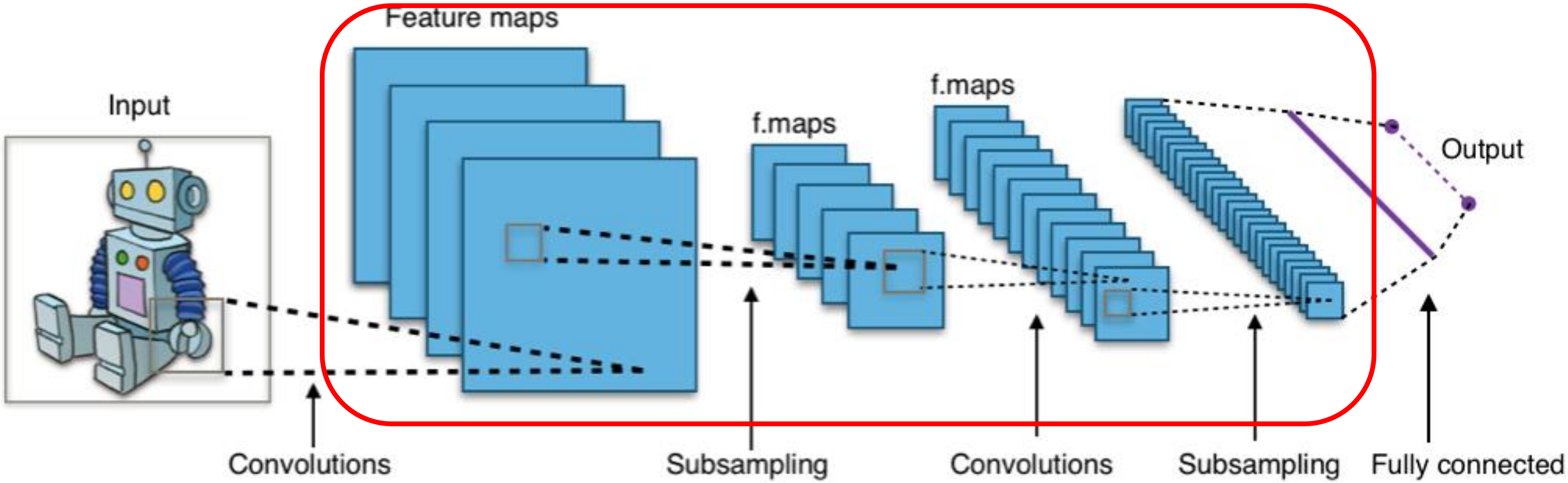
Testing

Image *not in training set*

Image Features → Apply classifier → Prediction

Slide credit: D. Hoiem and L. Lazebnik

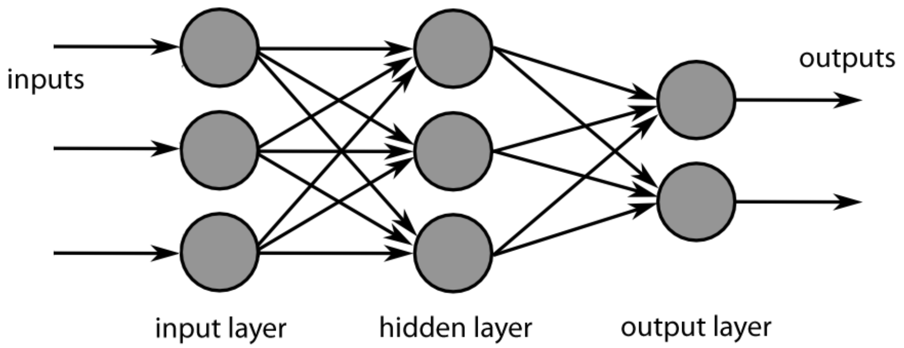
General CNN architecture



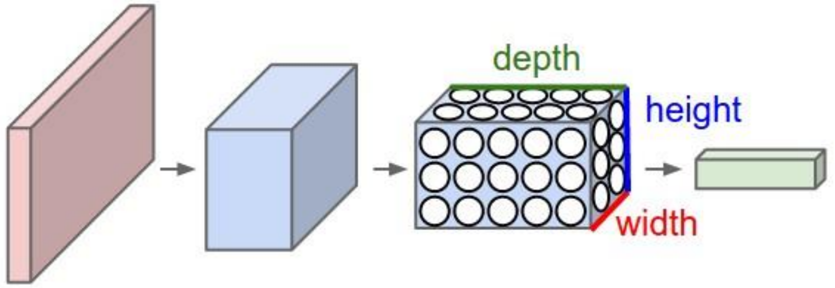
End to end learning!

Neural Network vs CNN

- Image as input in neural network
  - Size of feature vector = HxWxC
  - For 256x256 RGB image
    - 196608 dimensions



- CNN - Special type of neural network
  - Operate with volume of data
  - Weight sharing in form of kernels



Source: http://cs231n.github.io

CAP5415
Computer Vision

Yogesh S Rawat

yogesh@ucf.edu

HEC-241

Questions?

Introduction to Convolutional Neural Networks

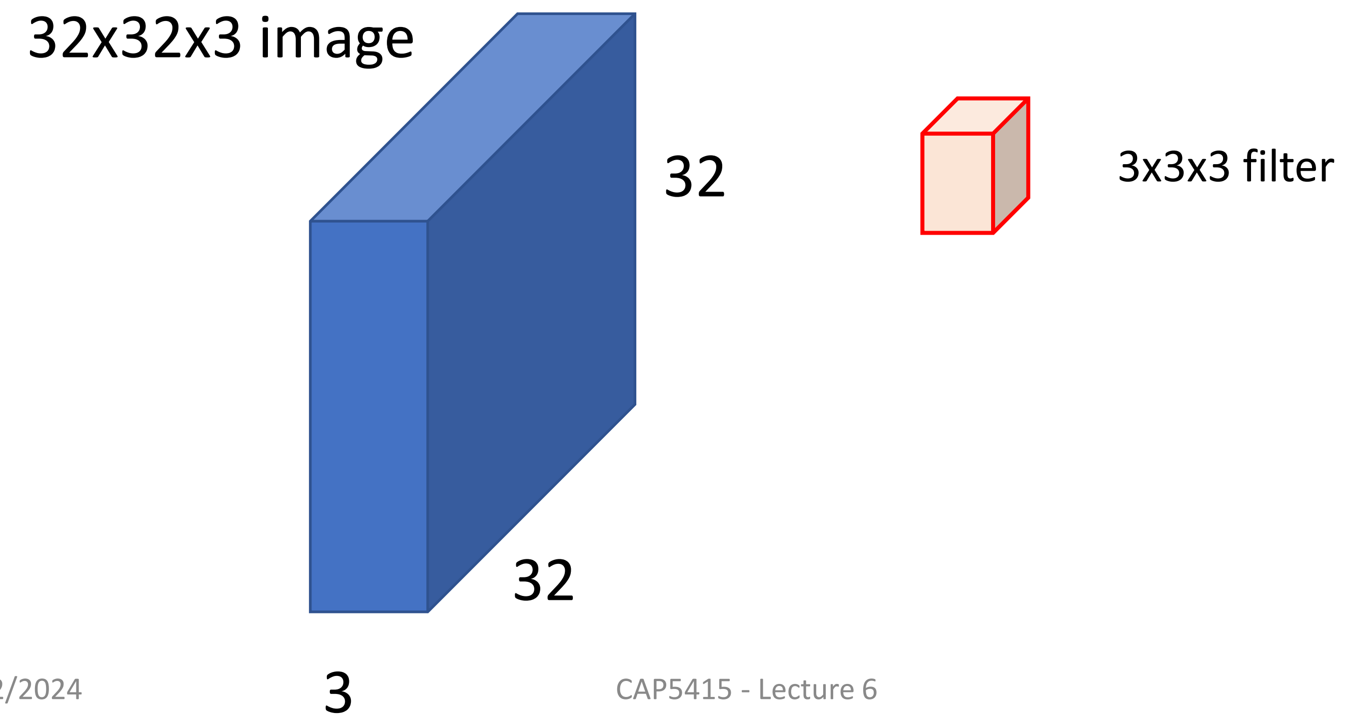
Lecture 6

Fundamental operation

Convolution

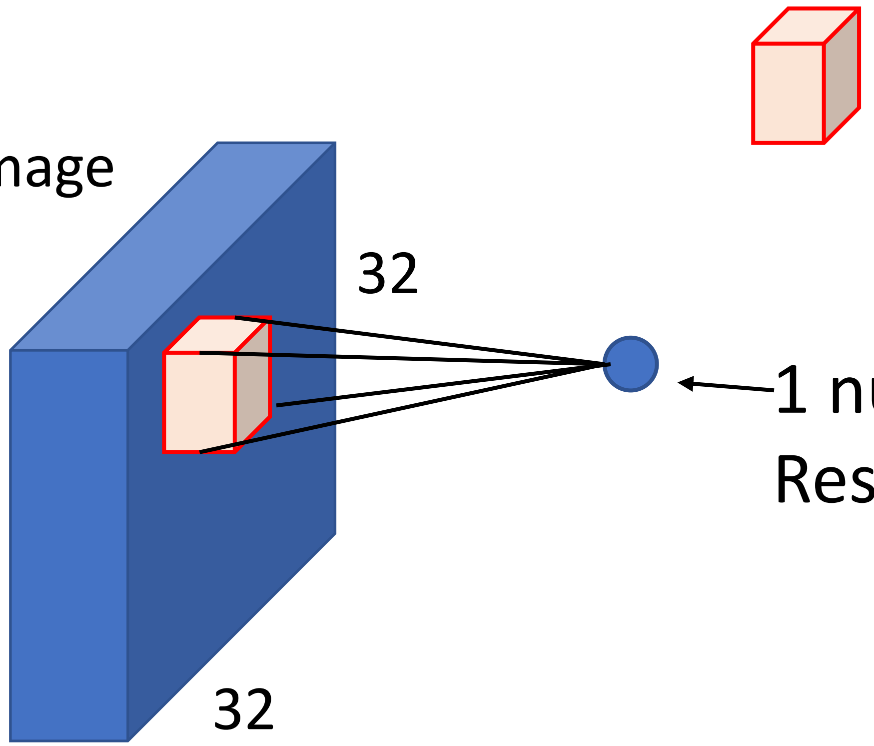
- Core building block of a CNN
  - Spatial structure of image is preserved

A filter/kernel is **convolved** with the image



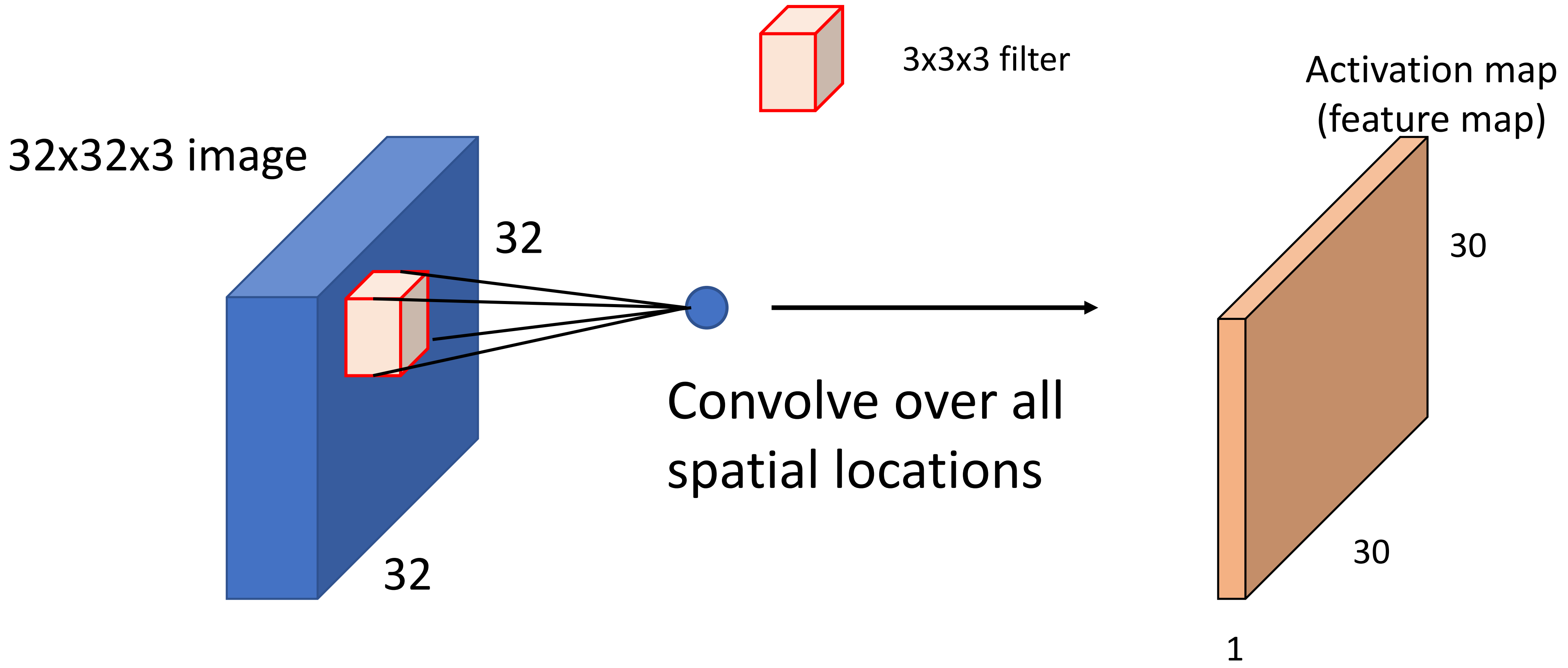
Convolution

- Convolution at one spatial location



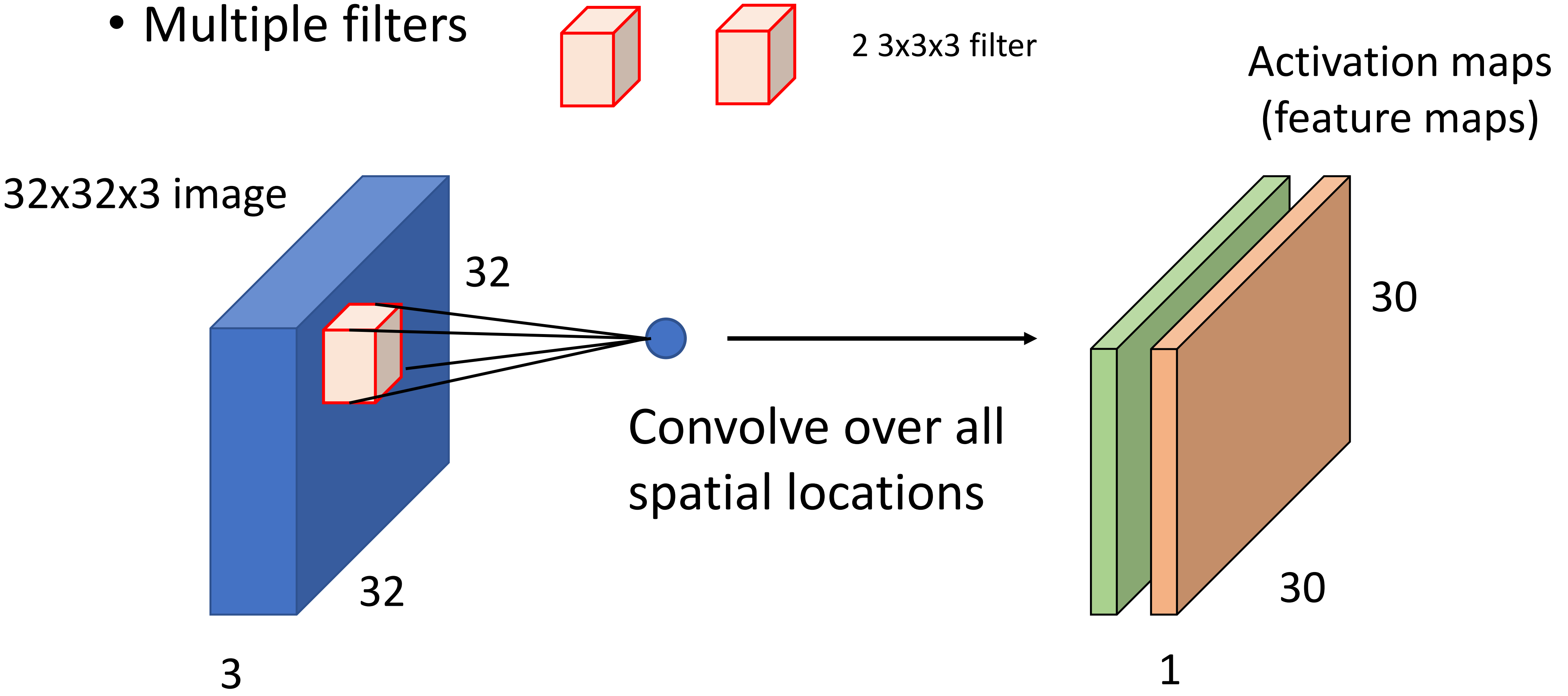
Convolution

- Convolution over whole image



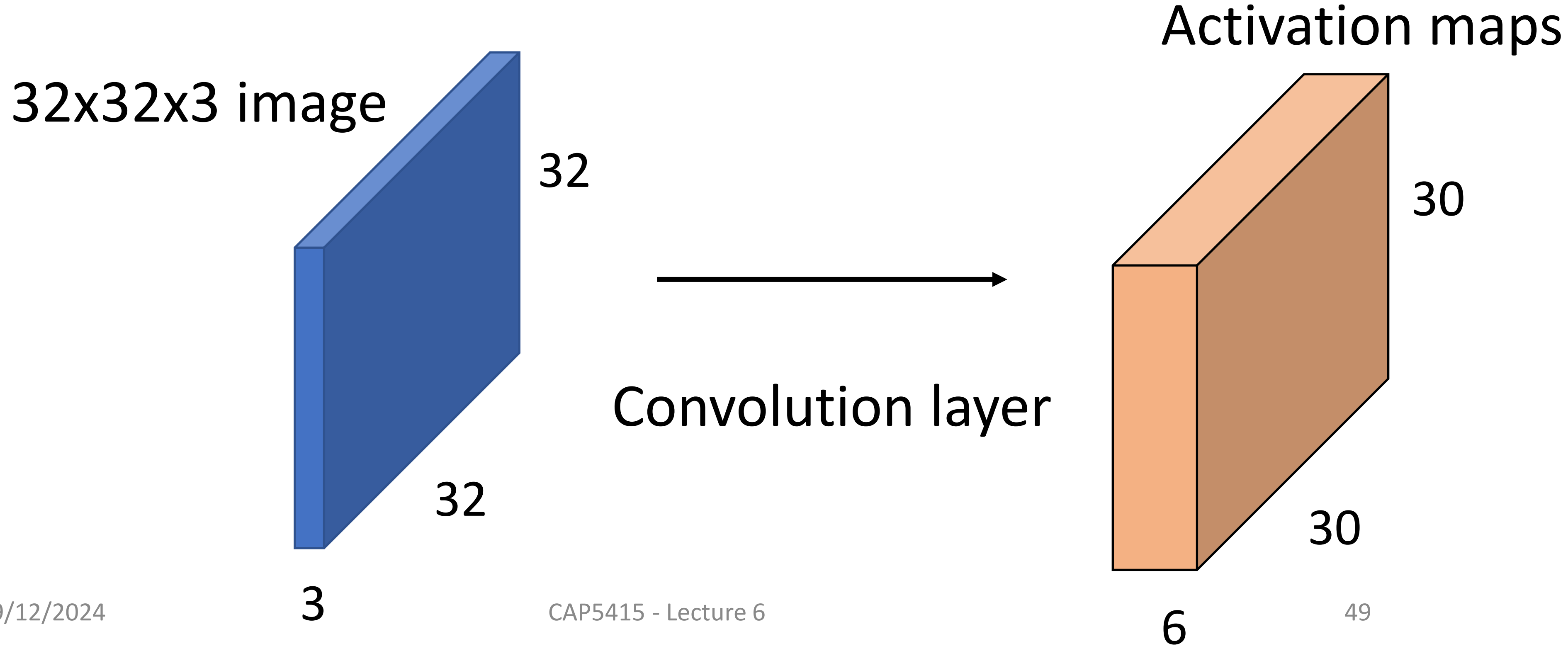
Convolution

- Multiple filters



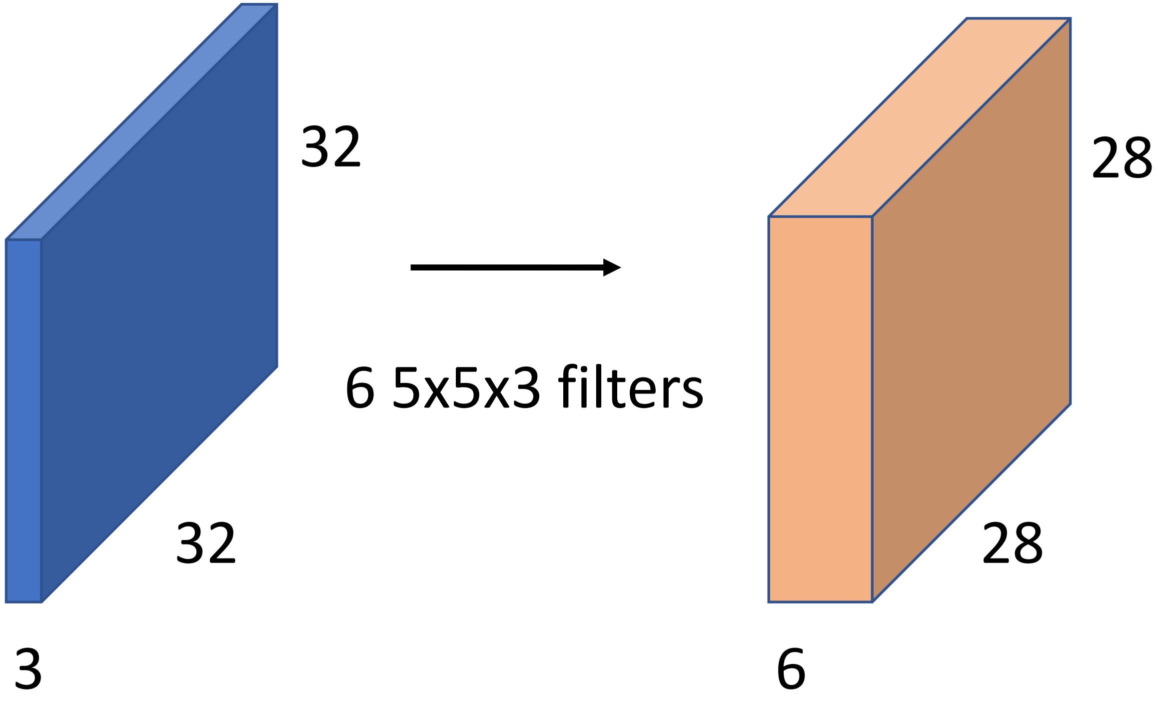
Convolution layer

- One convolution layer
  - 6 3x3x3 kernels



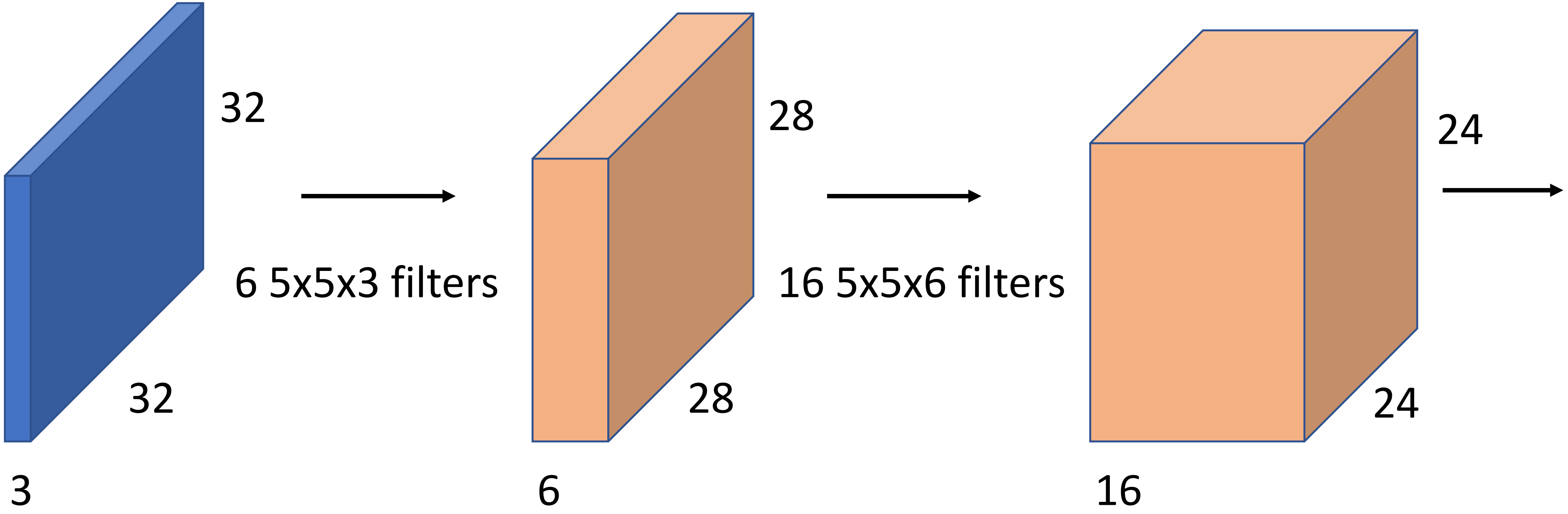
Convolutional Network

- Convolution network is a sequence of these layers

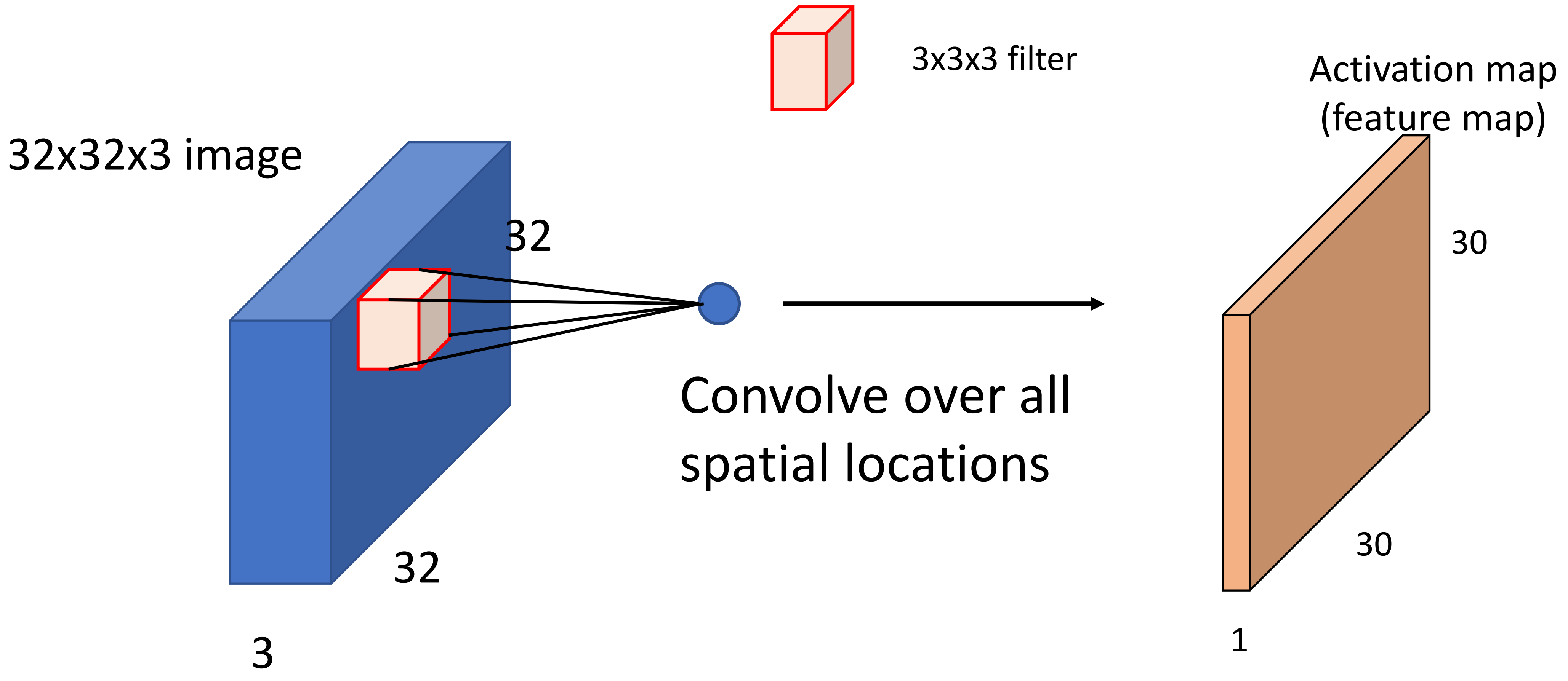


Convolutional Network

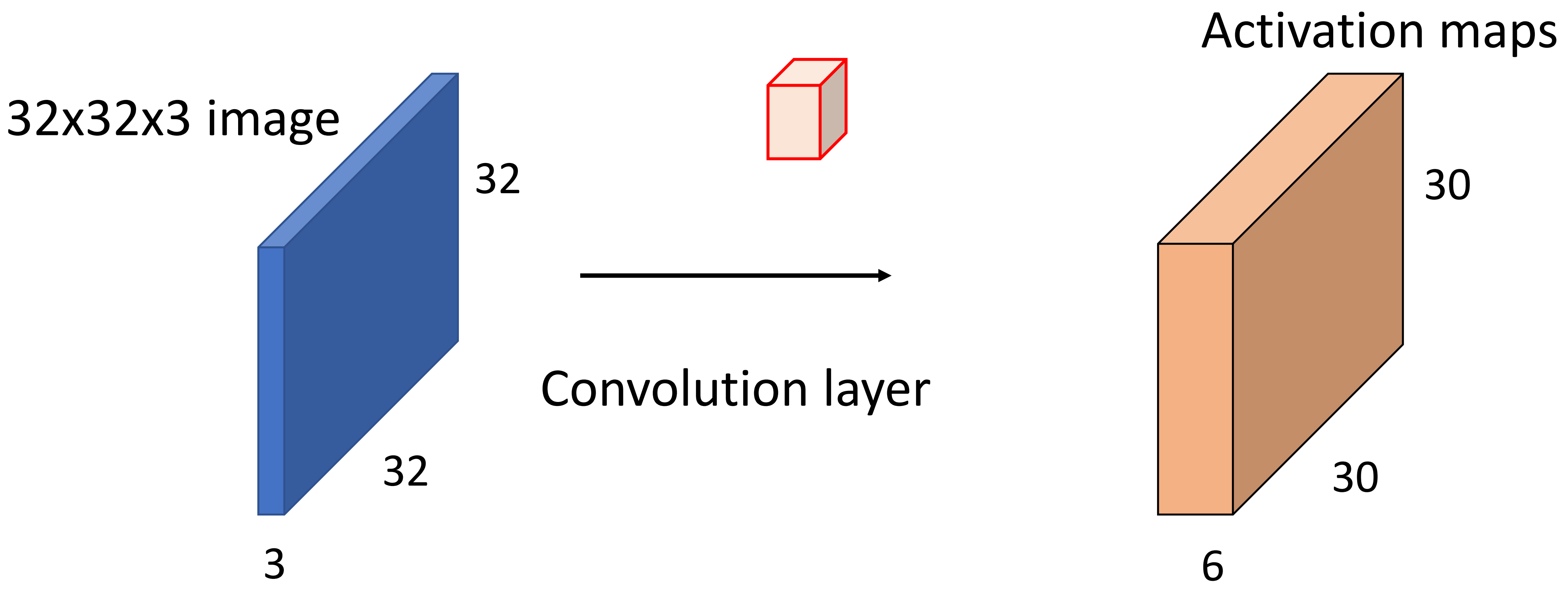
- Convolution network is a sequence of these layers



Parameters


3x3x3 filter
Activation map
(feature map)
32x32x3 image
32
30
Convolve over all
spatial locations
32
30
1
3

Parameters



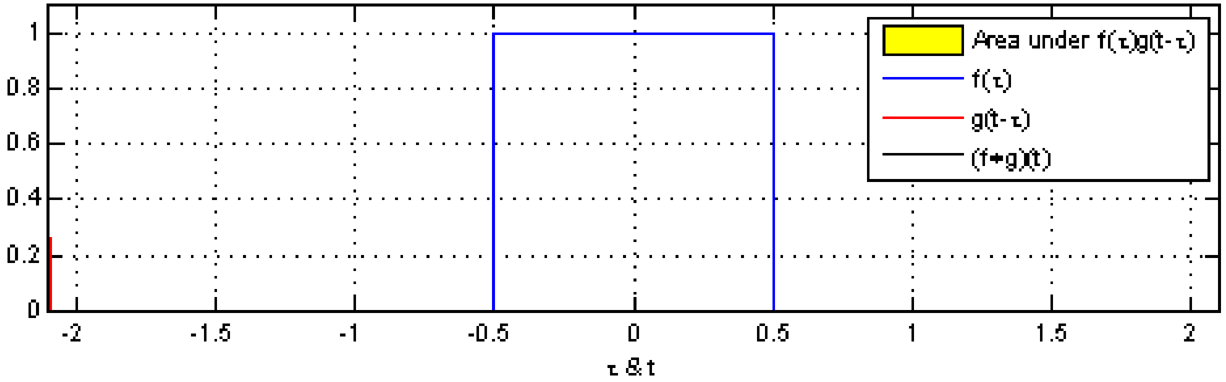
6 3x3x3 kernels – 6x3x3x3 parameters = 162

Convolution Operation

- Convolution of two functions f and g

function f(t)

kernel g(t)



$$(f * g)(t) = \int_{-\infty}^{+\infty} f(\tau)g(t-\tau)d\tau$$

In CNN we use 2D convolutions (mostly)

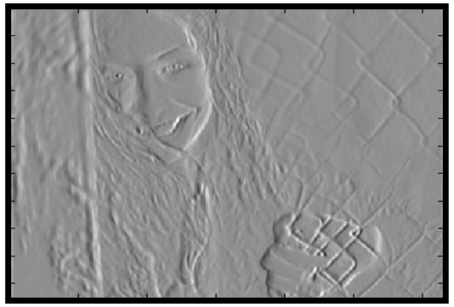
Sobel Edge Detector – recap

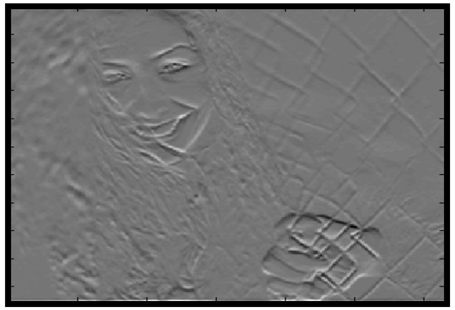
Image I

$$* \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \rightarrow \frac{d}{dx} I$$

$$* \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \rightarrow \frac{d}{dy} I$$

$$\sqrt{\left(\frac{d}{dx} I\right)^2 + \left(\frac{d}{dy} I\right)^2} \rightarrow \boxed{\text{Threshold}} \rightarrow \text{Edges}$$





Demo

filter

1	0	1
0	1	0
1	0	1

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

Input image

4		

output

Demo

filter

1	0	1
0	1	0
1	0	1

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

Input image

4	3	

output

Demo

filter

1	0	1
0	1	0
1	0	1

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

Input image

4	3	4

output

Demo

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

Input image

filter

1	0	1
0	1	0
1	0	1

4	3	4
2		

output

Demo

filter

1	0	1
0	1	0
1	0	1

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

Input image

4	3	4
2	4	3
1	3	3

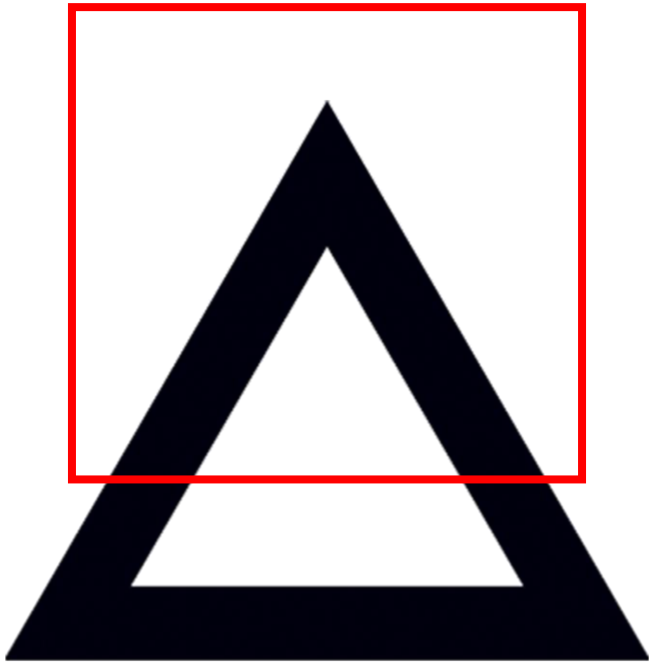
output

Convolution - Intuition

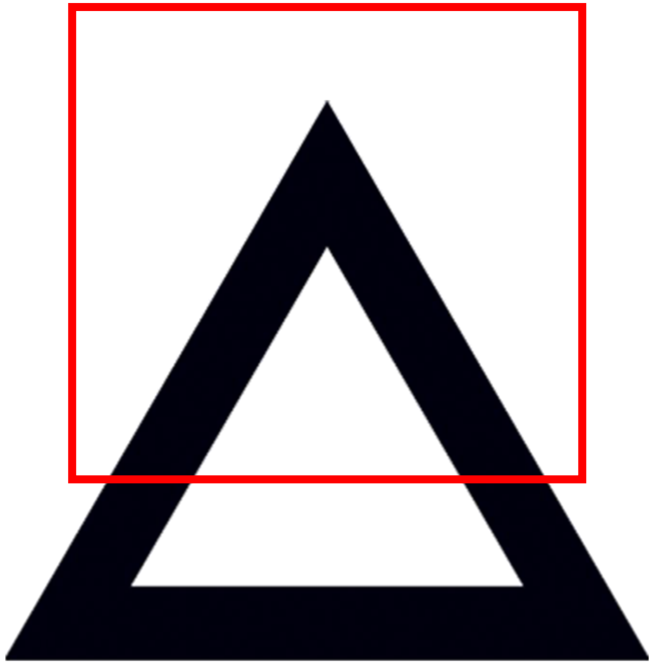
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	0	0	0	0

UCF Center for Research in Computer Vision

Convolution - Intuition



Convolution - Intuition



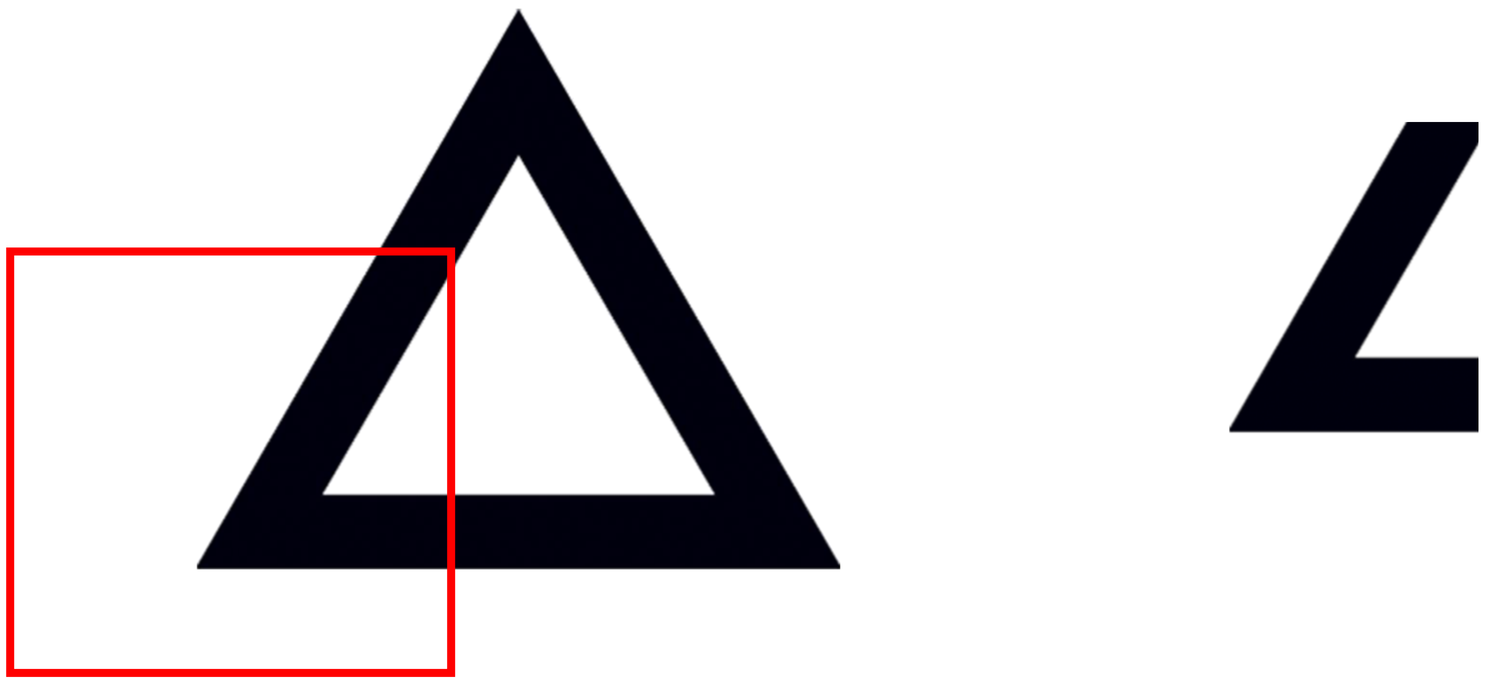
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	0	0	0	0

$*$

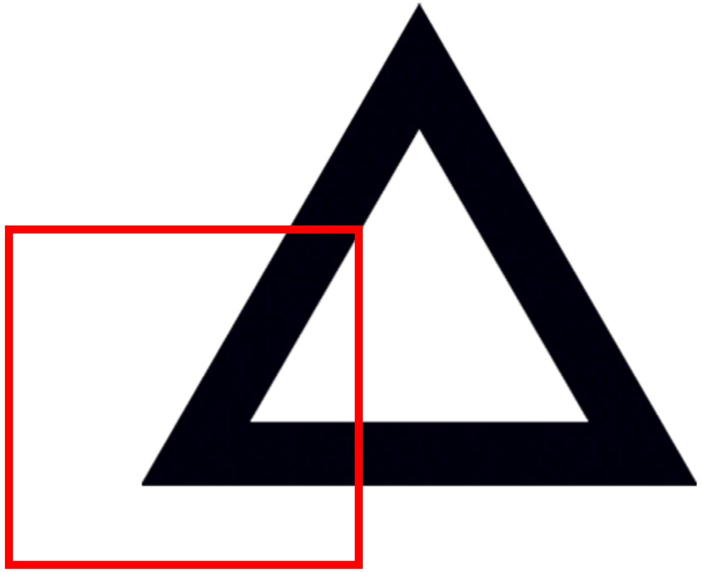
0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
1	0	0	0	1
0	0	0	0	0

$$1x1 + 1x1 + \ldots + 1x1 = 5$$

Convolution - Intuition



Convolution - Intuition



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

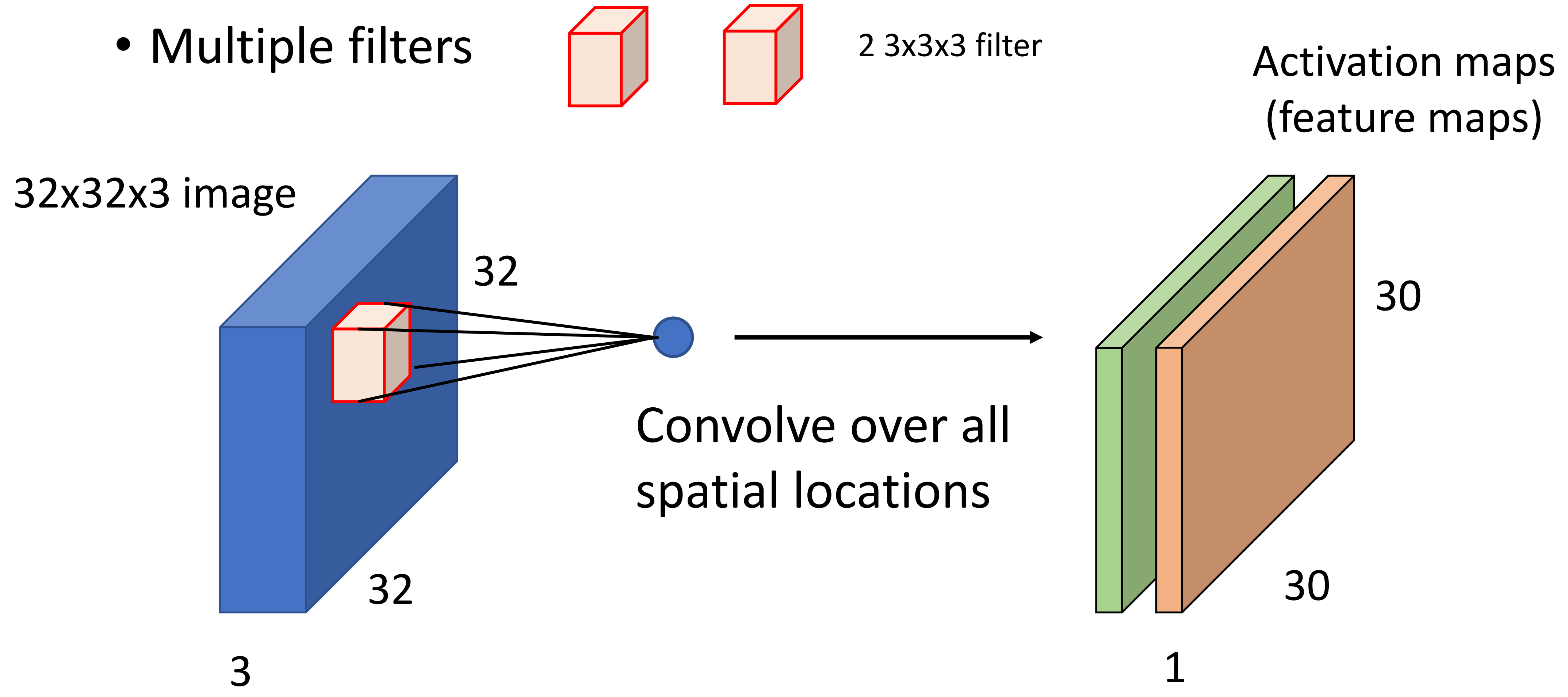
$*$

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

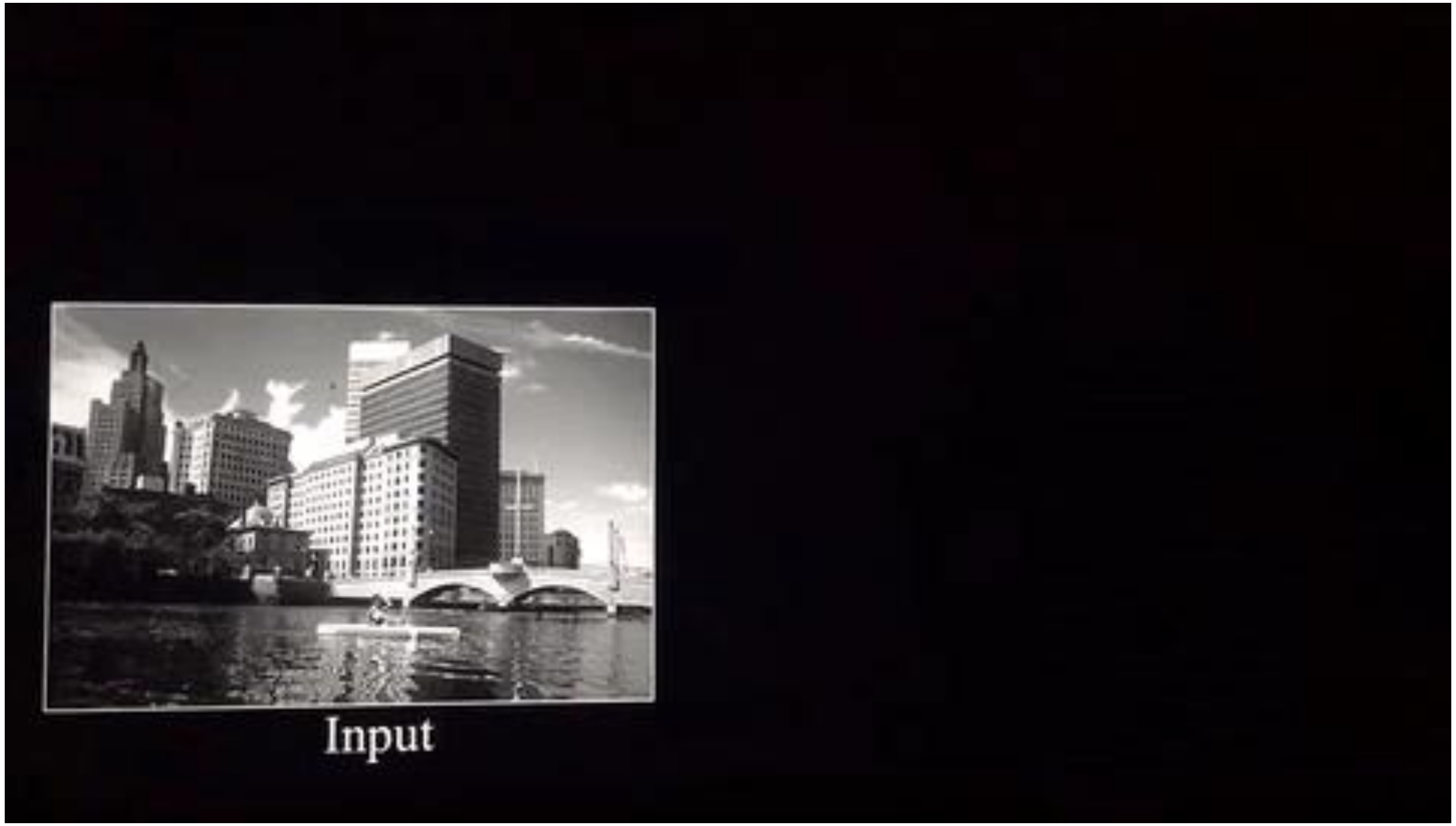
$1 \times 1 = 1$

Convolution

- Multiple filters



Convolution - Intuition



Source : https://cs.nyu.edu/~fergus/tutorials/deep_learning_cvpr12/

Questions?

Introduction to Convolutional Neural Networks

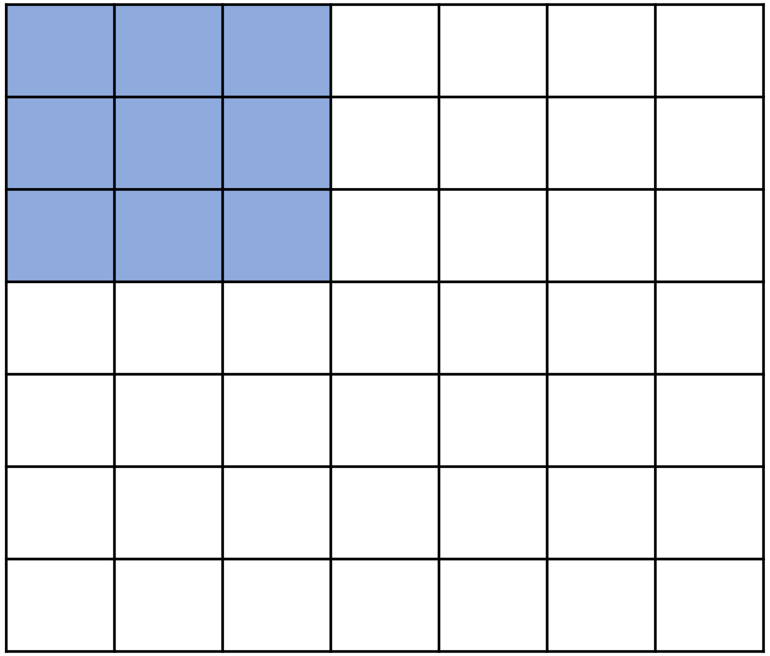
Lecture 6

Practical considerations

2D Convolution - dimensions

7x7 map

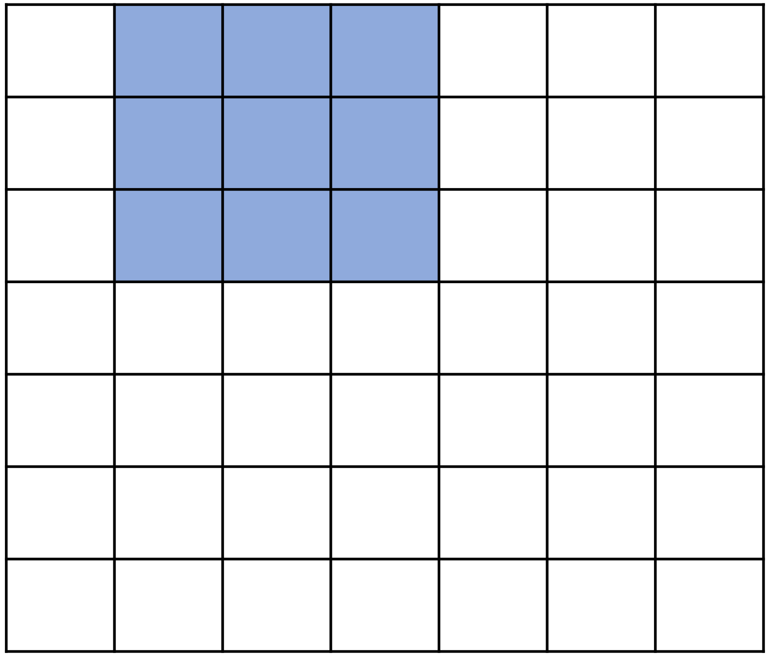
3x3 filter



2D Convolution - dimensions

7x7 map

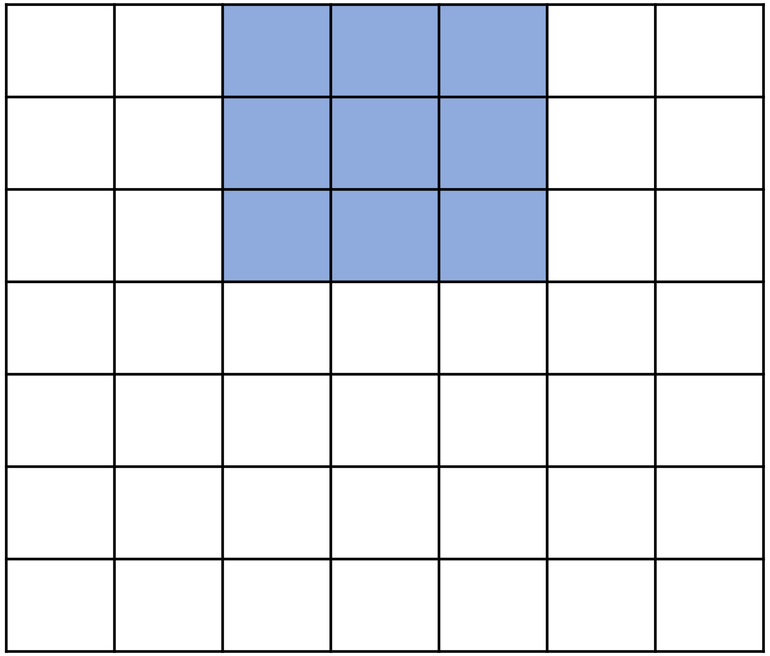
3x3 filter



2D Convolution - dimensions

7x7 map

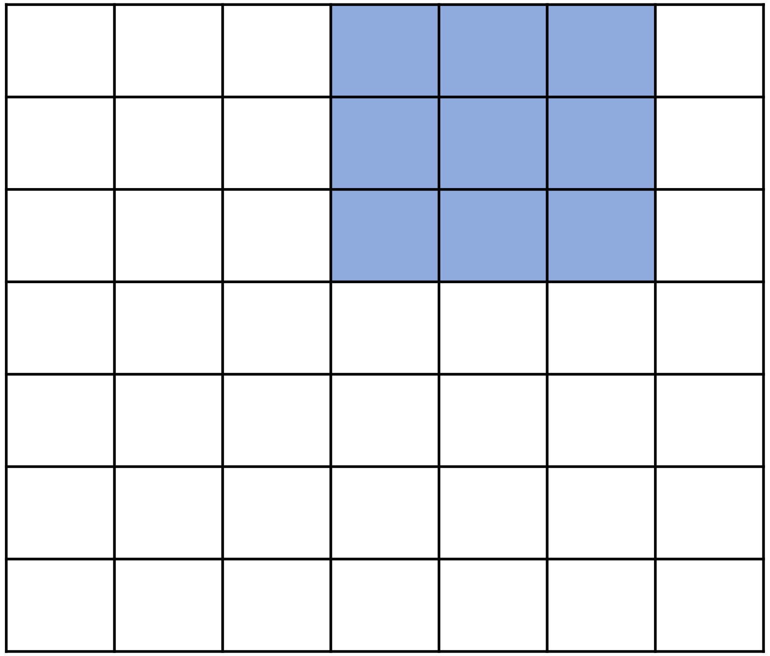
3x3 filter



2D Convolution - dimensions

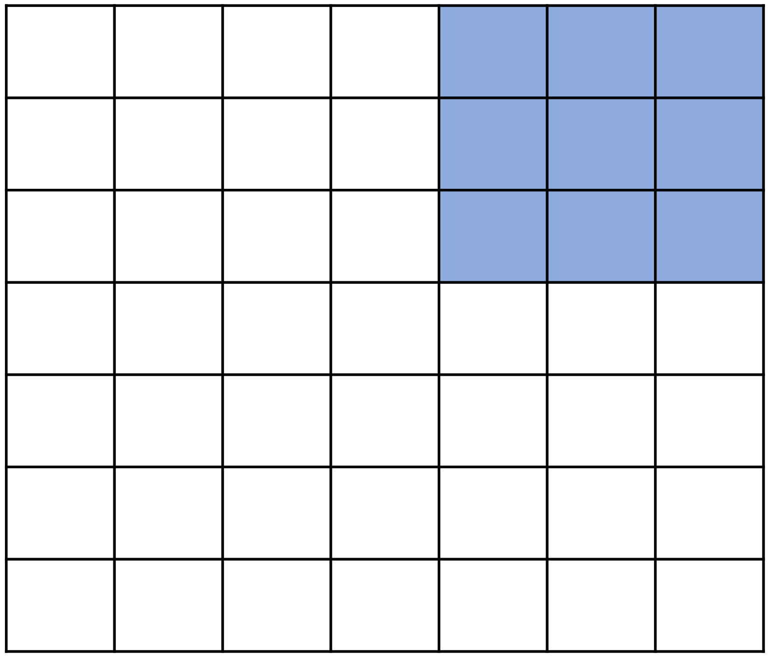
7x7 map

3x3 filter



2D Convolution - dimensions

7x7 map



3x3 filter

Output activation map 5x5

Output size

N-F+1

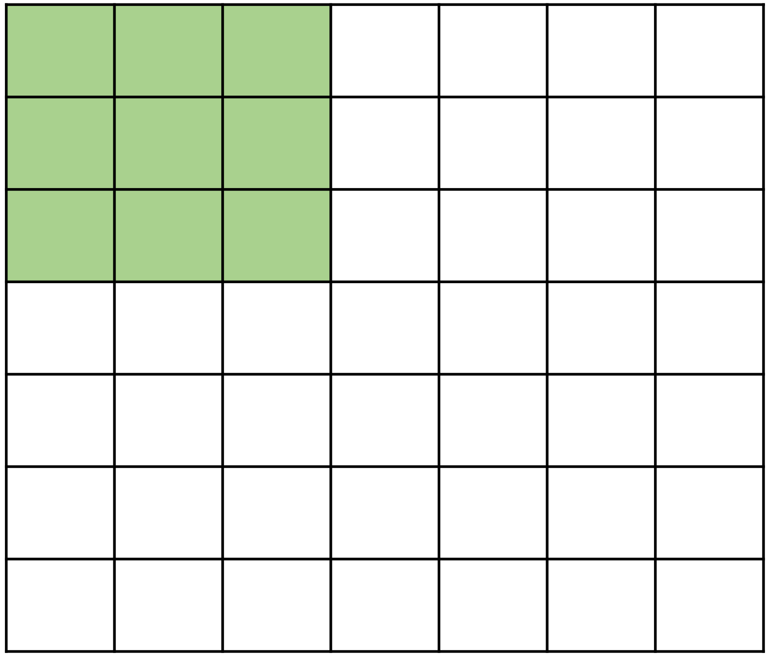
(7 – 3 + 1) = 5

N – input size

F – filter size

Stride

7x7 map

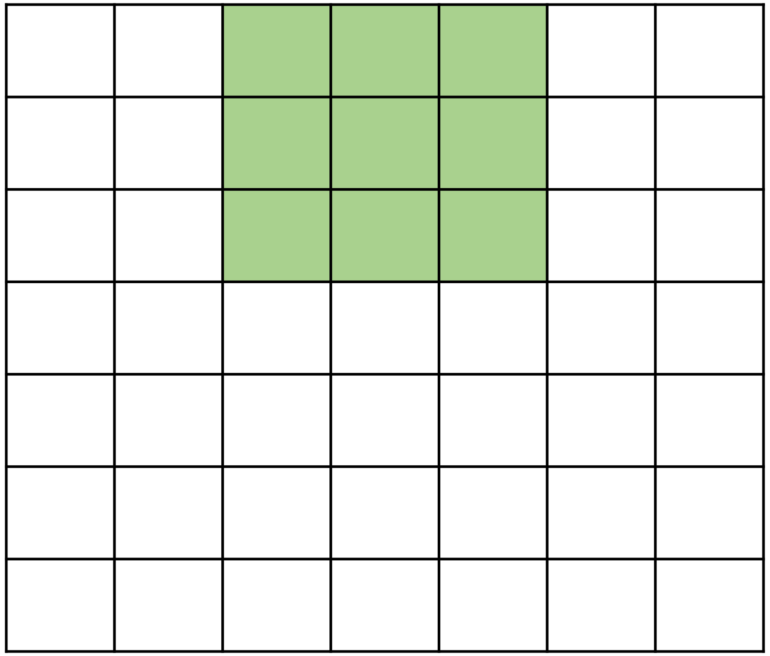


3x3 filter

Filter applied with stride 2

Stride

7x7 map

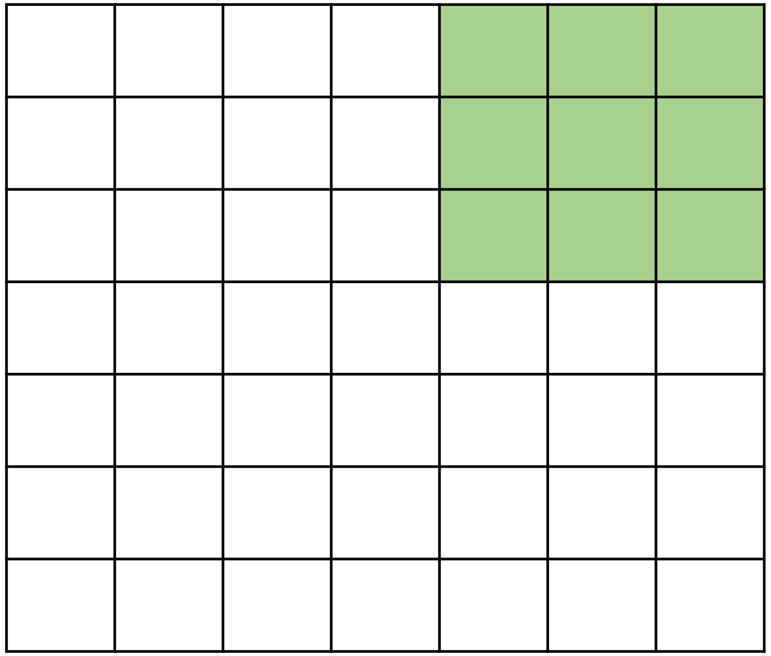


3x3 filter

Filter applied with stride 2

Stride

7x7 map



3x3 filter

Filter applied with stride 2

Activation map size 3x3

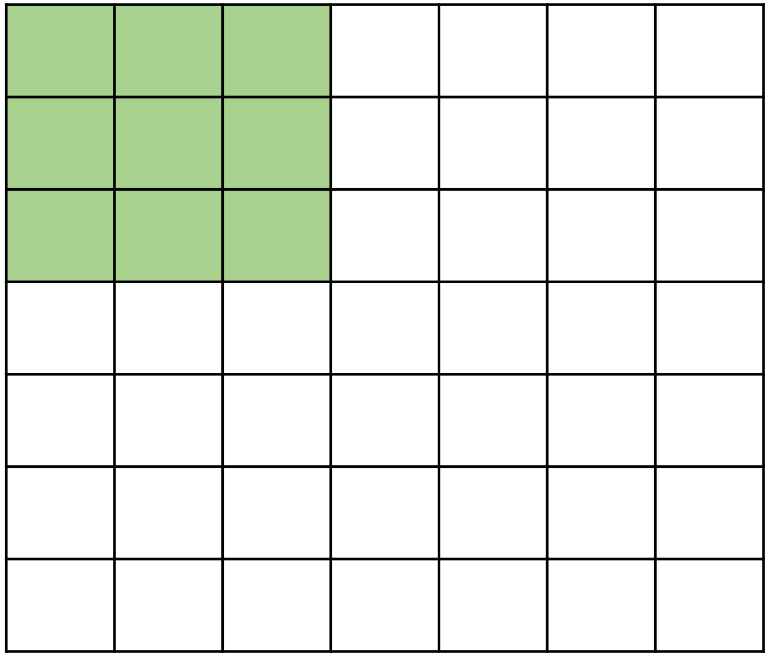
Output size

(7-3)/2 + 1 = 3

(N-F)/S + 1

Stride

7x7 map

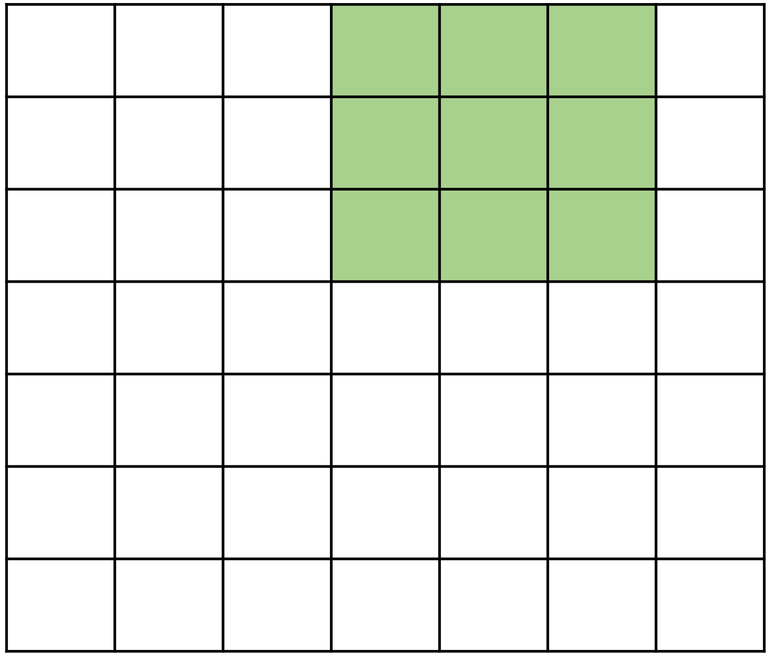


3x3 filter

Filter applied with stride 3

Stride

7x7 map



3x3 filter

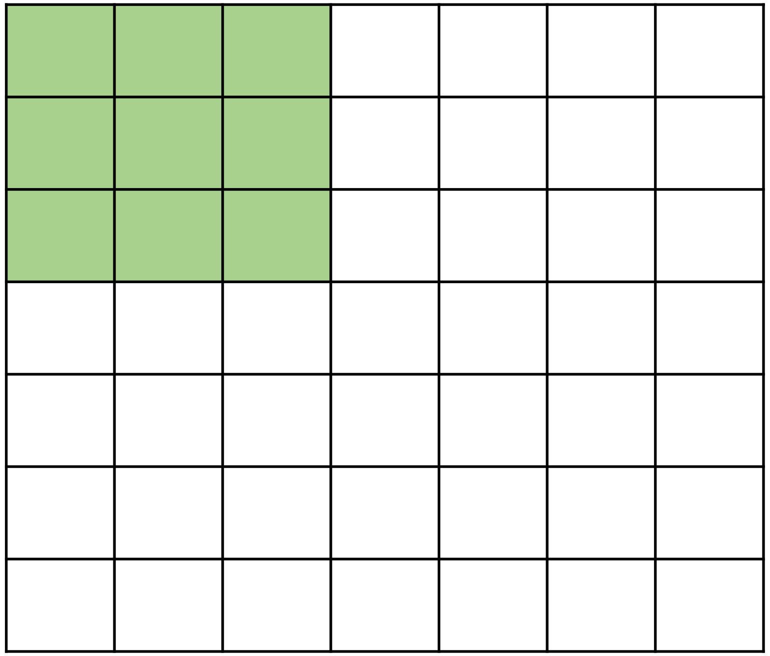
Filter applied with stride 3

Cannot cover perfectly

Not all parameters will fit

Stride

7x7 map



3x3 filter

Output size (N-F)/S + 1

N = 7, F = 3

Stride 1

(7-3)/1 + 1 => 5

Stride 2

(7-3)/2 + 1 => 3

Stride 3

(7-3)/3 + 1 => 2.33

Padding

- Zero padding in the input

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

For 7x7 input and 3x3 filter

If we have padding of one pixel

Output
7x7

Size (recall (N-F)/S+1)
(N-F+2P)/S + 1

Padding

- Zero padding in the input

0	0	0	0	0	0	0	0	0
0								0
0								0
0								0
0								0
0								0
0								0
0								0
0	0	0	0	0	0	0	0	0

Common to see,
(F-1)/2 padding with stride 1 to preserve the map size

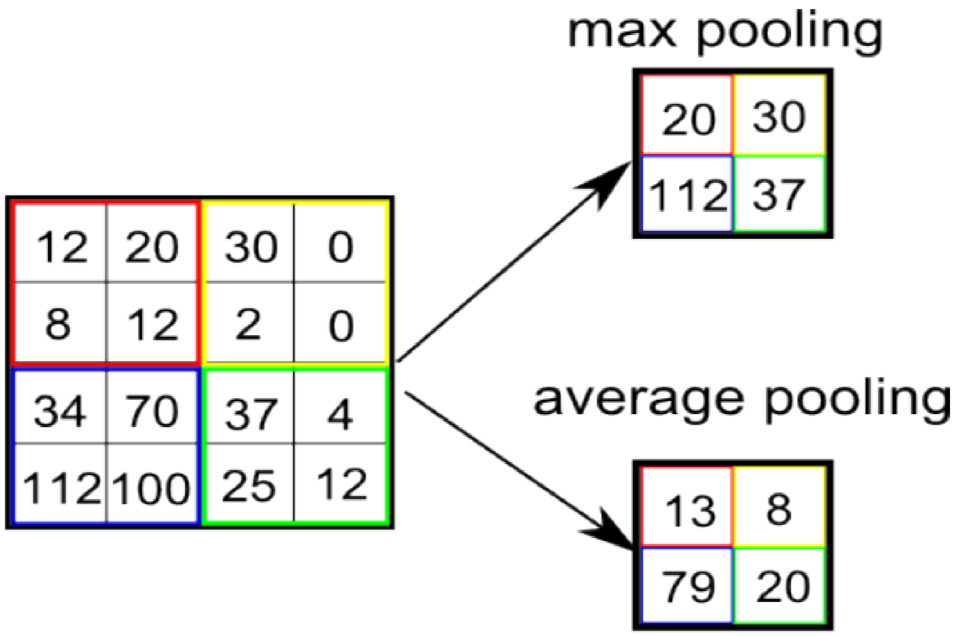
$N = (N-F+2P)/S + 1$

$\Rightarrow (N-1)S = N-F+2P$

$\Rightarrow P = (F-1)/2$

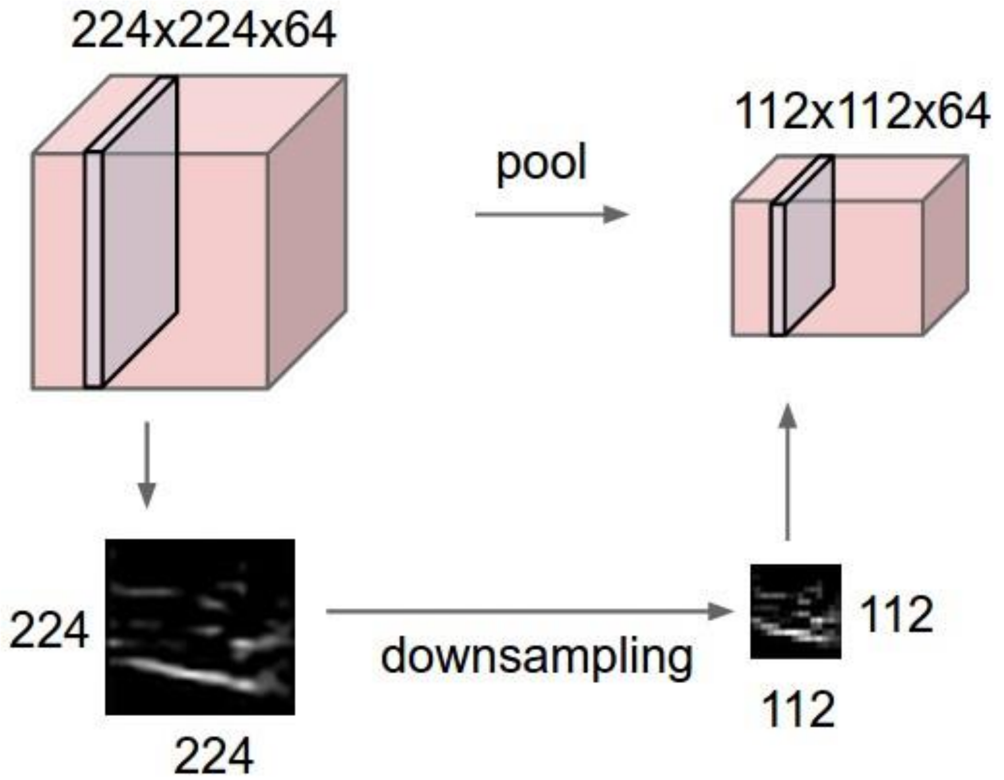
Pooling

- Invariance to small translations of the input



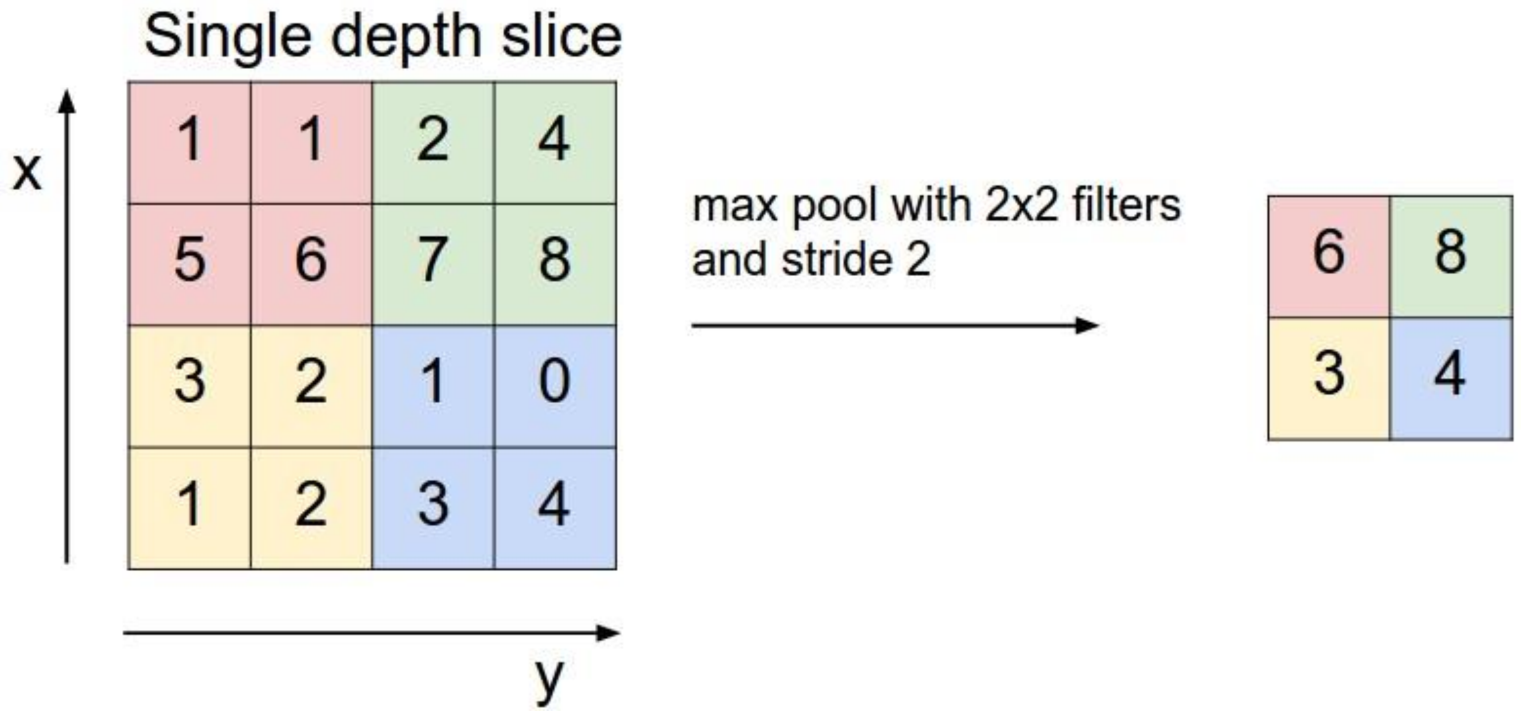
Pooling

- Makes the representations smaller
- Operates over each activation map independently



Pooling

- Kernel size
- Stride



Questions?

CAP5415
Computer Vision

Yogesh S Rawat

yogesh@ucf.edu

HEC-241

Questions?

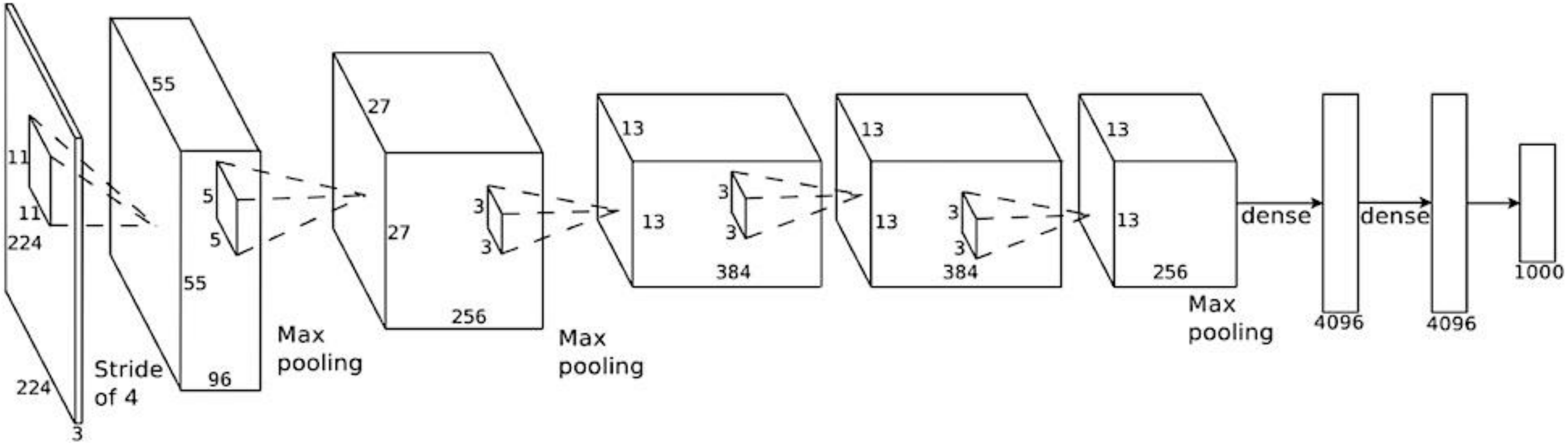
Introduction to Convolutional Neural Networks

Lecture 6

Case study

AlexNet : Network Size

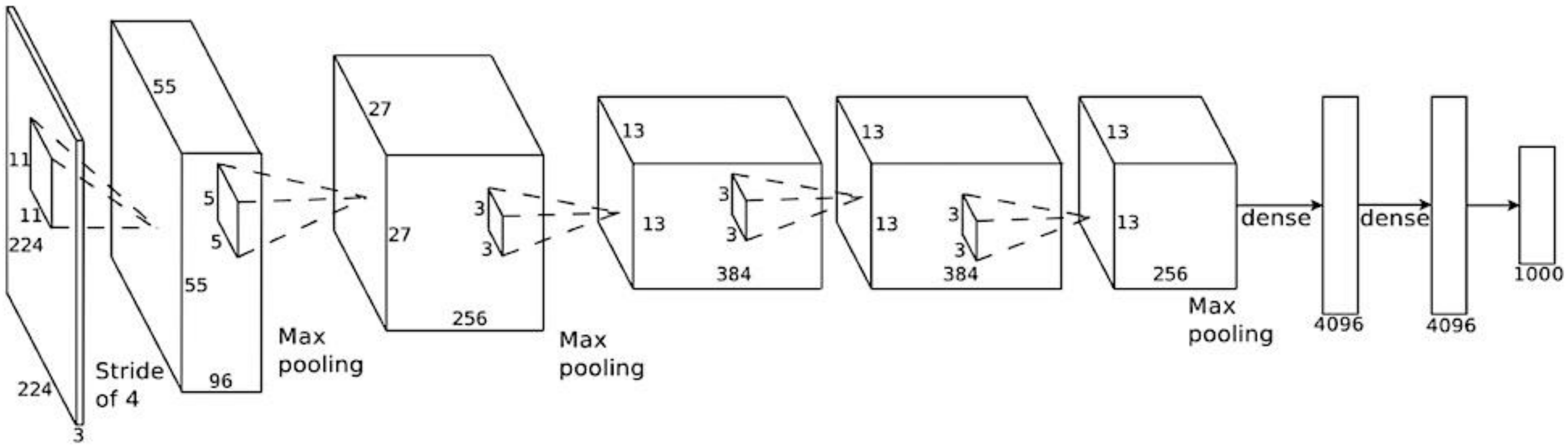
CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3
FC6
FC7
FC8



- Input 227x227x3
- 5 convolution layers
- 3 dense layers
- Output 1000-D vector

AlexNet : Network Size

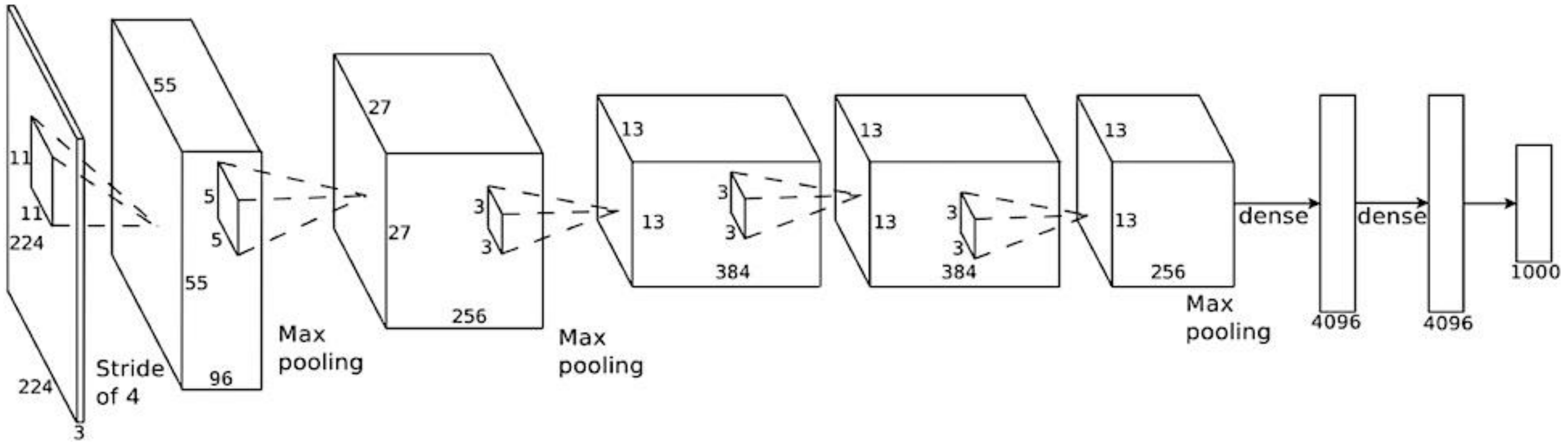
CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3
FC6
FC7
FC8



- Input: 227x227x3 images
- First layer (CONV1): 96 11x11 filters applied at stride 4
- What is the output volume size? (227-11)/4+1 = 55
- What is the number of parameters? 11x11x3x96 = 35K

AlexNet : Network Size

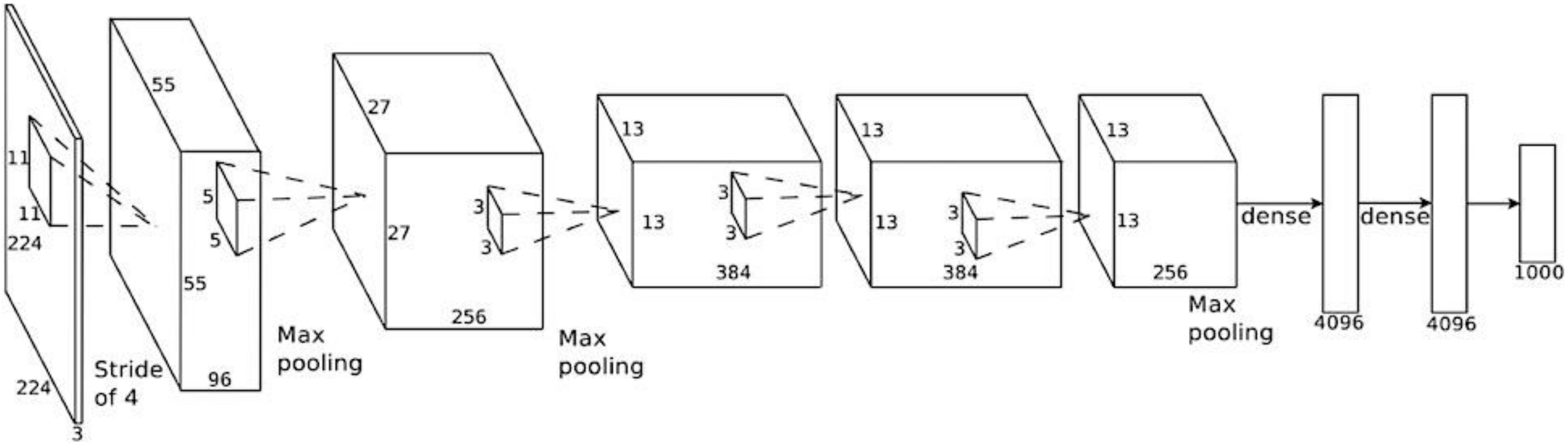
CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3
FC6
FC7
FC8



- After CONV1: 55x55x96
- Second layer (POOL1): 3x3 filters applied at stride 2
- What is the output volume size? (55-3)/2+1 = 27
- What is the number of parameters in this layer? 0

AlexNet : Network Size

CONV1
MAX POOL1
NORM1
CONV2
MAX POOL2
NORM2
CONV3
CONV4
CONV5
MAX POOL3
FC6
FC7
FC8



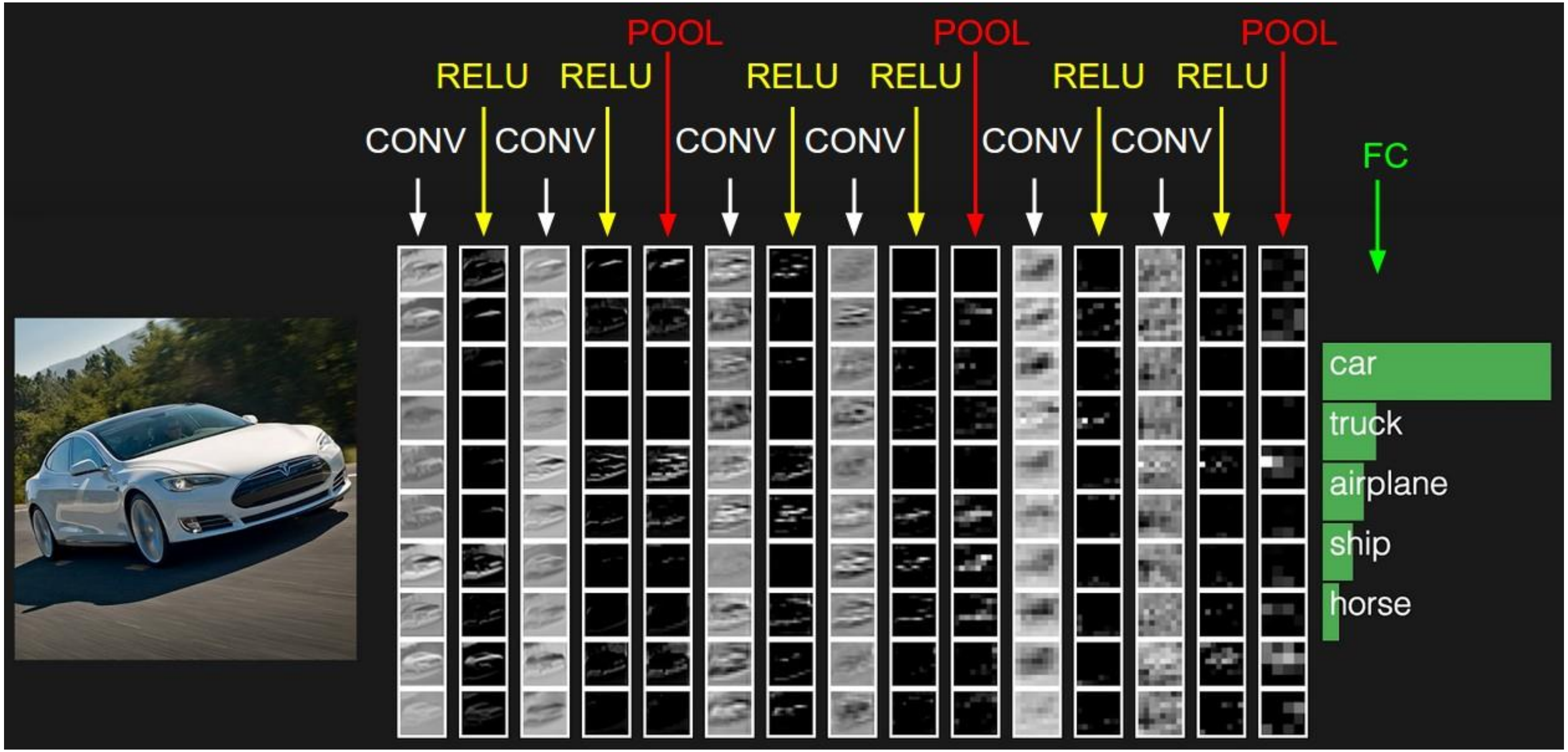
- After POOL1: 27x27x96
- Third layer (NORM1): Normalization
- What is the output volume size? 27x27x96

AlexNet : Network Size

CONV1	35K
MAX POOL1	
NORM1	
CONV2	614K
MAX POOL2	
NORM2	
CONV3	884K
CONV4	1.3M
CONV5	442K
MAX POOL3	
FC6	37M
FC7	16M
FC8	4M

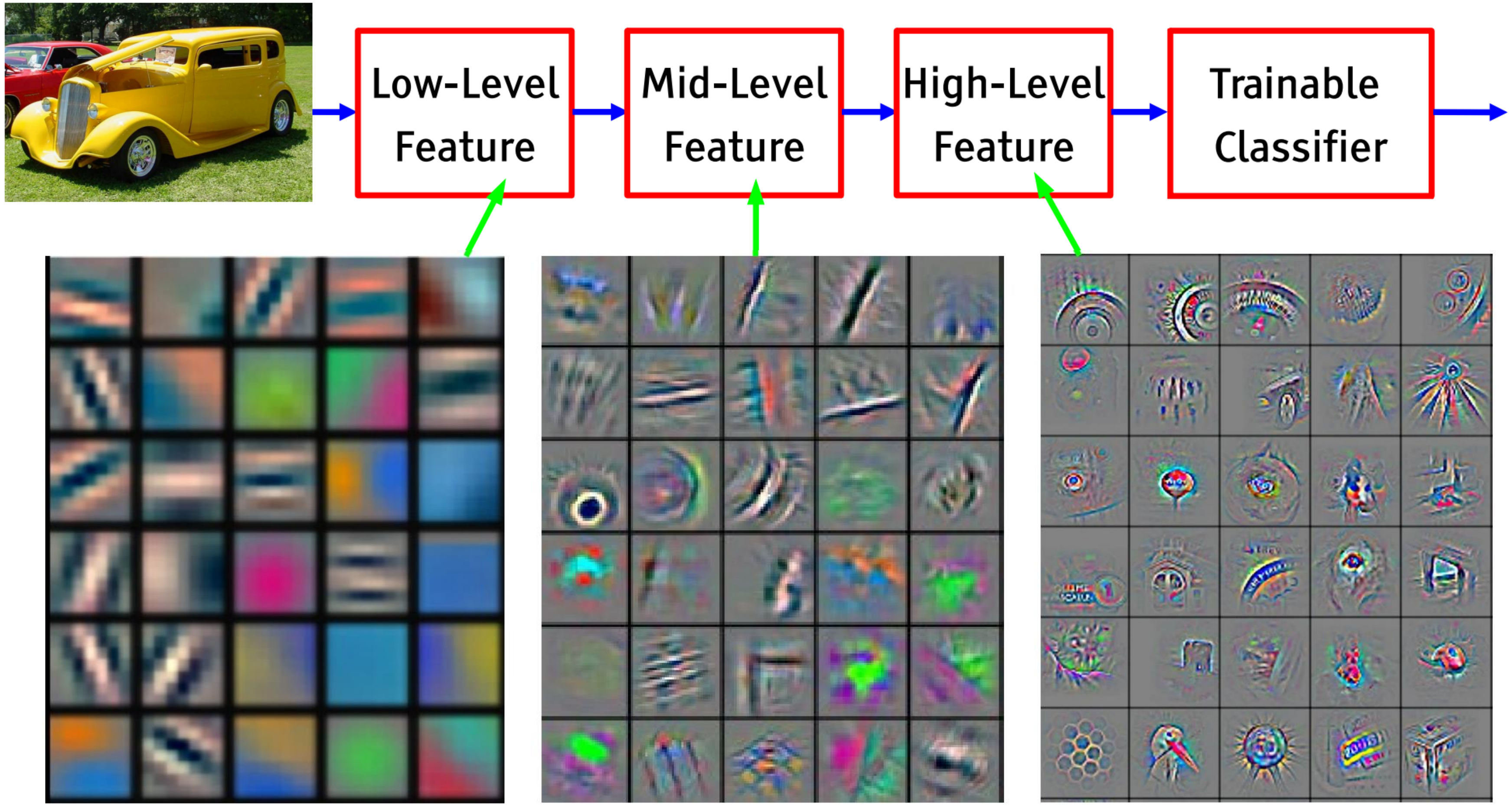
1. [227x227x3] INPUT
2. [55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
3. [27x27x96] MAX POOL1: 3x3 filters at stride 2
4. [27x27x96] NORM1: Normalization layer
5. [27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
6. [13x13x256] MAX POOL2: 3x3 filters at stride 2
7. [13x13x256] NORM2: Normalization layer
8. [13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
9. [13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
10. [13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
11. [6x6x256] MAX POOL3: 3x3 filters at stride 2
12. [4096] FC6: 4096 neurons
13. [4096] FC7: 4096 neurons
14. [1000] FC8: 1000 neurons (class scores)

Visualizing CNN



Source : http://cs231n.github.io

Visualizing Convolution



Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

Why not correlation neural network?

- It could be
  - Deep learning libraries actually implement correlation
- Correlation relates to convolution via a 180deg rotation
  - When we *learn* kernels, we could easily learn them flipped

Questions?

Sources for this lecture include materials from works by Abhijit Mahalanobis, Andrej Karpathy, and Fei Fei Li