Class: Machine Learning

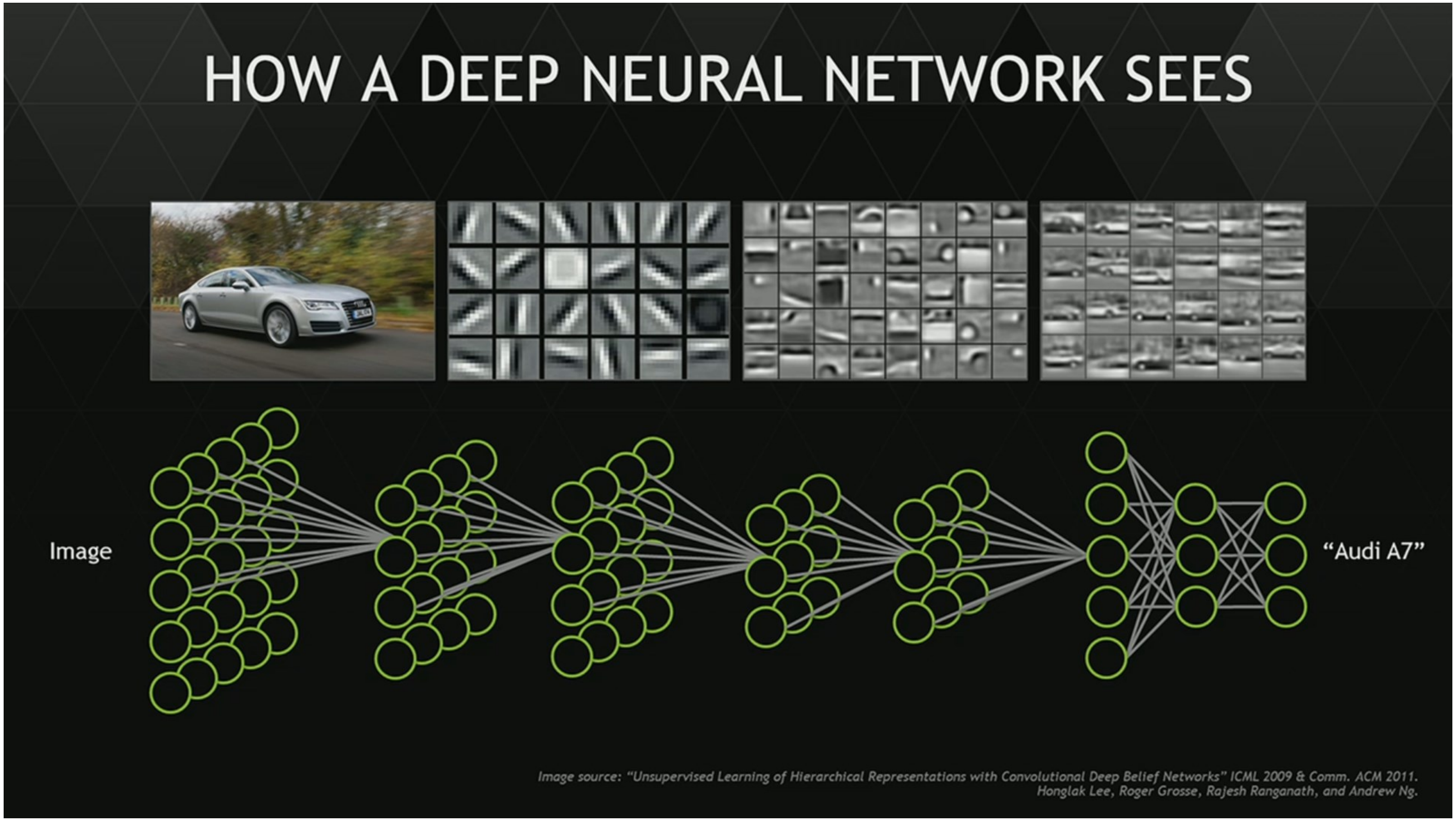
Convolutional Neural Networks

Instructor: Matteo Leonetti

Learning outcomes

- Describe the main elements of a Convolutional Neural Network (CNN)
- Compute the convolution between a filter and an image
- Assemble an architecture for a CNN

Why Convnets?

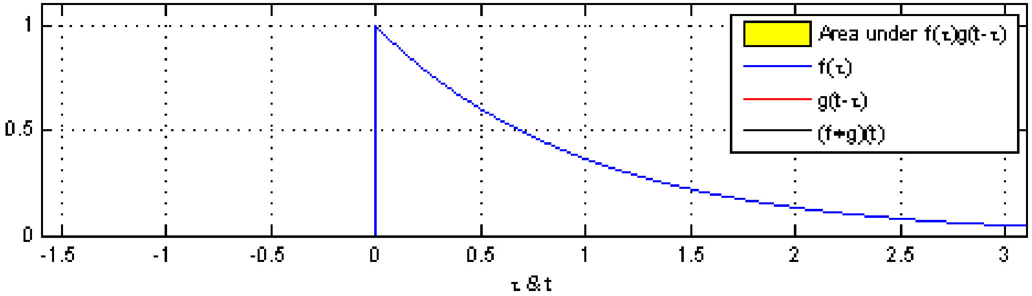


HOW A DEEP NEURAL NETWORK SEES

Image source: "Unsupervised Learning of Hierarchical Representations with Convolutional Deep Belief Networks" ICML 2009 & Comm. ACM 2011. Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Ng.

What is the convolution?

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



[From Wikipedia]

Filter application

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

1				

1	1	1
0	0	1
0	0	1

Filter application

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

1	1			

1	1	1
0	0	1
0	0	1

Filter application

$\boldsymbol{x} =$

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

$\boldsymbol{w} =$

1	1	1
0	0	1
0	0	1

$+ w_0$

1	1	1	1	0
1	1	1	2	0
3	4	4	5	2
2	2	1	2	1
2	3	3	3	2

$$= \boldsymbol{w}^T x + w_0$$

The filter can be implemented with a neuron!

However, the input is not “static” because the filter is slid across the image

Padding

$x =$

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

$w =$

1	1	1
0	0	1
0	0	1

0	0	0	0	0	0	0
1	1	1	1	1	0	
...	1	1	1	2	0	
	3	4	4	5	2	
	2	2	1	2	1	
	2	3	3	3	2	

The application of the filter would reduce the size of the image. This can be prevented by padding the image, typically with zeros.

Stride

$x =$

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

1	1	0
3	4	2
2	3	2

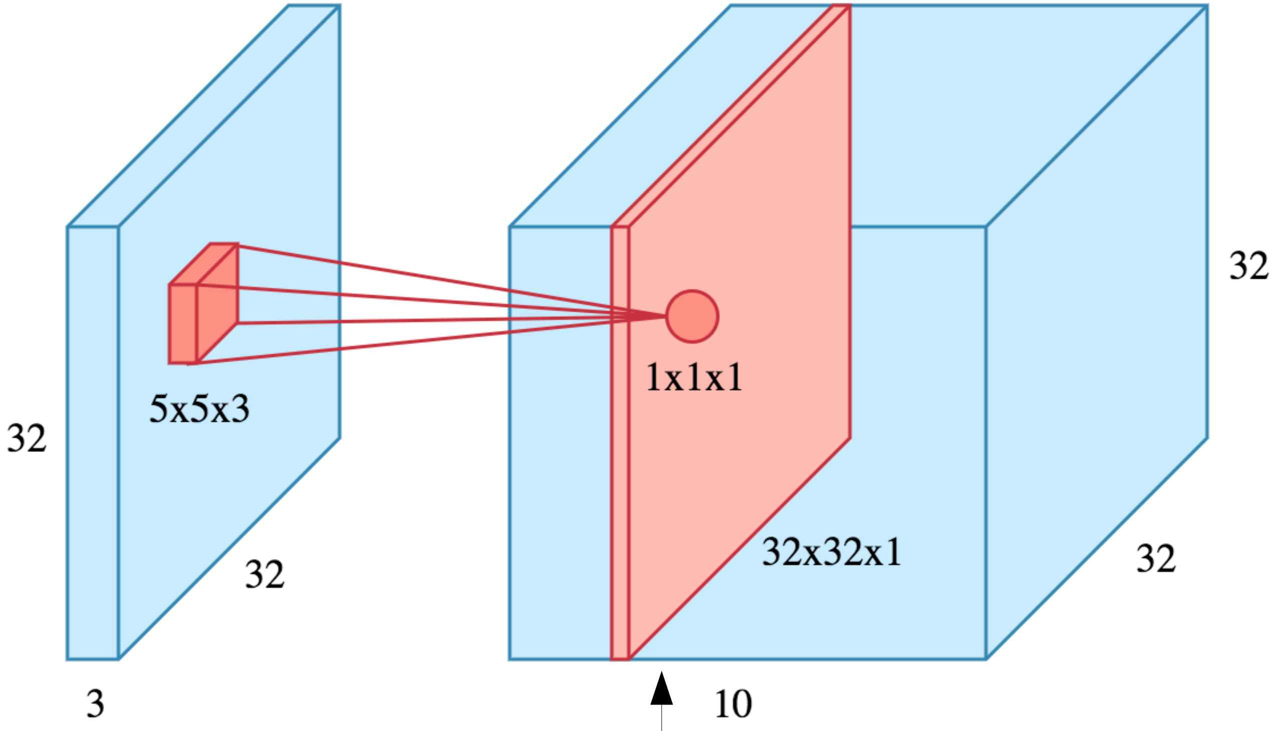
The stride can be more than 1, which downsamples the image.

Clearly not all strides are possible. For instance in this image 2 is ok, but 3 would not work.

$w =$

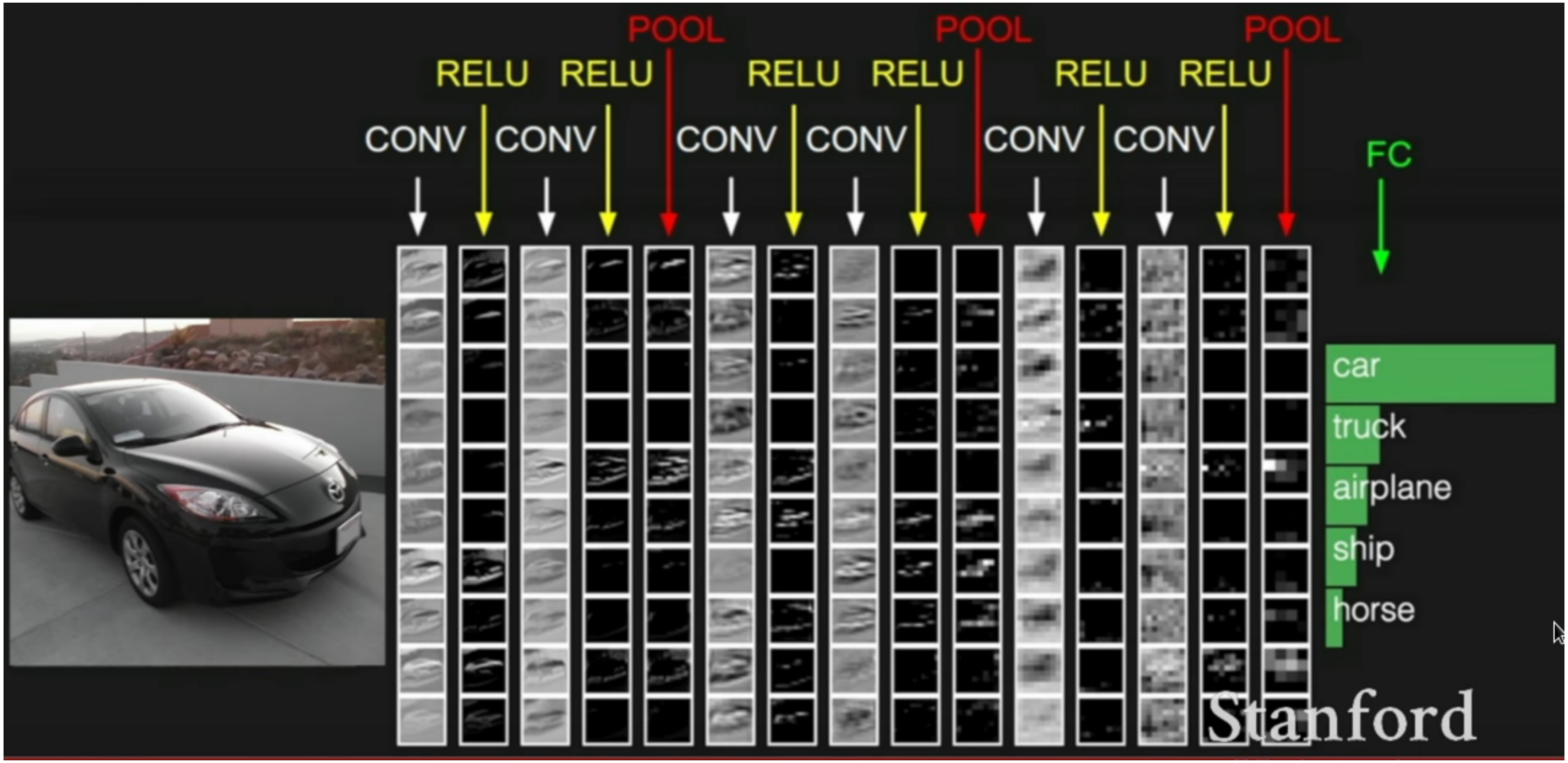
1	1	1
0	0	1
0	0	1

Images and filters

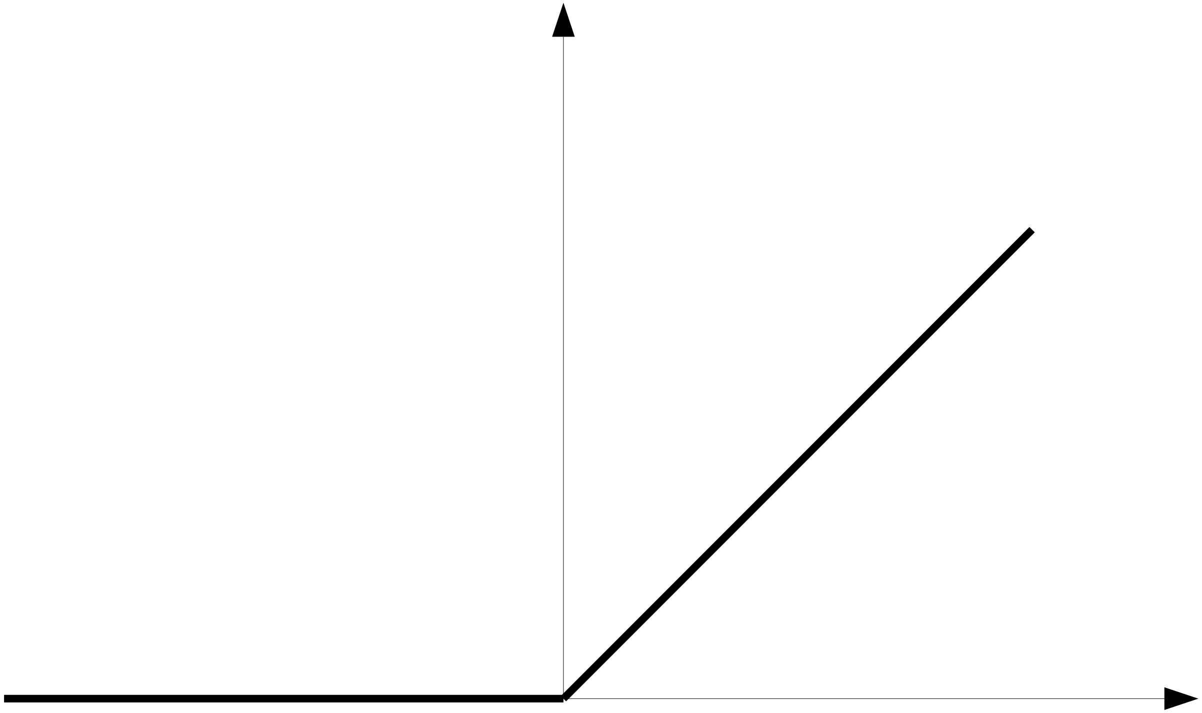


Activation layers (one per filter)

Architecture



Rectified Linear Units (ReLUs)



$$o(x)=\begin{cases} 0 & \text{if} \quad x<0 \\ x & \text{if} \quad x\geq 0 \end{cases}$$

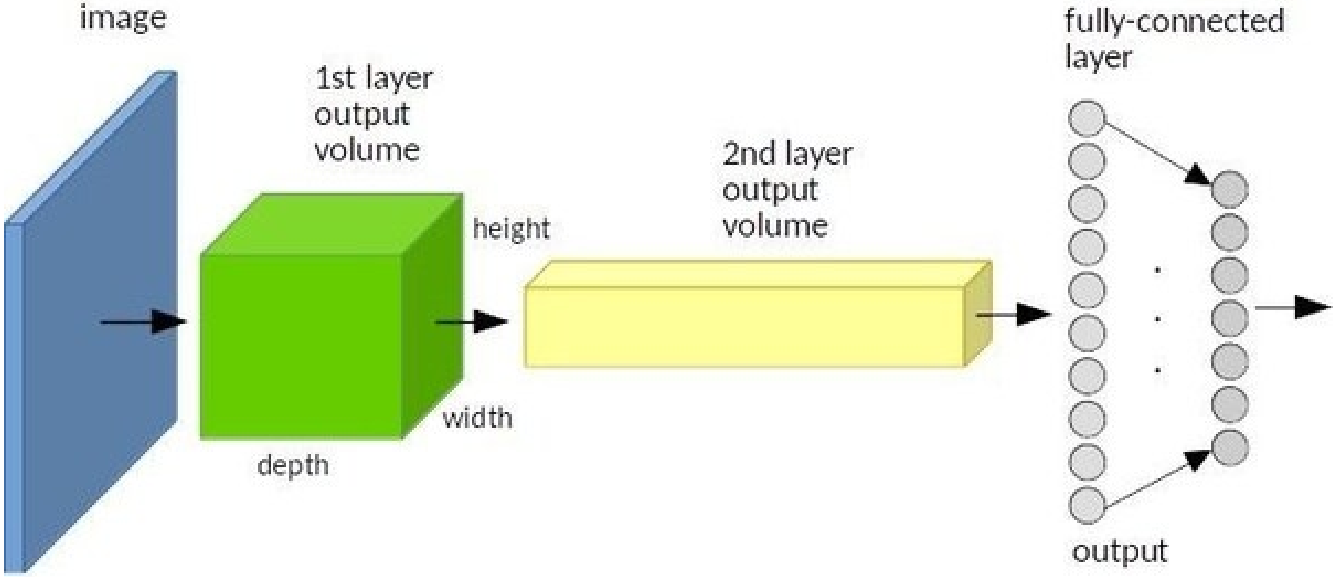
(Max) Pooling

1	1	1	1
1	1	1	2
3	4	4	5
2	2	1	2

1	2
4	5

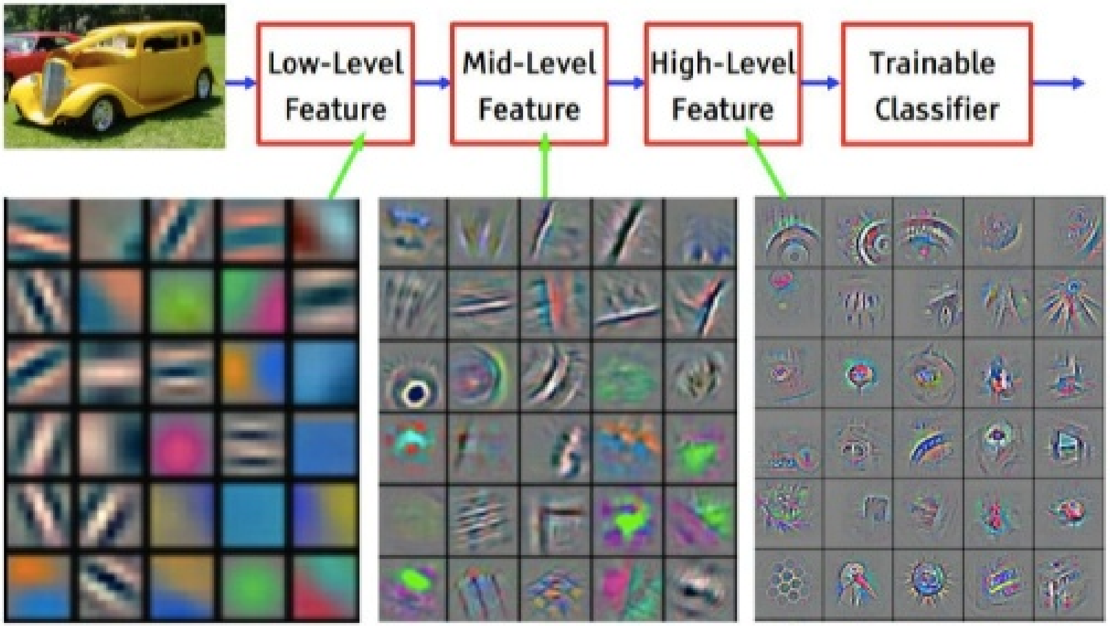
Max pooling is the most common way to downsample the image, in order to focus on higher-level patterns.

From convolutional to MLP



Features

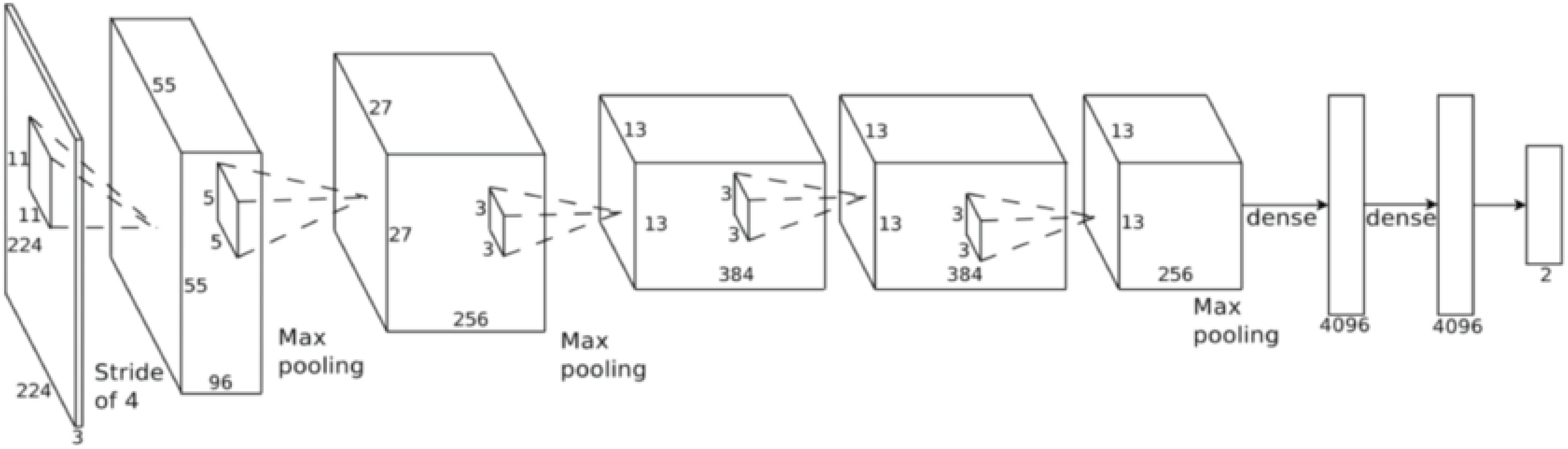
Convolutional Neural Network



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

Architectures

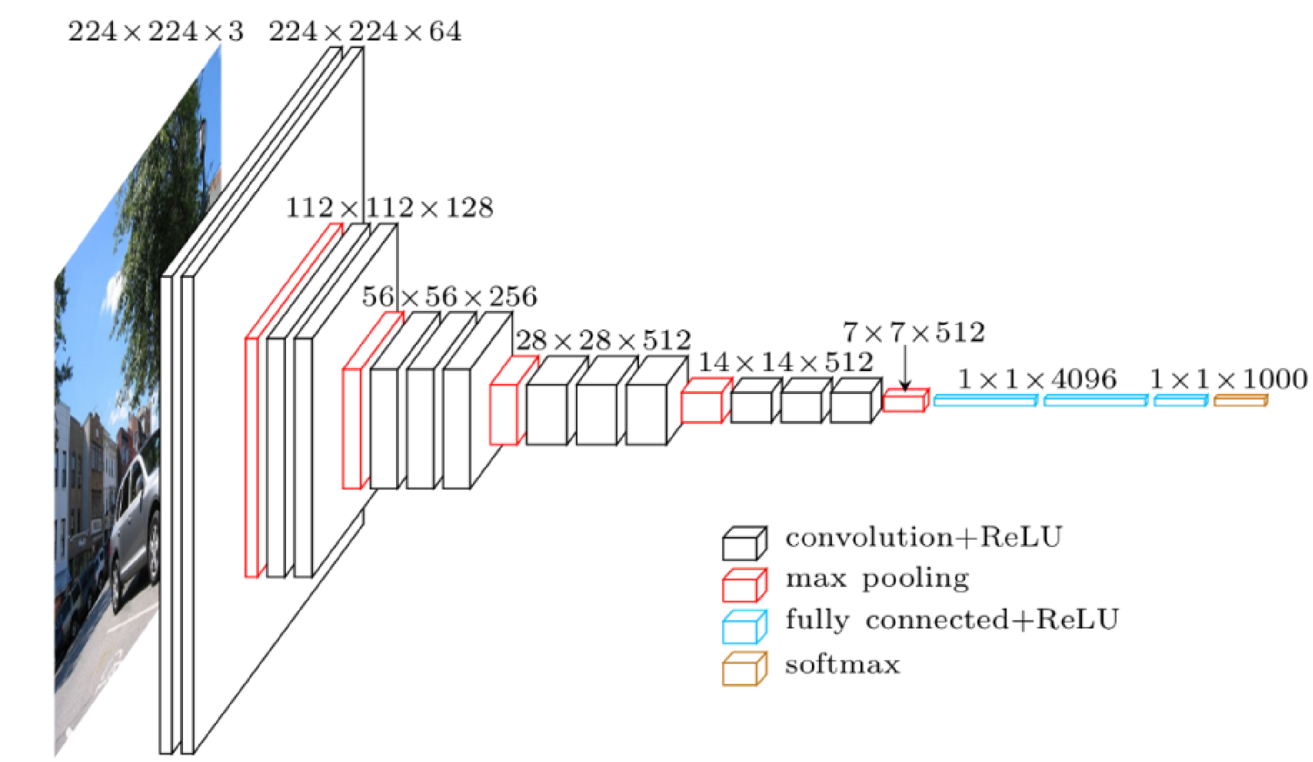
AlexNet Krizhevsky et al. in 2012



60 million parameters

Architectures

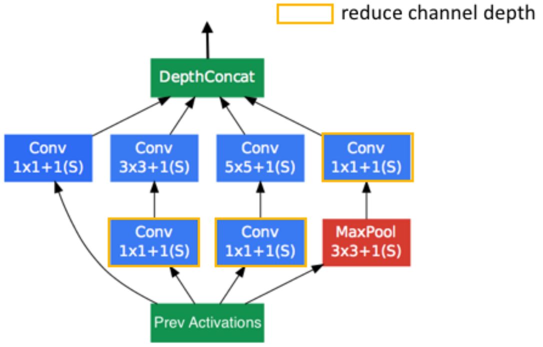
VGG16 Simonyan and Zisserman 2014



138 million parameters

Architectures

Inception (Google) Szegedy, et al 2015



🟦 convolution

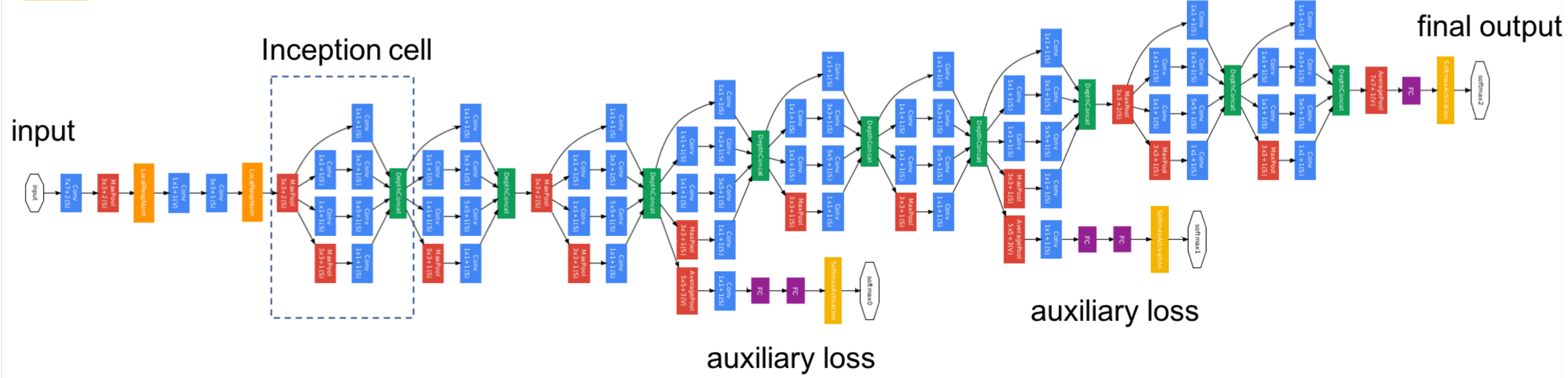
🟥 max pooling

🟩 channel concatenation

🟧 channel-wise normalization

🟪 fully-connected layer

🟨 softmax



5 million parameters, then revised with 23 million parameters

Conclusion

Learning outcomes

- Describe the main elements of a Convolutional Neural Network (CNN)
- Compute the convolution between a filter and an image
- Assemble an architecture for a CNN