

Convolutional Neural Networks

Olivier Dubrule/Navjot Kukreja

1

Objectives of the Day

- Understand what Convolutional Neural Networks (CNNs) are
- Be at ease with calculations associated with the CNN parameters
- Get familiar with classical CNN structures on well-known examples

2

Imperial College
London

Convolutional Neural Networks

1. Convolutional Neural Networks
2. Pooling and Fully Connected Layers
3. Examples

3

Imperial College
London

Computer Vision Topics

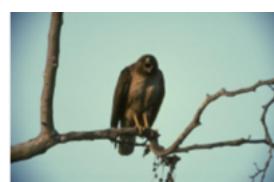
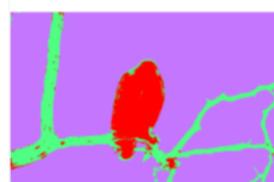


Image Segmentation



Image Classification



Object Localization (1 object) or Detection (several objects)

4

Imperial College
London

MNIST

6	5	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7
8	9	0	1	2	3	4	5	6	7	8	9	6	4	2	6	4	7	5	5
4	7	8	9	2	9	3	9	3	8	2	0	9	8	0	5	6	0	1	0
4	2	6	5	5	5	4	3	4	1	5	3	0	8	3	0	6	2	7	1
1	8	1	7	1	3	8	5	4	2	0	9	7	6	7	4	1	6	8	4
7	5	1	2	6	7	1	9	8	0	6	9	4	9	9	6	2	3	7	1
9	2	2	5	3	7	8	0	1	2	3	4	5	6	7	8	0	1	2	3
4	5	6	7	8	0	1	2	3	4	5	6	7	8	9	1	2	1	2	1
9	9	8	5	3	7	0	7	7	5	7	9	9	4	7	0	3	4	1	4
4	7	5	8	1	4	8	4	1	8	6	4	4	6	3	5	7	2	5	9

Extract from the MNIST dataset

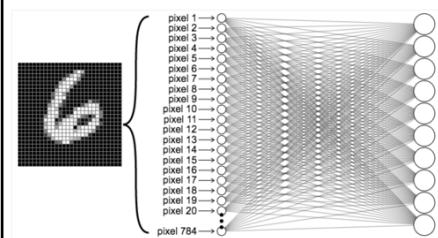
MNIST: Modified National Institute of Standards and Technology database

5

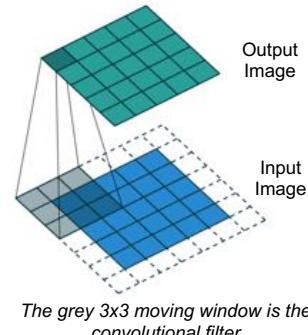
Imperial College
London

What are Convolutional Neural Networks?

FeedForward NN Approach on MNIST:
Spatial Organization is Lost



A CNN will take into account the
spatial organization of the dataset



The grey 3x3 moving window is the
convolutional filter

6

Imperial College London

What is the Convolution Operation?

The diagram illustrates the convolution operation. It shows an input image (3x3 grid) and a convolution filter (3x3 grid). The filter slides across the input image to produce an output image (3x3 grid).

Input Image:

3	3	2	1	0
0	0	1	1	1
1	1	2	2	2
2	0	0	2	2
2	0	0	0	1

Output Image:

12	10	9
10	8	7
9	7	6

Convolution Filter:

0	1	2
2	2	0
0	1	2

From Dumoulin & Visin, 2018

7

Imperial College London

Exercise: Calculate the Output Image

Input Image

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

* Convolution Filter (or Kernel)

2	-3	2
1	0	1
2	-3	3

= Output Image

12	20
13	9

8

Imperial College London

Filters (or Kernels) for Edge Detection

Filter for Detecting Vertical Edges

Input Image	Filter (or Kernel)	Output Image
$\begin{bmatrix} 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \\ 10 & 10 & 10 & 0 & 0 & 0 \end{bmatrix}$ 	$\begin{array}{c} * \\ \text{Convolution Operator} \end{array}$ $\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$ 	$=$ $\begin{bmatrix} 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \\ 0 & 30 & 30 & 0 \end{bmatrix}$ 

Inspired from Andrew Ng

9

Imperial College London

Which Filter should I use? (1)

$\begin{array}{c} \text{Blur} \\ \hline \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix} \end{array}$	 
$\begin{array}{c} \text{Horizontal Edges} \\ \hline \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \end{array}$	 

<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

10

Imperial College London

Which Filter should I use? (2)

Edges		
-1	-1	-1
-1	8	-1
-1	-1	-1

Sobel Filters					
-1	-2	-1	-1	0	1
0	0	0	-2	0	2
1	2	1	-1	0	1

Horizontal
Vertical

<https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

11

Imperial College London

Idea Behind Convolutional Neural Networks:

Parametrize the Filter and Optimize its Coefficients Based on the Objective!

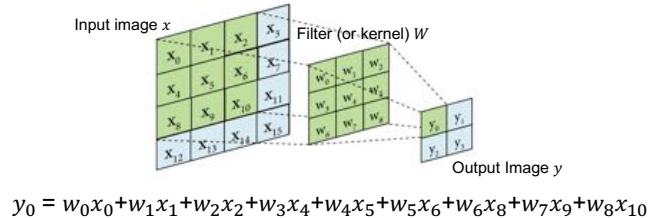
$$W * x = y$$

$$y_0 = w_0x_0 + w_1x_1 + w_2x_2 + w_3x_4 + w_4x_5 + w_5x_6 + w_6x_8 + w_7x_9 + w_8x_{10}$$

12

Imperial College
London

Making Each Kernel Operation Non-Linear



A bias term can be added

$$y_0 = w_0x_0 + w_1x_1 + w_2x_2 + w_3x_4 + w_4x_5 + w_5x_6 + w_6x_8 + w_7x_9 + w_8x_{10} + b$$

And a non-linear transformation is applied by an activation function g

$$y_0 = g(w_0x_0 + w_1x_1 + w_2x_2 + w_3x_4 + w_4x_5 + w_5x_6 + w_6x_8 + w_7x_9 + w_8x_{10} + b)$$

13

Imperial College
London

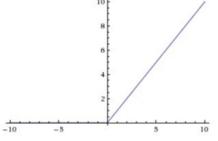
Linear Convolution Operation....



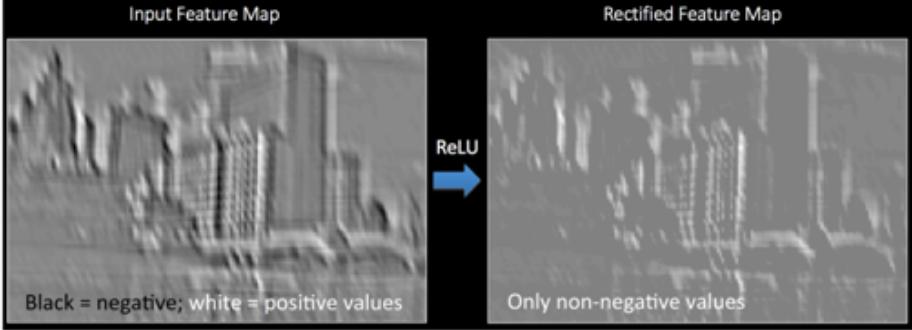
<https://ujwlkarn.files.wordpress.com/2016/08/giphy.gif?w=480&zoom=2>

14

Imperial College London



.....Followed by ReLU Activation



Input Feature Map
Rectified Feature Map

Black = negative; white = positive values
Only non-negative values

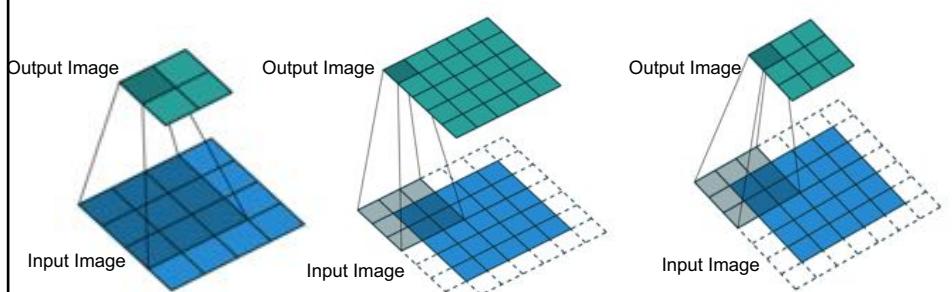
ReLU

<https://uijwalkarn.me/2016/08/11/intuitive-explanation-convnets/>

15

Imperial College London

Key Parameters of Convolutional Neural Networks

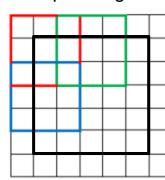


No padding, Stride 1 Padding 1, Stride 1 Padding 1, Stride 2

16

Imperial College
London

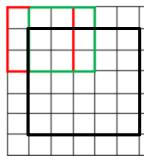
The Two-Dimensional Convolution Parameters

<i>Parameters</i>	<i>Output Image Size</i>
Input Image Size: $n \times n$	$\left(\frac{n+2p-f}{s} + 1\right) \times \left(\frac{n+2p-f}{s} + 1\right)$
Filter (or Kernel) Size: $f \times f$	
Padding: p	Input Image
Stride : s	Output Image
	 $n = 5 \ p = 1 \ f = 3 \ s = 2$

17

Imperial College
London

The Two-Dimensional Convolution Parameters

<i>Parameters</i>	<i>Output Image Size</i>
Image Size: $n \times n$	$\left(\frac{n+2p-f}{s} + 1\right) \times \left(\frac{n+2p-f}{s} + 1\right)$
Filter (or Kernel) Size: $f \times f$	
Padding: p	Input Image
Stride : s	Output Image
	 $n = 5 \ p = 1 \ f = 3 \ s = 1$

18

Imperial College London

If the Two-Dimensional image has 3 RGB channels...

$6 \times 6 \times 3$

$3 \times 3 \times 3$

$=$

$4 \times 4 \times 1$

27 operations each time (28 if bias term)

Each slice of the filter can detect different features in each of the 3 colour channels, for instance vertical edges in Red, horizontal edges in Green and vertical edges in Blue!

Andrew Ng https://www.youtube.com/watch?v=KTB_OFoAQcc&list=PLkDaE6sCZn6Gl29AcE31iwdVwSG-KnDzF&t=0s&index=7

19

Imperial College London

If Different Filters are Applied ...

$6 \times 6 \times 3$

$n_w = n_h = 6$

$n_c = 3$

$3 \times 3 \times 3$

Filter 1

$3 \times 3 \times 3$

Filter 2

$=$

4×4

$=$

4×4

2 output channels or features.
The depth of the output is 2.

Exercise: if my input Image is 50x50 with 10 channels, and if I apply 25 filters each of size 3x3x10 with a stride of 1, no padding, what is the size and number of channels of the output image?

Answer: size is 48x48. with 25 channels

Andrew Ng https://www.youtube.com/watch?v=KTB_OFoAQcc&list=PLkDaE6sCZn6Gl29AcE31iwdVwSG-KnDzF&t=0s&index=7

20

Imperial College
London

Number of Weights to Train for a Given Layer of the CNN

If I have 100 filters of size $4 \times 4 \times 3$ (each with bias term) in layer l , what is the total number of weights (or degrees of freedom) to train for this layer?

$$\text{Number of filters} \times (\text{size of filter} + 1 \text{ bias term}) =$$

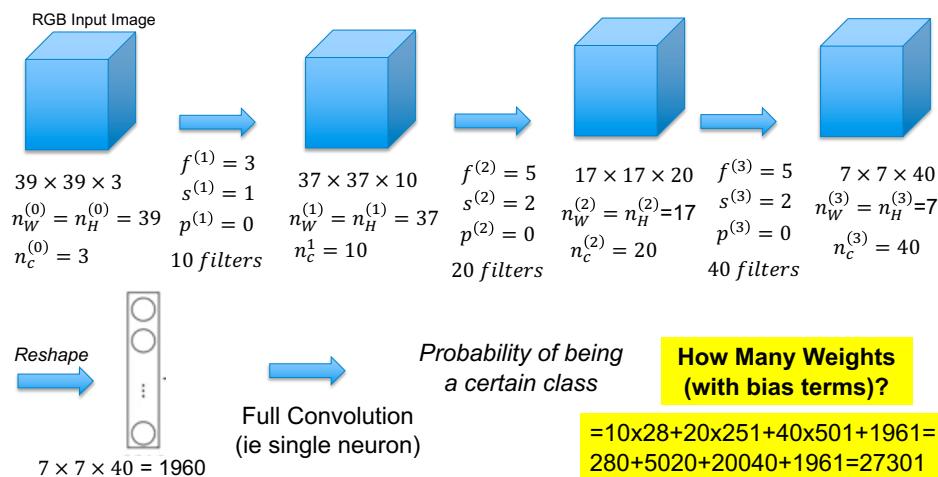
$$100 \times (4 \times 4 \times 3 + 1) = 100 \times 49 = 4900 \text{ weights}$$

Note that the number of parameters is independent of the size of the input image, which was not the case for feed-forward neural networks!

21

Imperial College
London

Our first CNN Example: Recognize Whether Image of Cat or Dog



Inspired from Andre Ng

22

Imperial College
London

Counting the Degrees of Freedom (or Parameters) for a layer l

Assume layer l has $n_c^{(l)}$ filters of size $f^{(l)}$, a padding $p^{(l)}$ and a stride $s^{(l)}$

Assume the input image has size $n_H^{(l-1)} \times n_W^{(l-1)} \times n_c^{(l-1)}$
and the output image has size $n_H^{(l)} \times n_W^{(l)} \times n_c^{(l)}$

The size of each filter is $f^{(l)} \times f^{(l)} \times n_c^{(l-1)}$ and the depth or number of output features is the number of filters $n_c^{(l)}$

And we can generalize the formula $n_H^{(l)} = \frac{n_H^{(l-1)} + 2p^{(l)} - f^{(l)}}{s^{(l)}} + 1$ and $n_W^{(l)} = \frac{n_W^{(l-1)} + 2p^{(l)} - f^{(l)}}{s^{(l)}} + 1$

And the total number of degrees of freedom (or parameters) of layer l is:

$$\text{Number of filters} \times (\text{size of filter} + 1 \text{ bias term}) = n_c^{(l)} \times (f^{(l)} \times f^{(l)} \times n_c^{(l-1)} + 1)$$

23

Imperial College
London

Convolutional Neural Networks

1. Convolutional Neural Networks

2. Pooling and Fully Connected Layers

3. Examples

24

Imperial College
London

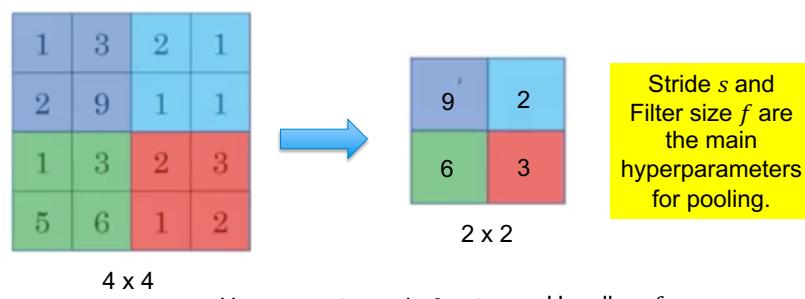
Three Types of Layers in a Convolutional Network

- Convolutional Layers (CONV)
- Fully Connected Layers (FC)
- Pooling Layers (POOL)

25

Imperial College
London

Example of Max Pooling Layer (for Downsampling)



Here $s = 2$ and $f = 2$ Usually $f = s$

Pooling makes the input representations (feature dimension) smaller and more manageable for the next layer. Max Pooling is used because it may be interesting to keep the high values for the activation of the next layer as they may characterize some important features. Pooling reduces the number of parameters and computations in the network, therefore controlling overfitting.

26

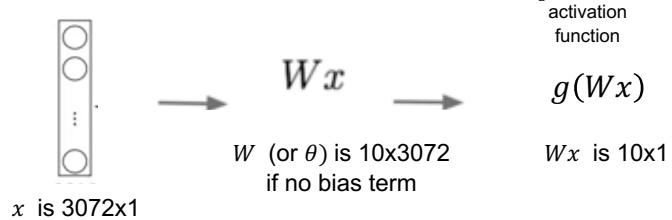
Imperial College
London

We already know Fully Connected Layers!

Transform for example $32 \times 32 \times 3$ image into 10×1 vector.

1. Reshape image into 1D vector x of dimension $32 \times 32 \times 3 = 3072$

2. Apply full convolution through matrix W

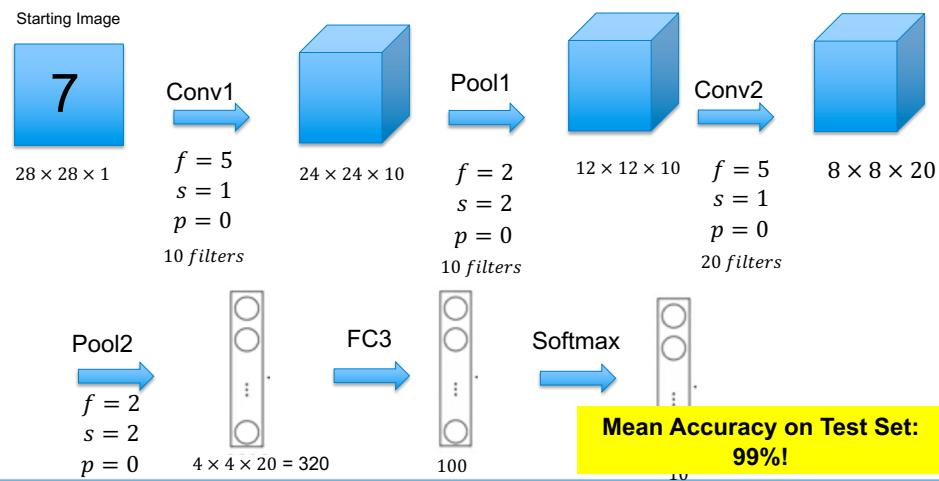


Fully connected layers often have many weights as they connect every neuron in input layer to every neuron in output layer.

27

Imperial College
London

MNIST CNN Example (The LeNet-5 Architecture from LeCun)



28

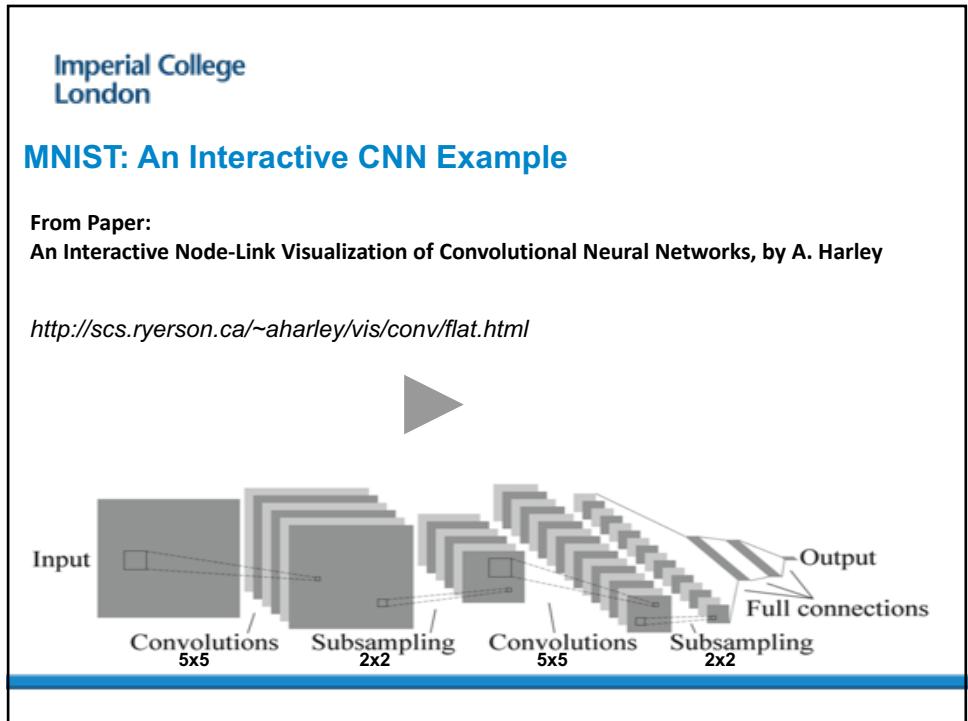
Imperial College London

MNIST: Summary of CNN Layers (Convolutions with Bias Terms)

	Size of input image n	Number of input channels	f	p	s	Size of output image $(n+2p-f)/s+1$	Number of output channels or filters	Number of output neurons	Size of Filter + 1	Number of Parameters
Conv1	28	1	5	0	1	24	10	5760	26	260
Pool1	24	10	2	0	2	12	10	1440		
Conv2	12	10	5	0	1	8	20	1280	251	5020
Pool2	8	20	2	0	2	4	20	320		
	Size of Input						Number of output neurons			
FC1	320						100	321	32100	
Softmax	100						10	101	1010	
							Total Neurons	8910	Total Parameters	38390

Formula used for last column:
Number of Parameters = Number of output channels * (Size of filter+1) (as we assume there is a bias term)

29



30

Imperial College London

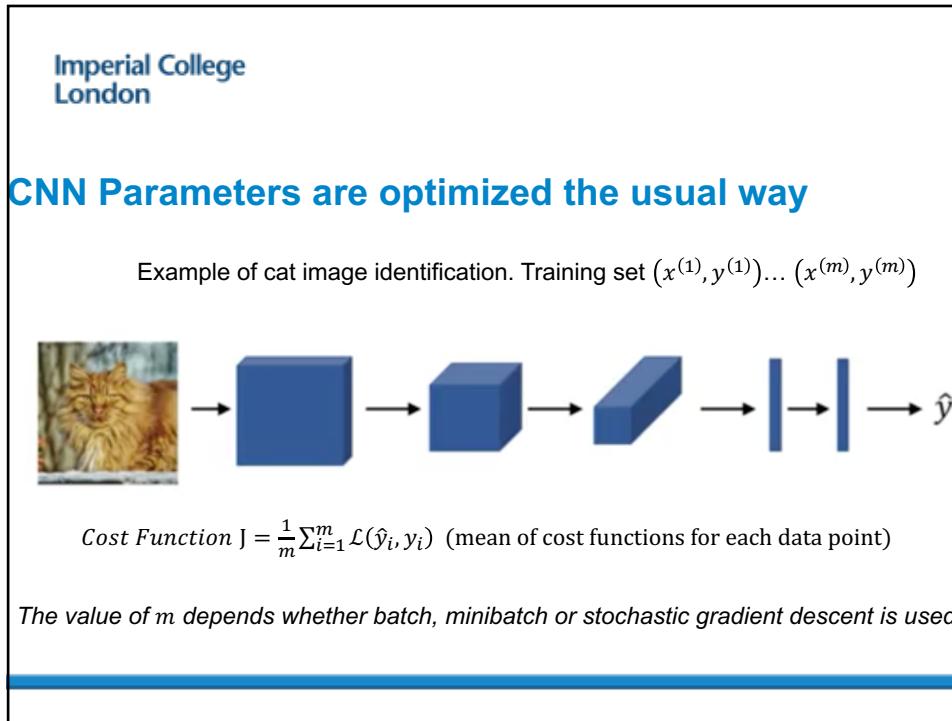
Structure of Harley's Network

From Paper:
An Interactive Node-Link Visualization of Convolutional Neural Networks, by A. Hartey

	Size of input image n	Number of input channels	f	p	s	Size of output image $(n+2p-f)/s+1$	Number of output channels or filters	Number of output neurons	Size of Filter + 1	Number of Parameters
Conv1	32	1	5	0	1	28	6	4704	26	156
MaxPool	28	6	2	0	2	14	6	1176		
Conv2	14	6	5	0	1	10	16	1600	151	2416
MaxPool	10	16	2	0	2	5	128	3200		
	Size of input						Number of output neurons			
FC1	3200						120	3202		384120
FC2	120						100	121		12100
Softmax	100						10	101		1010
							Total Neurons	10910	Total Parameters	399802

Formula used for last column:
Number of Parameters = Number of output channels × [Size of filter+1] (as we assume there is a bias term)

31



32

Imperial College
London

Convolutional Neural Networks

1. Convolutional Neural Networks
2. Pooling and Fully Connected Layers
3. Examples

33

Imperial College
London

Some of the Best-Known Datasets

Very deep neural networks work best when trained on very large datasets!

- MNIST: Handwritten digits, 60000 Training Images, 10000 Test Images
- CIFAR-10 / CIFAR-100: 50k Training, 10k Test Images of 10 (CIFAR-10) or 100 (CIFAR-100) classes

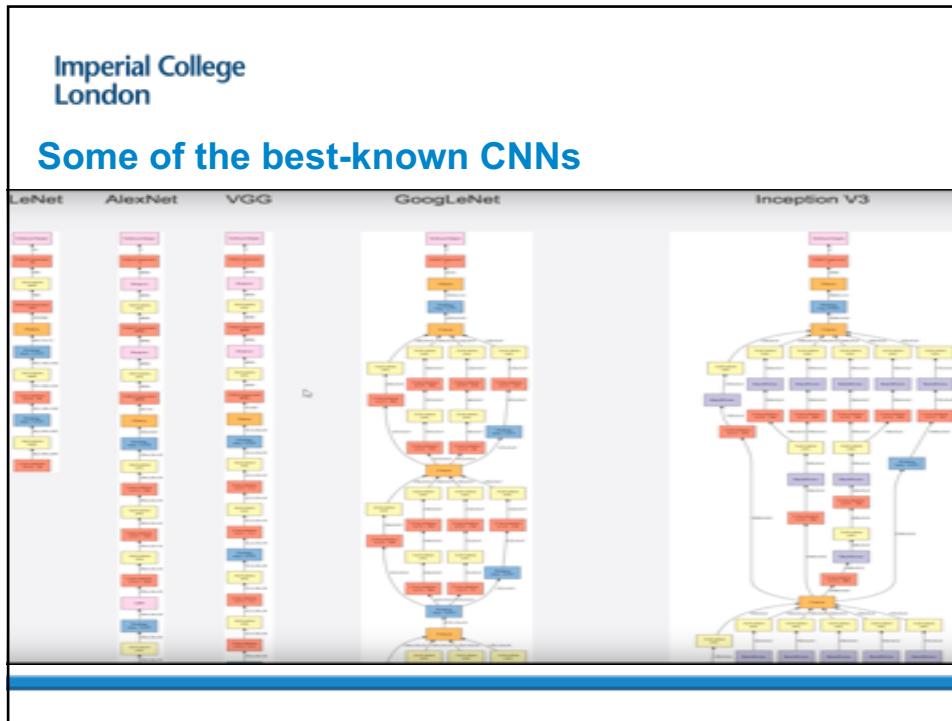
Color Images are 32x32, Task: Classification <https://www.cs.toronto.edu/~kriz/cifar.html>



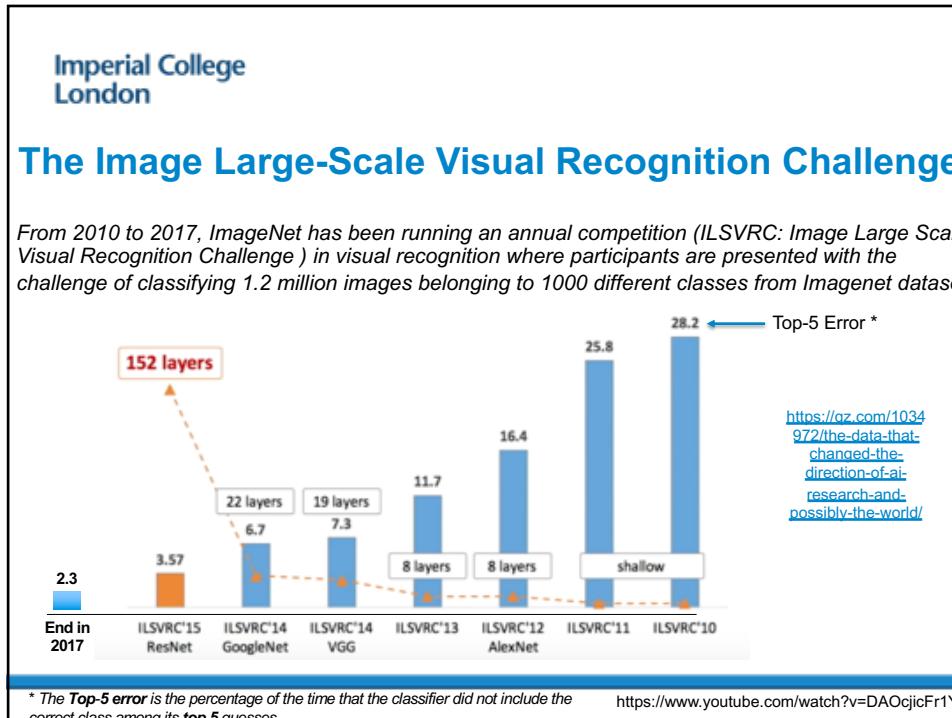
- Imagenet: > 15 Million Images in 20,000 classes! <https://en.wikipedia.org/wiki/ImageNet>



34



35



36

Imperial College
London

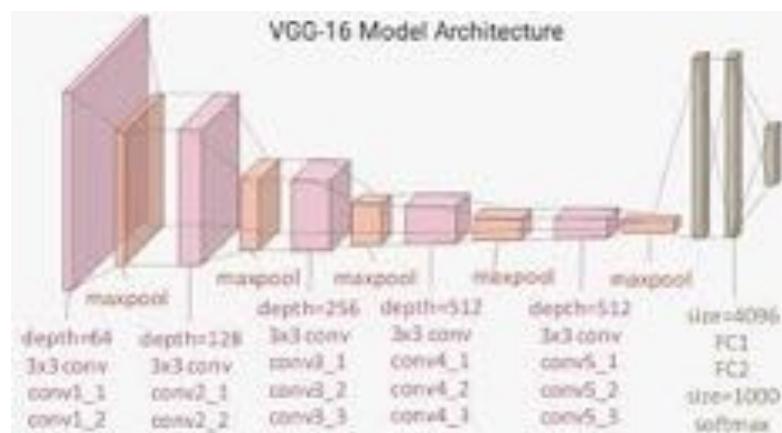
Some of the best-known CNNs

- **LeNet-5** (Le Cun et al, 1998) for Digit Recognition: Architecture is CONV-POOL-CONV-POOL-FC-FC
- **AlexNet** (ILSRVC 2012): Very similar to Le-Net-5, with just more layers (ReLU, Dropout 0.5 and Data Augmentation). About 60 MM parameters for 8 layers. First CNN-based winner of ILSRVC!
- **VGG** (co-winner of ILSRVC 2014): More layers (16 to 19) but very small filters for convolution (3x3, Pad 1, Stride 1, in stacked layers) and pooling (2x2, stride 2) . More non-linearities , with less parameters in CONV layers. But VGG16 has 138 MM parameters, mainly due to last Fully-Connected layers. VGG19 only slightly better than VGG16.

37

Imperial College
London

Some of the best-known CNNs VGG

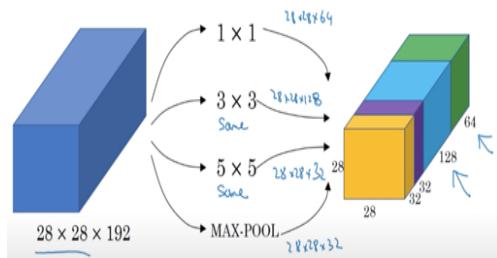


38

Imperial College
London

Some of the best-known CNNs: Inception Module

- **GoogLeNet/Inception** (co-winner of ILSRVC 2014): More layers (22), based on “Inception” module (“network within a network”), no Fully-Connected layer, only 5 Million parameters! General Philosophy based on basic Inception Module (“Network within a Network”)



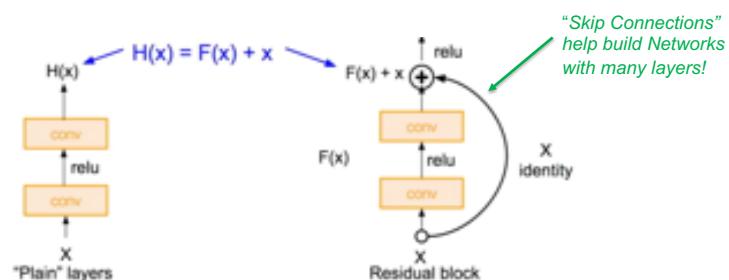
Example from Andrew Ng's C4W2L06

39

Imperial College
London

Some of the best-known CNNs: ResNet

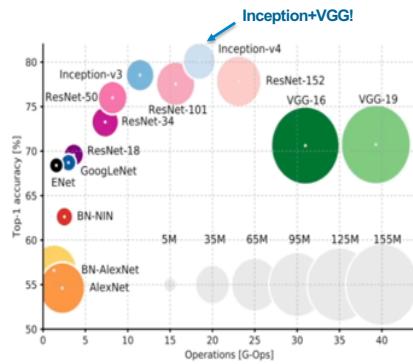
- **Resnet (winner of ILSRVC 2015):** Very deep network (152 layers) using residual connections. No FC layers. Did better than human performance!



40

Imperial College
London

Comparing Accuracy, Speed and Memory Usage



- Download open model implementations pre-trained on large datasets:
Pytorch: Torchvision <https://pytorch.org/docs/stable/torchvision/models.html>

<https://www.youtube.com/watch?v=DAOcjicFr1Y>

41

Imperial College
London

Transfer Learning (1)

The most well-known CNN designs are available on-line and have been successfully trained on very large number of images (1,000,000s).

In many applications we often work from a relatively small number of images.

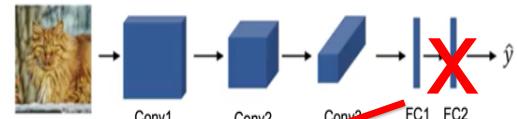
Why not start from an existing trained CNN sharing an objective of a similar nature (such as Classification) and tailor it to our application?

42

Imperial College
London

Example of Transfer Learning Approach

Take existing trained network such as Inception v3 model, trained on ImageNet dataset for differentiating between 1,000 different classes of images.



Rudstone Sample
Re-train last layer of the network
on images of interest, such as
pictures of carbonate cores .



From Sharinia Kanagandran and
Cedric John, Imperial College

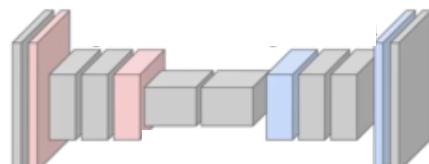
43

Imperial College
London

Semantic Segmentation with Down- and Up-Sampling



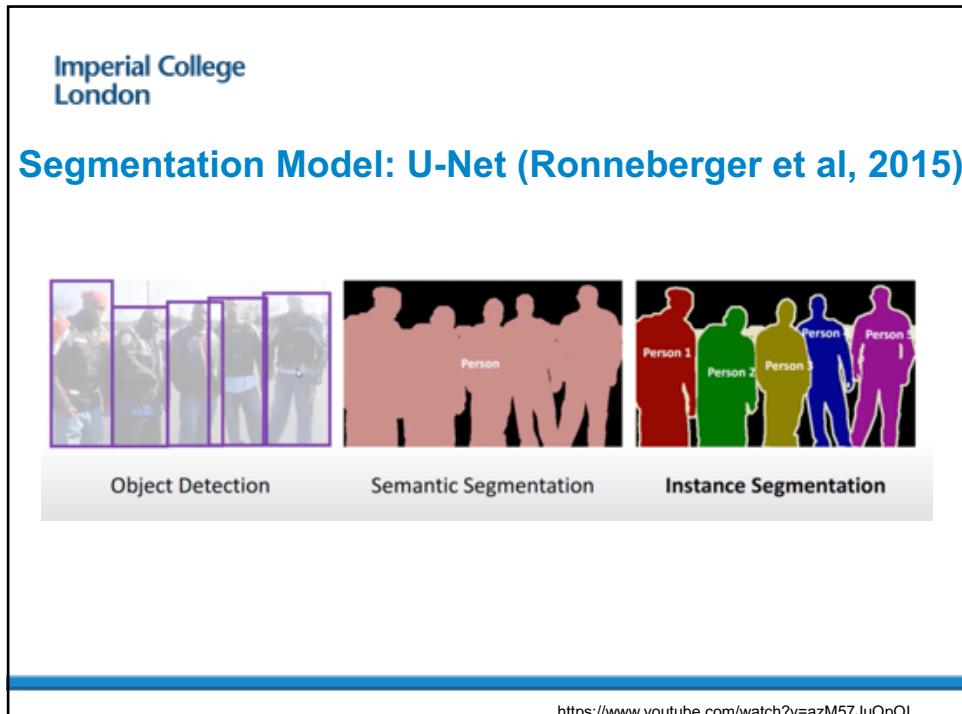
Input:
 $3 \times H \times W$



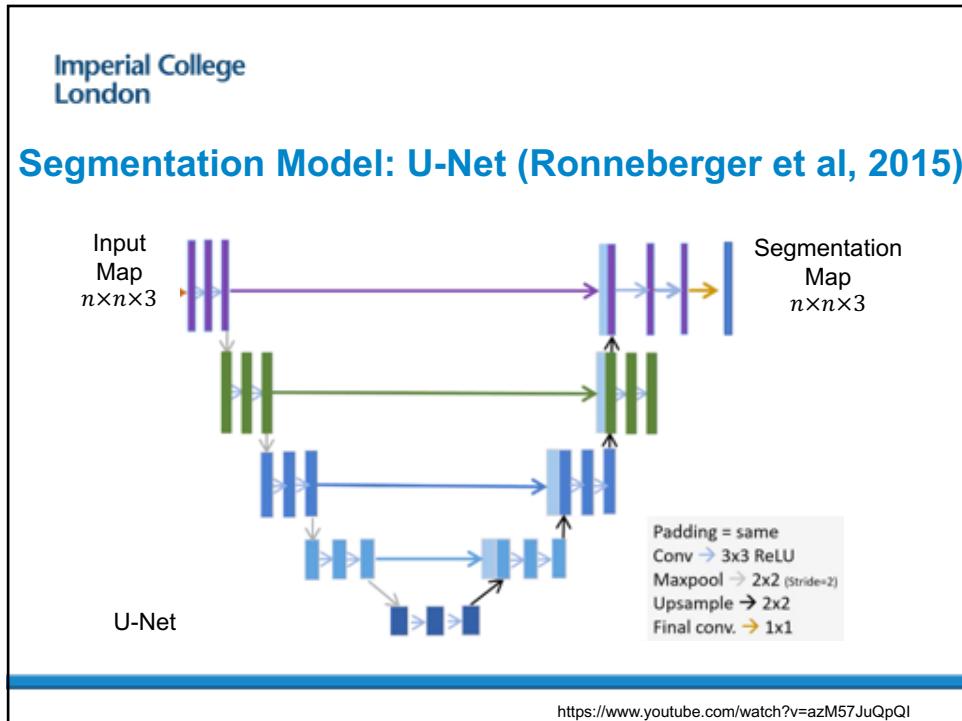
Predictions:
 $H \times W$

<https://www.youtube.com/watch?v=azM57JuQpQI>

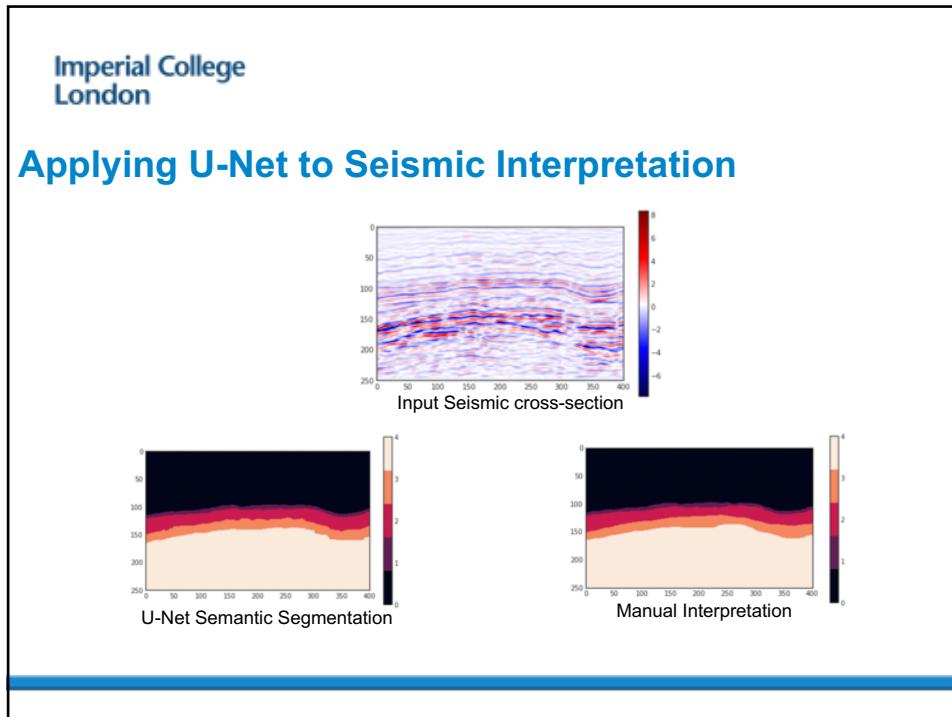
44



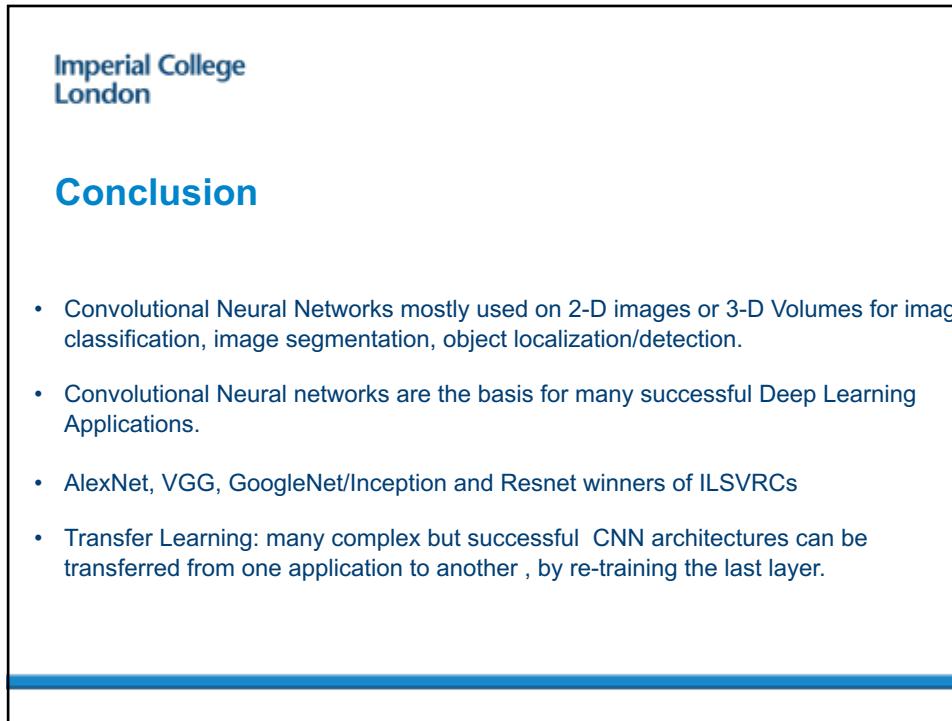
45



46



47



48