Convolutional Neural Networks

Training course: Machine learning for weather prediction

Vera Gahlen

ECMWF

vera.gahlen@ecmwf.int

Outline

- Motivation for convolutions
- What is a convolution?
- Convolution’s arithmetic
- Building a Convolutional Neural Network
- Connecting image structure and CNN architecture
- Popular CNN-based architectures - ResNets, U-nets

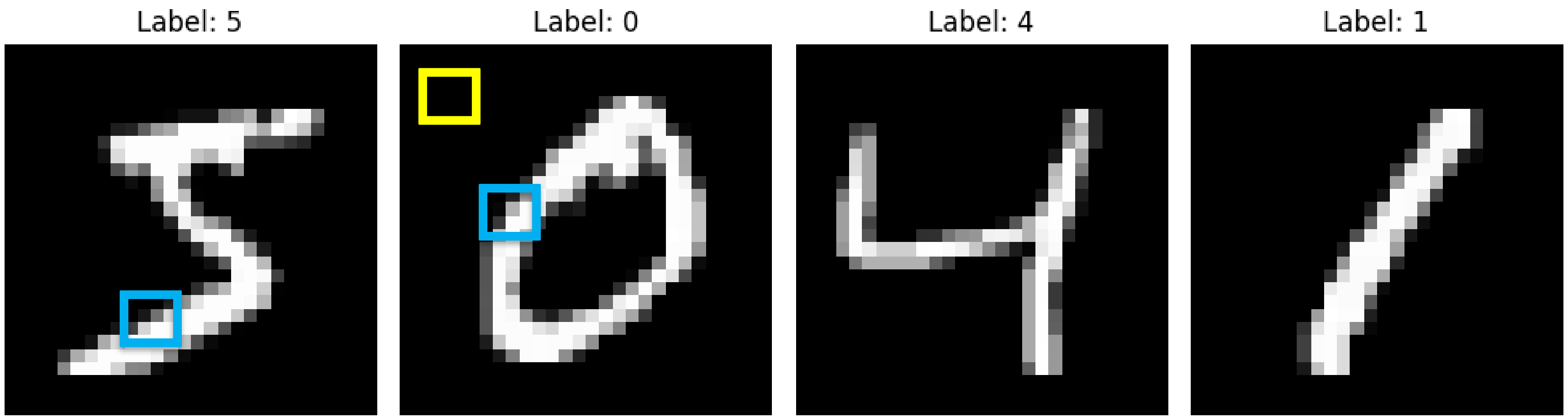
Motivation for convolutions

Why not use an MLP?

- Expressiveness?
  - MLPs are universal approximators — special architectures aren't needed to *represent* good solutions.
- Compute?
  - Well-designed architectures encode inductive biases that constrain the hypothesis space, enabling similar performance with less data and less computation.
  - Historically, before CNNs, training large networks for vision tasks was **infeasible — more compute alone wasn't enough**.
- Optimization!
  - Architectures shape the loss landscape, making good solutions ***discoverable* by gradient descent**.
  - Modern DL relies more on architectures than on optimizers to make training feasible.
- Generalization
  - Constraining the hypothesis space also helps prevent overfitting.

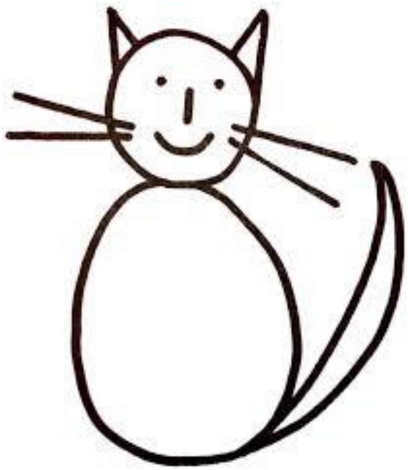
>> Architectures don't just tell us *what can be represented* — they determine *what can be **found***.

What structure do images have?



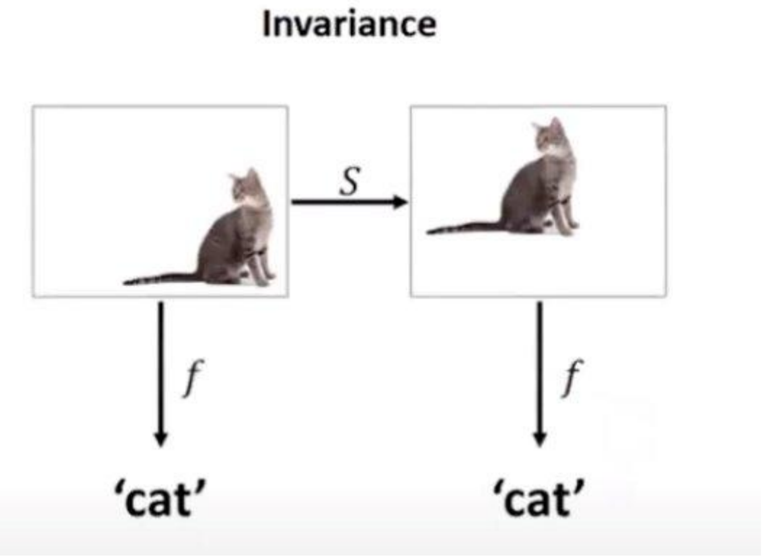
MNIST Dataset of handwritten digits

- **Spatial locality**
  To make sense of a pixel, we need (only!) the surrounding pixels
- **Compositional structure**
  Simple features (edges) combine into complex ones (textures, shapes)

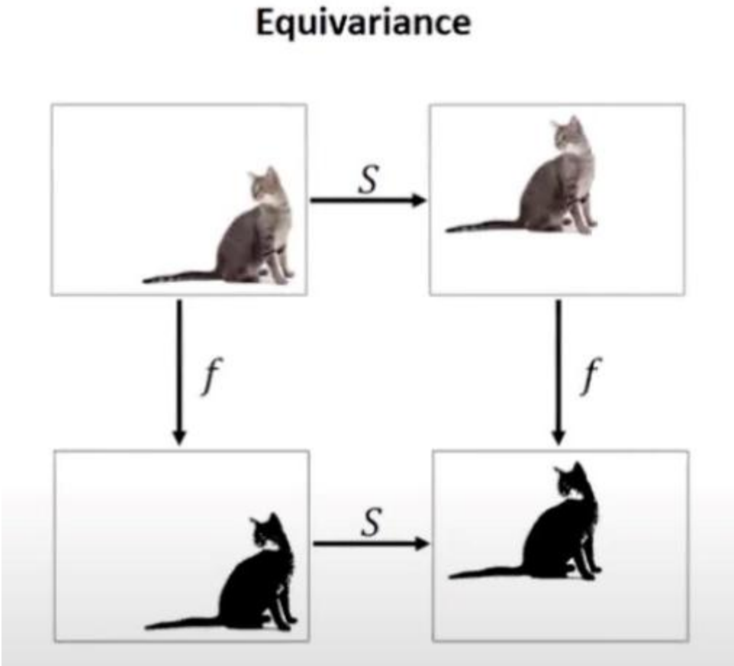


What structure do images have?

Translation Invariance



Translation Equivariance



05 Imperial's Deep learning course: Equivariance and Invariance
Bernhard Kainz
https://www.youtube.com/watch?v=a4Quhf9NhMY

Motivation for convolutions

What structure do images have?

- **Locality** — Neighboring pixels tend to form meaningful local structures (edges, corners, textures)
- **Compositional structure** — Simple features combine into complex ones
- **Translation equivariance** — Shifting the input shifts the output the same way

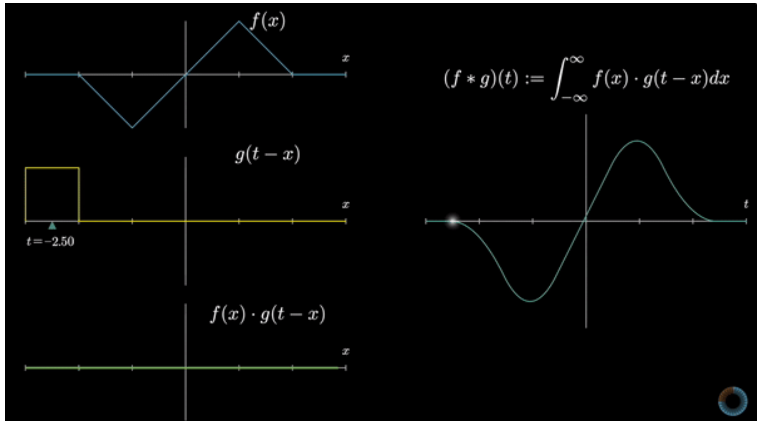
We want the model to perform **localized, translation-equivariant** *aggregation/extraction* of visual information

The architecture needs to suit the problem

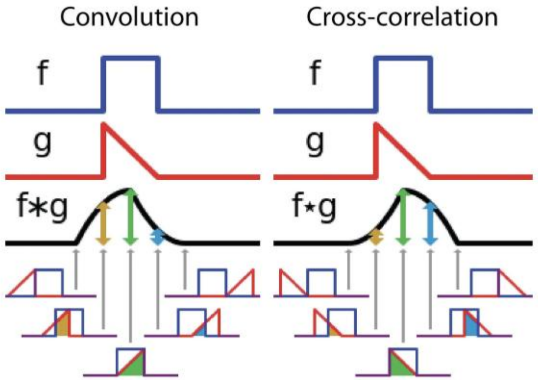
Sliding filters
apply the same local pattern extraction everywhere in the image

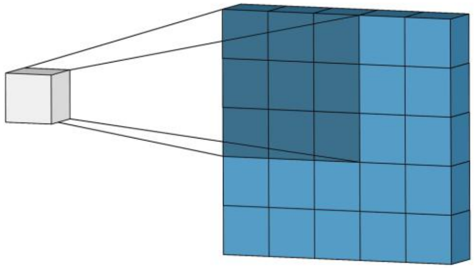
What is a convolution?

What is a convolution?



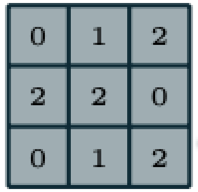
But what is a convolution?
3Blue1Brown





Intuitively Understanding Convolutions for Deep Learning
Towards Data Science

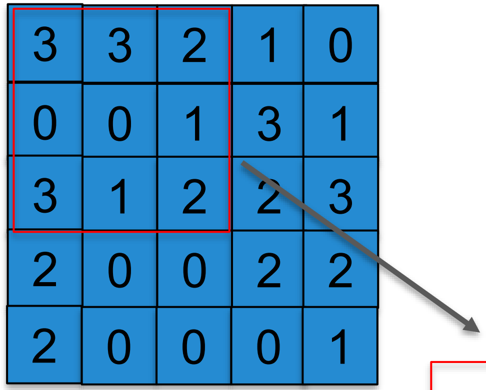
What is a convolution?

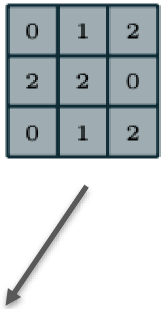


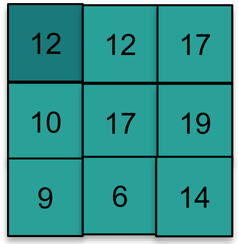
Kernel or Filter

Input Feature map

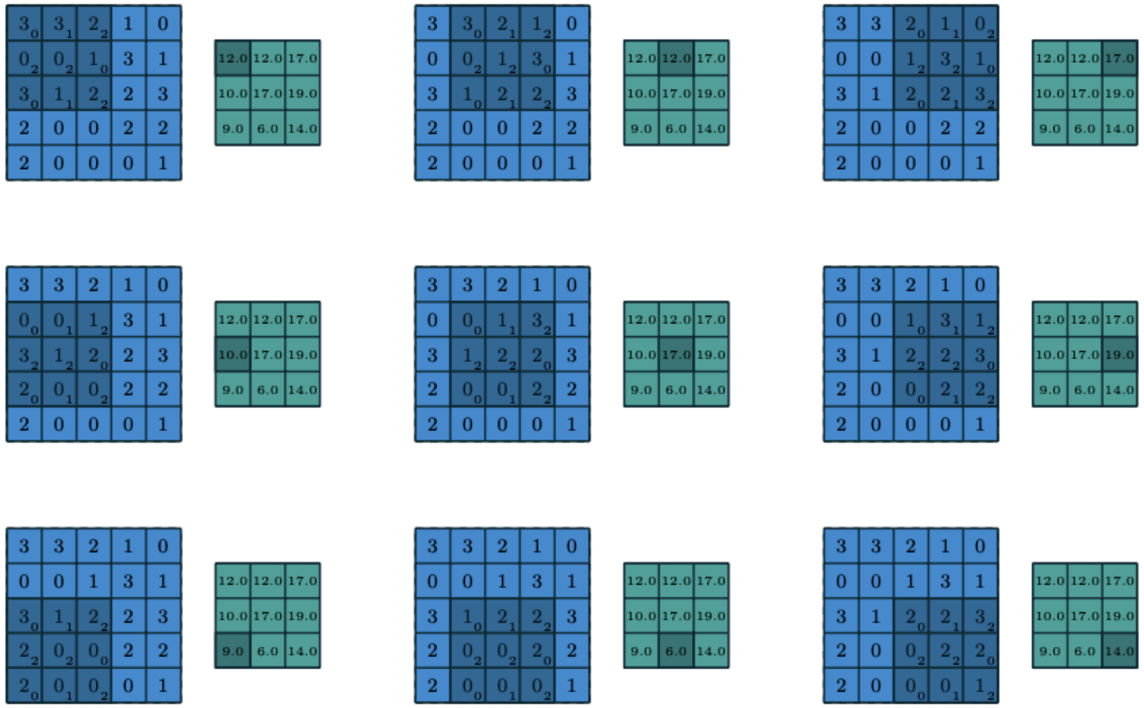
Output Activation Map







$$
\begin{aligned}
&= 0 * 3 + 1 * 3 + 2 * 2 \\
&+ 2 * 0 + 2 * 0 + 0 * 1 \\
&+ 0 * 3 + 1 * 1 + 2 * 2 \\
&= 12
\end{aligned}
$$



Convolve a filter with the image = spatially sliding it over the image and computing the dot product

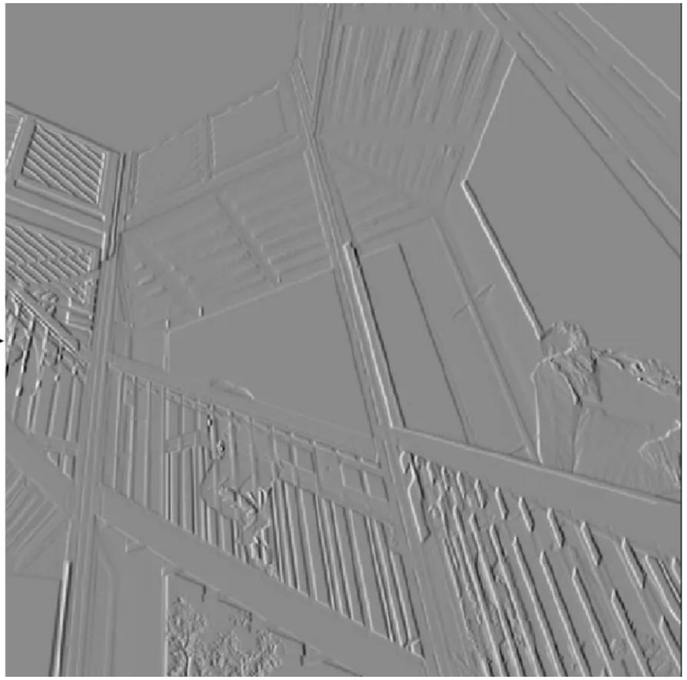
A guide to convolution arithmetic for deep Learning
Dumoulin V., Visin. F, 2018, arXiv:1603.07285

What is a convolution?

Automatic Feature Extraction

$$\begin{bmatrix} +1 & 0 & -1 \\ +2 & 0 & -2 \\ +1 & 0 & -1 \end{bmatrix}$$

Horizontal Sobel kernel



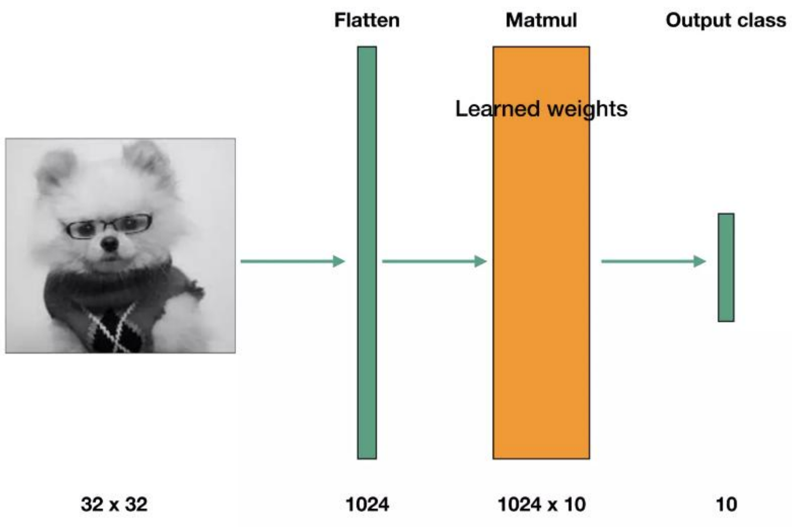
Applying a vertical edge detector kernel

https://setosa.io/ev/image-kernels/

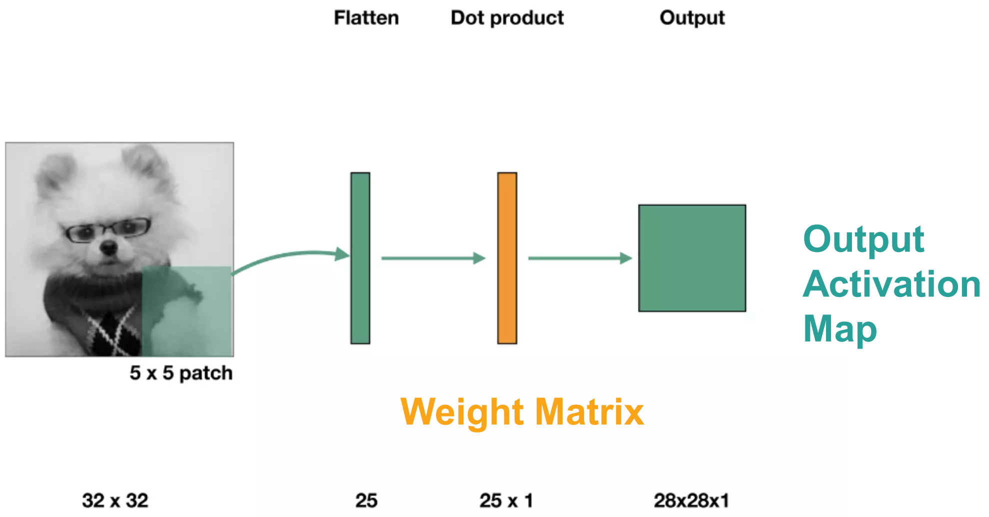
The kernel/filter is the **trainable** part of the convolutional layer

Convolution versus MLP

MLP(FC) Architecture



CNN Architecture

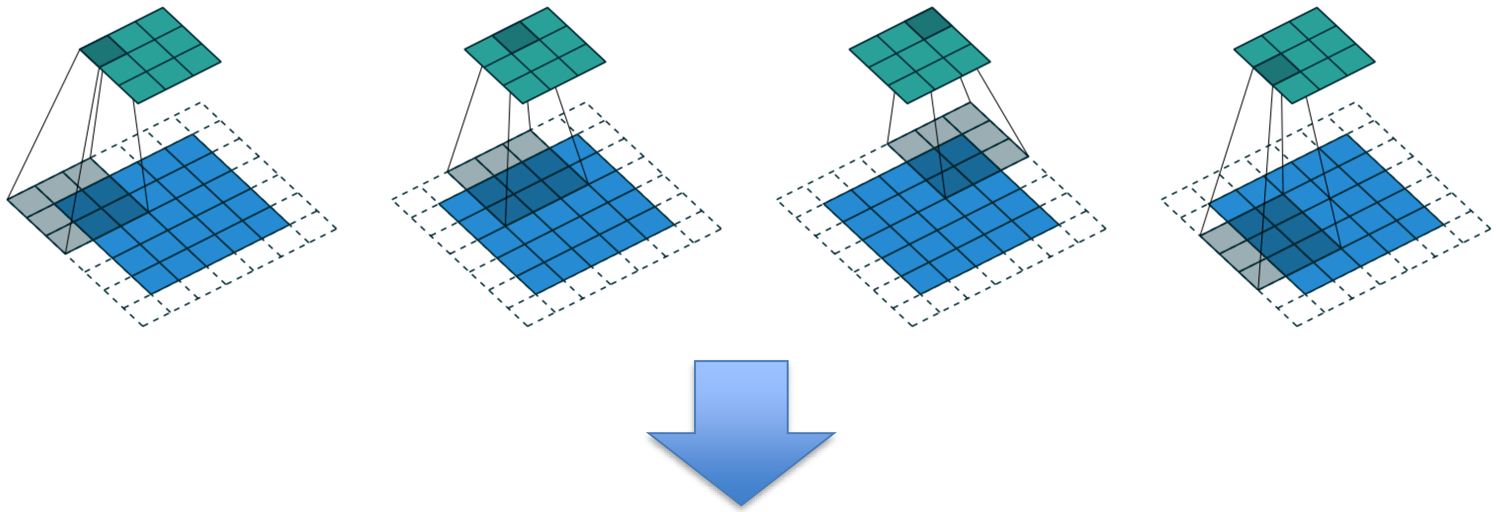


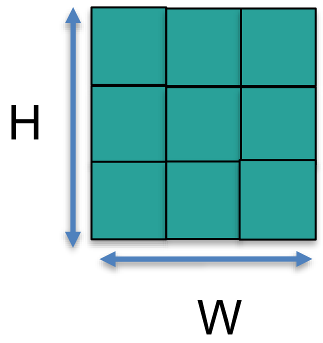
- Poor scaling with image size
- Inefficient weight use – no "weight sharing"
- FC do not provide translation invariance nor equivariance

Lecture 2A: Convolutional Neural Networks (Full Stack Deep Learning - Spring 2021)

Convolution’s Arithmetic

Convolution's arithmetic



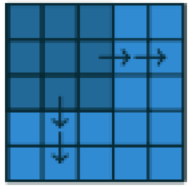


The shape of the output feature map (W,H) is defined based on:

- Shape of the input feature map (W,H)
- The Kernel size (w,h)
- The stride (s)
- The padding (p)

Convolution's arithmetic

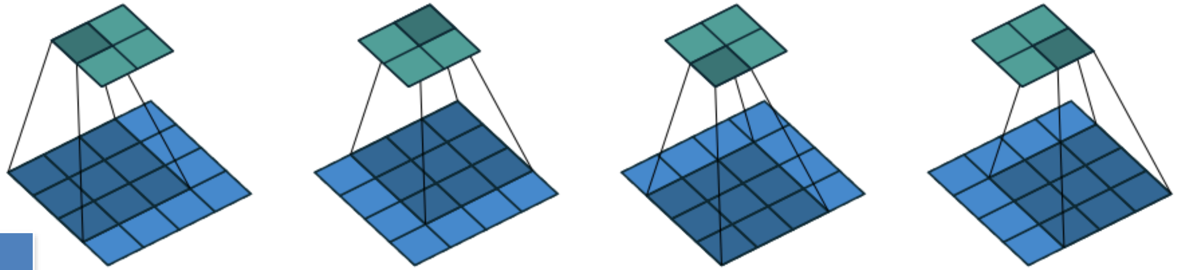
Stride



- Convolutions can subsample the image by jumping across some locations — this is called 'stride'

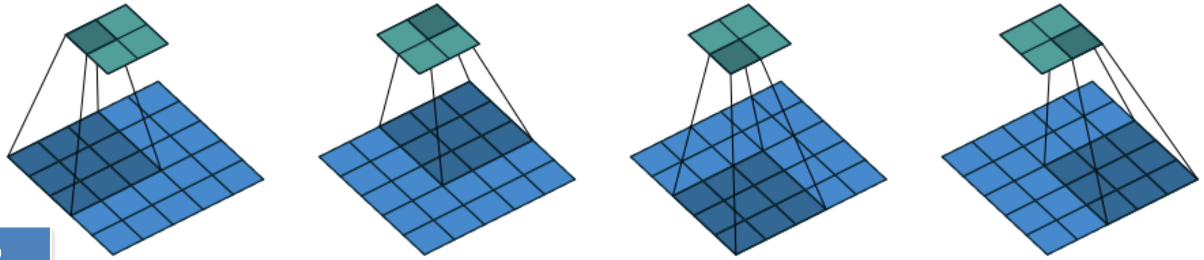
Input Feature map 4x4

s=1



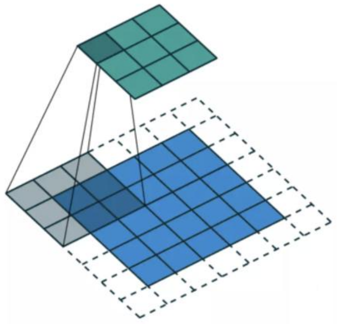
Input Feature map 5x5

s=2

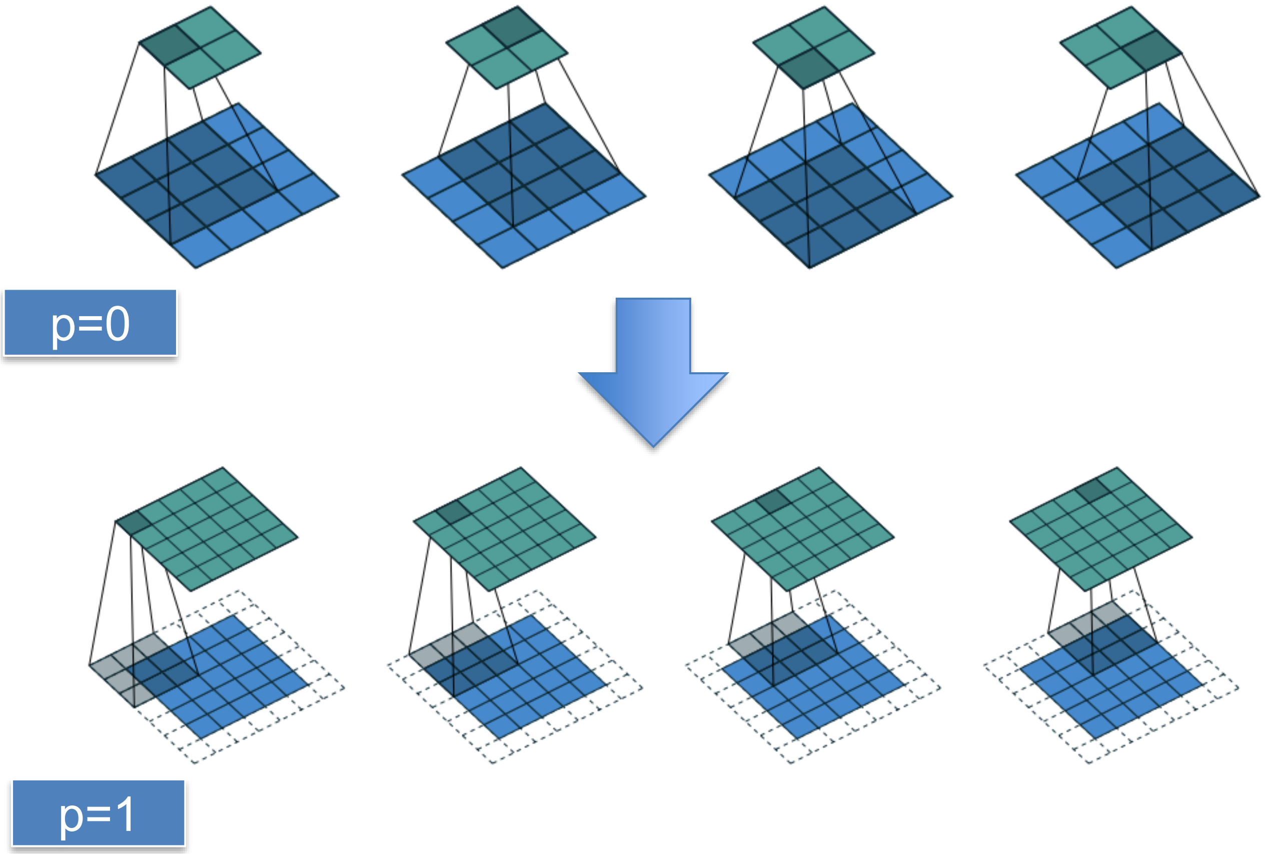


Convolution’s arithmetic

Padding



- Padding solves the problem of filters running out of image
- Done by adding extra rows/cols to the input (usually set to 0)



Convolution's arithmetic

```
import torch.nn as nn
m = nn.Conv2d(in_channels=16, out_channels=33, kernel_size=3, stride=1,padding=1)
m
```

✓ 0.0s

```
Conv2d(16, 33, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
```

- Input: $(N, C_{in}, H_{in}, W_{in})$ or (C_{in}, H_{in}, W_{in})
- Output: $(N, C_{out}, H_{out}, W_{out})$ or $(C_{out}, H_{out}, W_{out})$, where

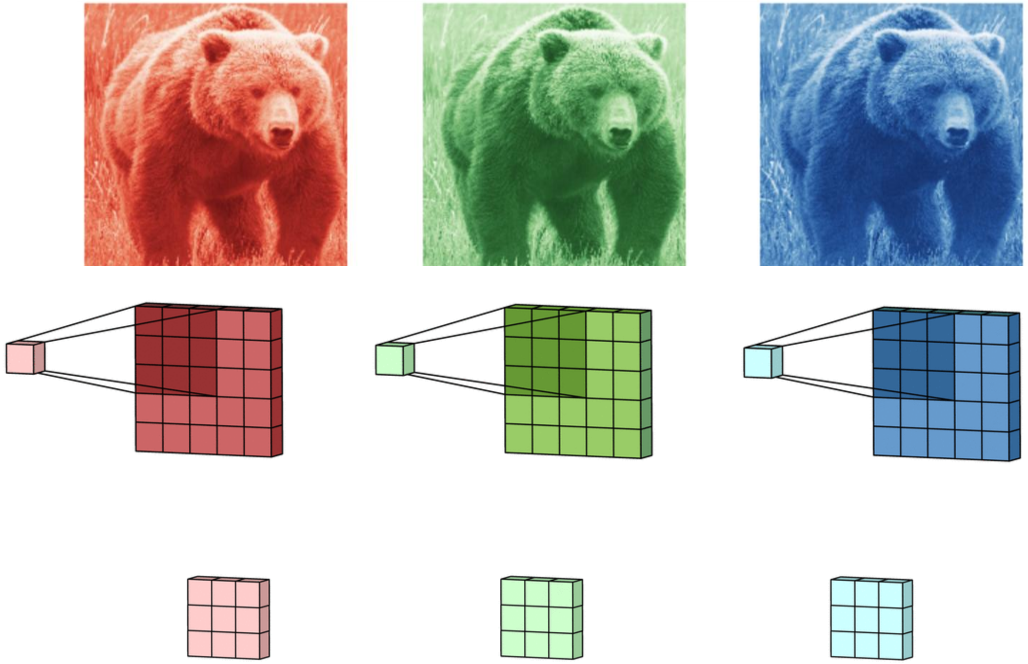
$$H_{out} = \left\lfloor \frac{H_{in} + 2 \times \text{padding}[0] - \text{dilation}[0] \times (\text{kernel\_size}[0] - 1) - 1}{\text{stride}[0]} + 1 \right\rfloor$$

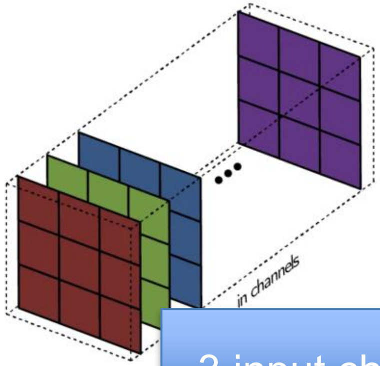
$$W_{out} = \left\lfloor \frac{W_{in} + 2 \times \text{padding}[1] - \text{dilation}[1] \times (\text{kernel\_size}[1] - 1) - 1}{\text{stride}[1]} + 1 \right\rfloor$$

- The same convolution can be applied to images of different sizes
- Stride, padding, and kernel_size determine how height and width change in one layer
- Numbers of in_channels and out_channels determine how many kernels/filters the layer has

Convolution’s arithmetic

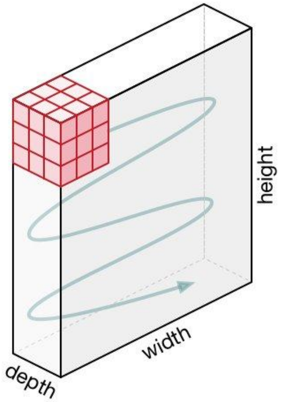
RGB Image



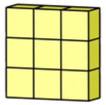


3 input channels

1 filter with
3 kernels



Can’t forget the bias term!



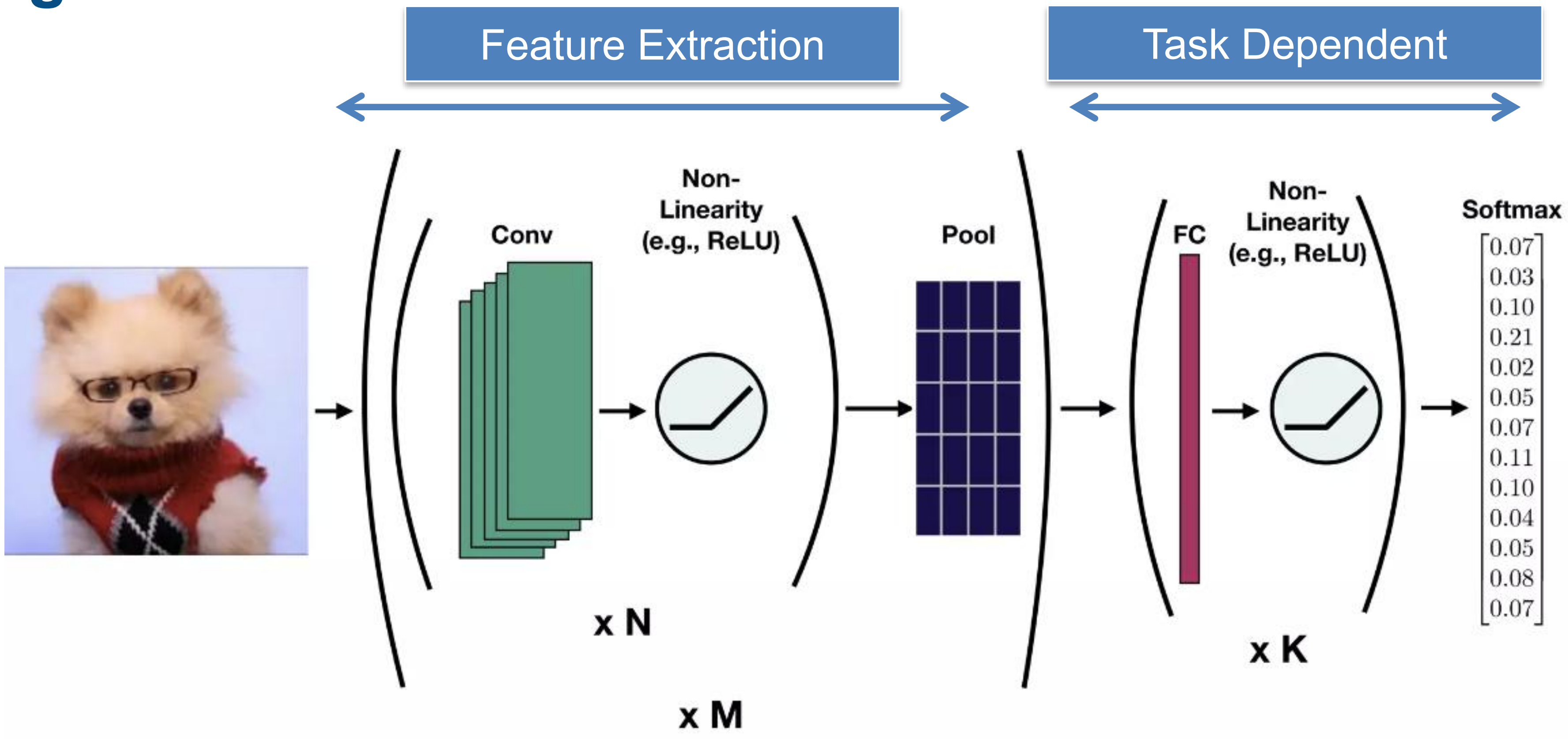
Multiple filters in one layer >> the same number of output channels
(~ different views of the image)

1 output channel

Intuitively Understanding Convolutions for Deep Learning
Towards Data Science

Building a Convolutional Neural Network (CNN)

Building a CNN

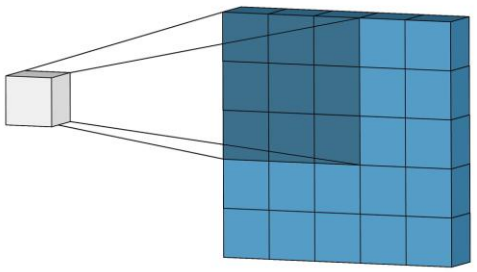


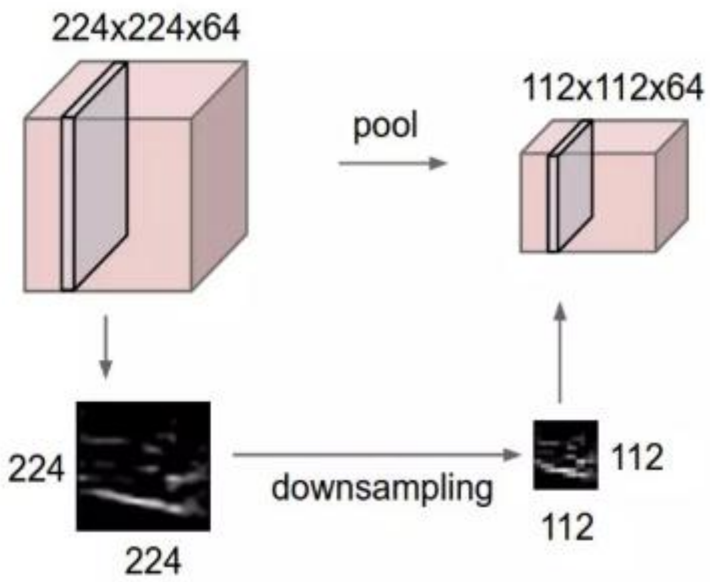
Lecture 2A: Convolutional Neural Networks (Full Stack Deep Learning - Spring 2021)

Feature Extraction Blocks:
- Convolutional layers extract local features
- Pooling layers aggregate the extracted local features

Building a CNN

Pooling

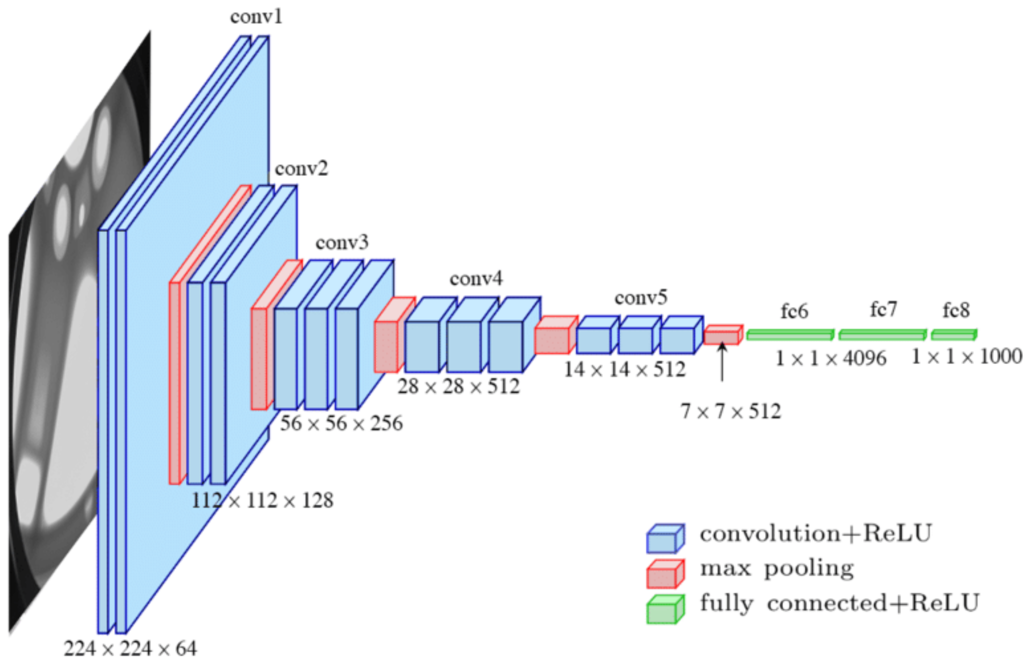




- Idea: downsample output activation maps from convolutional layers
- Similar to convolutional layer: Sliding window
- Instead of the dot product with the kernel, apply an aggregation function (e.g. max, avg)

- Reduces (H,W) by using bigger strides
- Preserves the number of channels

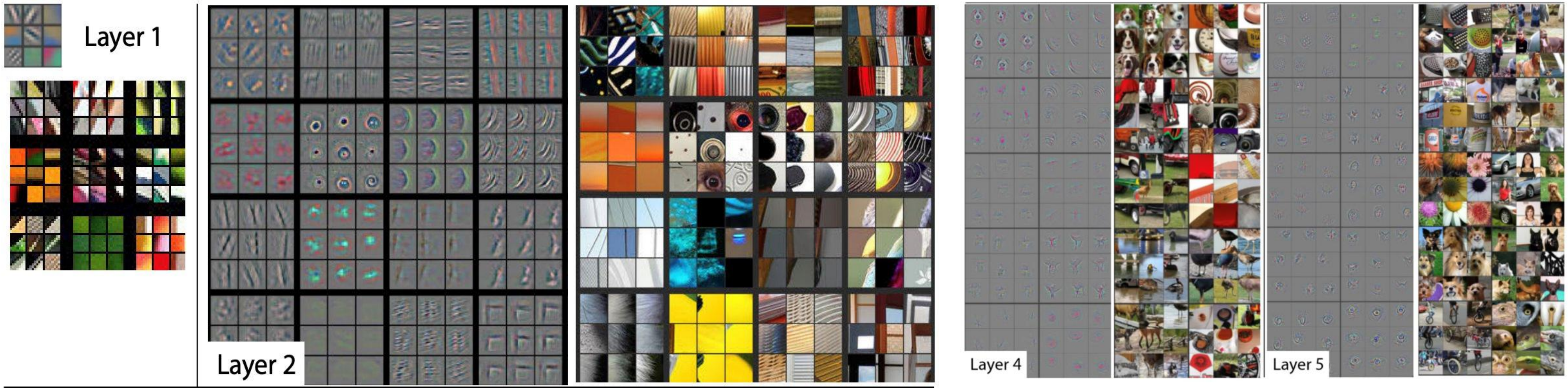
Building a CNN



Connecting image structure and CNN architecture

Connecting image structure and CNN architecture

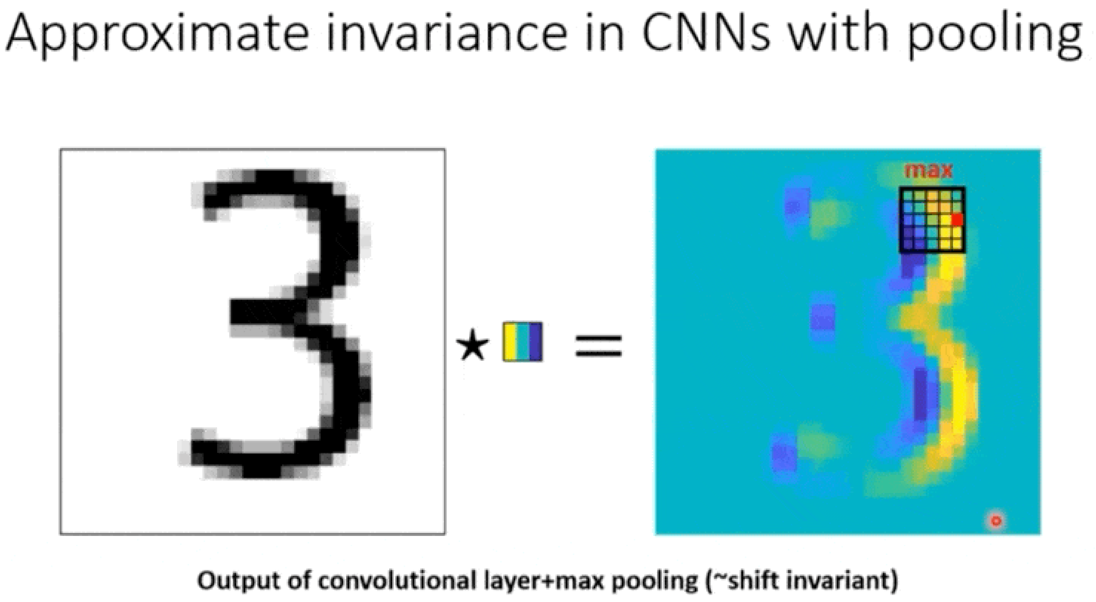
Hierarchical Feature Learning



Visualizing and Understanding Convolutional Networks
arXiv:1311.2901v3

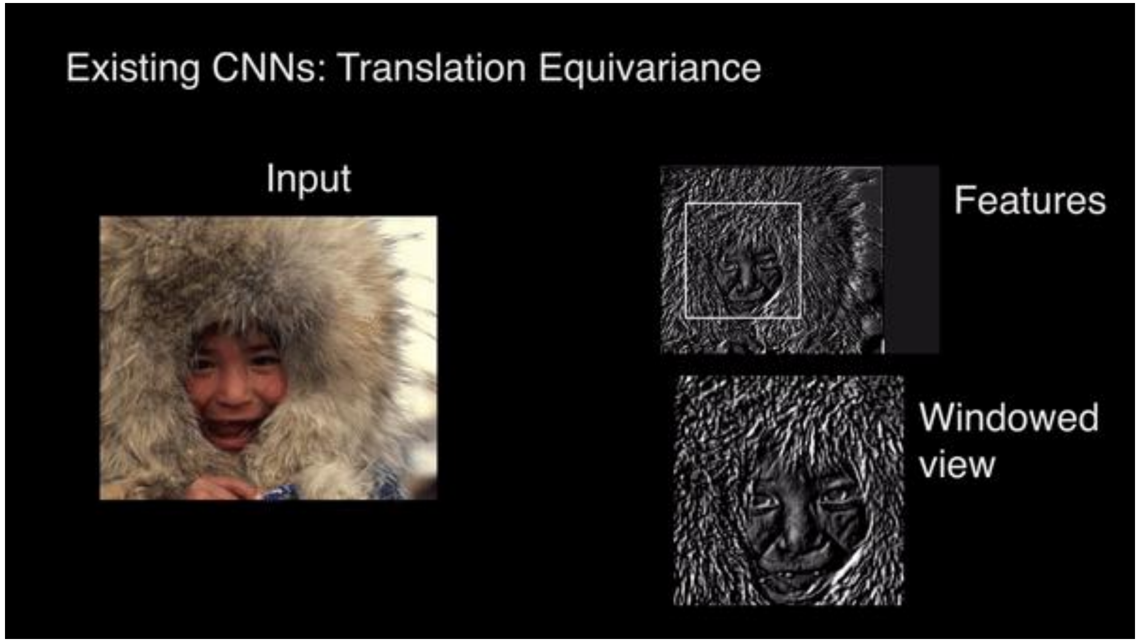
Connecting image structure and CNN architecture

Translation Invariance



05 Imperial's Deep learning course: Equivariance and Invariance
Bernhard Kainz
https://www.youtube.com/watch?v=a4Quhf9NhMY

Translation Equivariance



https://www.youtube.com/watchv=a4Quhf9NhMY&t=944s

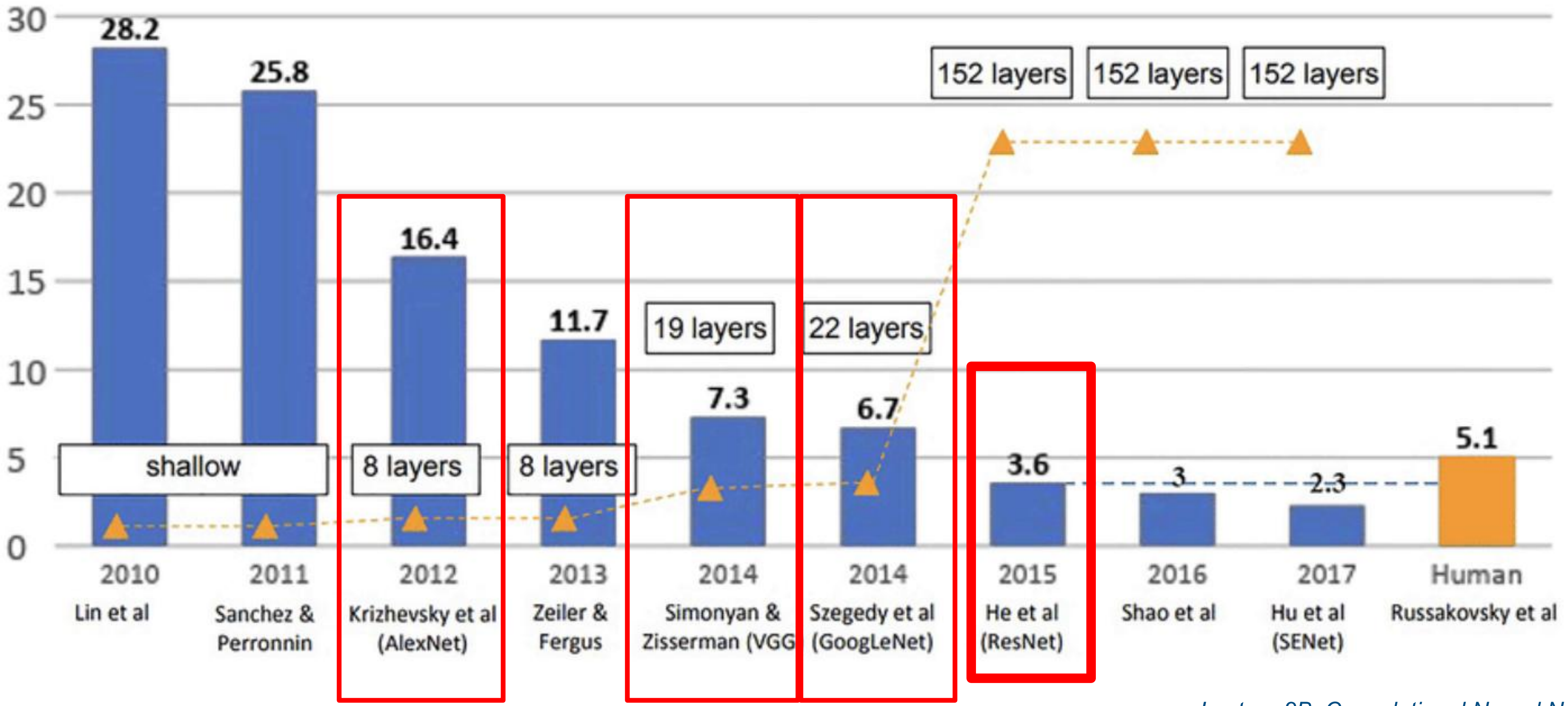
Connecting image structure and CNN architecture

Image property	Architectural reflection	Emergent result
Locality	Sparse local connections with shared filters	Early layers detect local features (edges, textures)
Translation equivariance	Sliding windows (convolution, pooling)	Feature maps shift with input
Compositional structure	Layered composition	Hierarchical representations (simple → complex)

Popular Convolutional Neural Network Architectures

Popular CNN-based architectures

ImageNet Large Scale Visual Recognition Challenge



Lecture 2B: Convolutional Neural Networks (Full Stack Deep Learning - Spring 2021)

Popular CNN-based architectures - ResNets

ResNet

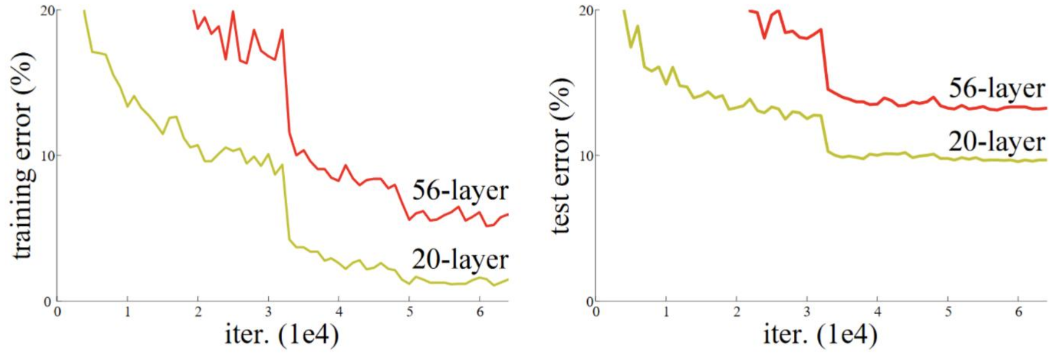
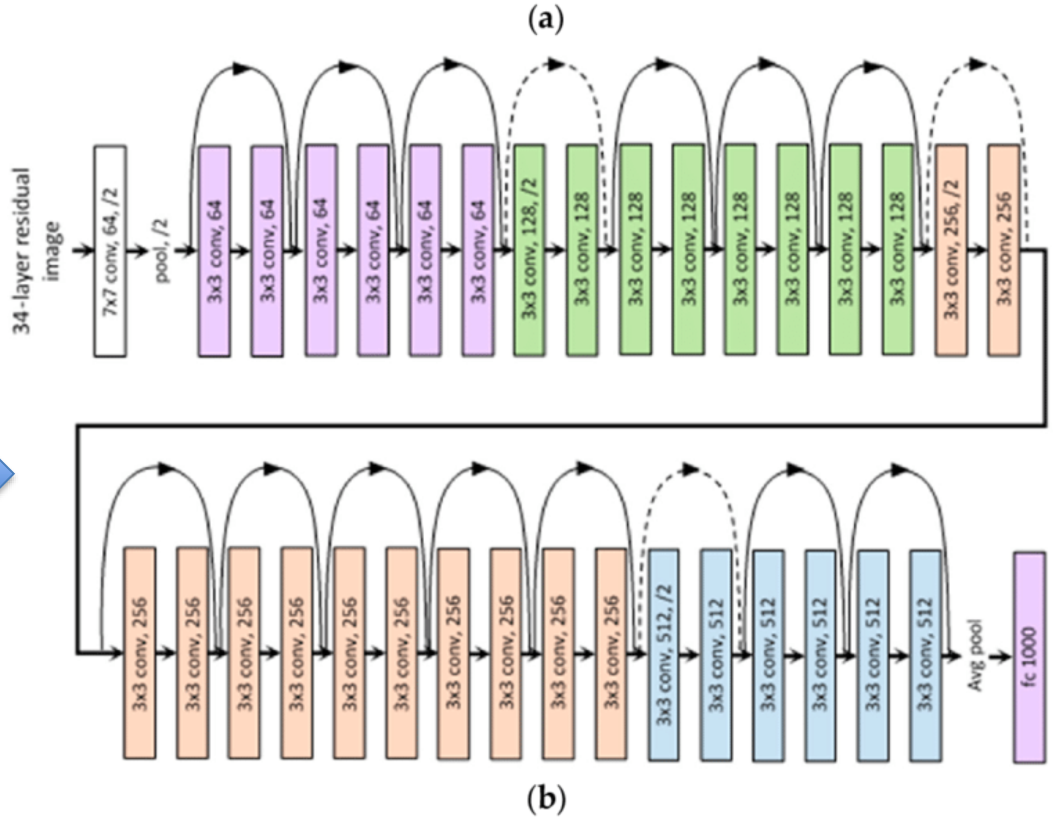


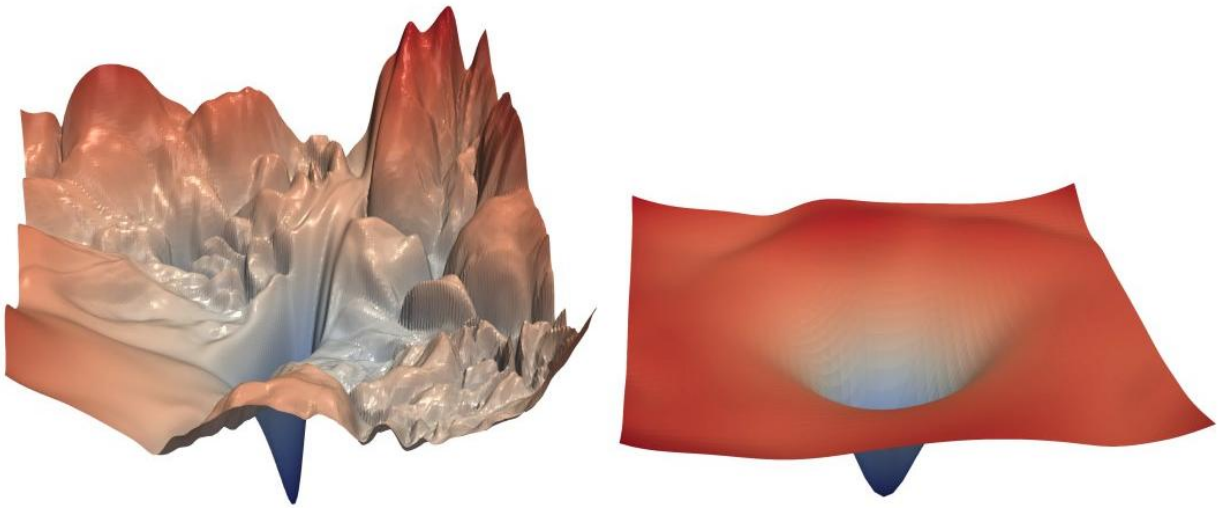
Figure 14.1: Training of networks of different depth (courtesy of Kaiming He et al.)

[“Deep Residual Learning for Image Recognition”]



ResNet-34 Layered architecture

Popular CNN-based architectures



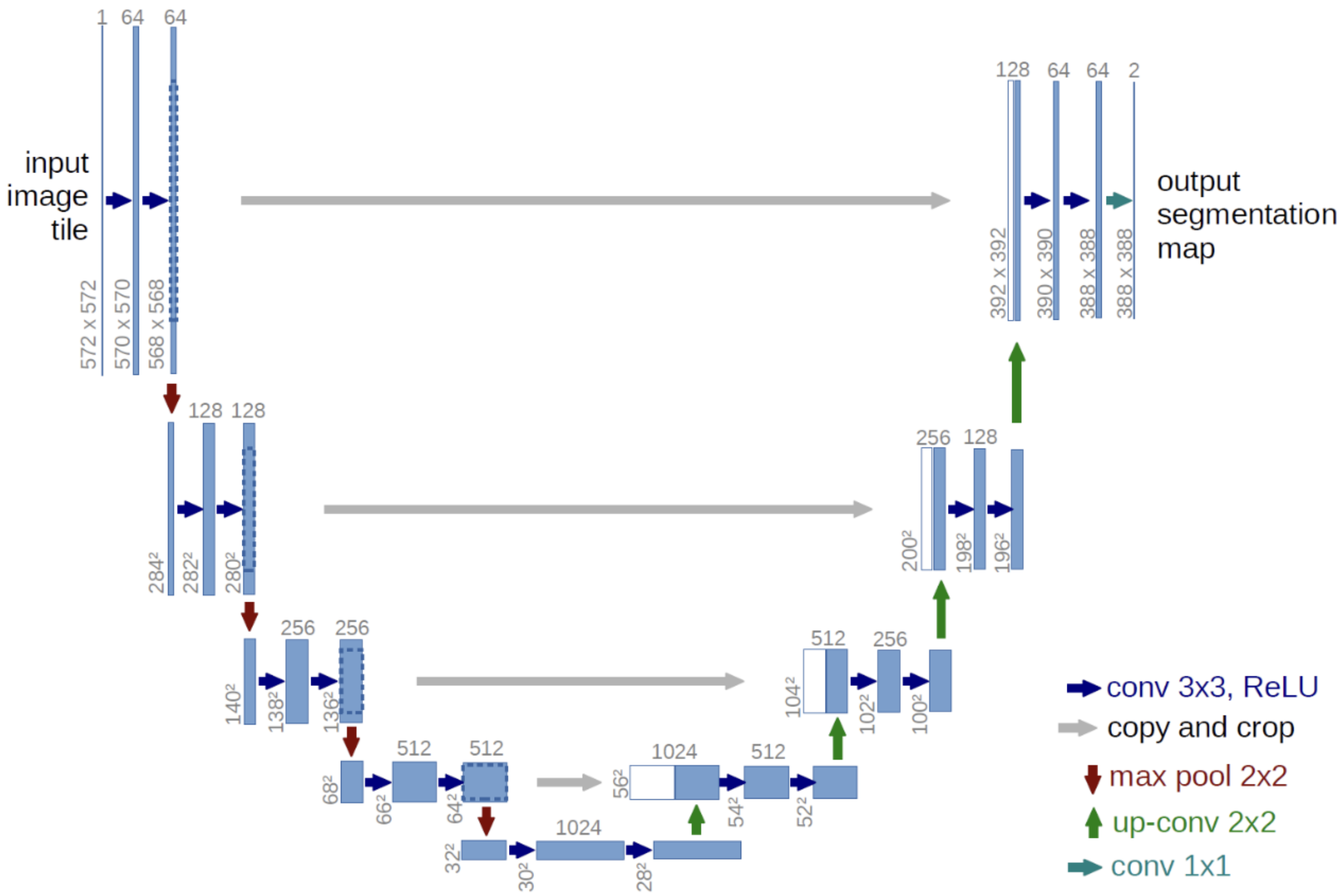
(a) without skip connections (b) with skip connections

The loss surfaces of ResNet-56 with and without skip connections

Using skip connections helps smooth the loss function, which makes training easier as it avoids falling into a very sharp area.

Visualizing the Loss Landscape of Neural Nets
https://arxiv.org/abs/1712.09913

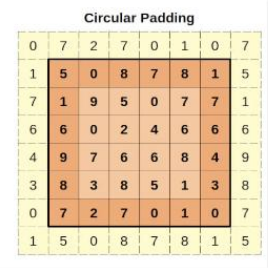
Popular CNN-based architectures – U-Nets



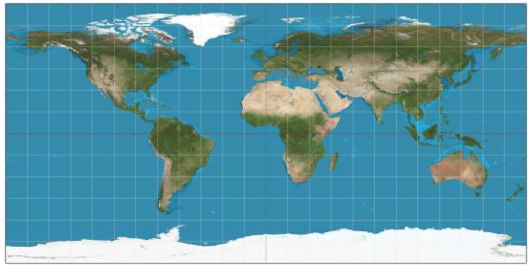
U-Net: Convolutional Networks for Biomedical Image Segmentation
arxiv.org/abs/1505.04597

Popular CNN-based architectures

Periodic Convolutions

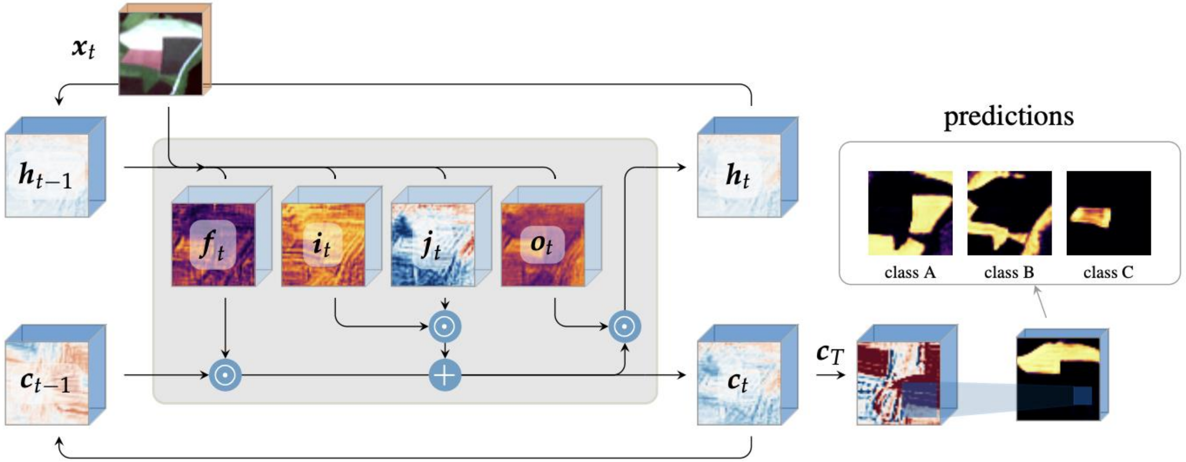


0	7	2	7	0	1	0	7
1	5	0	8	7	8	1	5
7	1	9	5	0	7	7	1
6	6	0	2	4	6	6	6
4	9	7	6	6	8	4	9
3	8	3	8	5	1	3	8
0	7	2	7	0	1	0	7
1	5	0	8	7	8	1	5



https://github.com/pangeo-data/WeatherBench/blob/master/src/train_nn.py#L102

Spatio-Temporal Data - ConvLSTMS



Convolutional LSTMs for Cloud-Robust Segmentation of Remote Sensing Imagery
Marc Rußwurm

References

- Deep Learning book CNNs - https://www.deeplearningbook.org/contents/convnets.html
- Understanding Deep Learning – CNN chapter - https://udlbook.github.io/udlbook/
- CNN feature visualization - https://distill.pub/2017/feature-visualization/
- CNN feature visualization - https://arxiv.org/pdf/1311.2901.pdf
- Intuitively Understanding Convolutions for Deep Learning
- A guide to convolution arithmetic for deep Learning - Dumoulin V., Visin. F, 2018, arXiv:1603.07285
- Lecture 2A and Lecture 2B Convolutional Neural Networks (Full Stack Deep Learning - Spring 2021)
- Invariance and equivariance - https://www.doc.ic.ac.uk/~bkainz/teaching/DL/notes/equivariance.pdf
- Imperial's Deep learning course: Equivariance and Invariance, Bernhard Kainz