Neural Networks in Computer Vision

NN in CV

CV Disclaimer

Computer Vision is about Neural Networks

Math + Physics
The traditional CV techniques are still widely used

CNN thrives only in CV

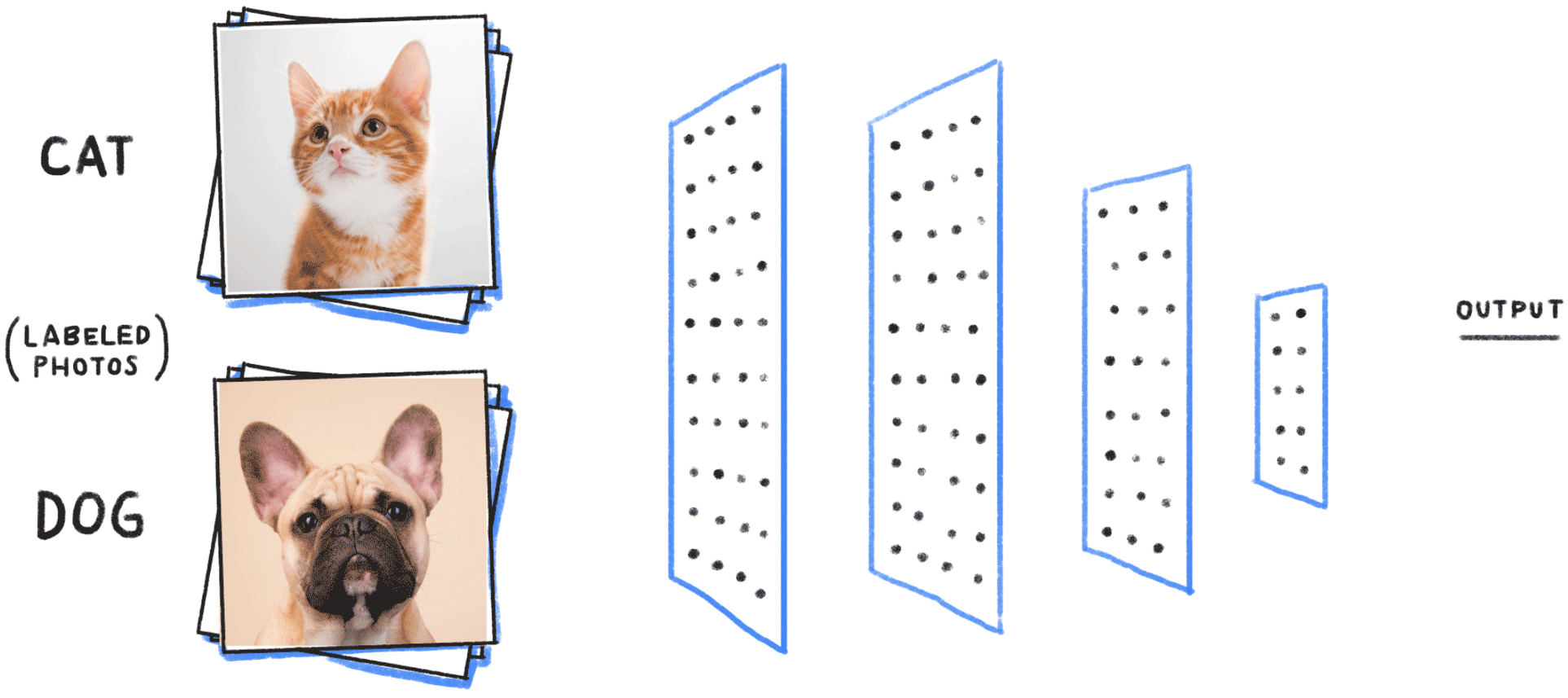
They’re fast and they’re efficient

Everything is approximate, better use traditional CV

Image classification task

Motivation

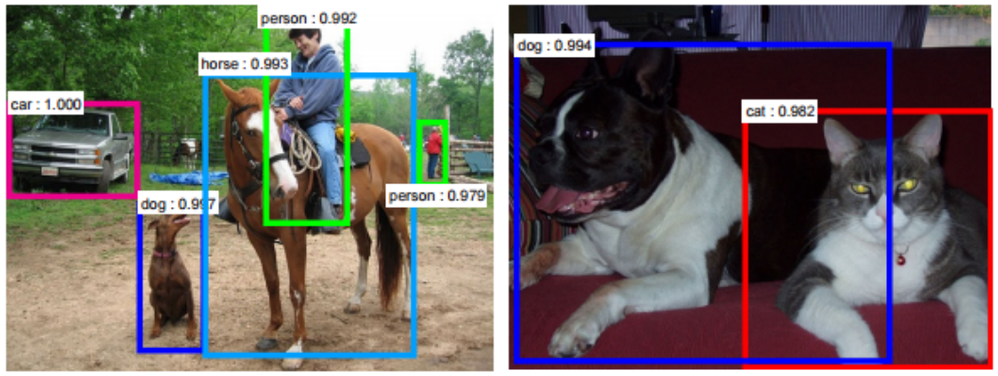
Image classification

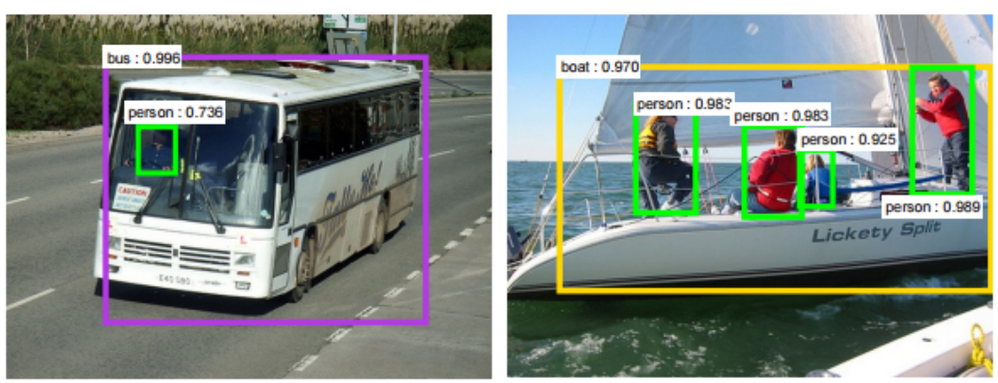


Source: http://jalammar.github.io/illustrated-bert/

Motivation. Part 2.

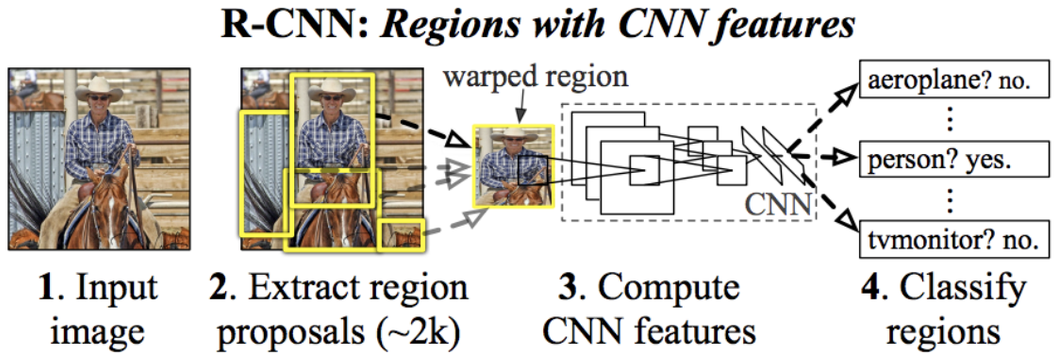
Object Detection





https://stats.stackexchange.com/questions/298389/probability-scores-threshold-usually-used-in-deep-learning-for-an-object-recogni

R-CNN
[Girshick et al., 2013]

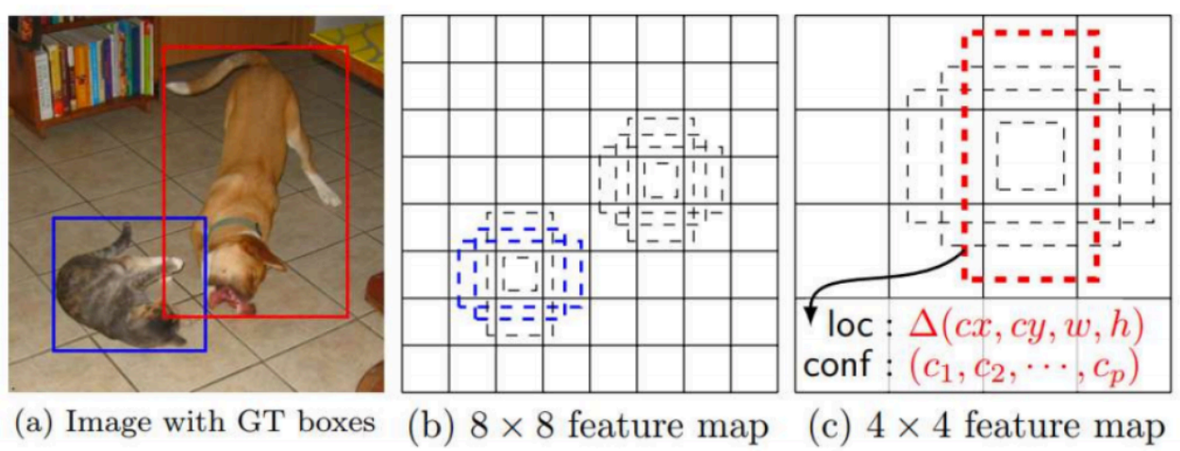


https://towardsdatascience.com/r-cnn-fast-r-cnn-faster-r-cnn-yolo-object-detection-algorithms-36d53571365e

Fast R-CNN
[Girshick 2015]

Region proposal network
[Ren et al., 2015]

YOLO (v3), SSD, RetinaNet
[Liu et al., 2016]

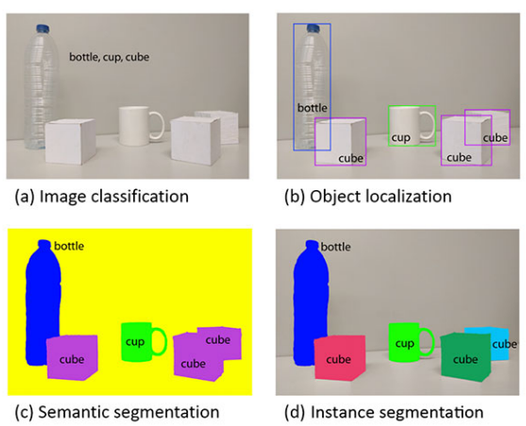


Motivation. Part 3.

Segmentation

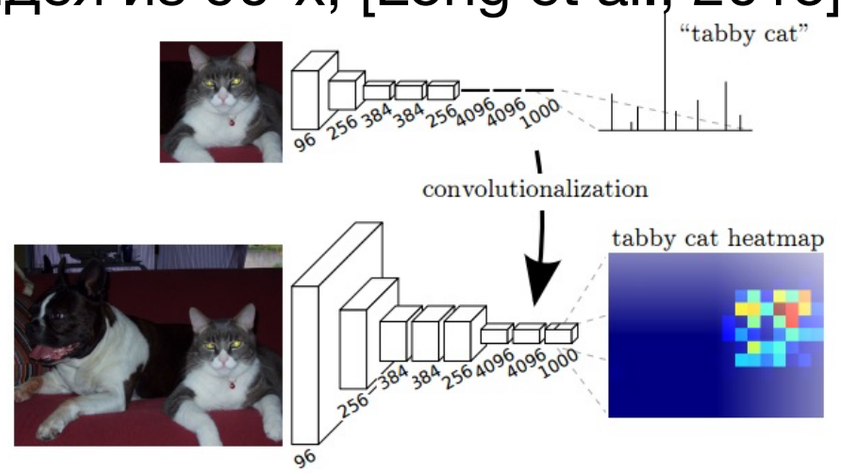
https://www.youtube.com/watch?v=qWl9idsCuLQ

Mask R-CNN
[He et al., 2017]



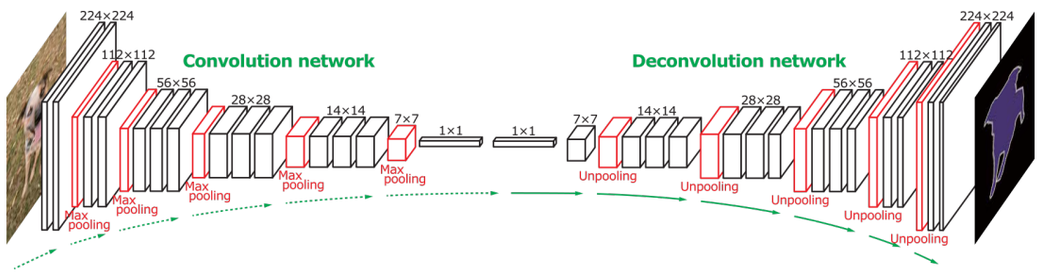
https://www.pyimagesearch.com/2018/11/19/mask-r-cnn-with-opencv/

Fully-convolutional CNN
Идея из 90-х, [Long et al., 2015]



https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/image_segmentation.html

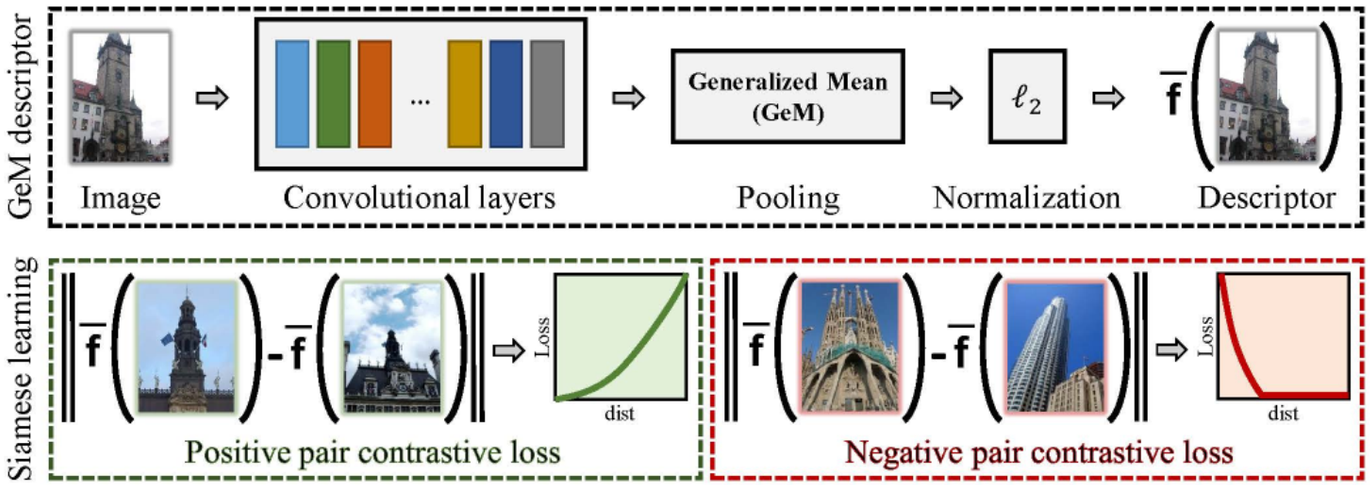
Extreme segmentation



https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/image_segmentation.html

Motivation. Part 4.

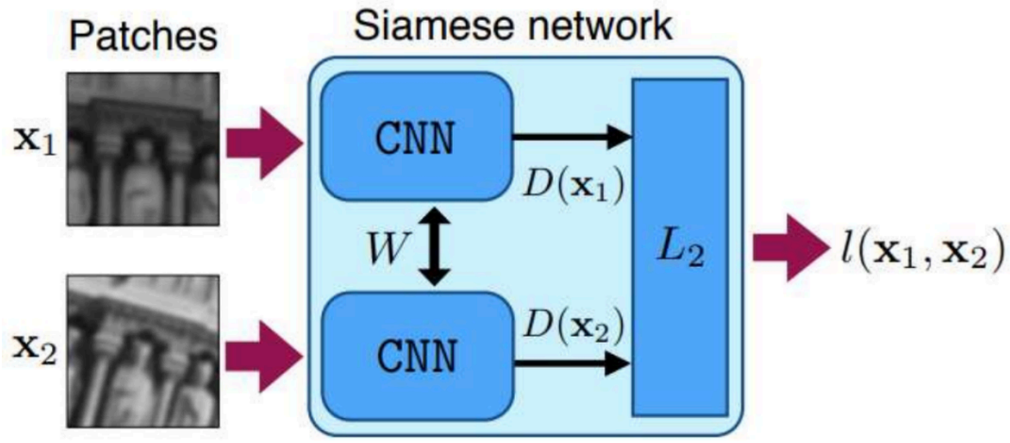
Image Retrieval

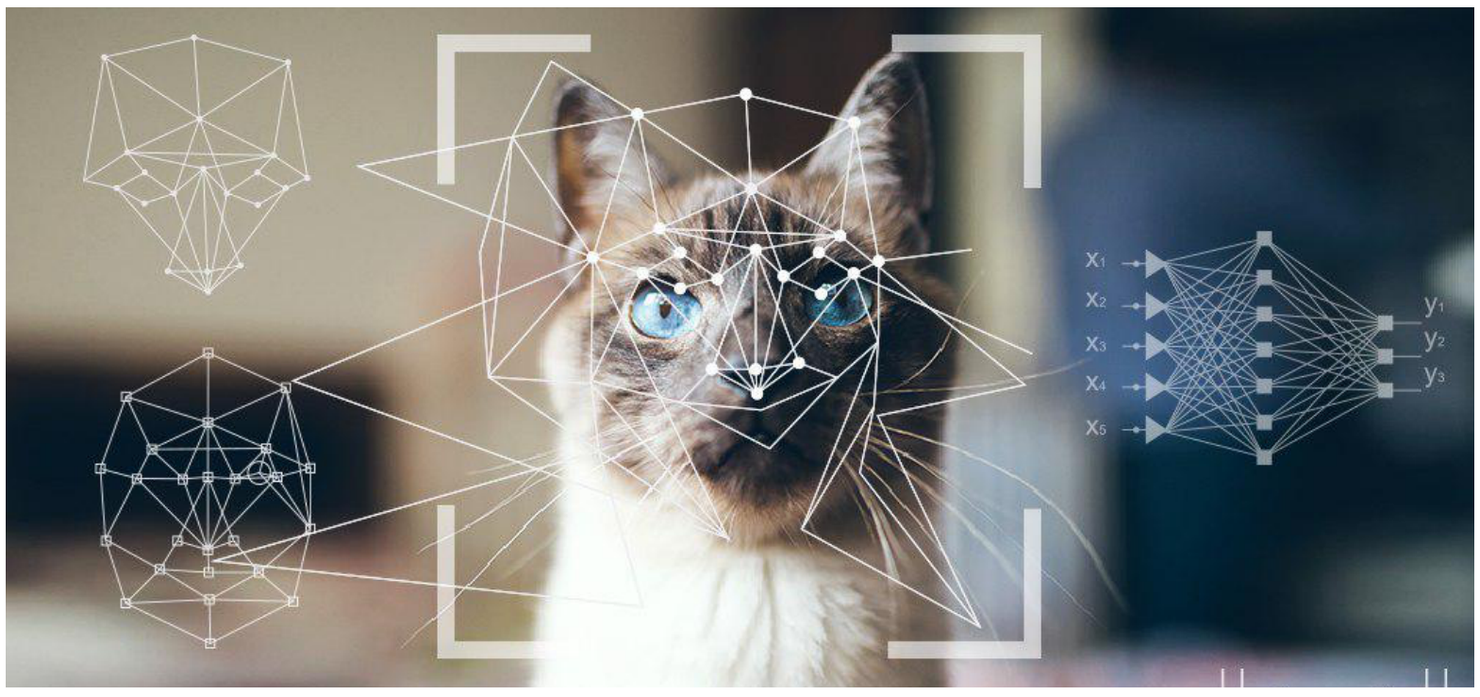


https://github.com/filipradenovic/cnnimageretrieval-pytorch

Siamese Networks

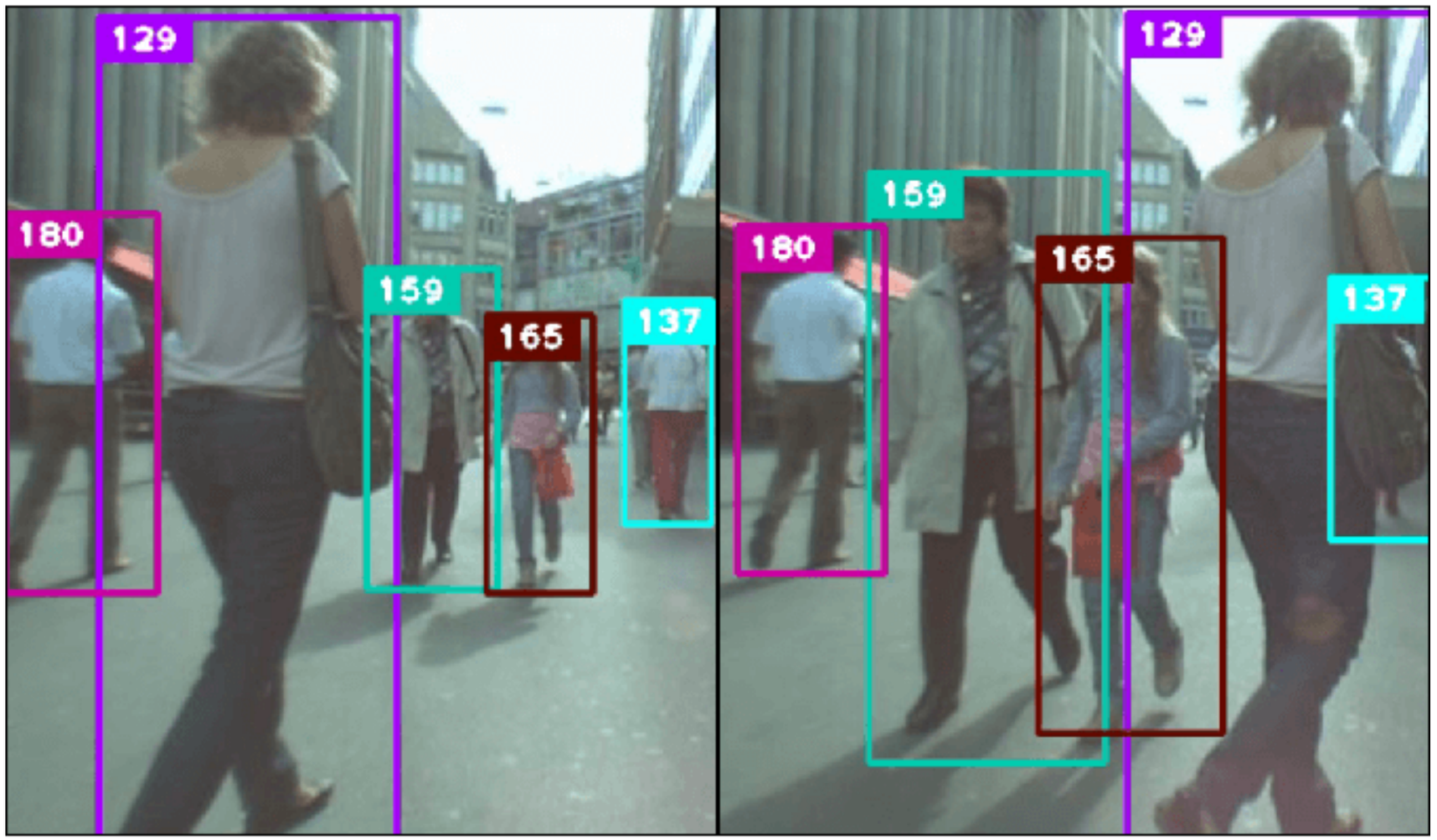
[Simo-Serra et al., 2015]





Motivation. Part 5.

Object Tracking



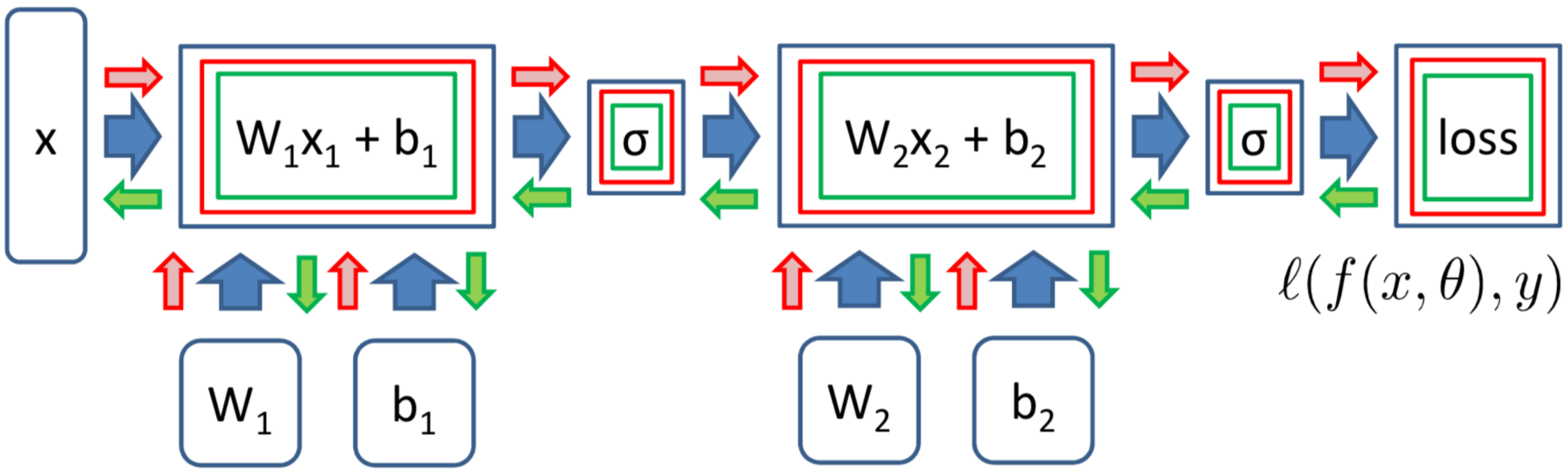
https://missinglink.ai/guides/computer-vision/object-tracking-deep-learning/

Object Detection + Segmentation + Siamese Nets + ...


To: 0
From: 0

Convolutional Neural Networks

Neural Networks



Anton Osokin, HSE, Deep Learning Course. Source: https://github.com/aosokin/dl_cshse_ami

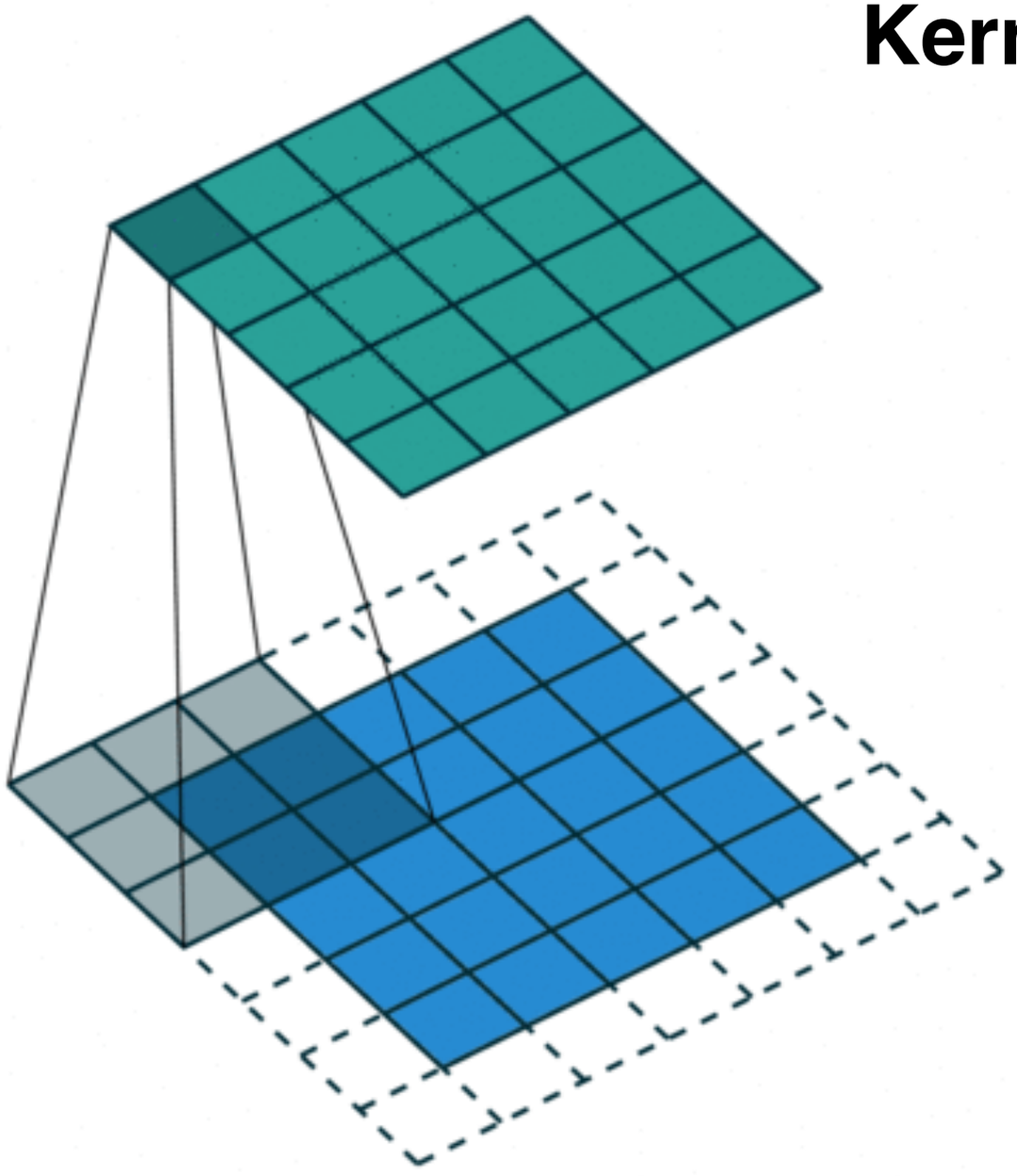
Too many parameters
Fast overfit

Convolutions

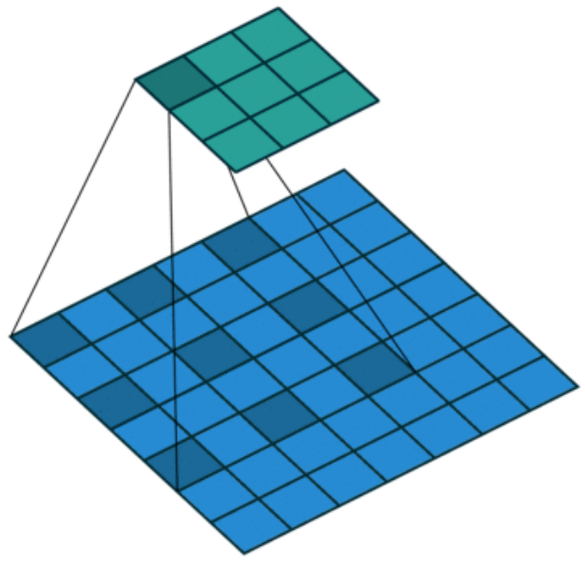
Kernel Size

Stride

Padding



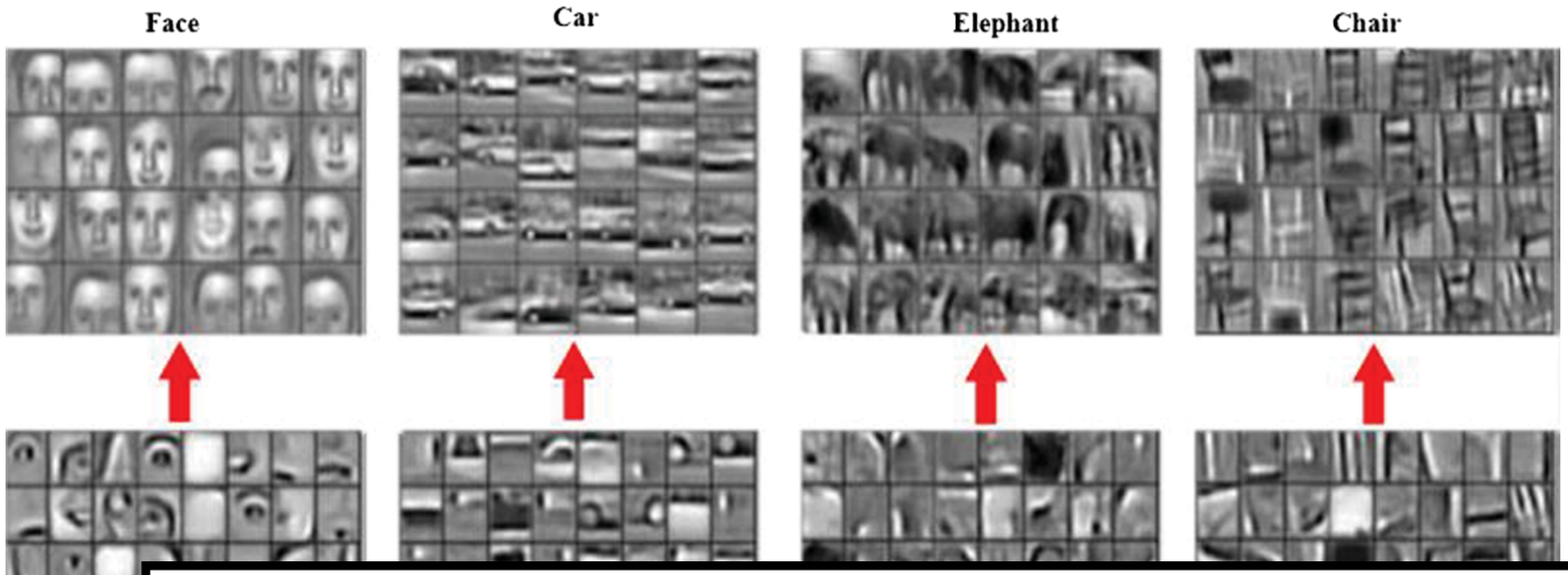
https://towardsdatascience.com/types-of-convolutions-in-deep-learning-717013397f4d



Dilated Convolutions

a.k.a. atrous convolutions

Feature Extraction



Different **optimization objectives** show what different parts of a network are looking for.

- `n` layer index
- `x,y` spatial position
- `z` channel index
- `k` class index

Neuron
`layer`$_n$`[x,y,z]`

Channel
`layer`$_n$`[:,:,z]`

Layer/DeepDream
`layer`$_n$`[:,:,:]`^{2}

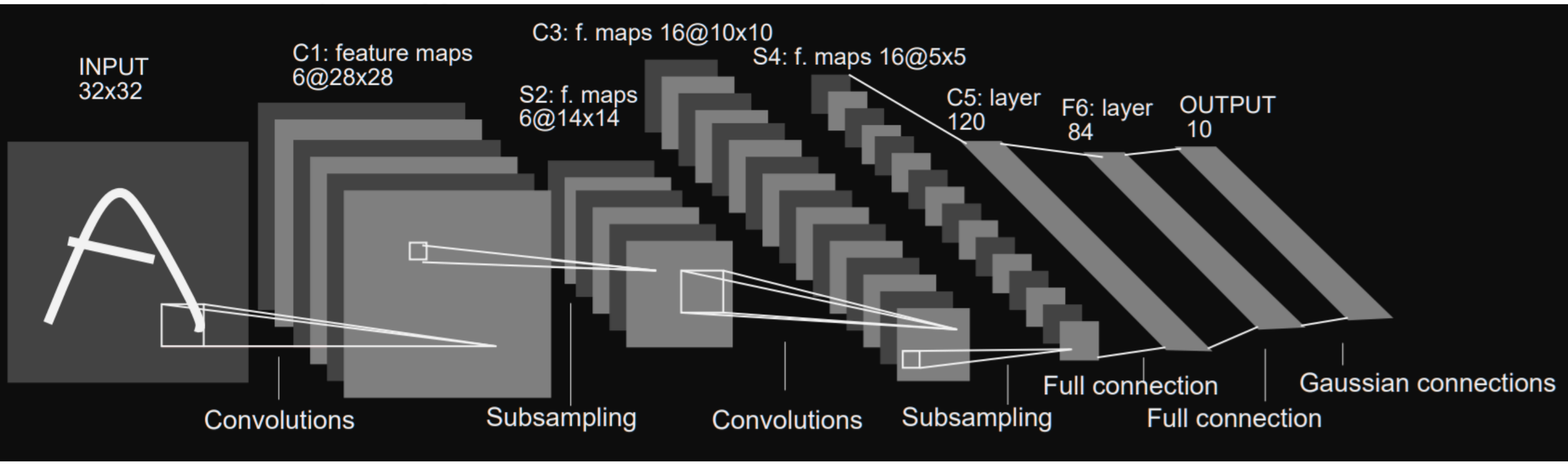
Class Logits
`pre_softmax[k]`

Class Probability
`softmax[k]`

https://distill.pub/2017/feature-visualization/

Know your history

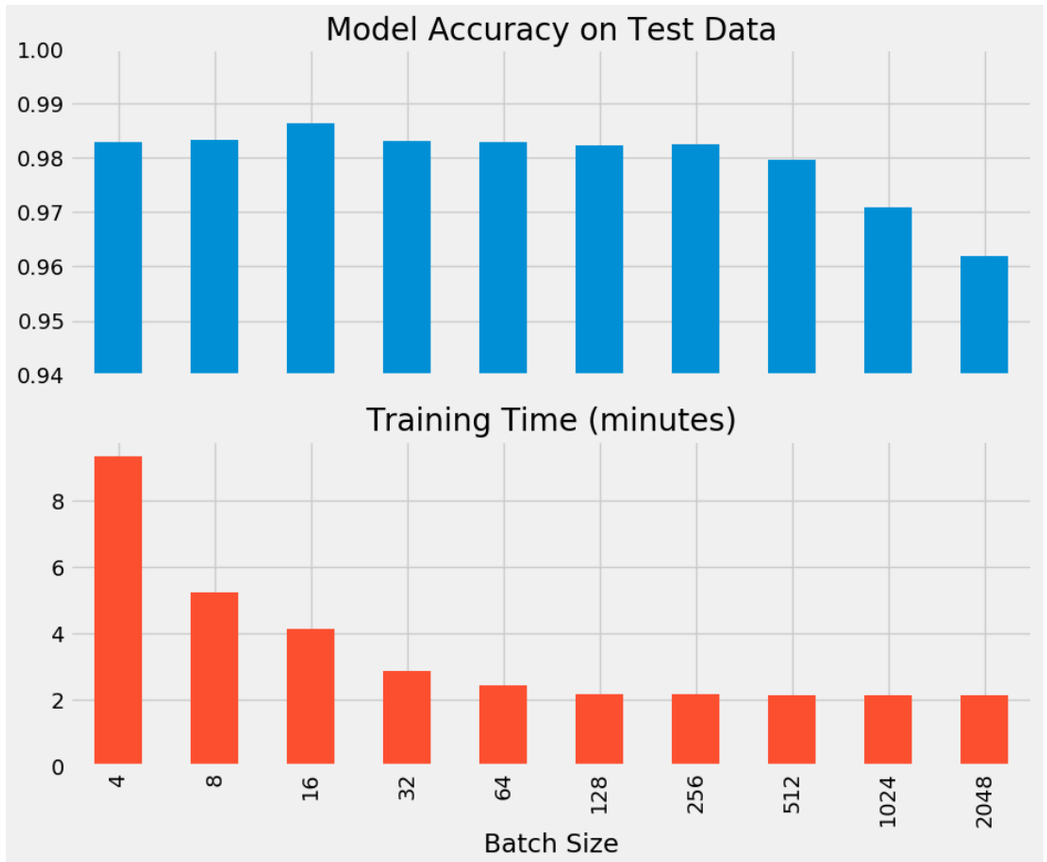
One of the first Convolutional Neural Networks – LeNet-5
[LeCun, Bottou, Bengio, Haffner 1998]



Source: Stanford CS class CS231n slides

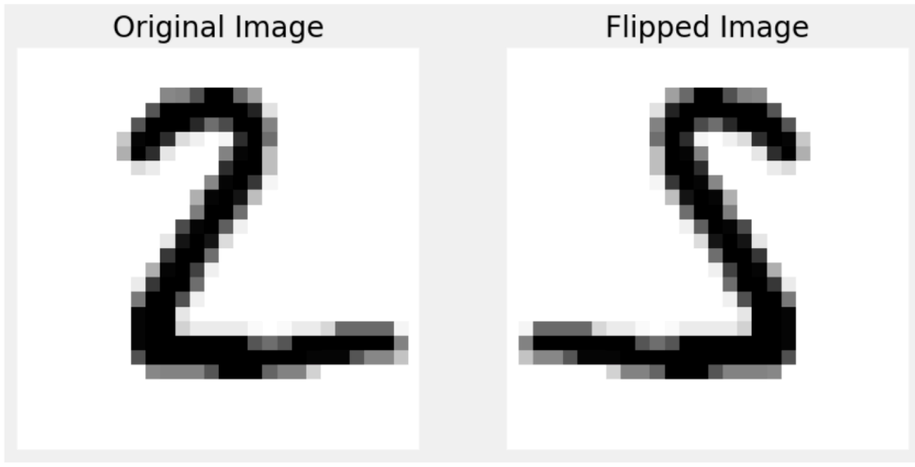
Important Notes

Batch Size



Effect of batch size on model accuracy (upper chart) and training time (lower chart)

Feature Rotation and Flipping



	Testing Accuracy
Base Model	0.9842
Flipped Images	0.706

Pooling Stategies

	Testing Accuracy
Avg Pooling	0.9842
Max Pooling	0.9831

What Matters

In the simplest case, the output value of the layer with input size (N, C_{in}, H, W) and output $(N, C_{\text{out}}, H_{\text{out}}, W_{\text{out}})$ can be precisely described as:

$$\text{out}(N_i, C_{\text{out}_j}) = \text{bias}(C_{\text{out}_j}) + \sum_{k=0}^{C_{\text{in}}-1} \text{weight}(C_{\text{out}_j}, k) \star \text{input}(N_i, k)$$

Shape:

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

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

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

Pytorch

Research