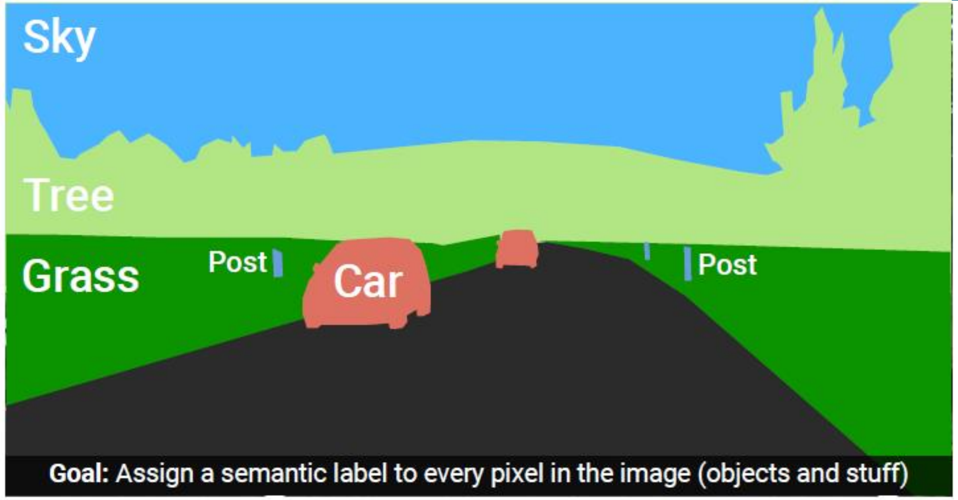
Semantic Segmentation

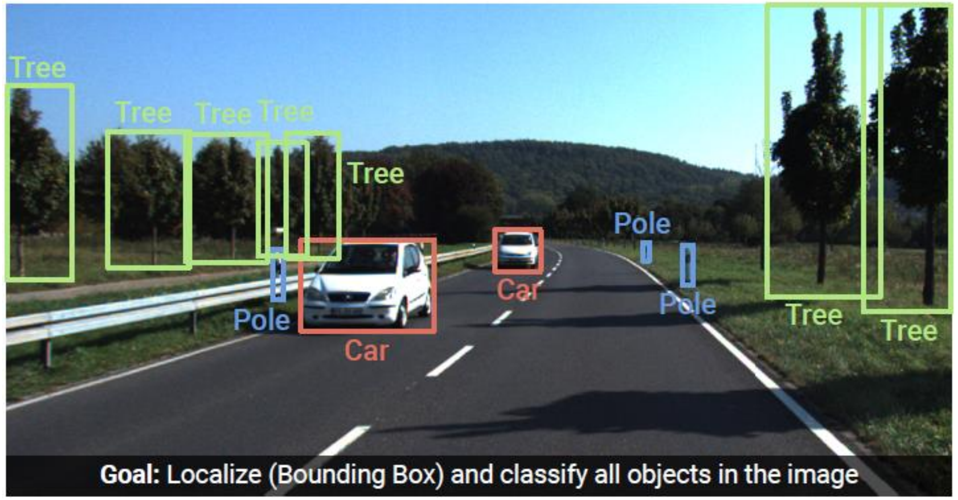
Deep Learning and Image Processing

Image Understanding (Recognition)

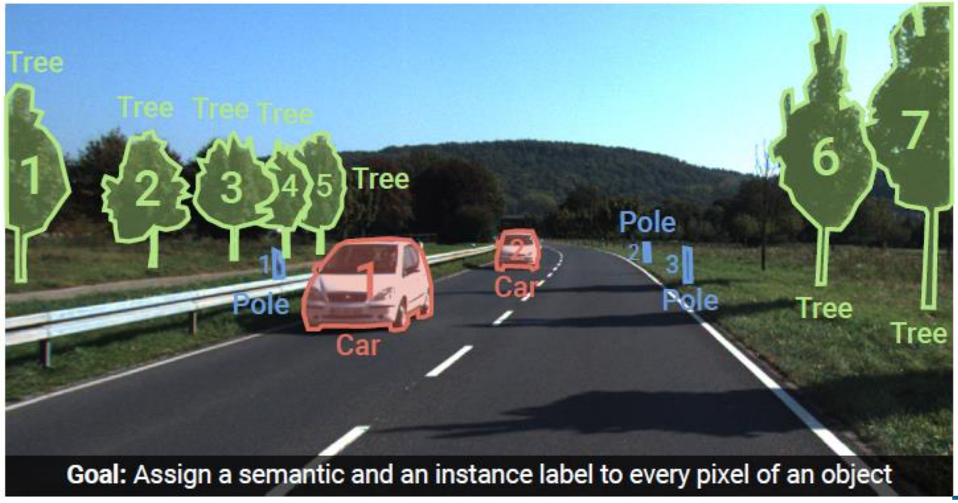
Image Classification



Semantic Segmentation



Object Detection



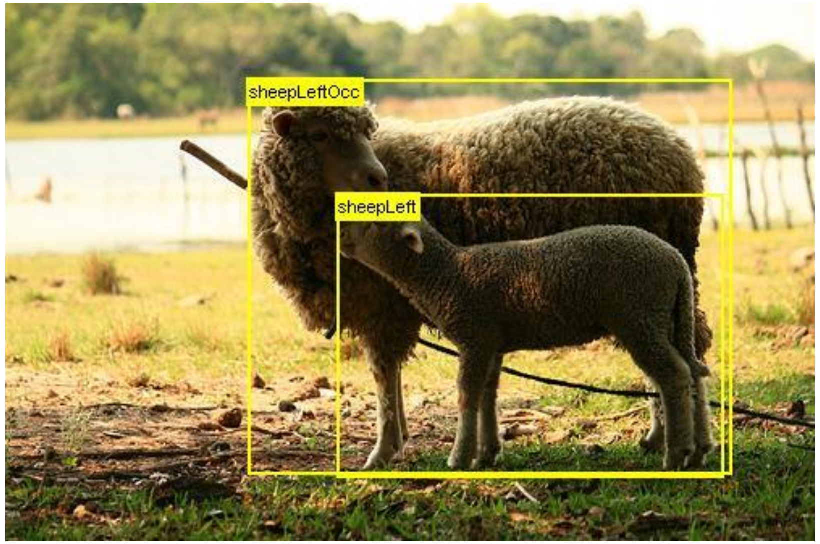
Instance Segmentation

combination of both: panoptic segmentation

A Few More Image Data Sets

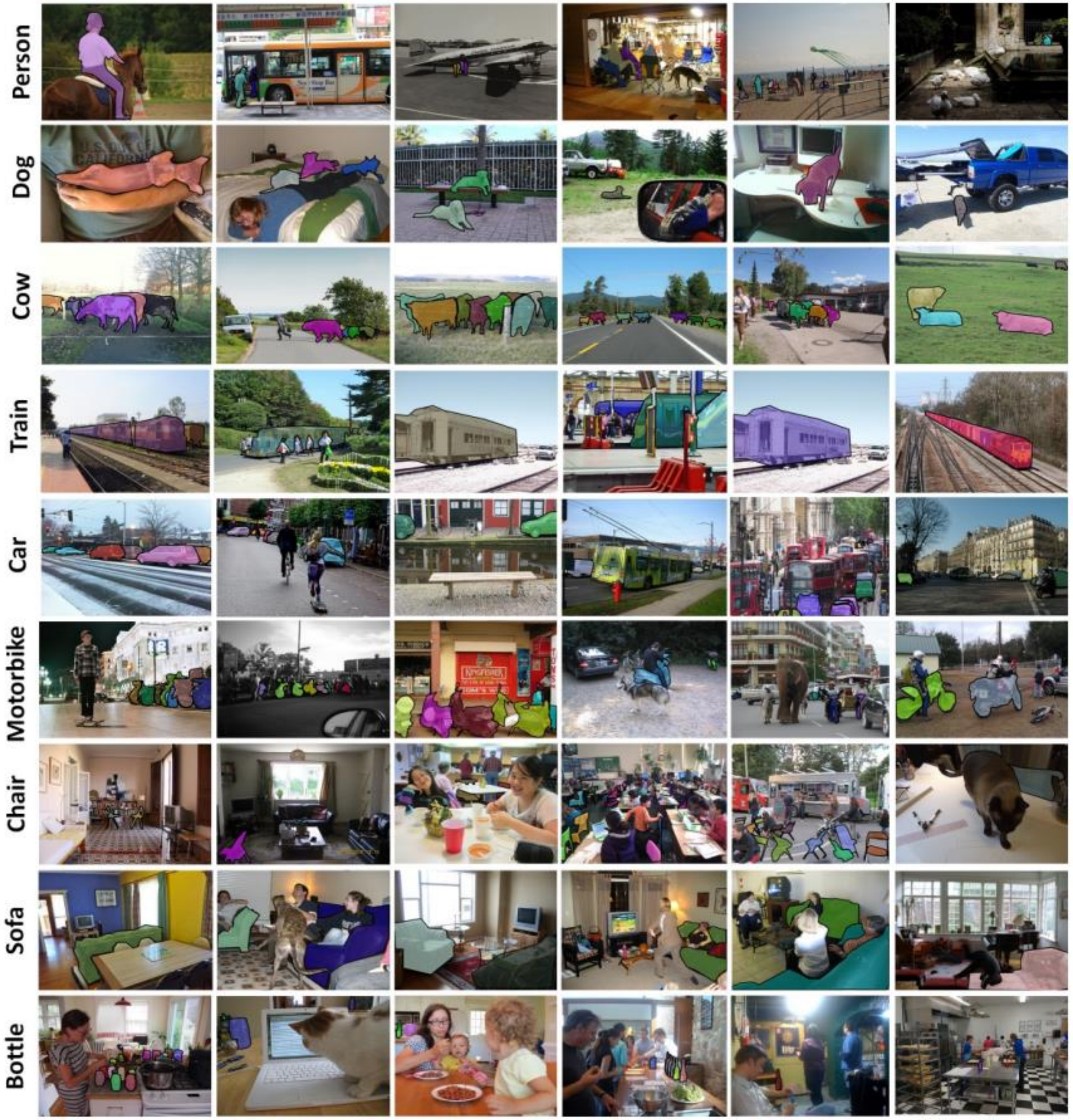
PASCAL VOC Data Set

- PASCAL Visual Object Class challenge
- widely used as benchmark for object detection and semantic segmentation
- 20 object categories such as person, sofa, sheep, car, …
- 11530 annotated images
- available annotations: pixel-level segmentation, bounding boxes, object classes



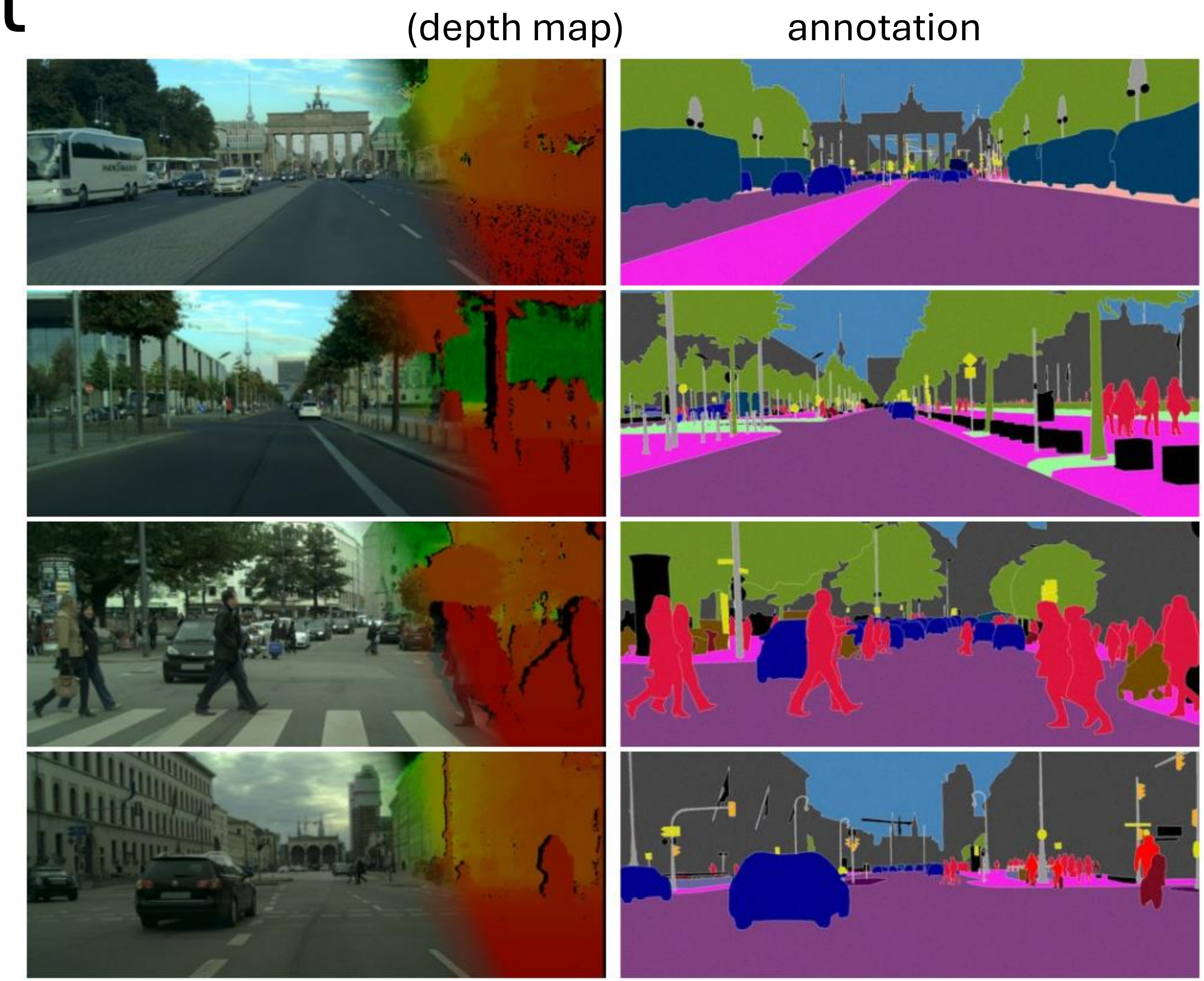
MS COCO Data Set

- Microsoft Common Objects in Context
- images of complex everyday scenes containing common objects in their natural context
- 91 objects types
- 2.5 million annotated instances in 328k images → instance segmentation



Cityscapes Data Set

- goal: semantic understanding of urban street scenes (captured in 50 cities)
- pixel annotations for 30 classes (person, car, building, ...)
- 5000 fine-annotated and 20000 coarse-annotated images



Semantic Segmentation

Object Segmentation from DINO

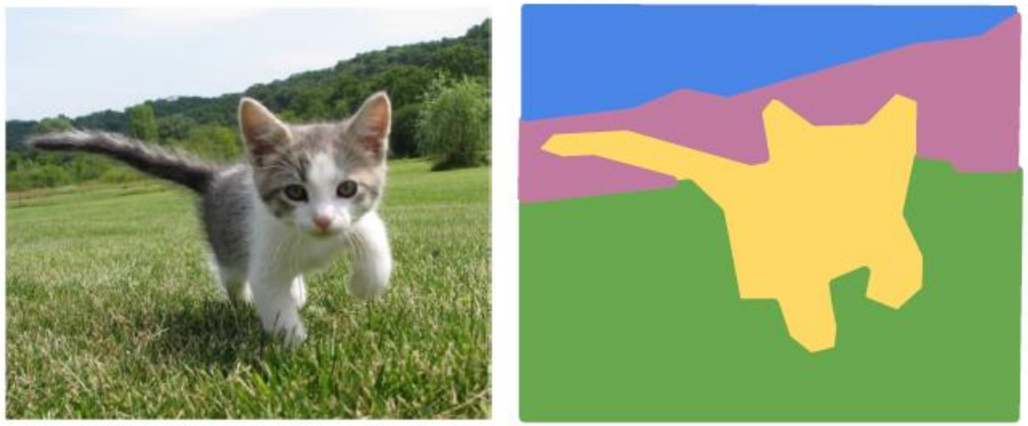
thresholding self-attention map of last layer:

source

not a full segmentation mask though ...

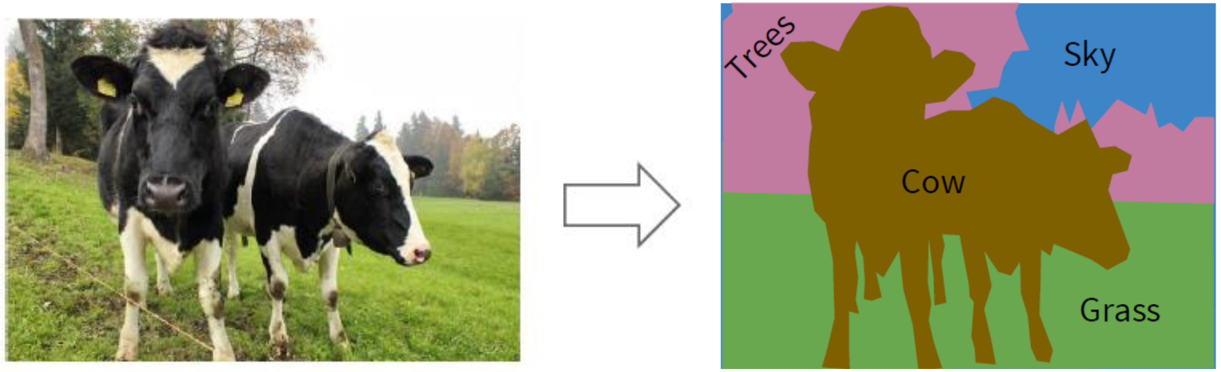
Classification of Each Pixel

segmentation: no objects, just pixels



GRASS, CAT, TREE,
SKY, ...

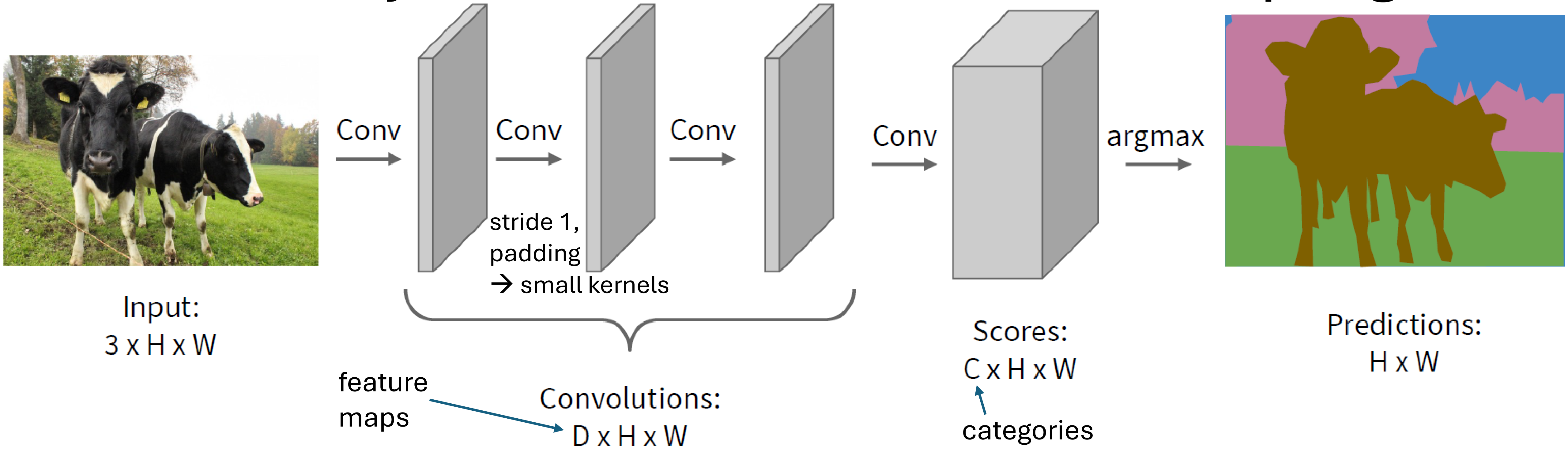
Paired training data: for each training image, each pixel is labeled with a semantic category.



At test time, classify each pixel of a new image.

minimize sum over classification losses (cross-entropy loss at every output pixel)

Idea: Fully Convolutional, No Downsampling



replace flattened, fully-connected classification layers with 1×1 convolutions
→ maintain spatial relationships and enable pixel-wise classification:
conversion of feature maps into classification heat maps (one for each class)

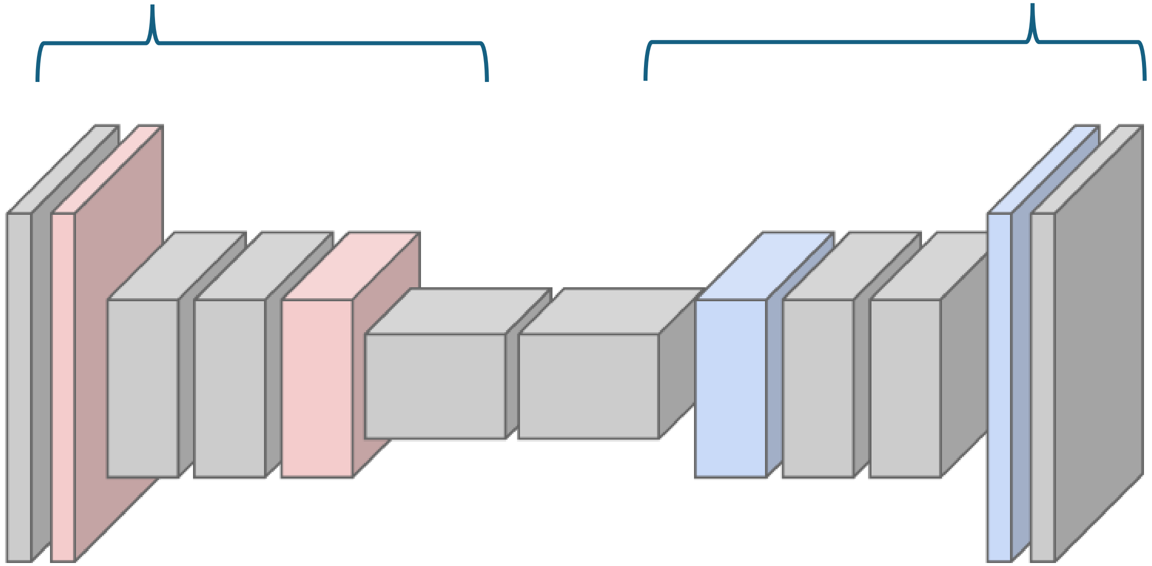
but no downsampling means small receptive field and no hierarchical learning

Upsampling to the Rescue

typical spatial feature extraction: convolution and pooling layers → downsampling

upsampling layers to restore original image size

Input:
3 x H x W



C x H x W

pixel-wise classification

Predictions:
H x W

different options for down- (pooling, strided convolution) and upsampling …

Reverse Pooling

resampling (no learned parameters)

Nearest Neighbor

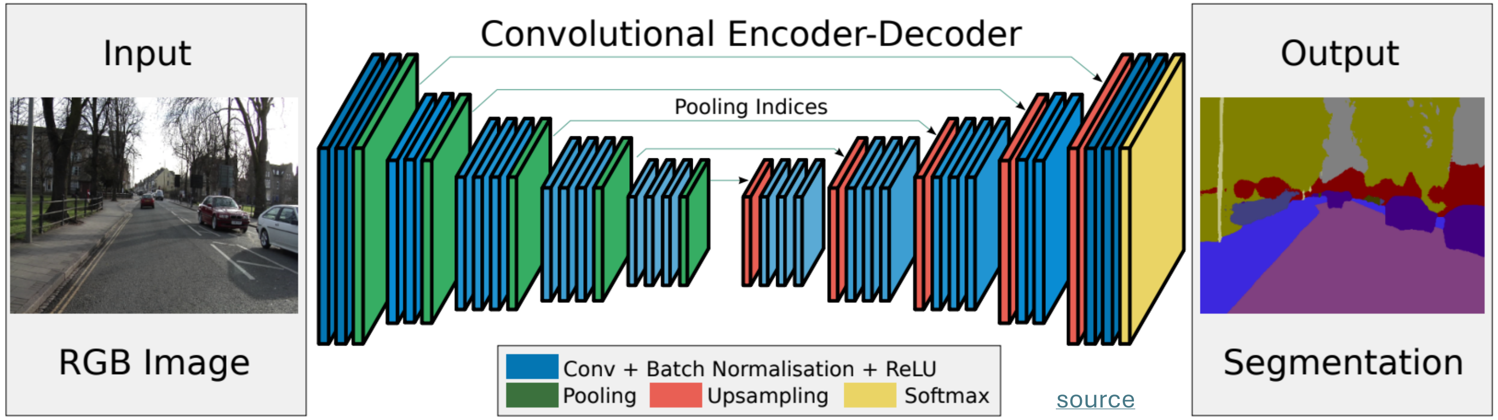
1	2
3	4

Input: 2 x 2

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

Output: 4 x 4

unpooling (recording max positions from pooling)



source

Max Pooling
Remember which element was max!

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

Input: 4 x 4

5	6
7	8

Output: 2 x 2

Rest of the network

Max Unpooling
Use positions from pooling layer

1	2
3	4

Input: 2 x 2

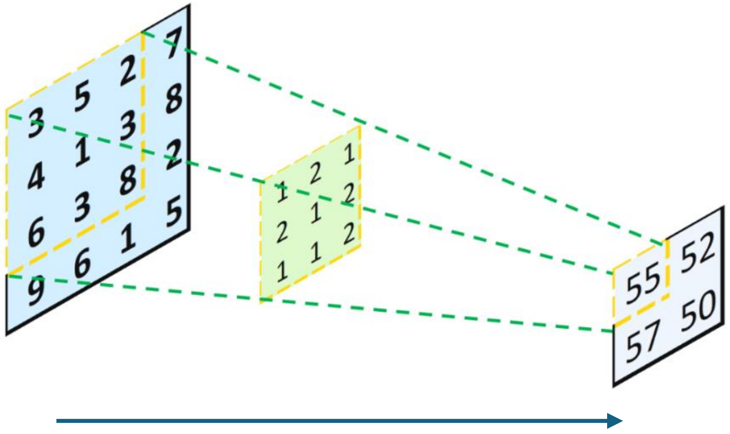
0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

Output: 4 x 4

filled with learned parameters in subsequent convolution

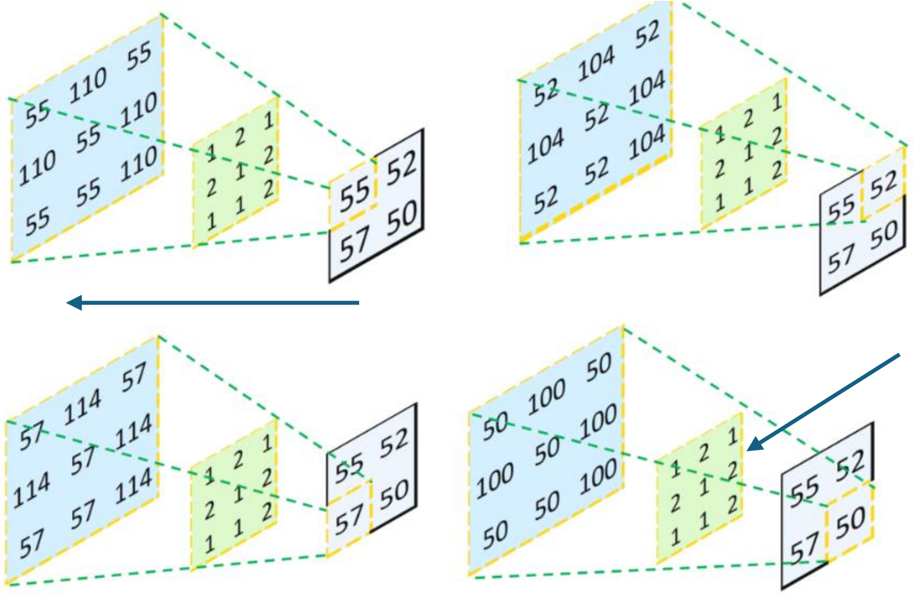
Reverse Convolution

convolution

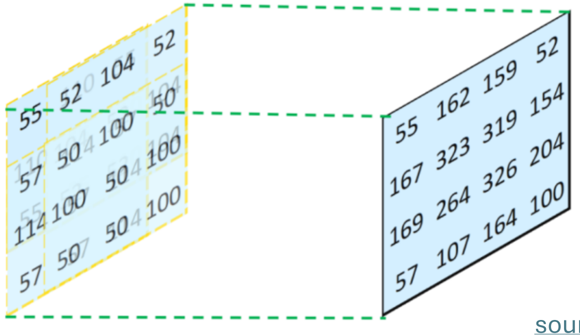


not inverse convolution
→ additional learned parameters

transposed convolution

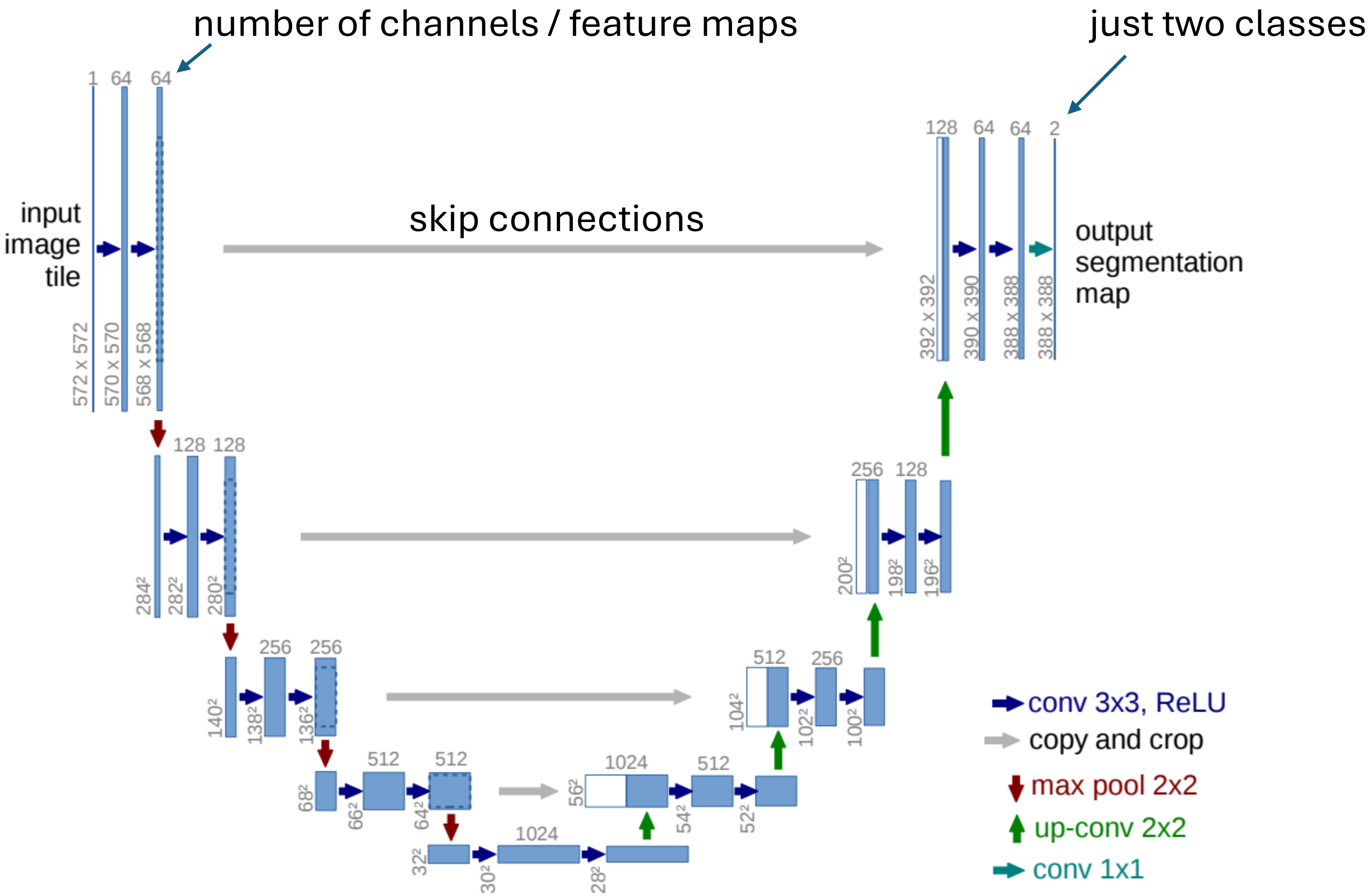


combine and sum overlaps:



source

U-Net



also used for learning of depth maps, image synthesis (diffusion), ...

Aside: Autoencoders

Autoencoder

(deep) encoder network

(deep) decoder network

learned together by minimizing differences between original input and reconstructed input (expressed as losses)

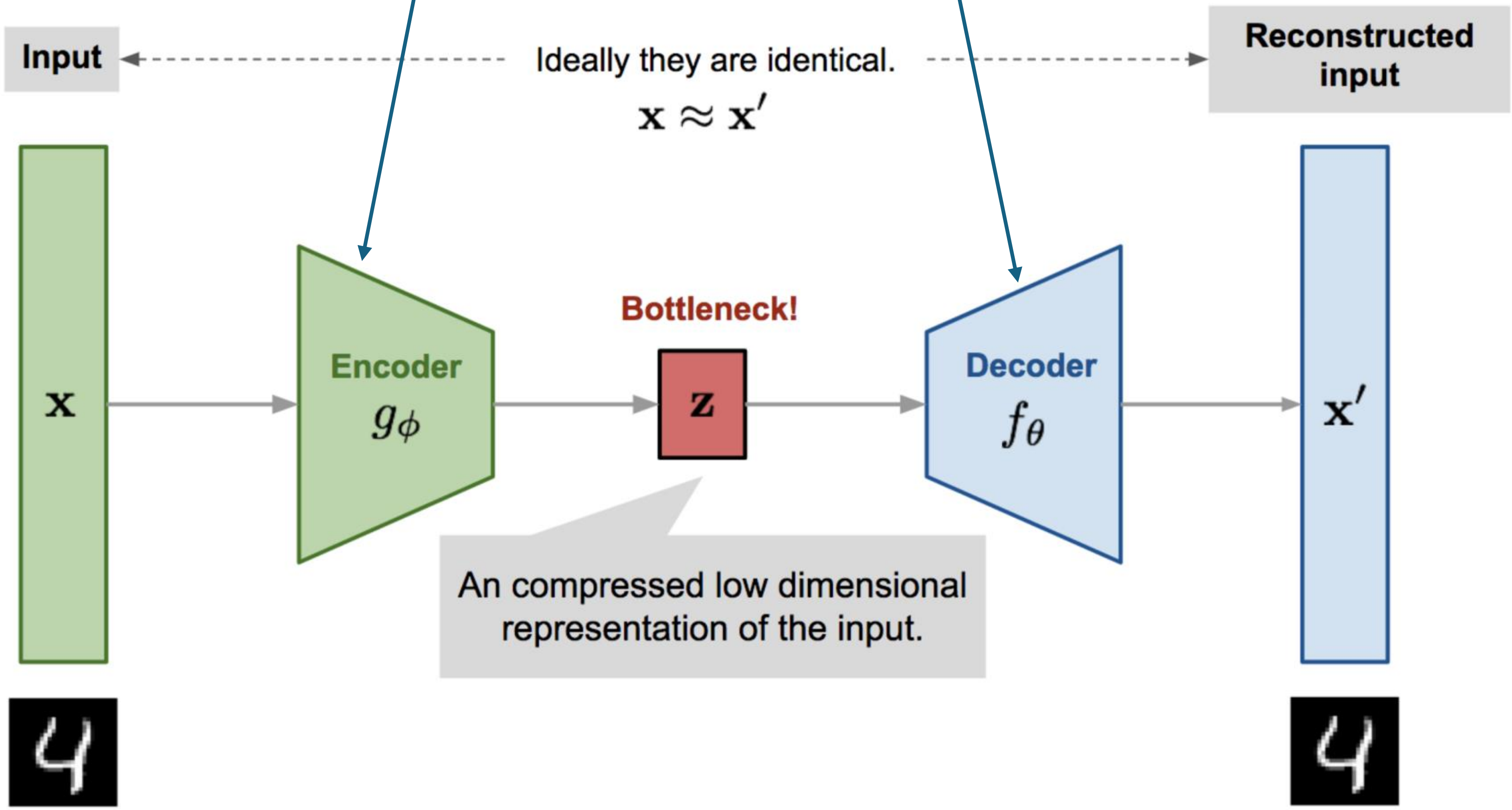
compressed intermediate representation: dimensionality reduction

(alternative to PCA)

for images:

convolutions

transposed convolutions



unsupervised learning

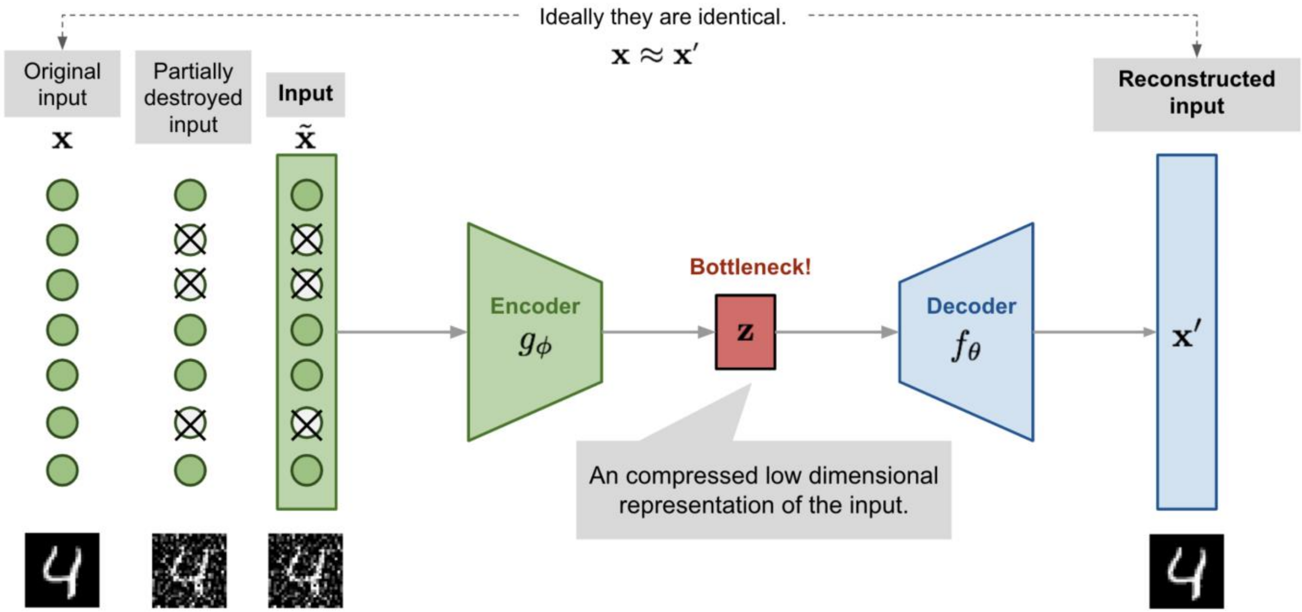
source

Denoising Autoencoder

goal: avoid overfitting and improve robustness of plain autoencoder

learn to remove noise of distorted input $\tilde{\boldsymbol{x}} \rightarrow$ restore original input $\boldsymbol{x}$

similar to dropout



source

alternative to deconvolution (image restoration)