



UNIVERSITY OF TECHNOLOGY  
IN THE EUROPEAN CAPITAL OF CULTURE  
CHEMNITZ

# Neurocomputing

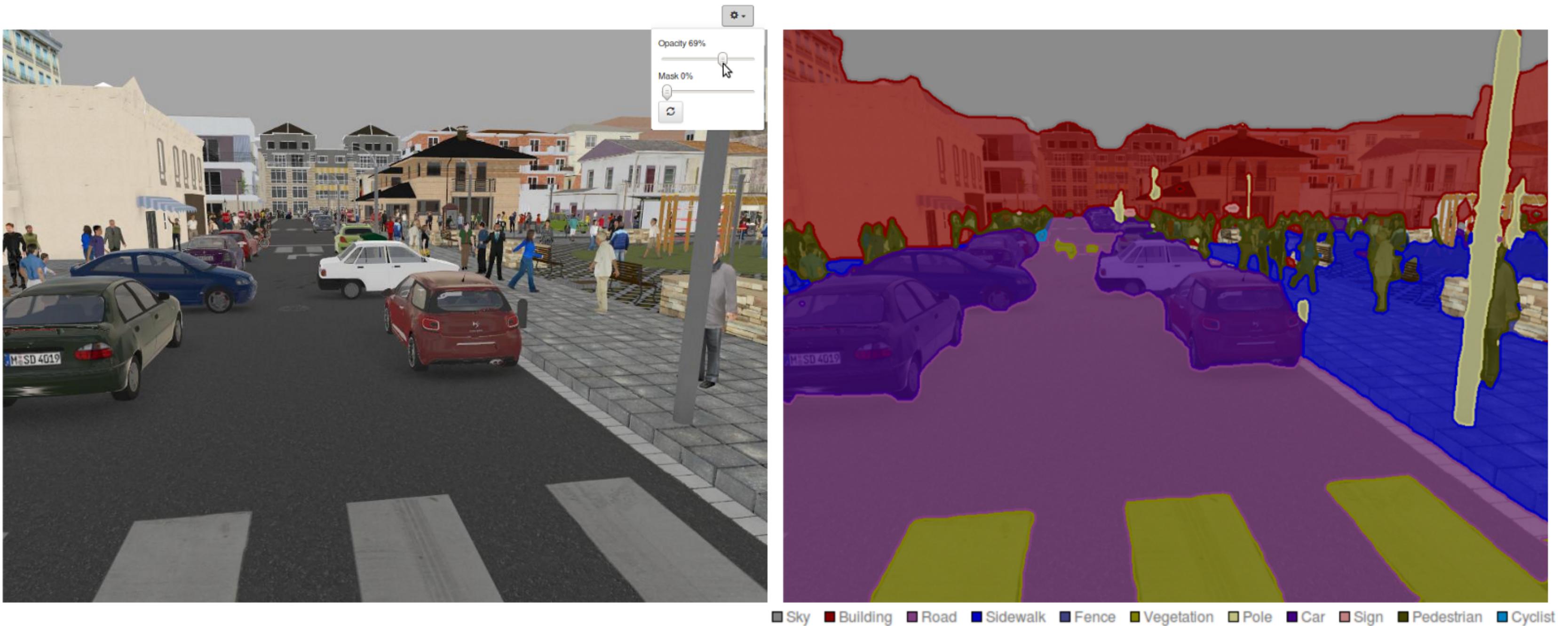
Semantic segmentation

Julien Vitay

Professur für Künstliche Intelligenz - Fakultät für Informatik

# Semantic segmentation

- **Semantic segmentation** is a class of segmentation methods where you use knowledge about the identity of objects to partition the image pixel-per-pixel.

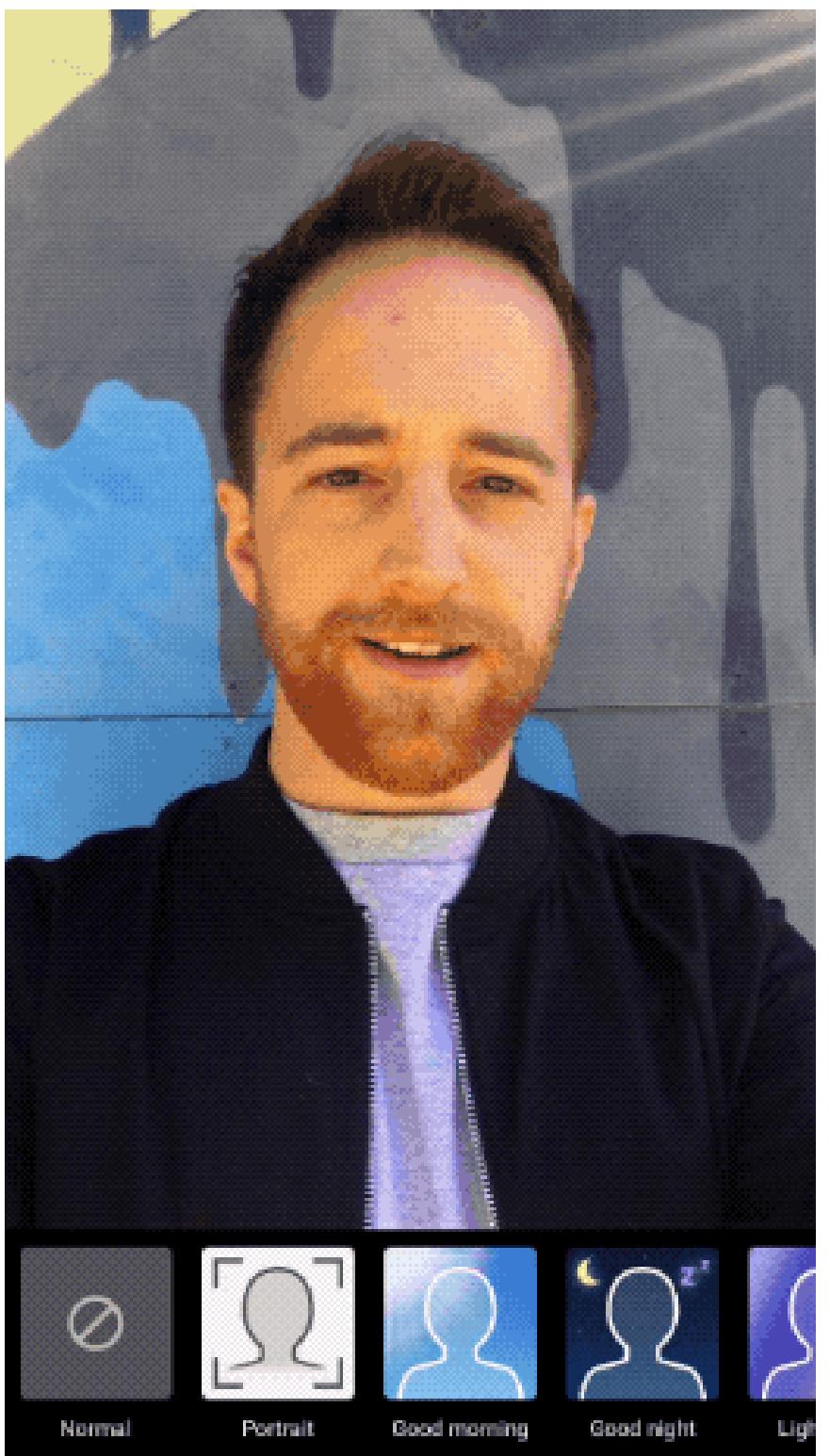


Source : <https://medium.com/nanonets/how-to-do-image-segmentation-using-deep-learning-c673cc5862ef>

- Classical segmentation methods only rely on the similarity between neighboring pixels, they do not use class information.
- The output of a semantic segmentation is another image, where each pixel represents the class.

# Semantic segmentation

- The classes can be binary, for example foreground/background, person/not, etc.
- Semantic segmentation networks are used for example in Youtube stories to add **virtual backgrounds** (background matting).



- Clothes can be segmented to allow for virtual try-ons.



Source: <https://ai.googleblog.com/2018/03/mobile-real-time-video-segmentation.html>

# Datasets for semantic segmentation



- There are many datasets freely available, but annotating such data is very painful, expensive and error-prone.
  - PASCAL VOC 2012 Segmentation Competition
  - COCO 2018 Stuff Segmentation Task
  - BDD100K: A Large-scale Diverse Driving Video Database
  - Cambridge-driving Labeled Video Database (CamVid)
  - Cityscapes Dataset
  - Mapillary Vistas Dataset
  - ApolloScape Scene Parsing
  - KITTI pixel-level semantic segmentation

# Output encoding

- Each pixel of the input image is associated to a label (as in classification).



segmented →

1: Person  
2: Purse  
3: Plants/Grass  
4: Sidewalk  
5: Building/Structures

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

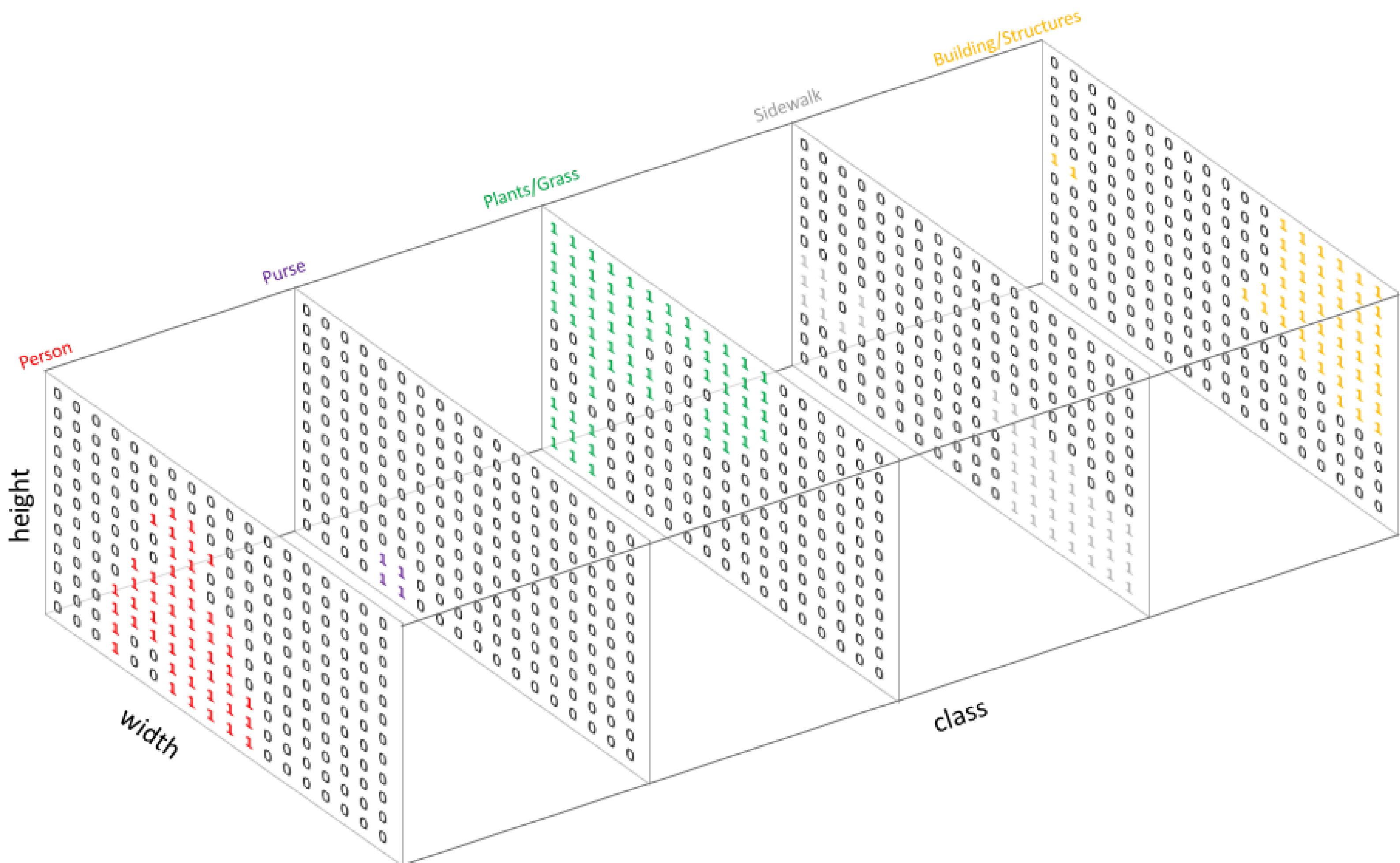
Input

Semantic Labels

Source : <https://medium.com/nanonets/how-to-do-image-segmentation-using-deep-learning-c673cc5862ef>

# Output encoding

- A **one-hot encoding** of the segmented image is therefore a tensor:

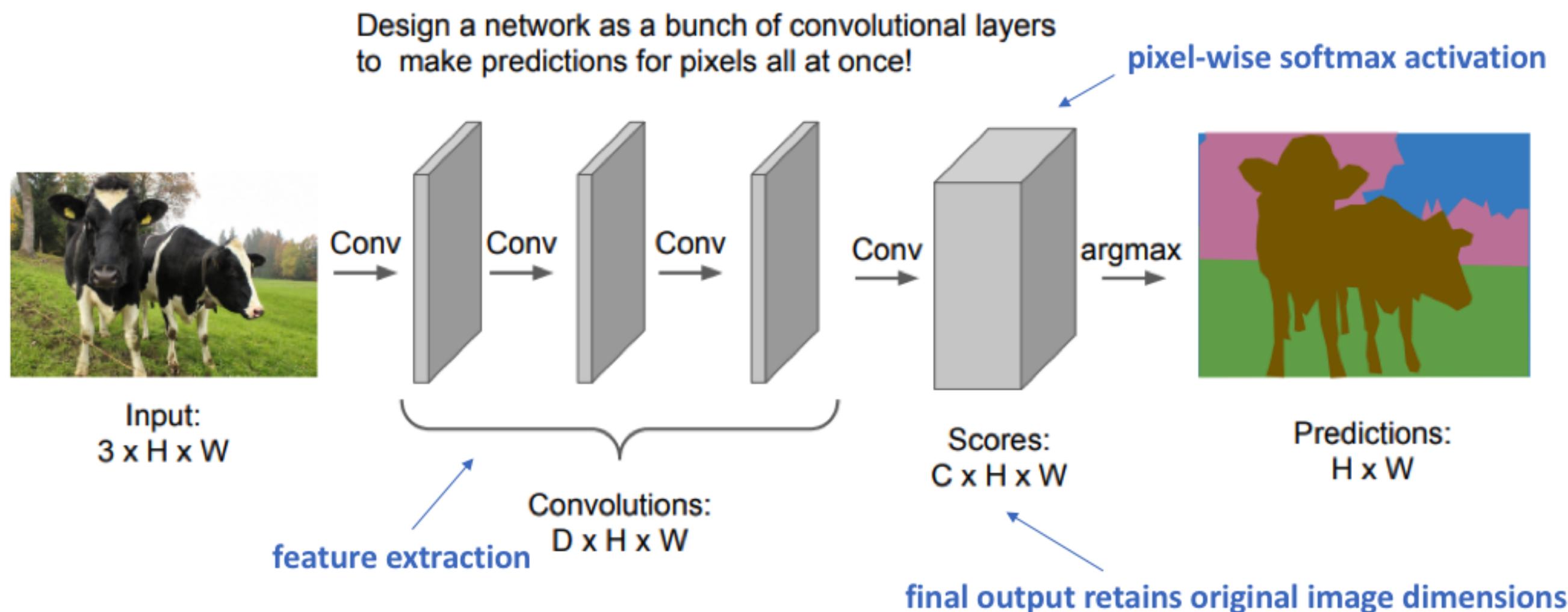


Source : <https://medium.com/nanoneets/how-to-do-image-segmentation-using-deep-learning-c673cc5862ef>

# Fully convolutional network

- A **fully convolutional network** only has convolutional layers and learns to predict the output tensor.
- The last layer has a pixel-wise softmax activation. We minimize the **pixel-wise cross-entropy loss**

$$\mathcal{L}(\theta) = \mathbb{E}_{\mathcal{D}} \left[ - \sum_{\text{pixels}} \sum_{\text{classes}} t_i \log y_i \right]$$

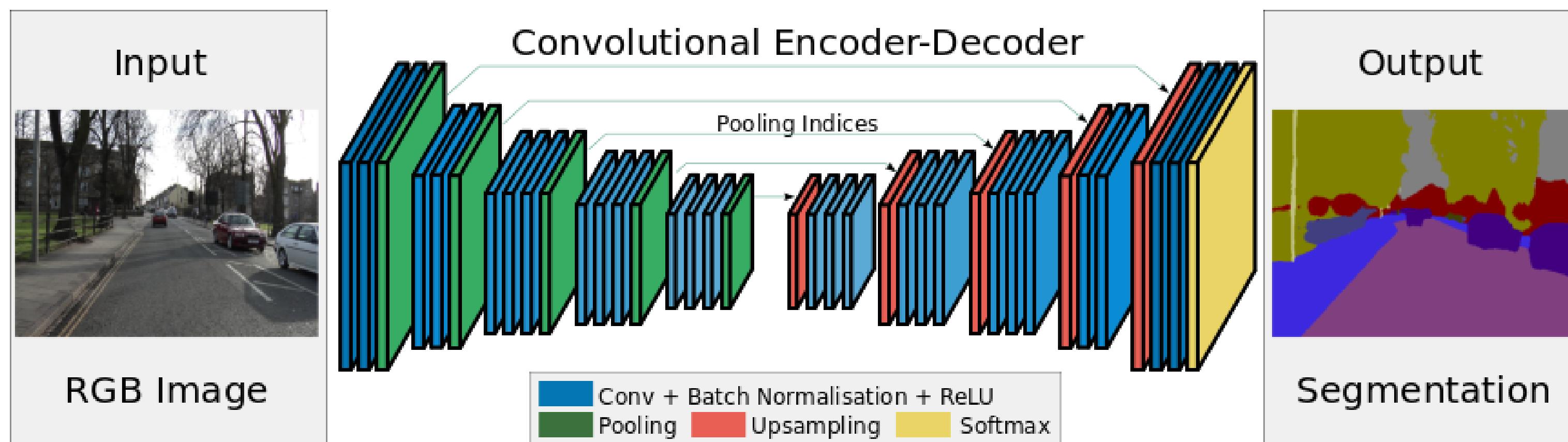


Source : [http://cs231n.stanford.edu/slides/2017/cs231n\\_2017\\_lecture11.pdf](http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf)

- Downside: the image size is preserved throughout the network: computationally expensive. It is therefore difficult to increase the number of features in each convolutional layer.

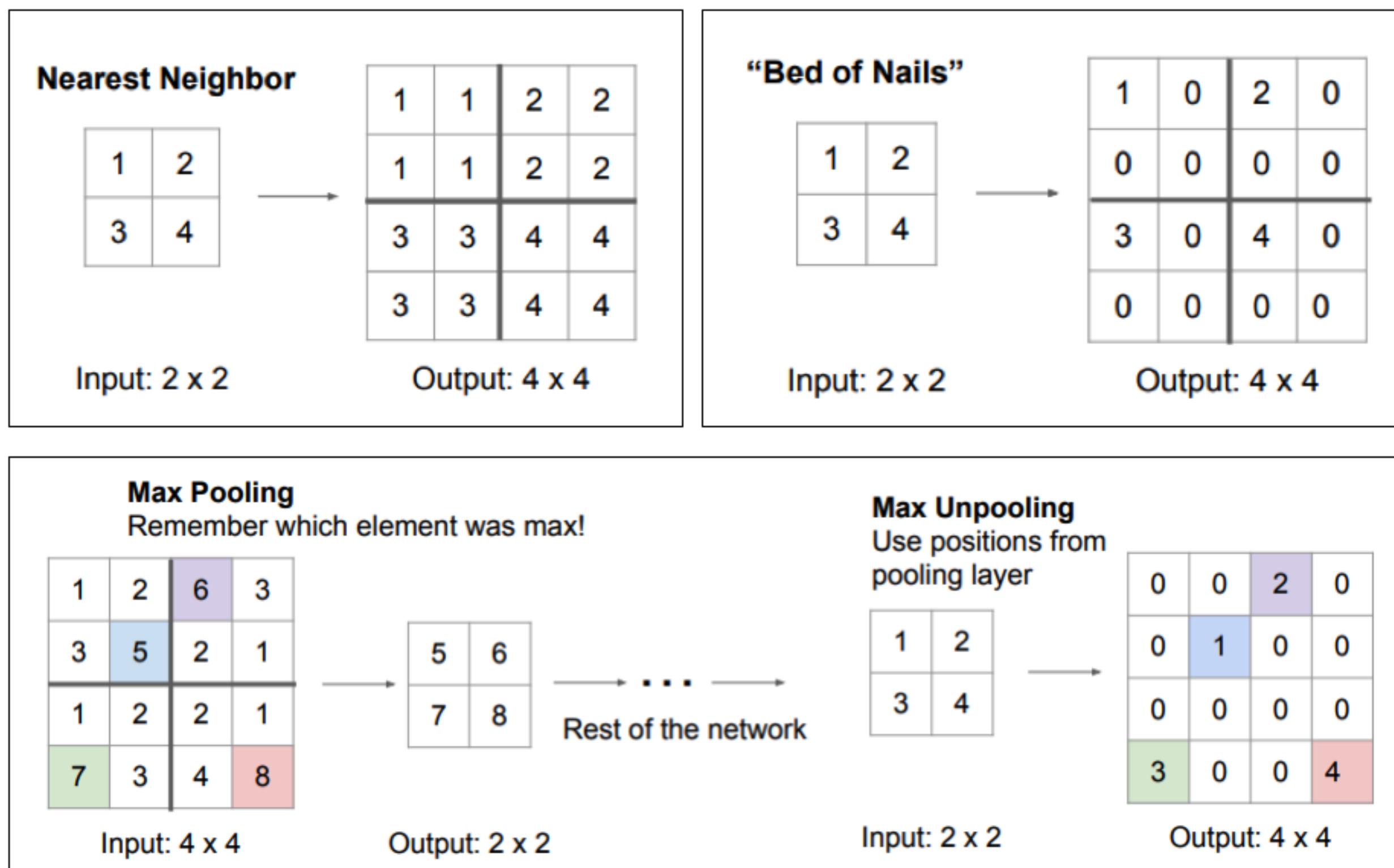
# SegNet: segmentation network

- SegNet has an **encoder-decoder** architecture, with max-pooling to decrease the spatial resolution while increasing the number of features.
- But what is the inverse of max-pooling? Upsampling operation.



# Upsampling: some methods

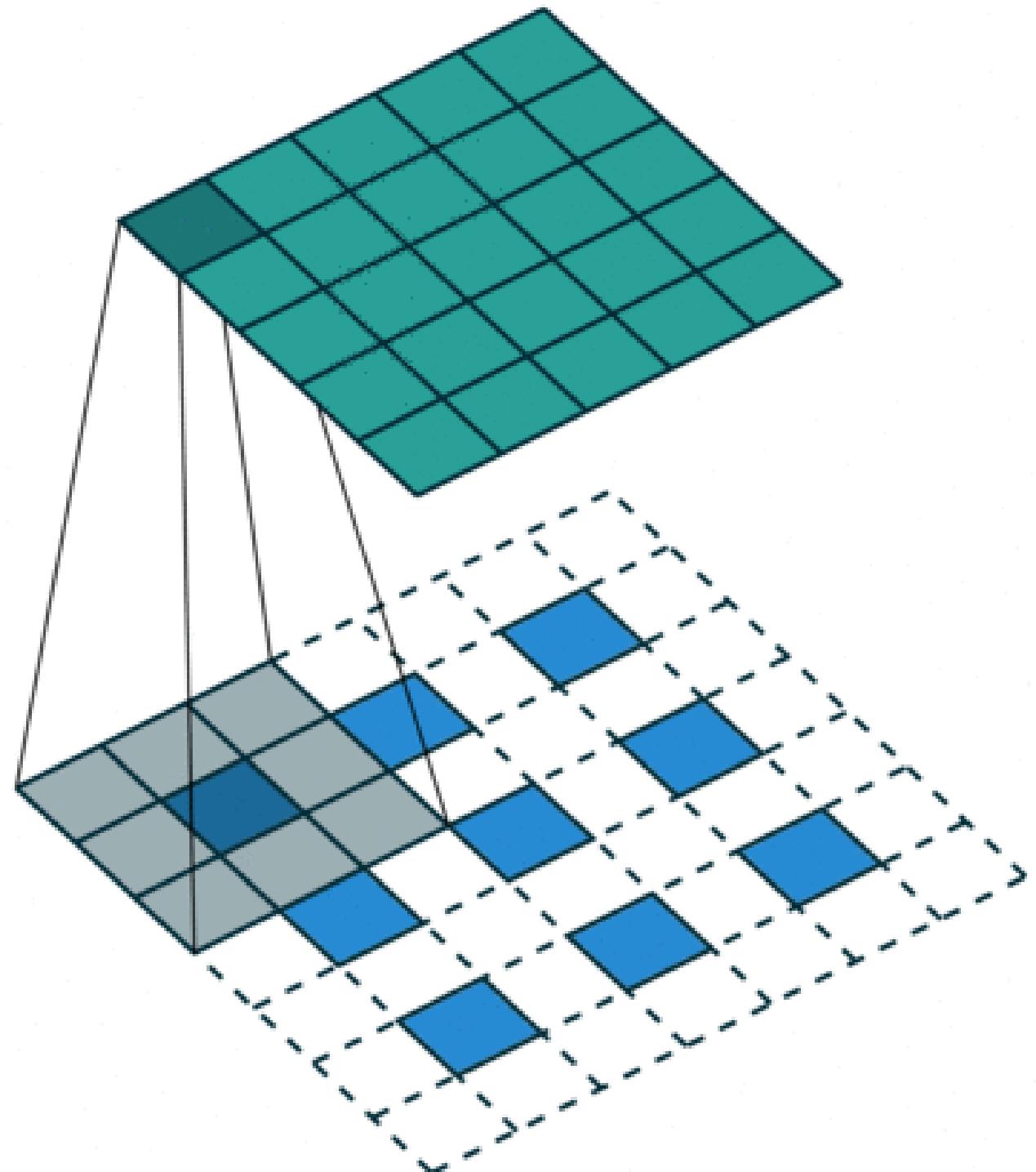
- Nearest neighbor and Bed of nails would just make random decisions for the upsampling.
- In SegNet, max-unpooling uses the information of the corresponding max-pooling layer in the encoder to place pixels adequately.



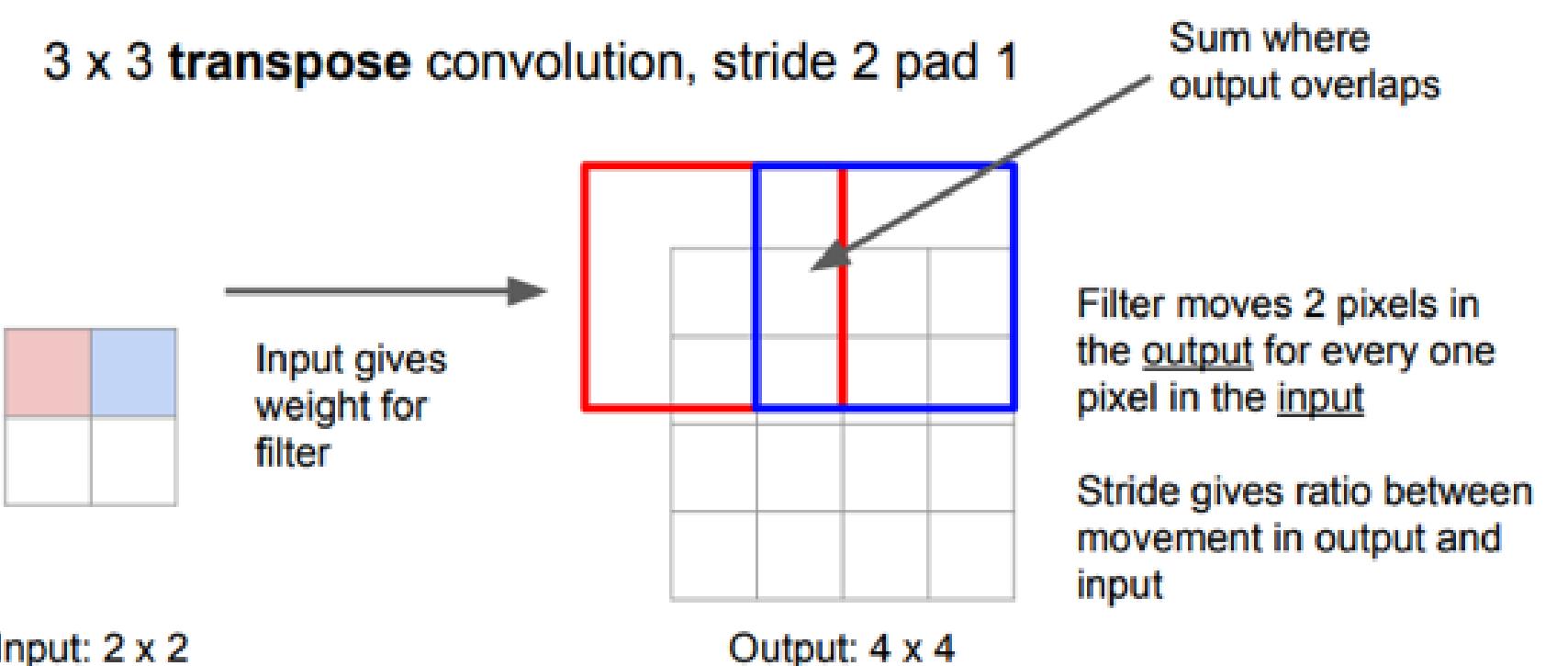
Source : [http://cs231n.stanford.edu/slides/2017/cs231n\\_2017\\_lecture11.pdf](http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture11.pdf)

# Upsampling: transposed convolution

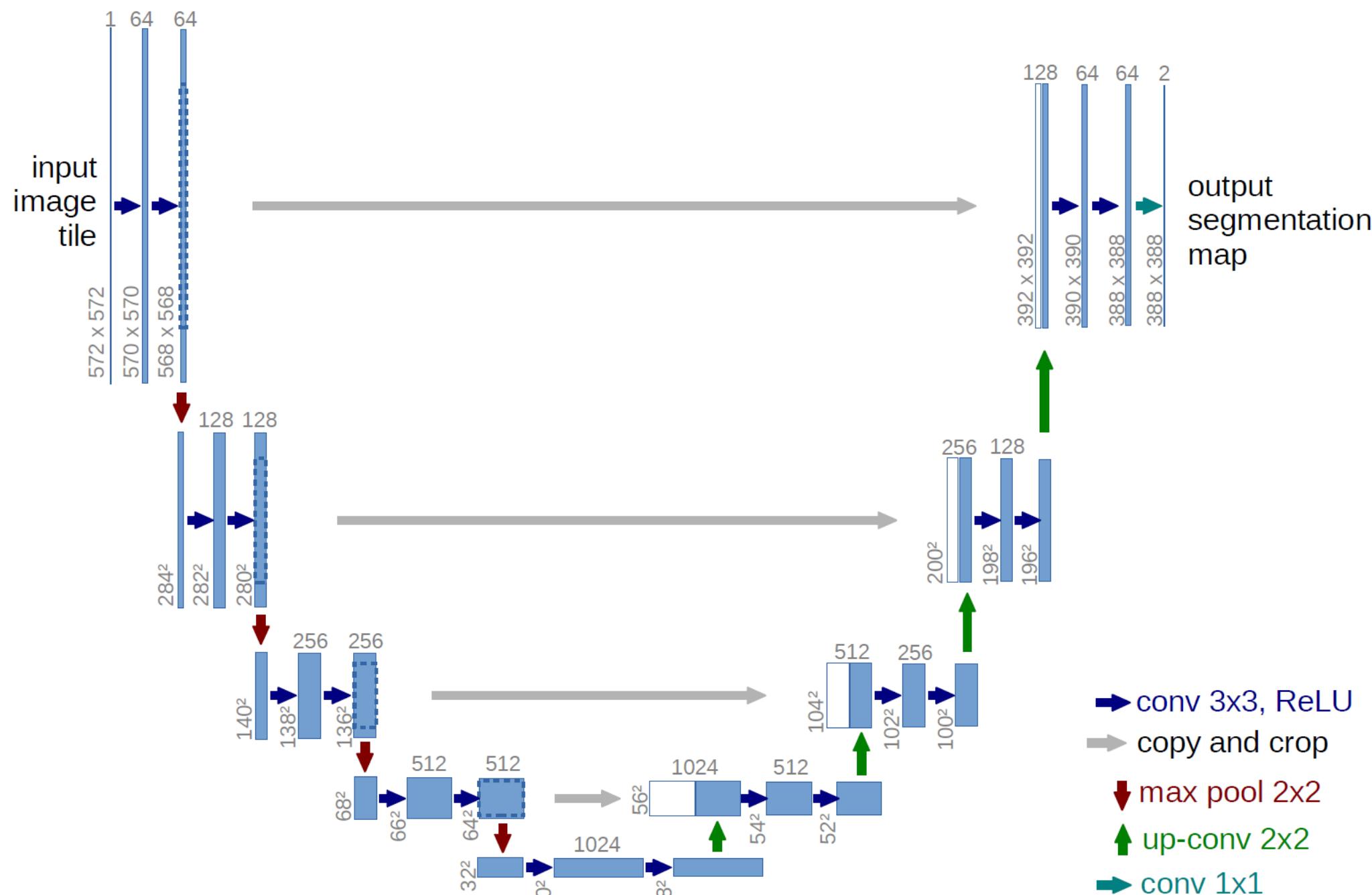
- Another popular option in the followers of SegNet is the **transposed convolution**.



- The original feature map is upsampled by putting zeros between the values.
- A learned filter performs a regular convolution to produce an upsampled feature map.
- Works well when convolutions with stride are used in the encoder.
- Quite expensive computationally.



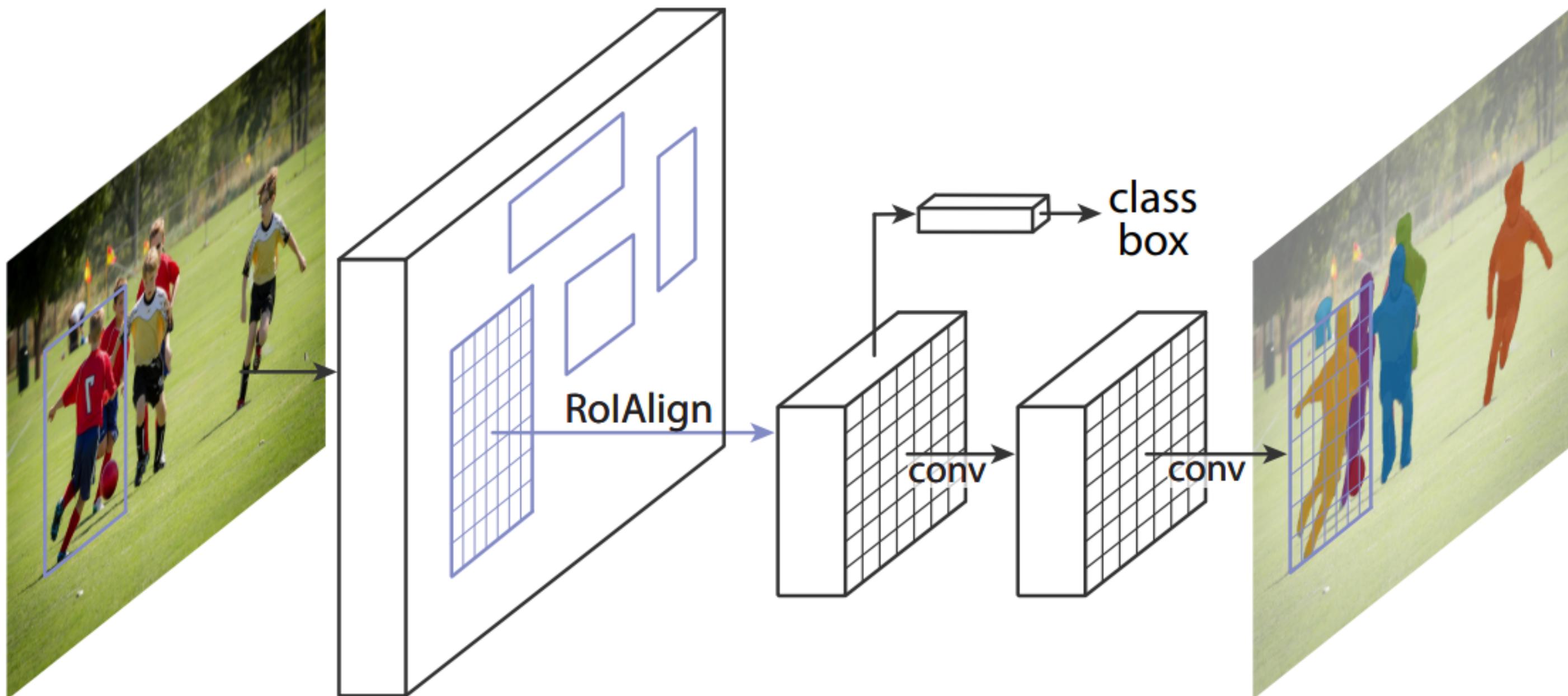
# U-Net



- The problem of SegNet is that small details (small scales) are lost because of the max-pooling. the segmentation is not precise.
- The solution proposed by **U-Net** is to add **skip connections** (as in ResNet) between different levels of the encoder-decoder.
- The final segmentation depends both on:
  - large-scale information computed in the middle of the encoder-decoder.
  - small-scale information processed in the early layers of the encoder.

# Mask R-CNN

- For many applications, segmenting the background is useless. A two-stage approach can save computations.
- **Mask R-CNN** uses faster R-CNN to extract bounding boxes around interesting objects, followed by the prediction of a **mask** to segment the object.



# Mask R-CNN



Figure 2. **Mask R-CNN** results on the COCO test set. These results are based on ResNet-101 [19], achieving a *mask AP* of 35.7 and running at 5 fps. Masks are shown in color, and bounding box, category, and confidences are also shown.

# Mask R-CNN

