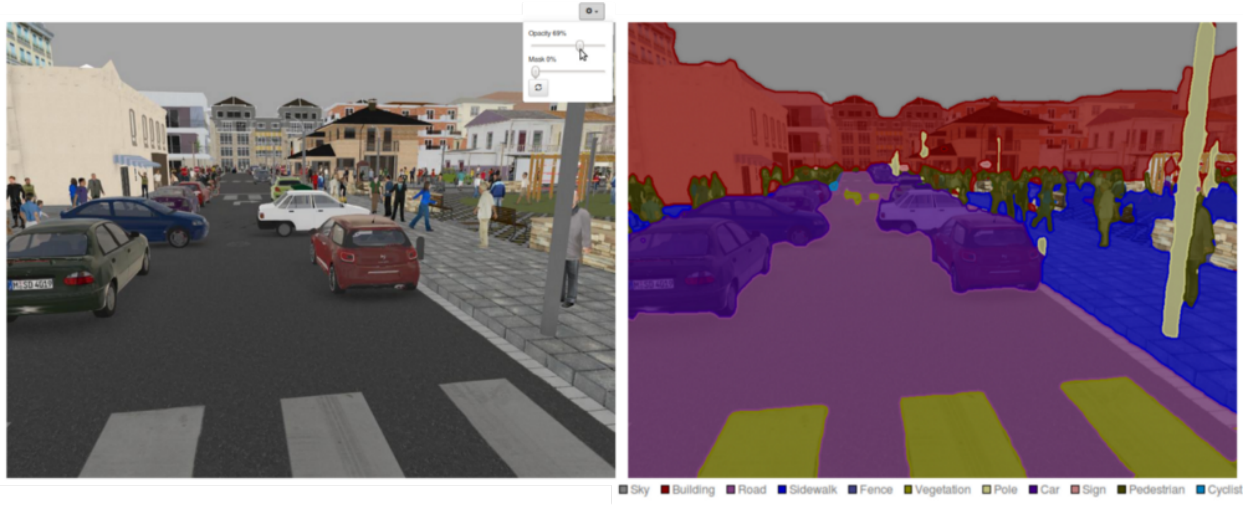
Advanced topics

1) Pre-training and fine tunning.
2) Image segmentation
3) Image restoration/denoising
4) Image detection
5) Working with sequences

2) Image segmentation



- Classification: Predict one label per image.
- Detection: Predict boxes where objects are and label them
- Segmentation: Predict one label per pixel

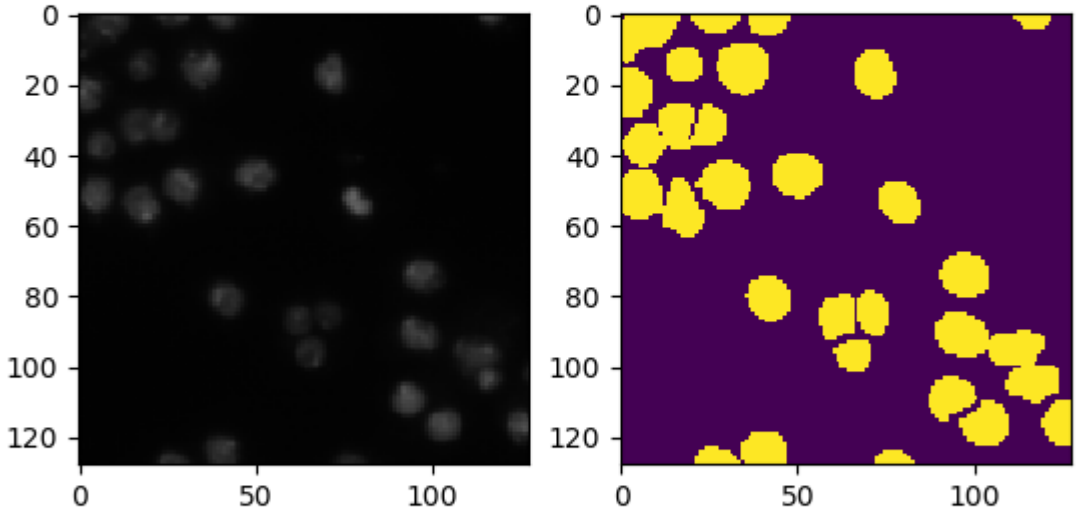
  - We need images and human curated masks to perform segmentation → difficult to collect big datasets

$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Typical evaluation metrics: IoU

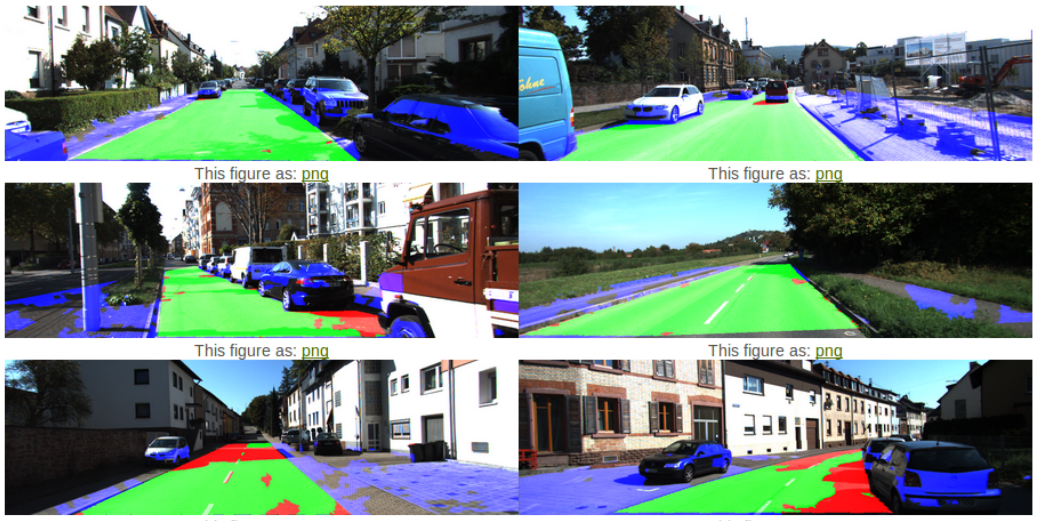
2) Image segmentation: Example

https://www.kaggle.com/c/data-science-bowl-2018



x_train

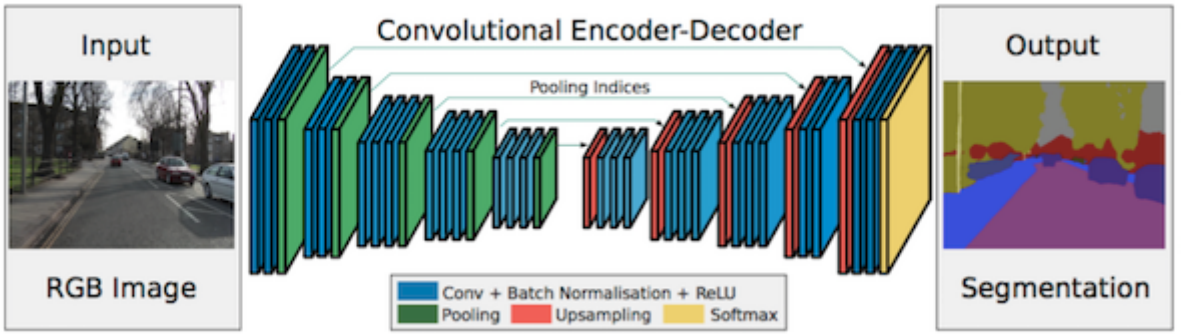
y_train



http://www.cvlibs.net/datasets/kitti/eval_road_detail.php?result=3748e213cf8e0100b7a26198114b3cdc7caa3aff

2) Image segmentation

Approaches: generally they have one encoder part and one decoder part, first part is usually pre-trained to perform classification



2) Image segmentation

One of the most popular architectures: U-Net

https://arxiv.org/pdf/1505.04597.pdf

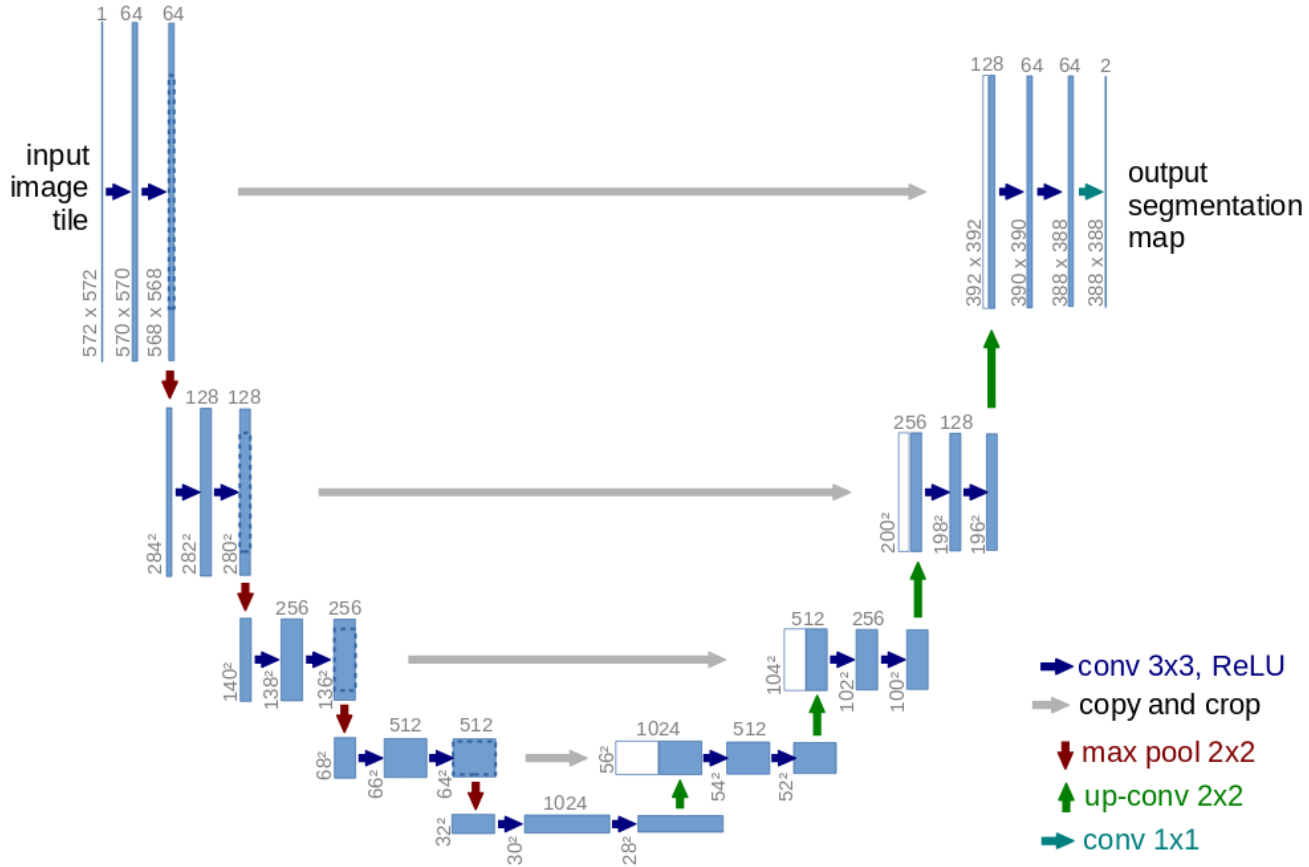


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows denote the different operations.

2) Image segmentation

One of the most popular architectures: U-Net

https://arxiv.org/pdf/1505.04597.pdf

My example: ./3_advanced_topics/code_segmentation/e1_nuclei_segm.py

2) Image restoration