Deep Learning Segmentation Structure

This document describes 3 structures selected for bio-image segmentation: **AlexNet**, **U-Net**, **SegNet**.

AlexNet

AlexNet is also called ImageNet / K-Net(paper: Deep Contour Segmentation)

Structure

ReLU Nonlinearity

For sigmoid and tanh activation function, there are 2 disadvantages: vanishing gredient and exploding gradient problem. More information

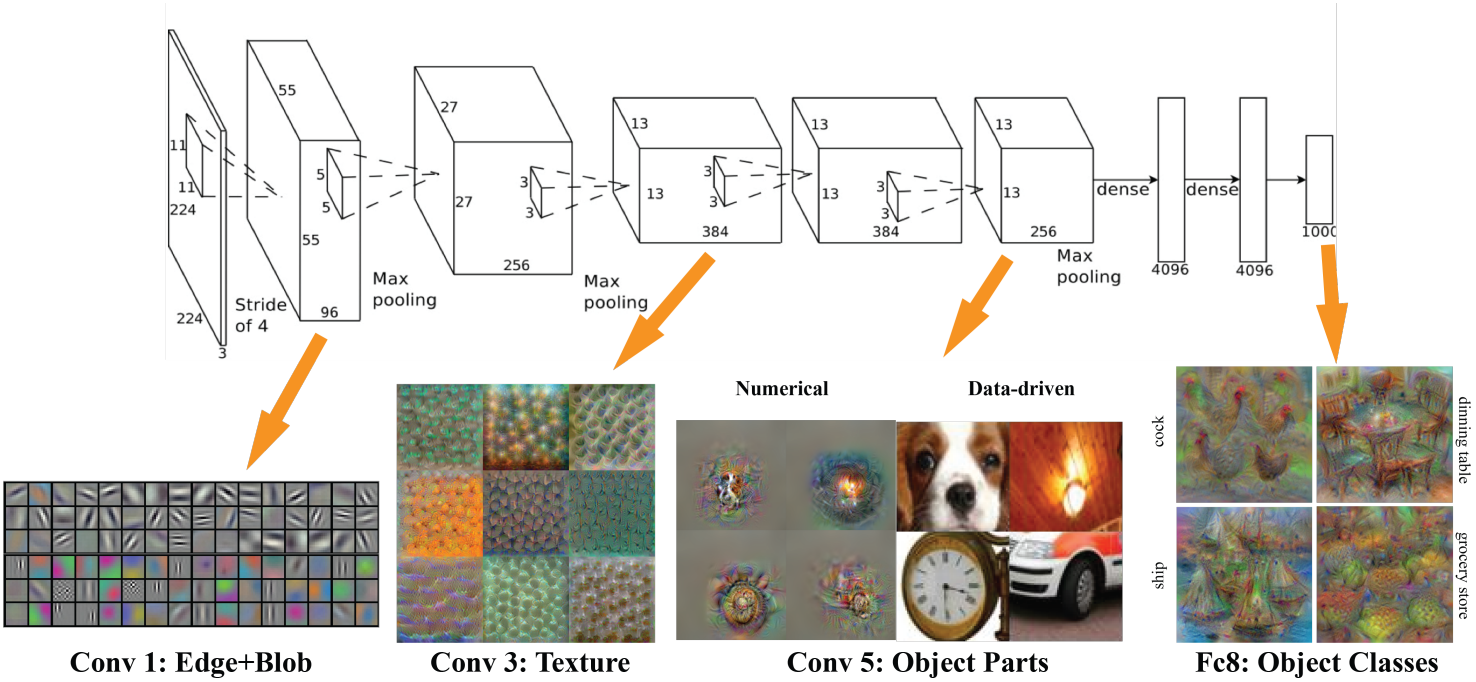
Training on Multiple GPUs

Local Response Normalization

Although ReLUs can prevent data from saturating without requiring input normalization (For sigmoid and tanh, gradient is the highest at 0), the unbounded activation of ReLUs requires local response normalization. This conception comes from bio-conception: lateral inhibition. More information

Overlapping Pooling

Overall Architecture



Reduce Overfitting

Data Augmentation

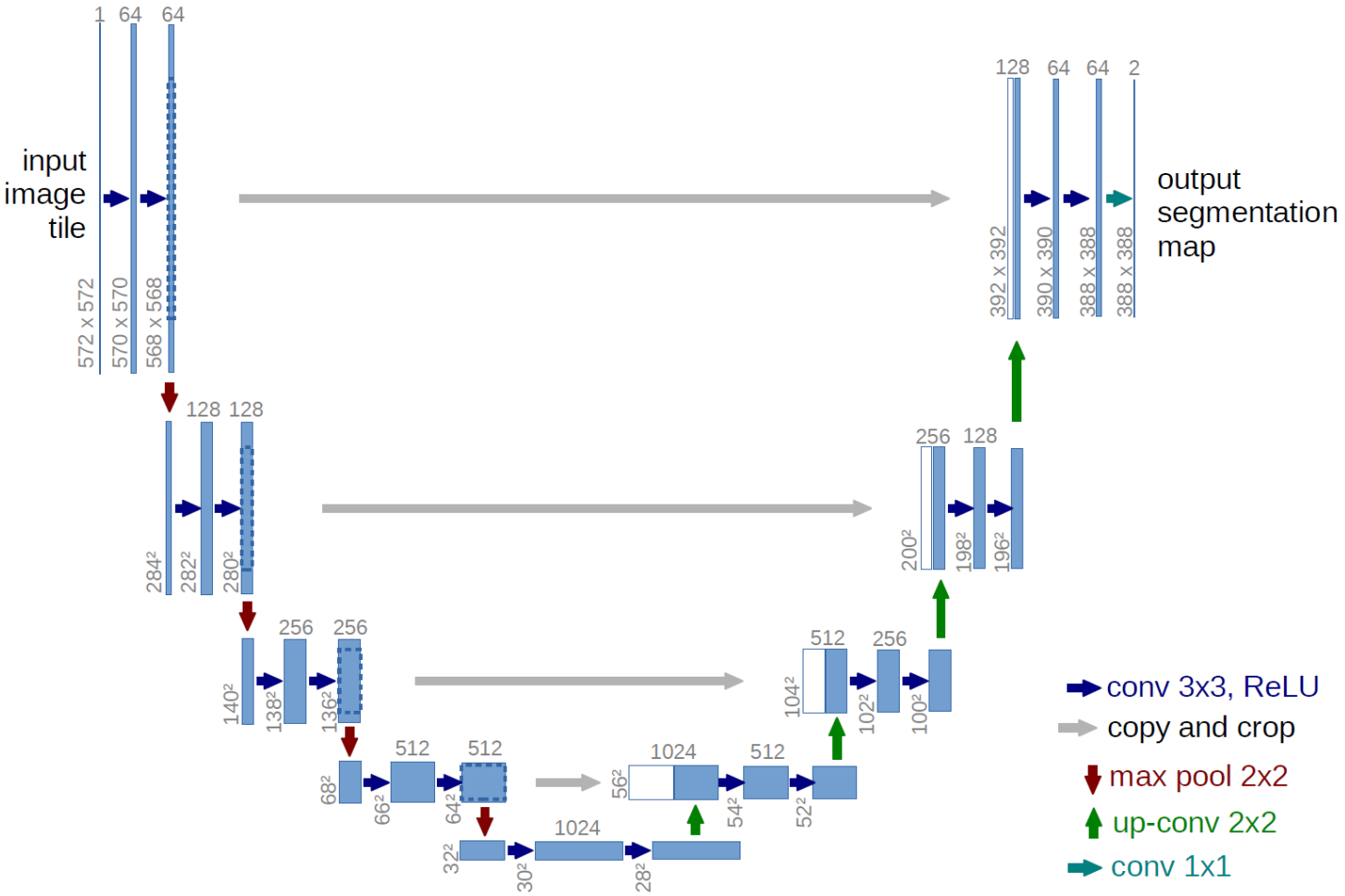
1. Image translations and horizontal reflections.
   Translation: Crop 224 * 224 patches(and their horizontal reflections) from 256 * 256 images. At test time, the network makes a prediction by the same way to get 10 patches per image and averaging the predictions made by the network's softmax layer.
2. Altering the intensities of the RGB channels in training images by PCA (add multiples of the found principal components, with magnitudes propotional to the corresponding eigenvalues times a random variable drawn from a standard Gaussian).

Dropout

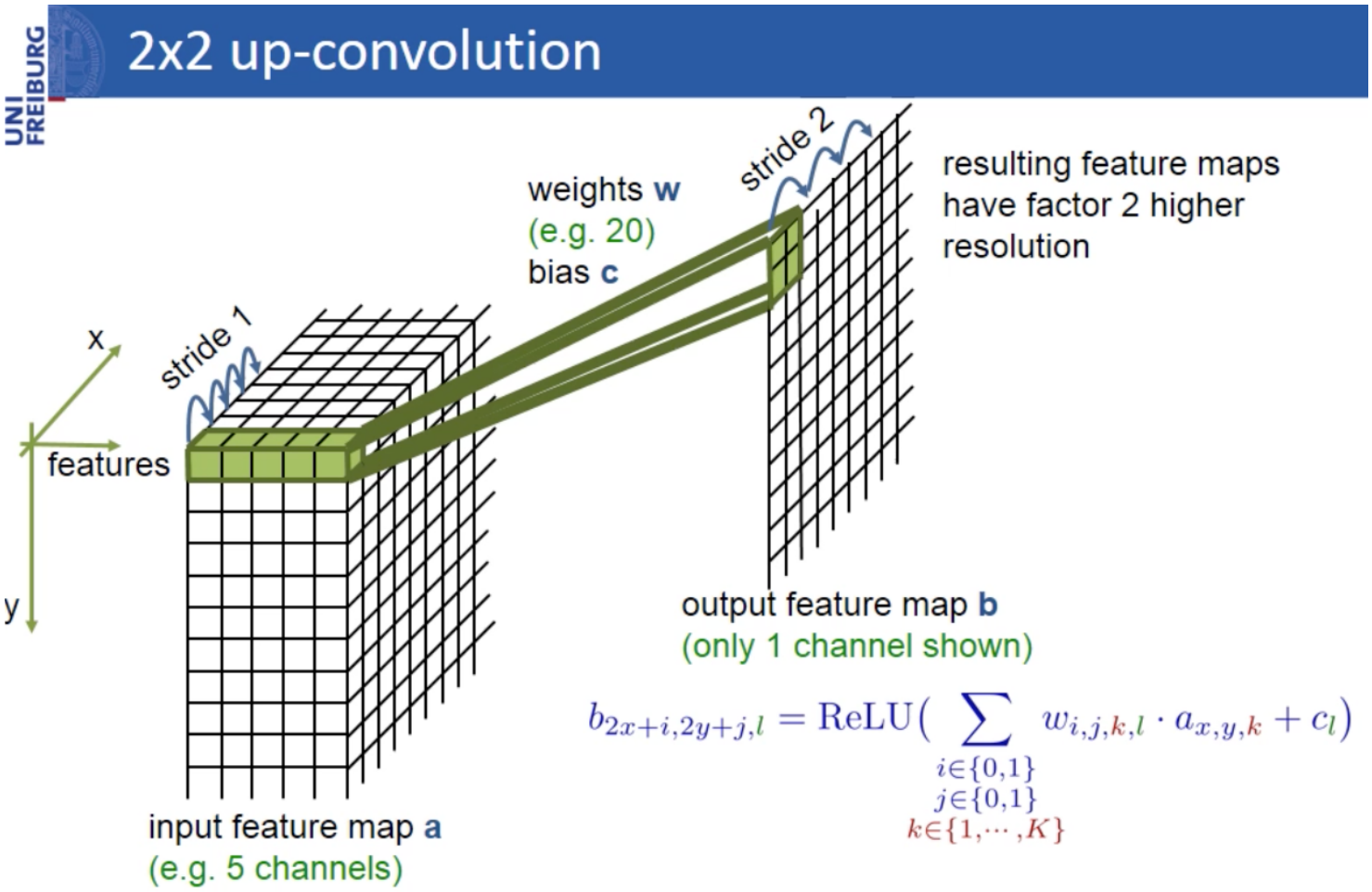
Set a node's output to zero with probabilty 0.5 while training. At test time, we use all nodes but multiply their outputs by 0.5.

U-Net

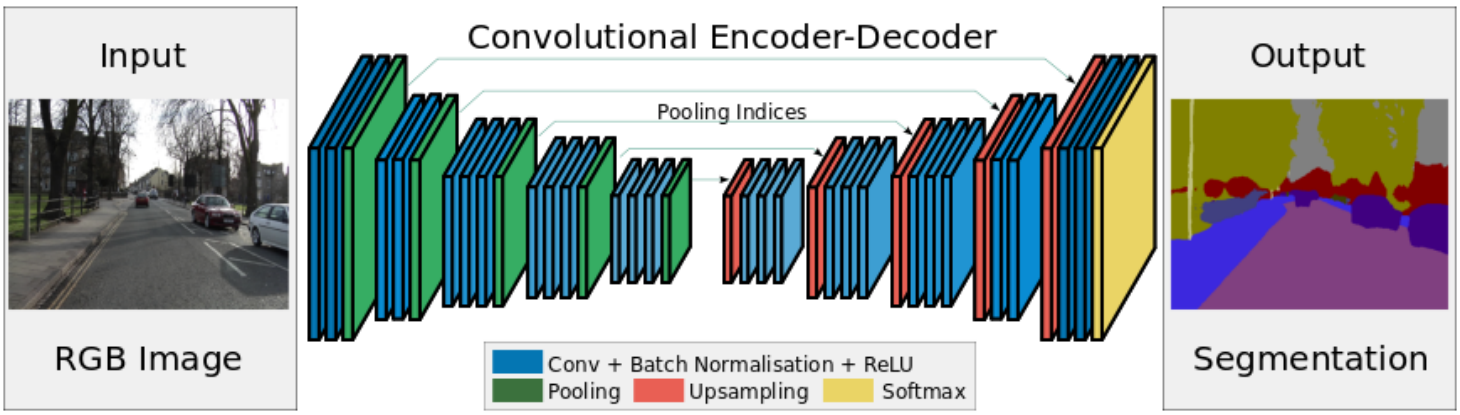
Architecture


input image tile
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1

Upsampling


UNI FREIBURG
2x2 up-convolution
stride 2
weights w
(e.g. 20)
bias c
resulting feature maps
have factor 2 higher
resolution
x
stride 1
features
y
output feature map b
(only 1 channel shown)
$b_{2x+i,2y+j,l} = \mathrm{ReLU}\Big(\sum_{\substack{i\in\{0,1\} \\ j\in\{0,1\} \\ k\in\{1,\cdots,K\}}} w_{i,j,k,l} \cdot a_{x,y,k} + c_l \Big)$
input feature map a
(e.g. 5 channels)

SegNet


Input
RGB Image
Convolutional Encoder-Decoder
Pooling Indices
Conv + Batch Normalisation + ReLU
Pooling
Upsampling
Softmax
Output
Segmentation

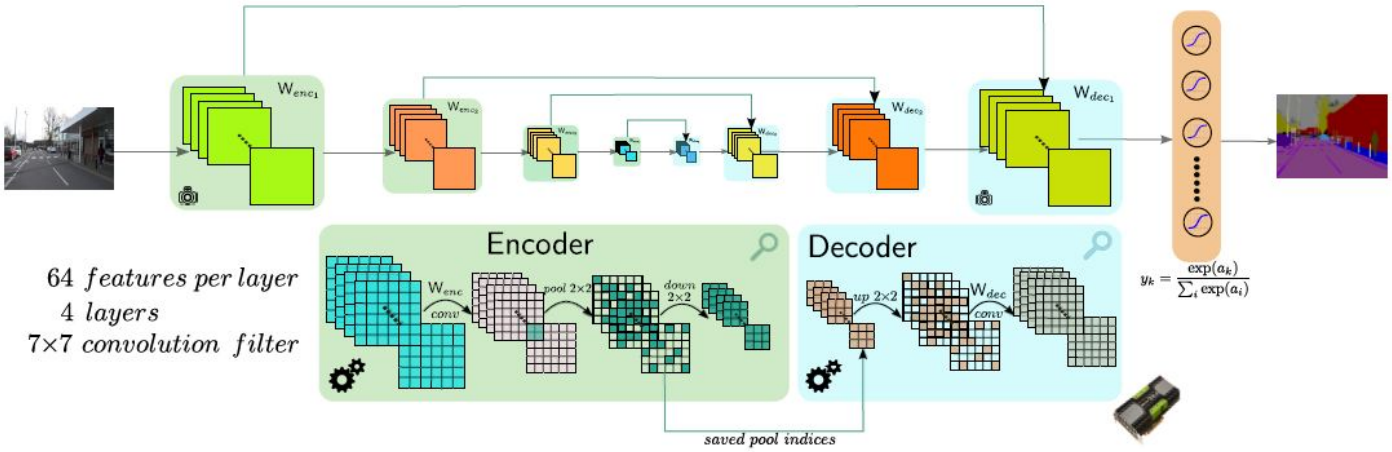


Figure 1. A 4 layer SegNet which takes in an RGB input image and performs *feed-forward* computation to obtain pixel-wise labelling. A stack of feature encoders is followed by a corresponding decoders. The soft-max layer classifies each pixel independently using the features input by the last decoder. An encoder uses the convolution-ReLU-max pooling-subsampling pipeline. A decoder upsamples its input using the transferred pool indices from its encoder. It then performs convolution with a trainable filter bank.