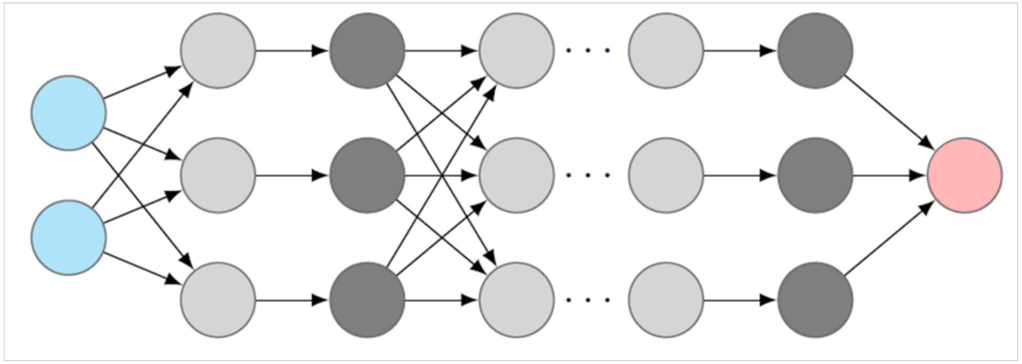
Convolutional Neural Networks

- Skip connections at the example of ResNet -

Lecturer: Michael Möller – michael.moeller@uni-siegen.de
Exercises: Hartmut Bauermeister – hartmut.bauermeister@uni-siegen.de

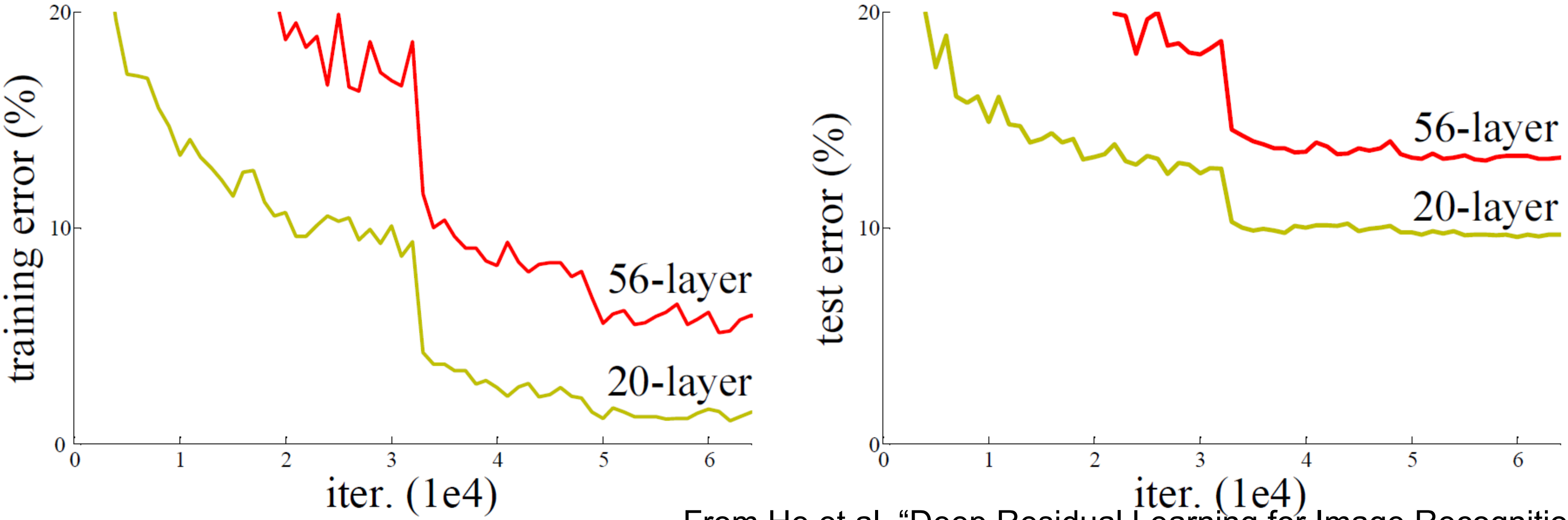


Going deeper?

Conclusion of the VGG-Team: The basic idea of the network architecture does not really differ from the ones proposed by LeCun et al. 1989, but is much, much deeper.

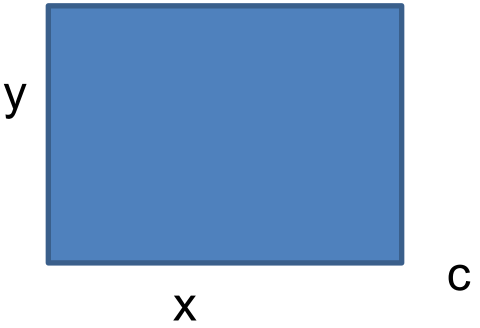
Deeper = better? Why don't we go from 19 to 40 to 400 or to 4000 layers?

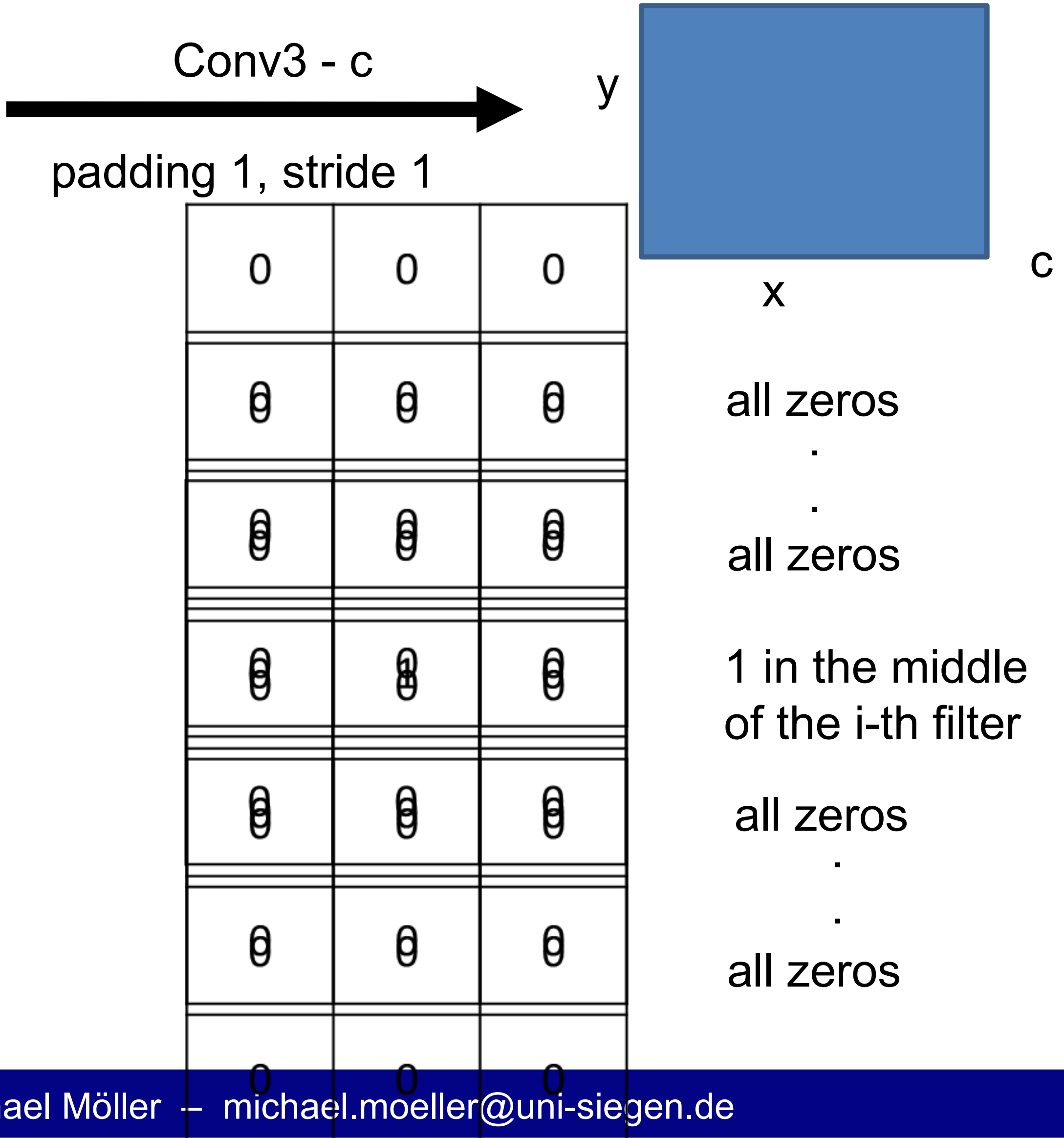
Strange behavior: At some point, going deeper does not improve the results DESPITE the network neither suffering from overfitting nor from an obvious problem in the training!



From He et al. “Deep Residual Learning for Image Recognition”

The problem still is the training!





all zeros
.
.
all zeros

1 in the middle of the i-th filter

all zeros
.
.
all zeros

This choice of the i-th kernel along with a zero bias makes this layer an identity!

More layers make the network strictly more expressive!

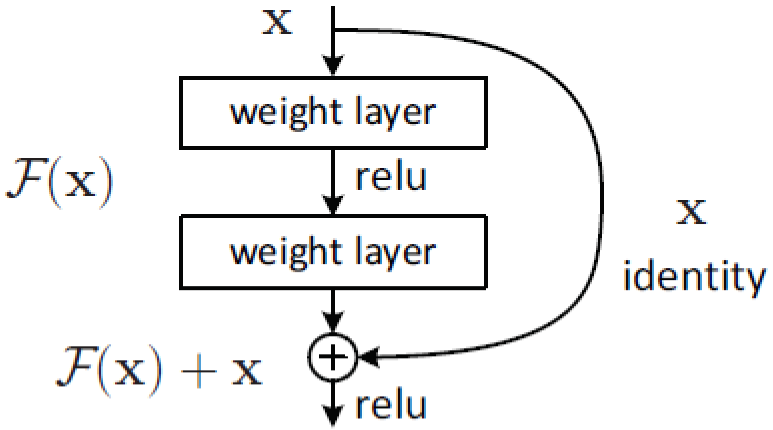
The problem still is the training!

Conclusion: The problem of training deep networks to perform well still remains!

Likely, it is still difficult to have meaningful gradients in deep layers.

To address this issue as well as to make it easier for the network not to use a layer if it is not needed, He et al. introduced the ResNet idea of certain skip connections in their paper "Deep Residual Learning for Image Recognition".

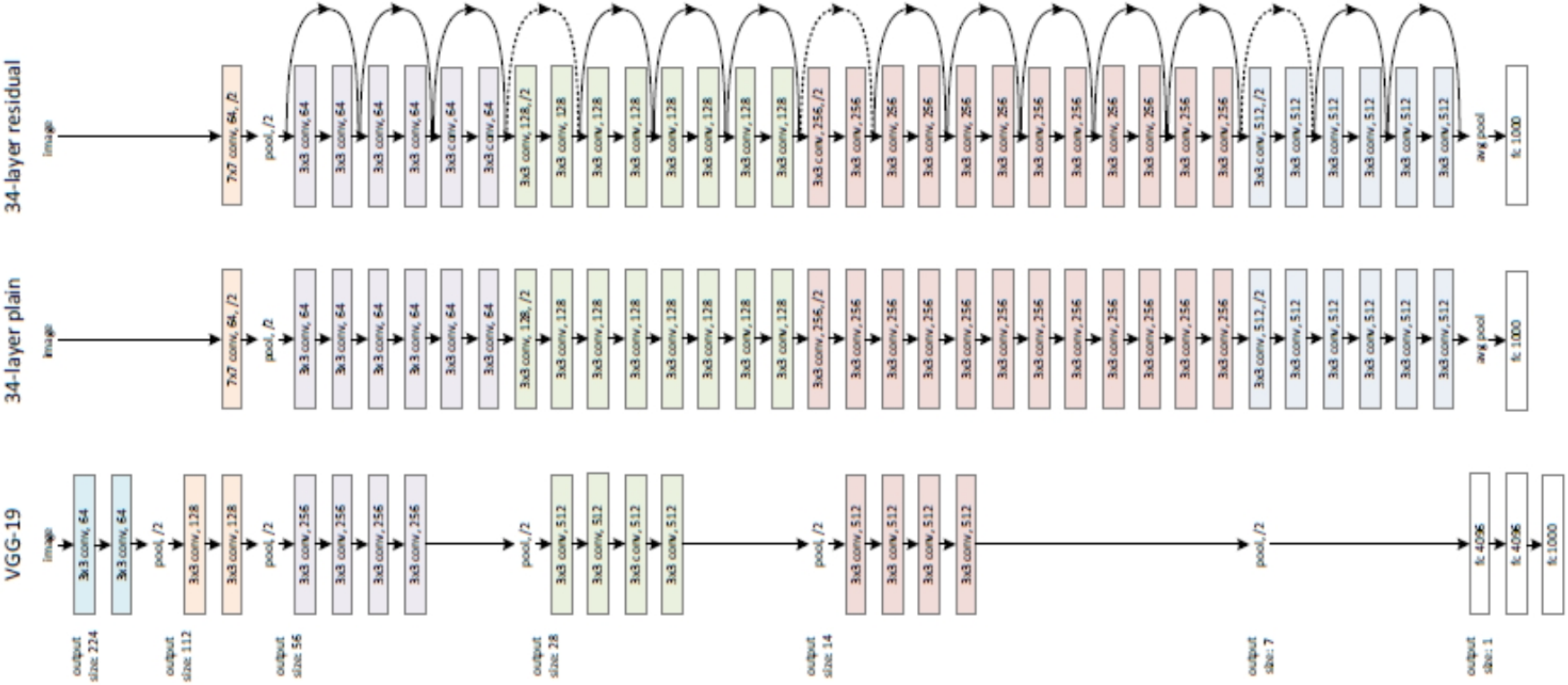
Assemble a network of building block like this:



1. The weights becoming zero leads to this block being identical. This seems to be easier for the network than learning the identity.
2. The full gradient is backpropagated and just modified by the nonlinear function.

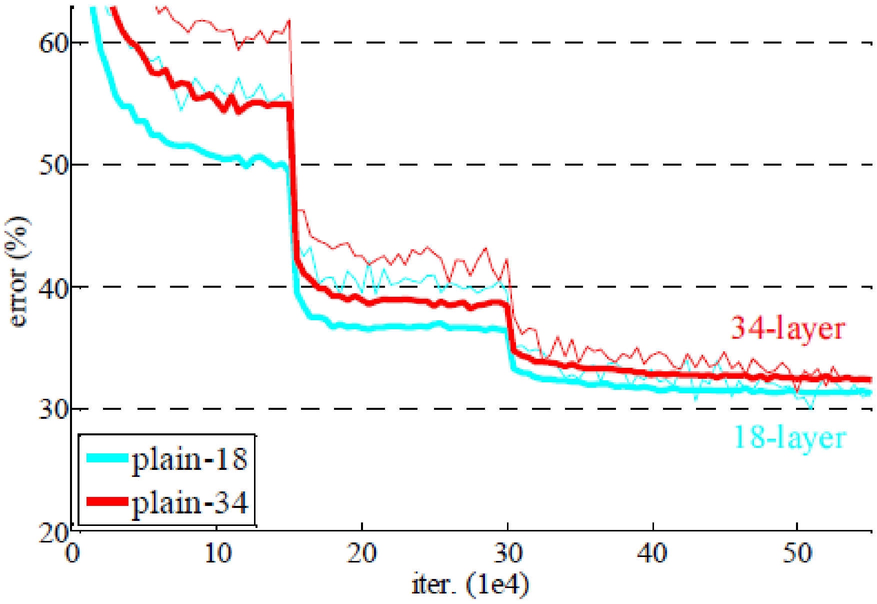
From He et al. "Deep Residual Learning for Image Recognition"

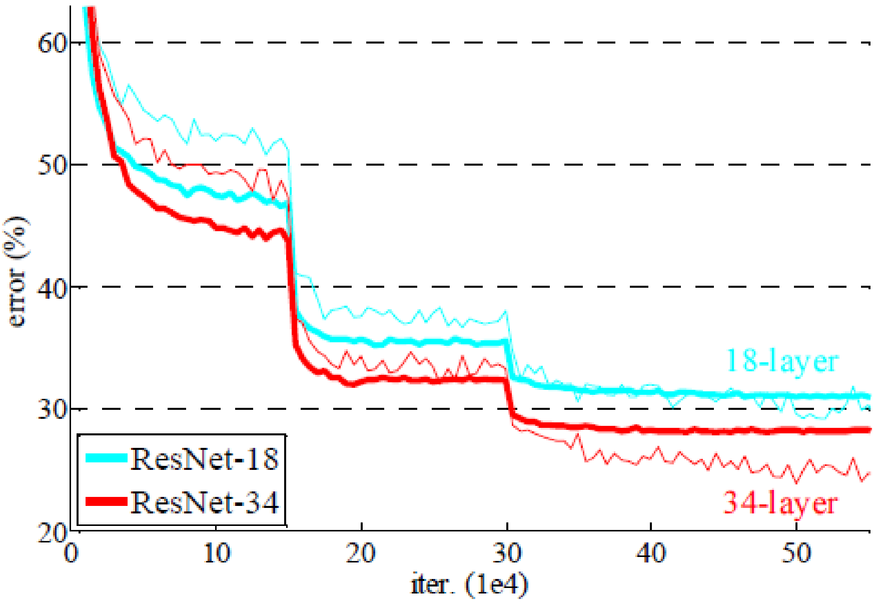
ResNet

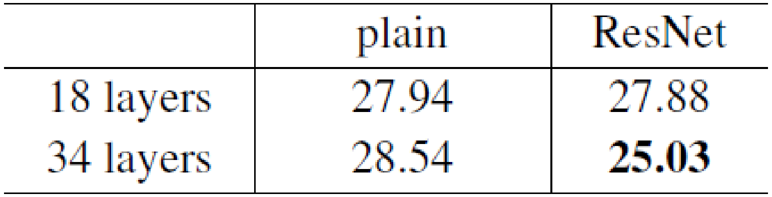


- Some fine tuning, e.g. one 7x7 convolution to start with. Otherwise only 3x3 convs.
- Only one fully connected layer (or 1x1 convolution) at the end.
- Shortcuts between layers of different sizes are either zero-padded or handled with an additional 1x1 convolution, both with stride 2 to reduce the spatial extend.

ResNet







	plain	ResNet
18 layers	27.94	27.88
34 layers	28.54	**25.03**

Interestingly, the difficulty to train the plain net is not due to vanishing gradients. It might be due to slow convergence, but even 3x more iterations did not equalize the accuracies.

ResNet

Details on the training procedure

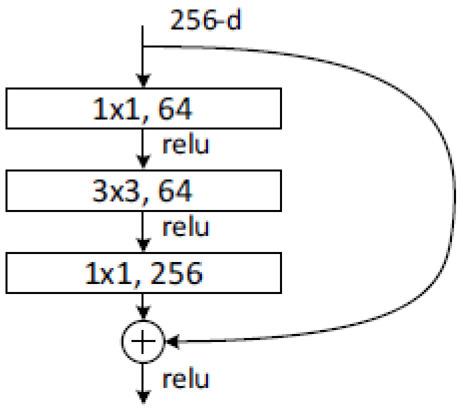
- The image is resized with its shorter side randomly sampled in [256; 480] for scale augmentation [41].
- A 224x224 crop is randomly sampled from an image or its horizontal flip, with the per-pixel mean subtracted [21].
- The standard color augmentation in [21] is used.
- We adopt batch normalization (BN) [16] right after each convolution and before activation, following [16].
- We initialize the weights as in [13] and train all plain/residual nets from scratch.
- We use SGD with a mini-batch size of 256. We use a weight decay of 0.0001 and a momentum of 0.9.
- The learning rate starts from 0.1 and is divided by 10 when the error plateaus, and the models are trained for up to 6x1e5 iterations.

Details on the test procedure

- In testing, for comparison studies we adopt the standard 10-crop testing [21].
- For best results, we adopt the fully convolutional form as in [41, 13], and average the scores at multiple scales (images are resized such that the shorter side is in {224; 256; 384; 480; 640}).

ResNet

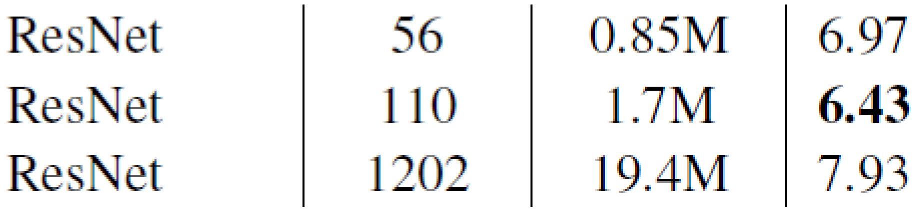
More efficient version for large feature maps: Bottleneck building blocks



This way a 152-layer resnet still has a lower computational complexity than the VGG-16 net!

ResNet won the imagenet challenge 2015 with 3.57% top-5 error on the test set.

Interesting study on CIFAR-10: At some point going deeper stopped improving the results, even with ResNets – likely due to overfitting (although both have similar training accuracy).

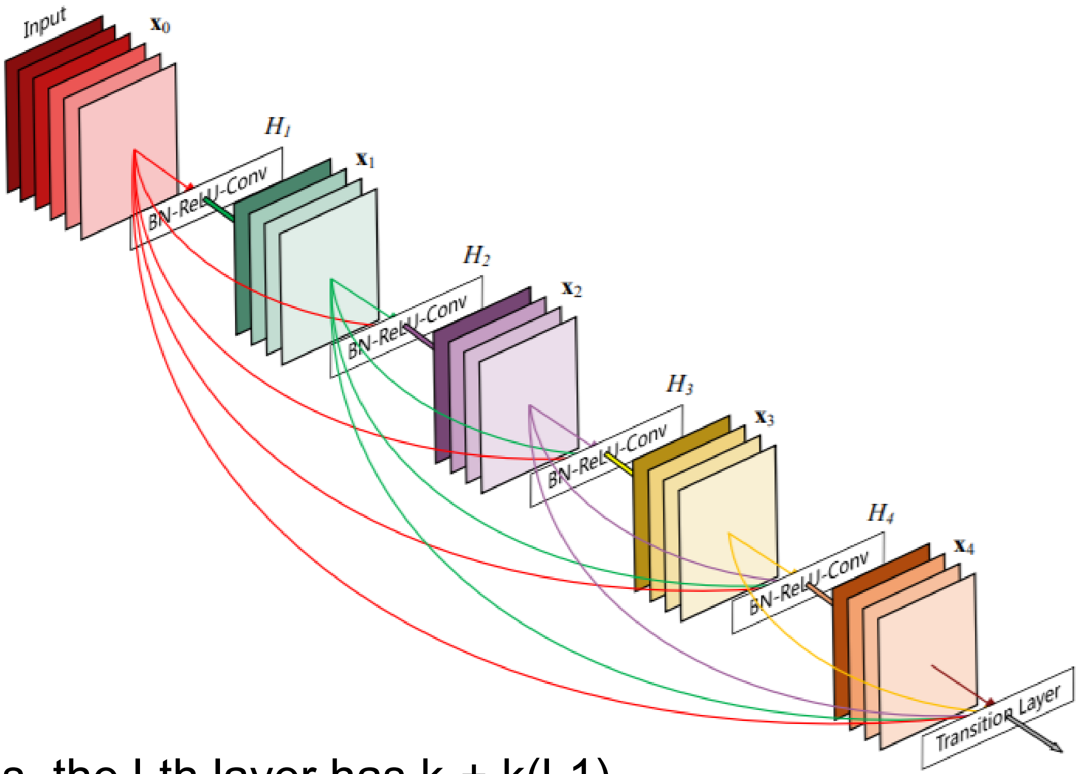


ResNet	56	0.85M	6.97
ResNet	110	1.7M	**6.43**
ResNet	1202	19.4M	7.93

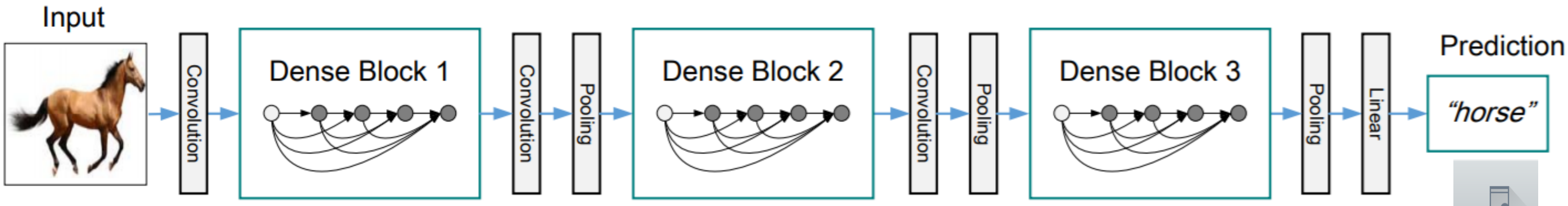
Conclusion: For deep nets, introduce shortcuts!

DenseNet

Interesting extension from Huang et al. “Densely Connected Convolutional Networks“ (DenseNets): For each layer, append the activations of all previous layer.

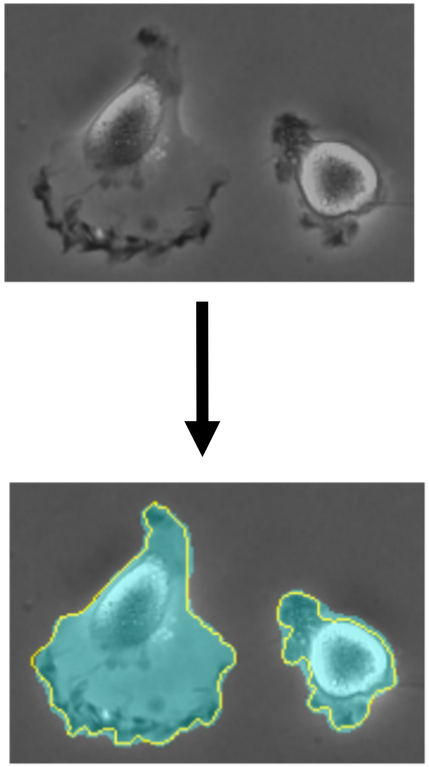


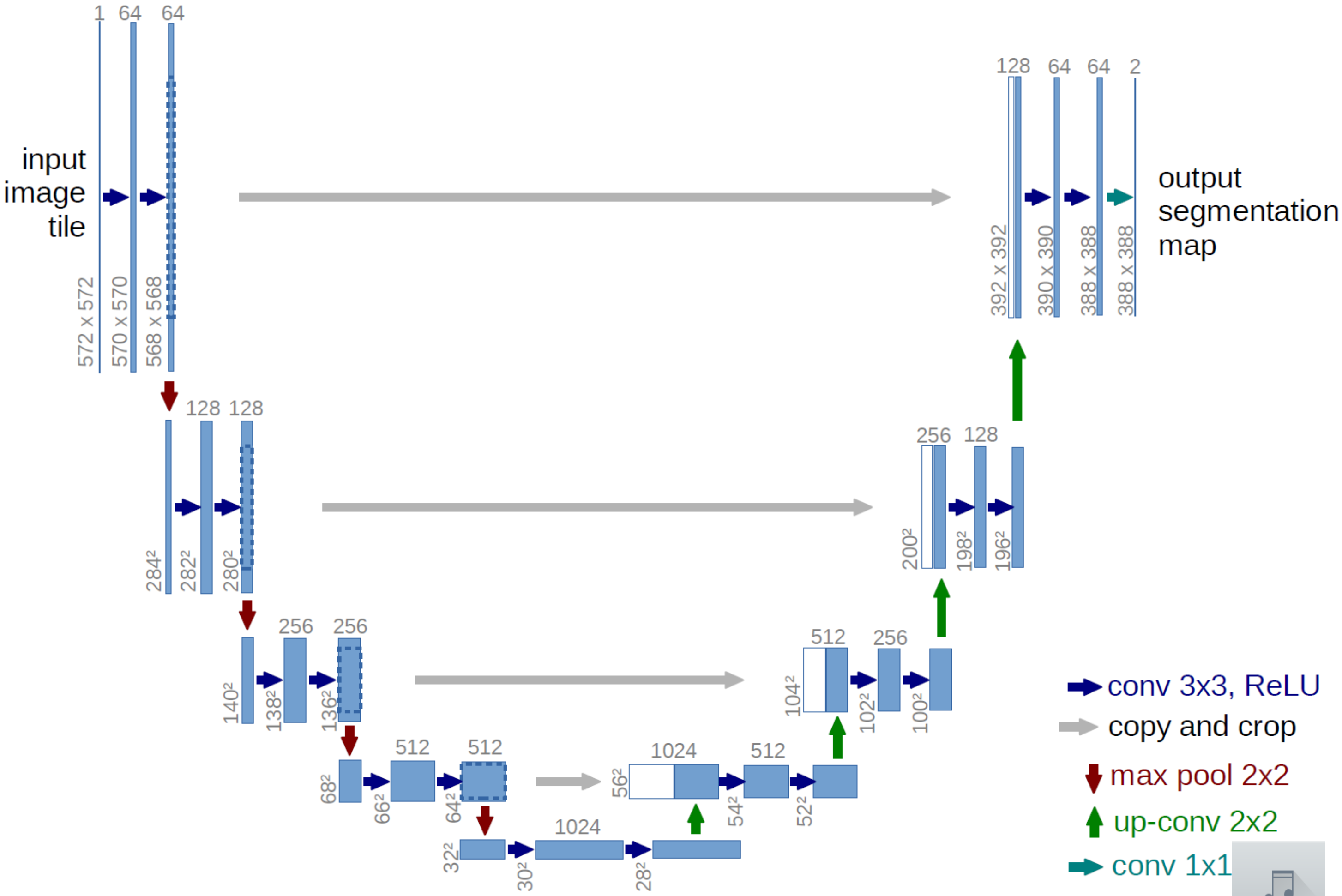
Starting with k_0 many feature maps, the l-th layer has k_0+ k(l-1) many feature maps. The parameter k is called groth-rate.



U-Net

Related publication in the area of image segementation: Ronneberger, Fischer, Brox, „U-Net: Convolutional Networks for Biomedical Image Segmentation", 2015.




1
64
64
input image tile
572 x 572
570 x 570
568 x 568
128
128
284^2
282^2
280^2
256
256
140^2
138^2
136^2
512
512
68^2
66^2
64^2
1024
32^2
30^2
28^2
1024
512
56^2
54^2
52^2
512
256
104^2
102^2
100^2
256
128
200^2
198^2
196^2
128
64
64
2
392 x 392
390 x 390
388 x 388
388 x 388
output segmentation map
conv 3x3, ReLU
copy and crop
max pool 2x2
up-conv 2x2
conv 1x1