U-Net: Convolutional Networks for Biomedical Image Segmentation

Olaf Ronneberger, Philipp Fischer,
and Thomas Brox

INDEX

HYUNHP

INTRODUCTION

What is U-NET

Abstract. There is large consent that successful training of deep networks requires many thousand annotated training samples. In this paper, we present a network and training strategy that relies on the strong use of data augmentation to use the available annotated samples more efficiently. The architecture consists of a contracting path to capture context and a symmetric expanding path that enables precise localization. We show that such a network can be trained end-to-end from very few images and outperforms the prior best method (a sliding-window convolutional network) on the ISBI challenge for segmentation of neuronal structures in electron microscopic stacks. Using the same network trained on transmitted light microscopy images (phase contrast and DIC) we won the ISBI cell tracking challenge 2015 in these categories by a large margin. Moreover, the network is fast. Segmentation of a 512x512 image takes less than a second on a recent GPU. The full implementation (based on Caffe) and the trained networks are available at http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net.

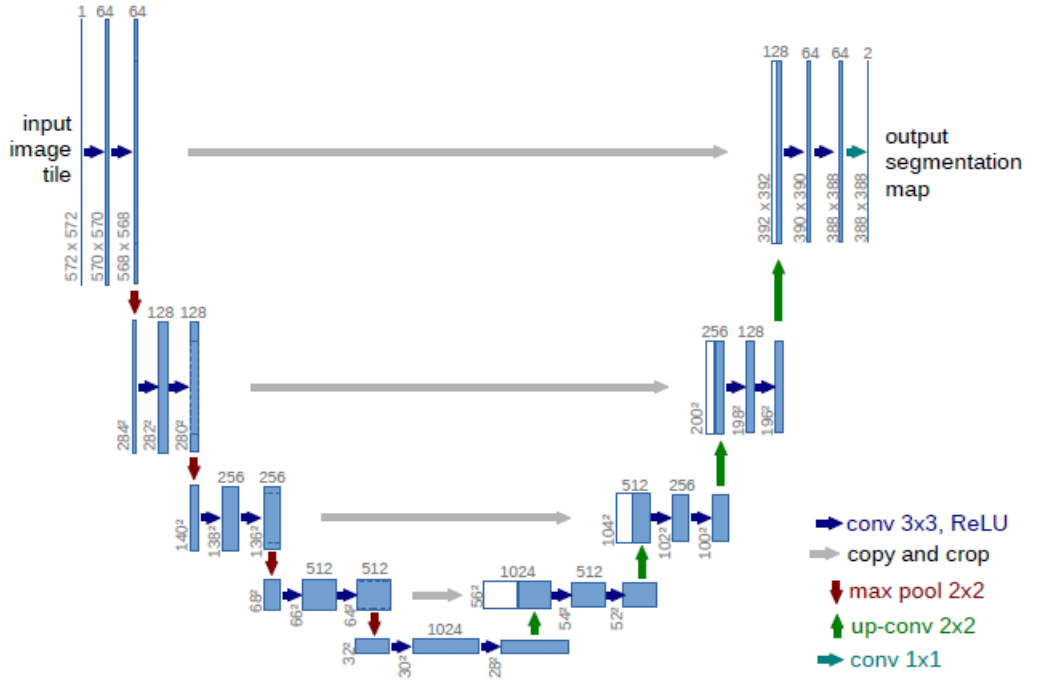


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows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

ARCHITECTURE

U-Net Architecture

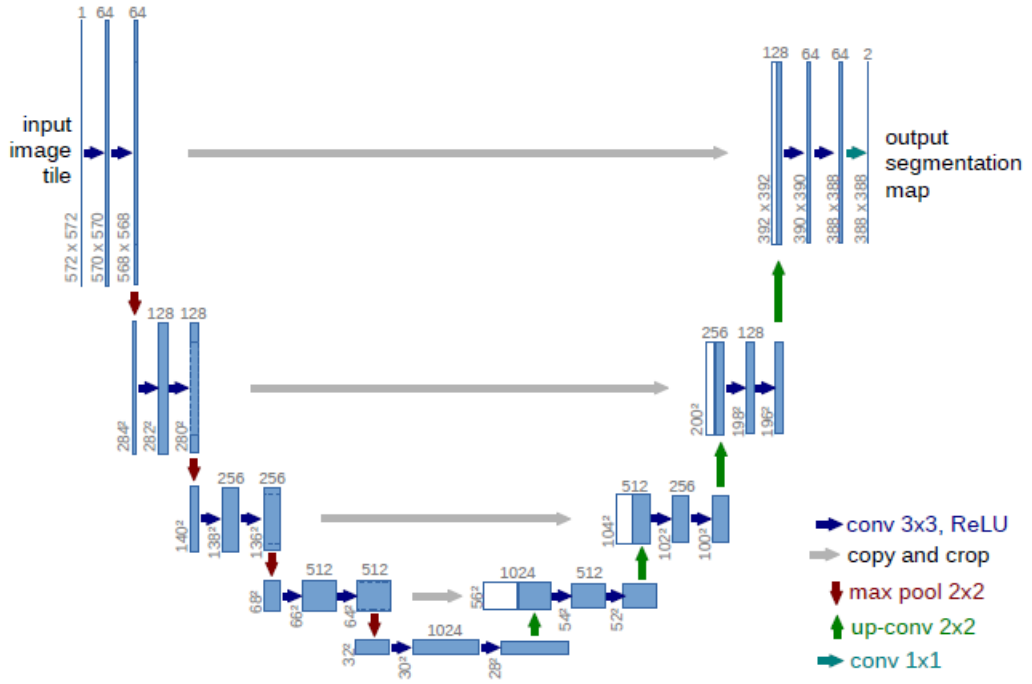


Fig. 1. U-net architecture (example for 32x32 pixels in the lowest resolution). Each blue box corresponds to a multi-channel feature map. The number of channels is denoted on top of the box. The x-y-size is provided at the lower left edge of the box. White boxes represent copied feature maps. The arrows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

U-Net consists of 3 parts

1. ***Contracting Path (Left side)***
   - Typical architecture of a CNN
2. ***Bottleneck (Lowest part)***
   - Path from contracting to expansive
3. ***Expansive Path (Right side)***
   - After every 2x2 convolution (up-convolution) upsampling
   - Concatenation with the correspondingly cropped feature map from the contracting path

Detailed Network Architecture

The network architecture is illustrated in Figure 1. It consists of a contracting path (left side) and an expansive path (right side). The contracting path follows the typical architecture of a convolutional network. It consists of the repeated application of two 3x3 convolutions (unpadded convolutions), each followed by a rectified linear unit (ReLU) and a 2x2 max pooling operation with stride 2 for downsampling. At each downsampling step we double the number of feature channels. Every step in the expansive path consists of an upsampling of the feature map followed by a 2x2 convolution ("up-convolution") that halves the number of feature channels, a concatenation with the correspondingly cropped feature map from the contracting path, and two 3x3 convolutions, each followed by a ReLU. The cropping is necessary due to the loss of border pixels in every convolution. At the final layer a 1x1 convolution is used to map each 64-component feature vector to the desired number of classes. In total the network has 23 convolutional layers.

To allow a seamless tiling of the output segmentation map (see Figure 2), it is important to select the input tile size such that all 2x2 max-pooling operations are applied to a layer with an even x- and y-size.

Unpadded convolutions

The input images and their corresponding segmentation maps are used to train the network with the stochastic gradient descent implementation of Caffe [6]. Due to the unpadded convolutions, the output image is smaller than the input by a constant border width. To minimize the overhead and make maximum use of the GPU memory, we favor large input tiles over a large batch size and hence reduce the batch to a single image. Accordingly we use a high momentum (0.99) such that a large number of the previously seen training samples determine the update in the current optimization step.

Trade-off between model accuracy and training efficiency,
paper suggested of using unpadded convolutions and reduce the batch to a single image.

Is it possible or okay to use padding in U-Net and when we use, is it efficiency?

Appendix. Unpadding

In the U-Net architecture, instead of using padding, the paper chose to use a series of convolutional layers with a stride of 2 to reduce the spatial dimensions of the feature maps. This results in a downsampled feature map that captures a coarse representation of the input image.

To compensate for the loss of information due to the downsampling, the paper introduced a series of upsampling layers that are **combined with the corresponding feature maps from the downsampled path**. This allows the model to **recover the spatial resolution and fine-grained details of the input image.**

In this way, the U-Net architecture balances **the trade-off between computational efficiency and the preservation of spatial information and enables accurate segmentation of the input images.**

Appendix. What if we use padding?

Padding can help to maintain the spatial dimensions of the feature maps and prevent information loss at the edges of the image, which can be important for accurate segmentation. By using padding, the model can effectively capture more context around each pixel, which can help to improve the accuracy of the segmentation predictions.

However, it's also worth considering the **computational cost of using padding**. Padding can increase the memory requirements of the model and the computation time for each forward and backward pass, which can have a significant impact on the overall training time. Therefore, **it is important to carefully consider the trade-off between accuracy and computational cost when using padding in the U-Net architecture.**

If computation power is not an issue, using padding in U-Net may indeed lead to improved accuracy in image segmentation, as the model will have access to more information about the input image. However, if computation resources are limited, it may be more efficient to use the original U-Net architecture, which balances the trade-off between computational efficiency and accuracy.

TRAINING

Loss Function

The energy function is computed by a pixel-wise soft-max over the final feature map combined with the cross entropy loss function. The soft-max is defined as $p_k(\mathbf{x}) = \exp(a_k(\mathbf{x}))/\left(\sum_{k'=1}^{K}\exp(a_{k'}(\mathbf{x}))\right)$ where $a_k(\mathbf{x})$ denotes the activation in feature channel k at the pixel position $\mathbf{x} \in \Omega$ with $\Omega \subset \mathbb{Z}^2$. K is the number of classes and $p_k(\mathbf{x})$ is the approximated maximum-function. I.e. $p_k(\mathbf{x}) \approx 1$ for the k that has the maximum activation $a_k(\mathbf{x})$ and $p_k(\mathbf{x}) \approx 0$ for all other k. The cross entropy then penalizes at each position the deviation of $p_{\ell(\mathbf{x})}(\mathbf{x})$ from 1 using

$$E = \sum_{\mathbf{x}\in\Omega} w(\mathbf{x})\log(p_{\ell(\mathbf{x})}(\mathbf{x})) \tag{1}$$

Weight Map

We pre-compute the weight map for each ground truth segmentation to compensate the different frequency of pixels from a certain class in the training data set, and to force the network to learn the small separation borders that we introduce between touching cells (See Figure 3c and d).

The separation border is computed using morphological operations. The weight map is then computed as

$$w(\mathbf{x}) = w_c(\mathbf{x}) + w_0 \cdot \exp\left(-\frac{(d_1(\mathbf{x}) + d_2(\mathbf{x}))^2}{2\sigma^2}\right) \tag{2}$$

Model Performance

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

IOU stands for Intersection Over Union, and it is a metric used to evaluate the accuracy of object detection algorithms. The IOU is calculated as the ratio of the area of overlap between the predicted bounding box and the ground truth bounding box to the total area of the union of both boxes.

The IOU value is between 0 and 1, where 1 means that the predicted bounding box perfectly matches the ground truth box, and 0 means that the predicted box and the ground truth box have no area of overlap.

The IOU is a commonly used performance measure in object detection tasks, especially in the context of computer vision and image processing.

Data Augmentation

Overlap-tile strategy

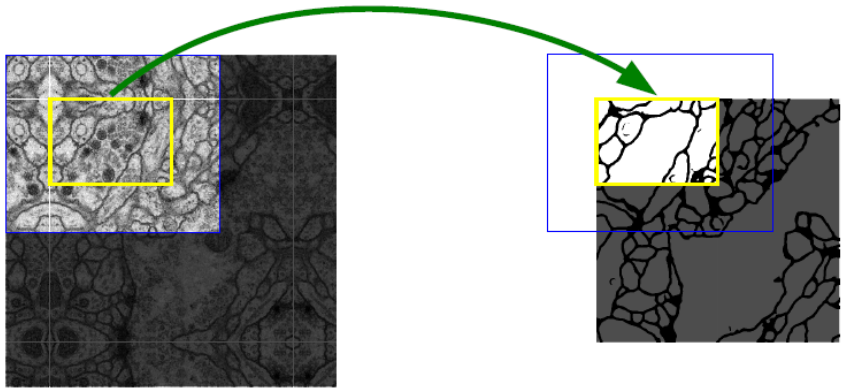
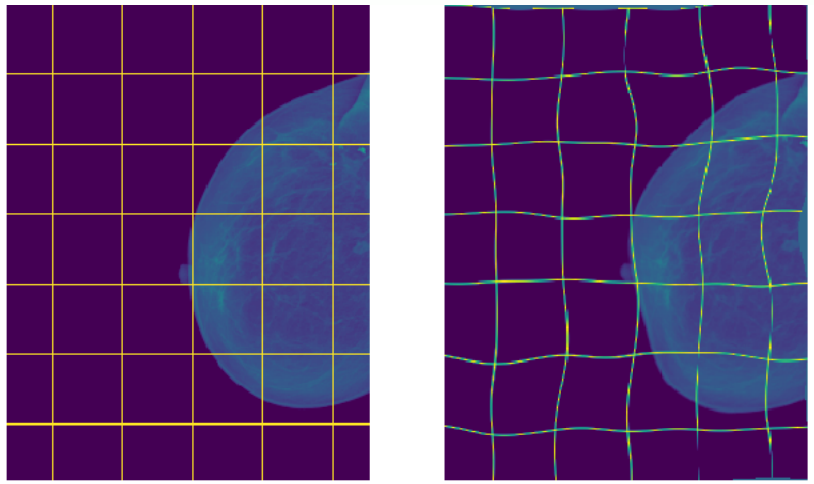


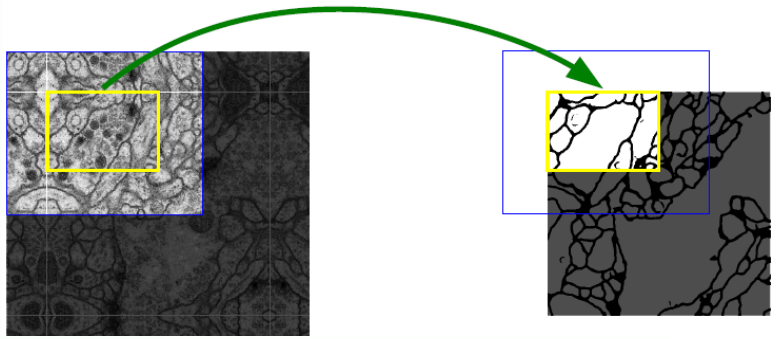
Fig. 2. Overlap-tile strategy for seamless segmentation of arbitrary large images (here segmentation of neuronal structures in EM stacks). Prediction of the segmentation in the yellow area, requires image data within the blue area as input. Missing input data is extrapolated by mirroring

Elastic augmentation



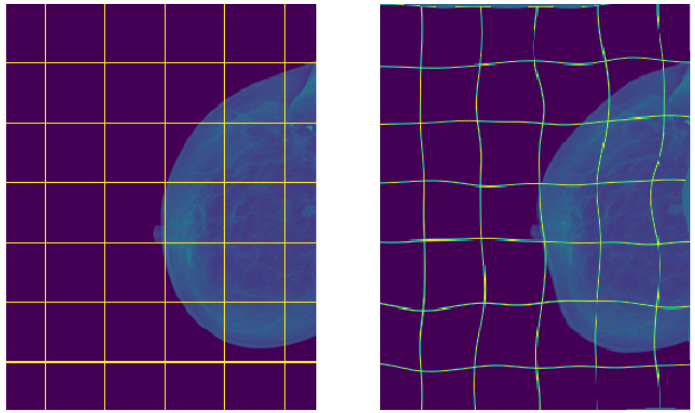
(a) Original (b) Deformed

Data Augmentation (1) : Overlap-tile strategy



One important modification in our architecture is that in the upsampling part we have also a large number of feature channels, which allow the network to propagate context information to higher resolution layers. As a consequence, the expansive path is more or less symmetric to the contracting path, and yields a u-shaped architecture. The network does not have any fully connected layers and only uses the valid part of each convolution, i.e., the segmentation map only contains the pixels, for which the full context is available in the input image. This strategy allows the seamless segmentation of arbitrarily large images by an overlap-tile strategy (see Figure 2). To predict the pixels in the border region of the image, the missing context is extrapolated by mirroring the input image. This tiling strategy is important to apply the network to large images, since otherwise the resolution would be limited by the GPU memory.

Data Augmentation(2) : Elastic deformation



(a) Original (b) Deformed

Elastic deformation is a type of deformation that occurs in materials when they are subjected to stress or strain. When a material is subjected to stress, it experiences a **change in shape or size that is proportional to the applied stress. However, once the stress is removed, the material returns to its original shape**. This property of materials is known as elasticity, and it is a result of the restoring force that acts on the material when it is subjected to stress.

In the context of the U-Net article, elastic deformation mentioned in reference to the deformation of images or objects within images. For example, U-Net is a type of deep learning architecture that is used for image segmentation tasks. When images are transformed or warped, the model may need to account for elastic deformations in order to maintain accurate segmentations.

Why deformation is useful in the image segmentation field?

One of the main reasons why elastic deformation can be useful in supervised learning is because of the expensive labeling cost. In many applications, obtaining annotated data for training deep learning models can be a time-consuming and labor-intensive process. Labeling data requires a human annotator to manually segment each image and label the different objects within it. This process can be particularly challenging for medical imaging, where a high level of accuracy is required, and the data can be complex and multi-modal.

By using elastic deformation, it is possible to generate additional annotated data from a limited set of annotated images. The idea is to apply small random deformations to the original images, which are then used to generate new annotated images that can be used to train the model. This can help to overcome the problem of limited annotated data, as well as improve the robustness of the model to small variations in the input data.

In other words, elastic deformation can be seen as a data augmentation technique that can help to improve the generalization performance of the model. By training the model on a diverse set of deformations, it can learn to handle a range of variations in the input data and better generalize to unseen data.

Appendix. Normalization

It's worth noting that both min-max scaling and Z-score normalization have their advantages and disadvantages, and the choice of the normalization method depends on the specific problem and the nature of the data. For example, min-max scaling is often used for image classification problems, while Z-score normalization is used for image segmentation and object detection problems.

In conclusion, image normalization is an important step in preparing images for use with deep learning models, as it helps to standardize the range and distribution of the pixel intensity values. This makes the input data more consistent and helps to improve the performance of the deep learning models.

EXPERIMENT

Ranking on the EM segmentation challenge 2015

Table 1. Ranking on the EM segmentation challenge [14] (march 6th, 2015), sorted by warping error.

Rank	Group name	Warping Error	Rand Error	Pixel Error
	** human values **	0.000005	0.0021	0.0010
1.	u-net	**0.000353**	0.0382	0.0611
2.	DIVE-SCI	0.000355	0.0305	0.0584
3.	IDSIA [1]	0.000420	0.0504	0.0613
4.	DIVE	0.000430	0.0545	**0.0582**
⋮				
10.	IDSIA-SCI	0.000653	**0.0189**	0.1027

Segmentation results (IOU) on the ISBI cell tracking challenge 2015

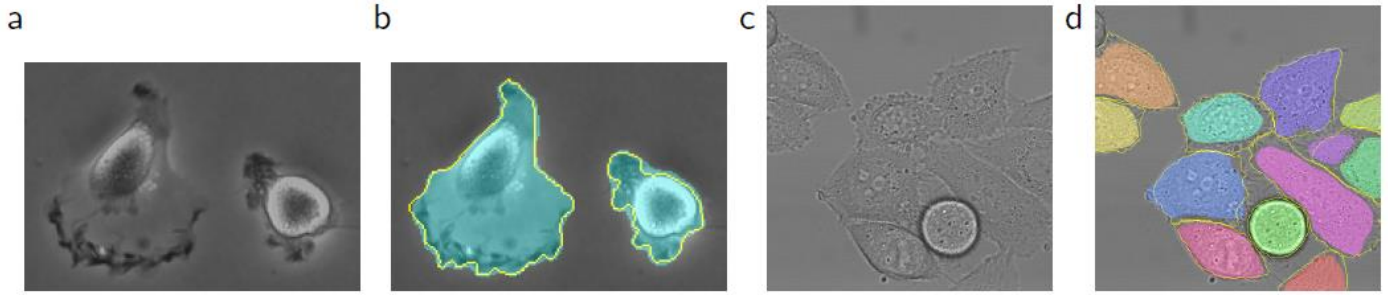


Fig. 4. Result on the ISBI cell tracking challenge. (**a**) part of an input image of the "PhC-U373" data set. (**b**) Segmentation result (cyan mask) with manual ground truth (yellow border) (**c**) input image of the "DIC-HeLa" data set. (**d**) Segmentation result (random colored masks) with manual ground truth (yellow border).

Table 2. Segmentation results (IOU) on the ISBI cell tracking challenge 2015.

Name	PhC-U373	DIC-HeLa
IMCB-SG (2014)	0.2669	0.2935
KTH-SE (2014)	0.7953	0.4607
HOUS-US (2014)	0.5323	-
second-best 2015	0.83	0.46
u-net (2015)	**0.9203**	**0.7756**

APPENDIX

CLASSIC BLUE

HYUNHP

RELATED U-NET PAPERS

1. U-Net: Convolutional Networks for Biomedical Image Segmentation (2015) by Olaf Ronneberger, Philipp Fischer, and Thomas Brox.

2. Image Segmentation Using U-Net and Segmentation-Aware Transfer Learning (2019) by Shuyue Liu, Jianmin Li, Jianfeng Liu, and Jianyong Wang.

3. High-Resolution U-Net for Medical Image Segmentation (2018) by N. K. Dhungel and N. Zhang.

4. Spatial U-Net with Attention Mechanism for Multi-Organ Segmentation in CT Volumes (2019) by Weijia Chen, Wei Cheng, and Hongliang Li.

5. Deep U-Net for Lesion Segmentation in Brain Magnetic Resonance Images (2017) by F. Xia and L. Yin.

6. A Deep Learning Framework for Lung Segmentation in Computed Tomography Images Using U-Net (2019) by Q. Wang and X. Liu.

7. Low-Dose CT Image Denoising with a Deep U-Net (2020) by B. Li and C. Wei.