Smart image denoising with noise2void

Alexander Krull, Tim-Oliver Buchholz, Florian Jug. *Noise2Void - Learning Denoising from Single Noisy Images*. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019.

Alexander Krull, Tim-Oliver Buchholz, Florian Jug. *https://arxiv.org/abs/1811.10980*. GitHub repository.

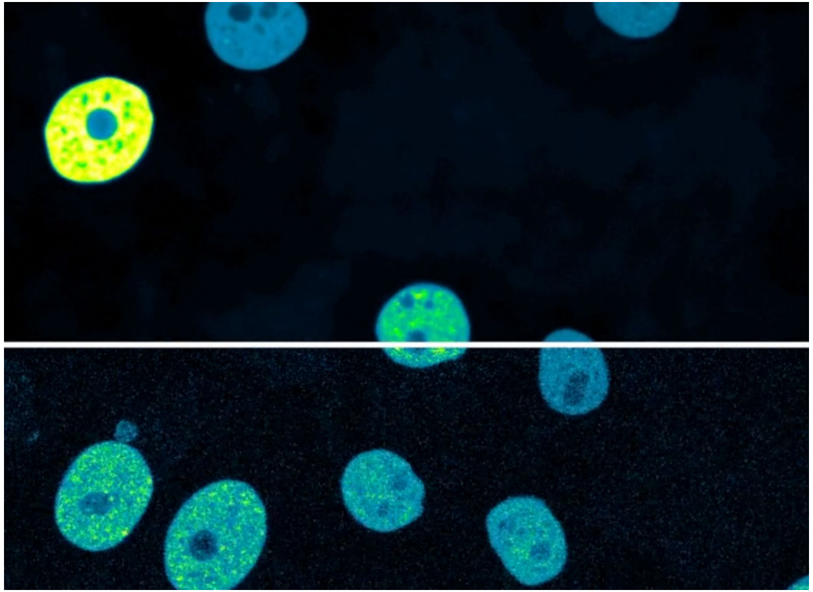
Motiation

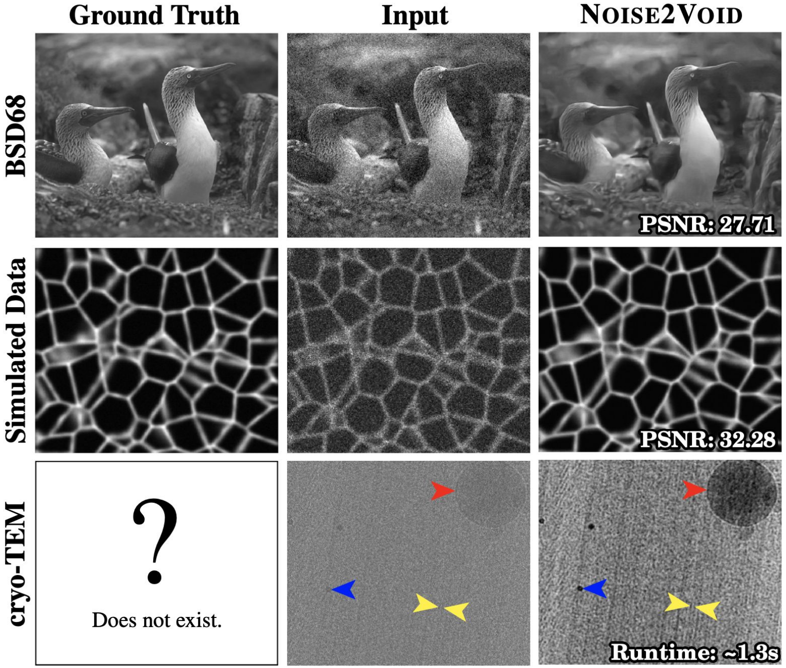
- Smart and universal denoising of various image types for visualization.
- Ready-to-use tool in ImageJ, open source code on GitHub
- The original article also described the limitations of the method.
- Availability of test data from the article.

- Deep Learning model (black box)
  - with very well described dependencies
  - focused on re-usability
- It has the potential to use the resulting data for the quantization of brightness values, but has not yet been tested.

Noise2Void (N2V)

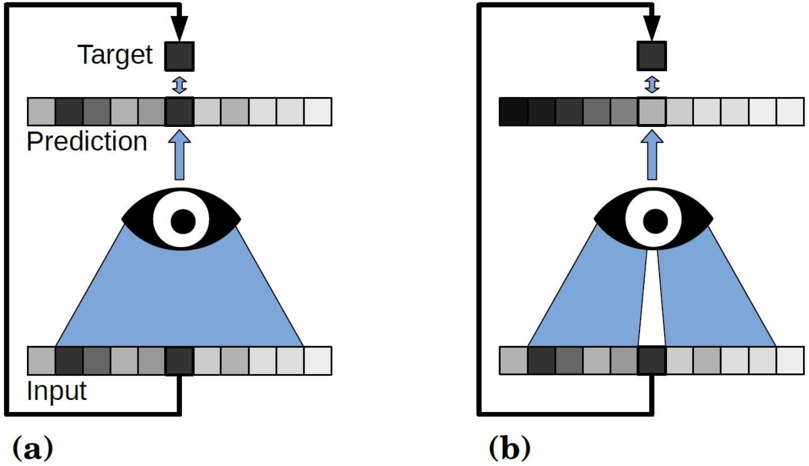
Machine learning model for denoising. Designed based on the search for statistical dependencies in the image.





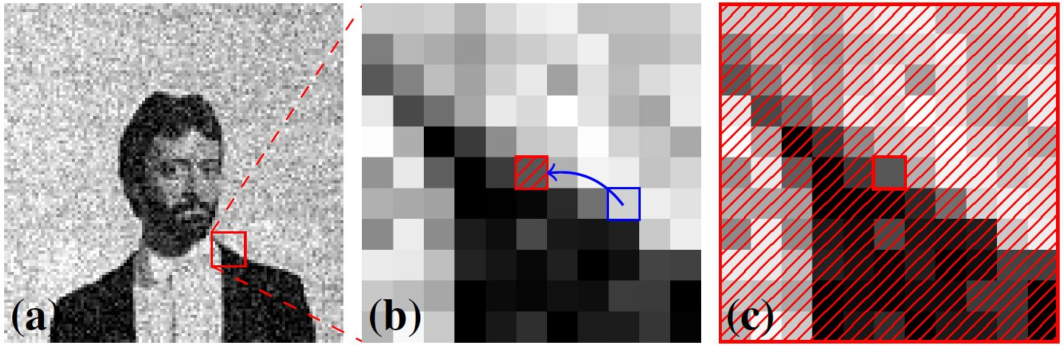
Implementation of Noise2Void

- Assumption: noise changes but image information remains.
  - The signal is statistically dependent on itself in the image.
  - Noise is statistically independent of the signal.
- Creating blind spots in the training image **(b)** will help the model recognize dependencies in the image.



Supervised Learning

A classical learning approach: the input (an image with noise) is compared against a target (ground through)



N2V innovation: input **(a)** is divided into subsections **(b)**. In training part b, a random pixel is copied to the center. Target **(c)** has no center pixel.

Using the N2V model

It is necessary to take into account:

- DL model is a "black box"
- It will only be as good as the data we use to train it (“copying the teacher”)

The following must be observed:

- Apply to same data as training (bit depth, luminance distribution)
- To share the model, it is good to keep the training data + description.

Practical use:

- Visualization, not the basis for segmentation. Not yet usable for brightness quantification!

Limitations

Structured noise

N2V assumes that the noise has no statistical dependence, and therefore does not remove structured noise (checker/mosquito noise, bad frequency filter).

Seemingly easy to use

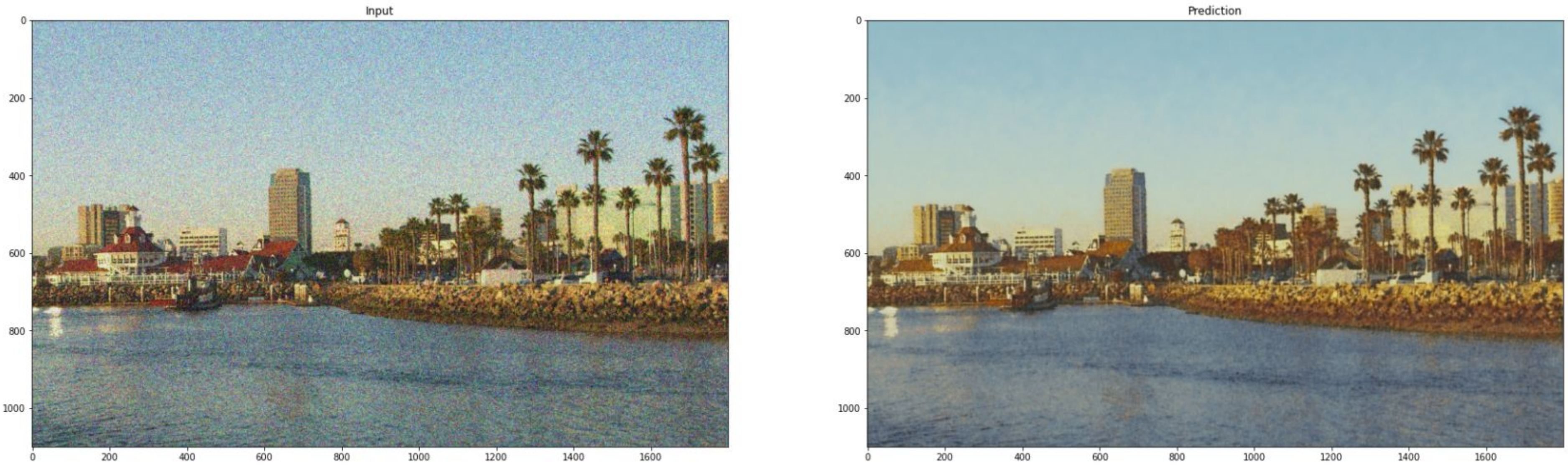
DL methods are relatively new, very effective, but there is still no standardized description of their use (mainly for scientific data). It is very easy to misuse them.

Ethical rules of image analysis

Due to the complexity of Deep Learning, it is necessary to be careful for now.

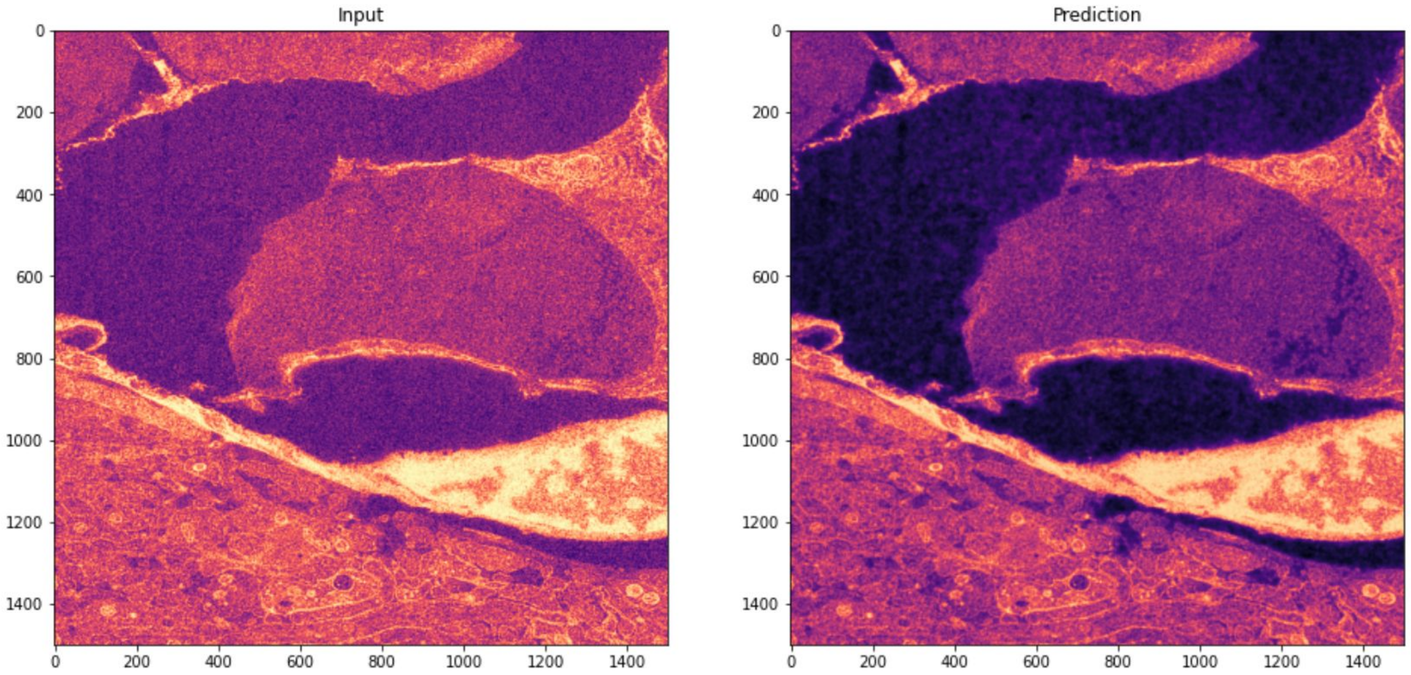
Python - Jupyter notebook

https://github.com/juglab/n2v/blob/master/examples/ 2D RGB



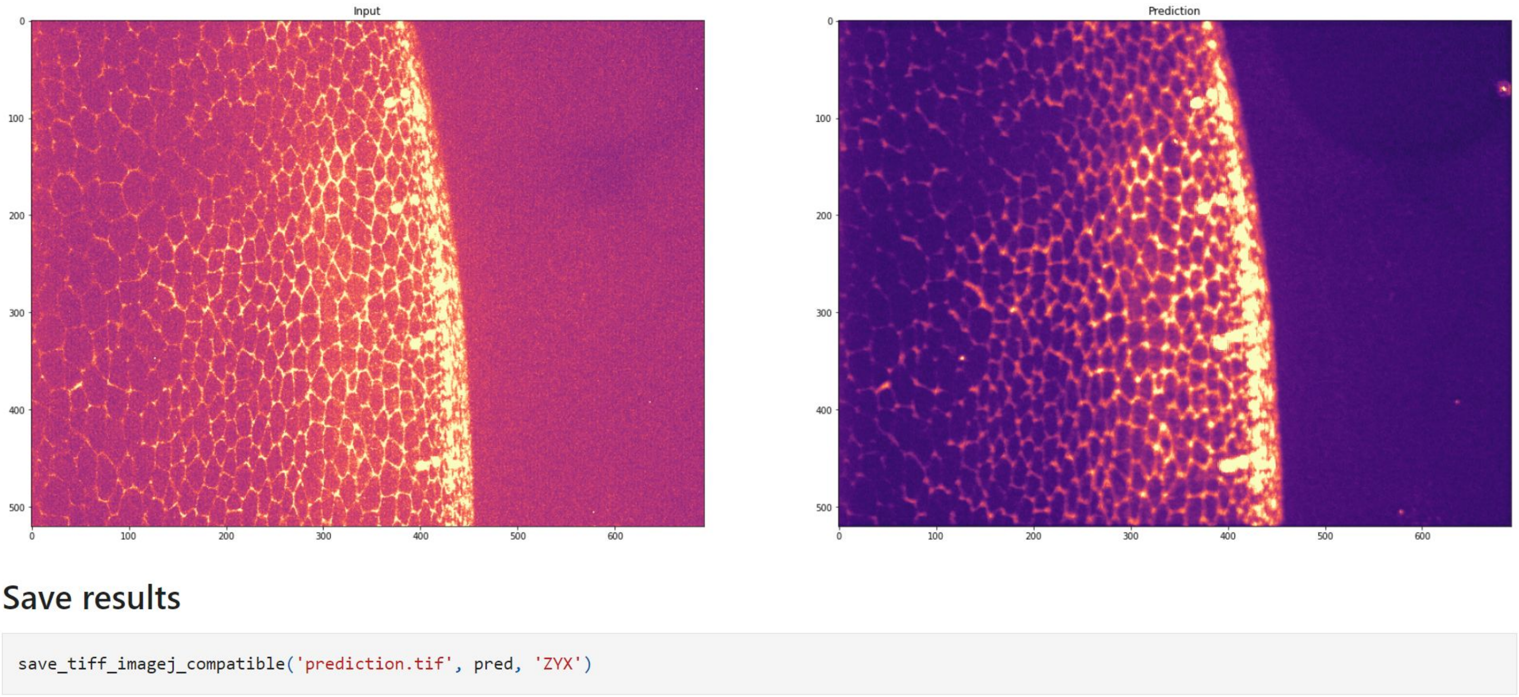
Python - Jupyter notebook

https://github.com/juglab/n2v/blob/master/examples/ 2D SEM



Python - Jupyter notebook

https://github.com/juglab/n2v/blob/master/examples/ 3D



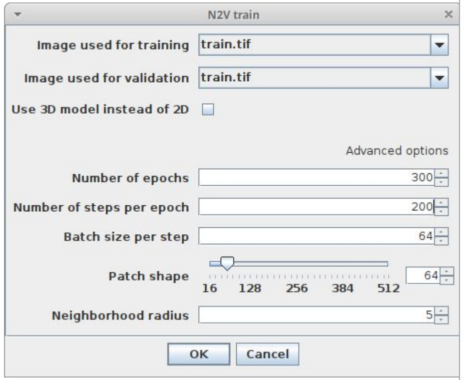
Save results

```
save_tiff_imagej_compatible('prediction.tif', pred, 'ZYX')
```

ImageJ plugin

N2V Fiji plugin - https://imagej.net/plugins/n2v

Dependent on the CSBDeep package



N2V train

Image used for training: train.tif

Image used for validation: train.tif

Use 3D model instead of 2D ☐

Advanced options

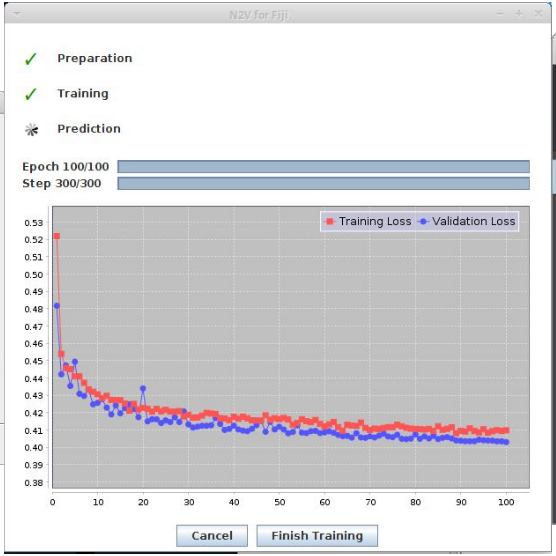
Number of epochs: 300

Number of steps per epoch: 200

Batch size per step: 64

Patch shape: 64

Neighborhood radius: 5



N2V for Fiji

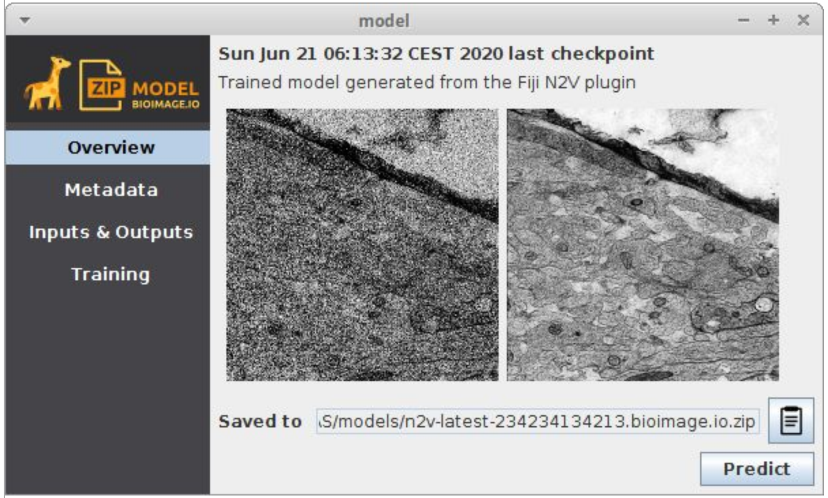
✓ Preparation

✓ Training

Prediction

Epoch 100/100

Step 300/300



model

Sun Jun 21 06:13:32 CEST 2020 last checkpoint

Trained model generated from the Fiji N2V plugin

Saved to .S/models/n2v-latest-23423413421 3.bioimage.io.zip

Summary

Ideal for:

- Visualization
- Preprocessing for segmentation
- Preprocessing for counting objects

Simple use through ImageJ - both training and application. Or use in other Python-enabled applications.

But:

- Black box
- May produce artifacts (statistically dependent noise)

Thanks for your attention!