Deep Image Prior for Microscopy Images

Caroline Magg

Computer Vision Lab
Institute of Visual Computing & Human-Centered Technology
TU Wien

June 16, 2021

Supervisor: Roman Pflugfelder

Abstract

Noise and low-resolution are problems that often occur in medical and biological imaging. Convolutional neural networks have shown expressive results for reconstruction tasks, such as denoising and super resolution. One of the drawback of traditional supervised deep learning approaches is the need for a large training dataset. In some domains, such as the medical or biological, large datasets are difficult to acquire and generating labels or data pairs is time- and resource-consuming. In addition, using a large training dataset can enhance the effect of hallucination, that is when features exist in a joint distribution, but not in the statistic of a single image. Developed for natural images, the Deep Image Prior method can overcome these limitations. A single image is used to train a network in an unsupervised way. Since the approach is based on the network architecture, different settings for denoising and super resolution are tested for single natural images and a microscopy dataset. The results show that the baseline settings from the original work can be outperformed if the model capacity is increased and other activation functions are used. It is shown, that the best setting for an image is image-specific not dataset-specific. This means that although a setting performs well on average on a dataset, single images might have different best settings.

1 Introduction

All modalities used in medical and biological imaging, such as X-ray, computed tomography (CT), magnetic resonance imaging (MRI), ultrasound (US) and optical coherence tomography (OCT), microscopy imaging, are affected by noise and speckles [13, 17]. Aside from the state-of-the-art non-learned denoising approaches [9, 4], convolution neural networks (CNNs) have proven their value [37]. One of the problems with common deep-learning based denoising methods is an effect called hallucinations [22]. Based on the provided training dataset and the therefore represented joint distribution, false image features looking like valid features can be generated in the input image. In natural images, the effect of hallucinations might be acceptable [34], but for medical images this could lead to false diagnoses or conclusions and should therefore be avoided [22]. In addition to that, for some domains, such as the medical and biological domain, it might be difficult to obtain a large dataset with data sample pairs for training a neural network.
Traditional deep learning approaches, such as for denoising, super resolution and inpainting, have a dedicated training phase using a large (labeled) dataset, and an independent inference phase, where predictions for new data are generated [16, 24]. In contrast to that, the Deep Image Prior (DIP) method by Ulyanov et al. [31] introduces an unsupervised way to solve inverse problems. Although, the authors' goal is not to beat other state-of-the art methods in the respective restoration task, the method shows powerful results considering that the network is initialized randomly and no pre-training is necessary. Similar to a generative network approach, the DIP network generates an image. It does so by using a random input vector and a corrupted version of a single image. Since the network has a low impedance to signal and a high impedance to noise, the low frequencies of a natural image are fitted faster than the high frequency of the noise [5].
The contributions of this work are:

- implementation of the DIP method for denoising and super resolution in Tensorflow [1].
- comparison of different network architecture settings, such as activation functions, feature map settings, down- and upsampling methods, input channels
- evaluation and experiments with a cell microscopy dataset

The work shows that:

1. For denoising, the results of the baseline setting can be improved and the optimal setting is the same for the natural image and the microscopy dataset. This behavior can not be translated to super resolution where the optimal setting for the natural image depends on the performance measure and the baseline setting is already the best setting for microscopy images.
2. The best architecture setting for an entire dataset is not necessarily the optimal setting for all individual images.
3. The inference time is quite long for a large dataset without parallelization.

The report is organized as follows. In Section 2, related work about Deep Image Prior and Deep Internal Learning are covered. The theoretical background, the application tasks, the workflow, and the model architecture of the DIP method

are introduced in Section 3. The data is described in Section 4. The performance measures and the experimental results are presented in Section 5 with an analysis and discussion of advantages and disadvantages of the method in Section 6. Finally, an outlook for future work is provided and a conclusion is drawn in Section 7.

2 Background

Deep Image Prior (DIP) was introduced by Ulyanov et al. [31] in 2018. Their work focuses on the a-priori knowledge that is introduced by the network architecture and how this fact is used to solve inverse tasks, such as denoising, super resolution and impainting. A single image is used during inference time to train a network. The image-specific trained model predicts the reconstruction with a random noise input vector. In the following paragraphs, further research on a better understanding, combinations with other methods and improvements are presented together with applications, focusing on the medical and biological domain.

Cheng et al. [7] provide a Bayesian perspective on DIP. To include Bayesian posterior inference in DIP, they use stochastic gradient Langevin dynamics (SGLD), a method based on stochastic gradient descent (SGD) that adds noise to the parameters at each gradient update step. The original DIP already uses random input perturbations at each training step. However, SGLD adds noise to all parameters, including the weights θ. The authors show that they achieve better results for natural image denoising and inpainting with SGDL compared to experiment setups with SGD. Furthermore, SGDL stabilizes the training over several 10k training iterations and eliminates the need for early stopping.
Laves et al. [22] show that the SGDL approach for natural images by Cheng et al. is not applicable for medical imaging modalities. They test a Bayesian approach with Monte Carlo dropout and full negative log-likelihood against SGDL and non-Bayesian DIP for optical coherence tomography (OCT), chest X-ray and ultrasound images. Their results show that the overfitting of SGDL is similar to non-Bayesian DIP. Furthermore, their approach does not only consider epistemic uncertainty, that is uncertainty about model parameters, but also aleatoric uncertainty, that is uncertainty in the data due to noise.
Another aspect of deep prior approaches have been investigated by Dittmer et al. [10]. They study different interpretations of DIP and focus more on the intrinsic regularizing properties. Along the analysis on a more mathematical level, numerical verifications are presented.

DIP-TV by Liu et al. [19] is a combination of DIP with Total Variation (TV). Anisotropic TV is used as additional regularization for the optimization problem and is added to the data term. This leads to piecewise smooth solutions, this are uniform regions in the resulting image.
Another method combination with DIP is introduced by Mataev et al. [21] - DeepRED incorporates Regularization by Denoising (RED) into DIP. RED is a framework where a regularization term is defined by means of a denoiser, which means a denoiser is turned into a regularization.

There is some research on improving the hand-designed SkipNet architecture of the original paper [31]. Chen et al. [6] use Neural Architecture Search (NAS) with a recurrent neural network (RNN) controller to find an optimal architecture within their designed search space. The search space of the NAS-DIP is designed to find an upsampling layer in the decoder and the cross-level feature connection from encoder to decoder. An improved performance compared to conventional U-Net design [26] is shown for denoising, super resolution, dehazing, image-to-image translation, and matrix factorization.
Segawa et al. [28, 27] have studied the activation function used in the SkipNet for super resolution and denoising. They propose a new activation function RSwish, which is based on Swish [25] but with a random slope for negative x values.

Deep Image Prior is already used for various medical imaging and reconstruction applications, such as in positron emission tomography (PET) images reconstruction [14, 35, 8, 15, 29], time-dependent dynamic magnetic resonance imaging (MRI) reconstruction [36], undersampled photoacoustic microscopy (PAM) [33], reconstruction of sparse microscopy images [38, 30], OCT image restoration [11], computer tomography (CT) reconstruction [3], and compressed sensing in CT and retinopathy images [32].
Gandelsmann et al. [12] introduce Double DIP, a unified framework for image decomposition. They combine multiple DIP networks, to decompose a natural image into its layers. Tasks, such as foreground/background segmentation, image dehazing, watermark-removal and transparency separation, are tested.

3 Deep Image Prior (DIP) method

In this chapter, the problem formulation and the learning principle of the DIP method are explained. Then, the options for the model architecture are reviewed and finally, the training is explained in a step-by-step manner.

3.1 Problem formulation

In general, a deep neural network can be interpreted as a parametrization $x = f_\theta(z)$ with an image $x \in R^{C \times H \times W}$, an input vector $z \in R^{C' \times H' \times W'}$ and the network f with weights θ. The channel number for RGB images is $C = 3$, for grayscale images $C = 1$. [31]

Ulyanov et al. [31] consider an image restoration problem to be an energy minimization problem:

$$x^* = min_x E(x; x_0) + R(x). \tag{1}$$

$E(x; x_0)$ is a data term depending on the considered inverse task, x_0 the corrupted image, and $R(x)$ a regularization term that should capture a general prior on natural images. The search for the optimal x^* is performed in image space. By using a mapping function $g(\theta) = x$, the optimization can be performed in another search space:

$$\theta^* = min_\theta E(g(\theta); x_0) + R(g(\theta)), x^* = g(\theta). \tag{2}$$

For the DIP method, we use the model parameter space as optimization space. The separate regularization term $R(x)$ is neglected and an implicit network prior $f(.)$ is introduced as implicit regularization which results in

$$\theta^* = min_\theta E(f_\theta(z); x_0), x^* = f_\theta^*(z). \tag{3}$$

Equation 3 is an optimization problem in θ which can be solved by using a gradient-based optimizer. The restored image is given by $x^* = f_\theta^*(z)$ with optimal parameter θ^*. The input z is a random vector filled with uniform noise.

The DIP method was originally applied to different image restoration problems, such as denoising and super resolution.
The data term for denoising is:

$$E(x; x_0) = ||x - x_0||^2. \tag{4}$$

For super resolution, the data term is:

$$E(x; x_0) = ||d(x) - x_0||^2 \tag{5}$$

with $d(x)$ as the downsampled version of the high-resolution image x and x_0 as the low-resolution image. The downsampling operator $d(.) : R^{C \times tH \times tW} \longrightarrow R^{C \times H \times W}$ resizes the image by a factor t.
In both cases, the data term is the MeanSquaredError (MSE) between the corrupted image and the (resized) original image.

3.2 Learning principle

The workflow used in the DIP method [31] is different from a traditional supervised deep learning approach and the self-supervised training in DIL (see Figure 1). The supervised framework (1a) has a clearly separated training and test (inference) phase. A network is pre-trained on a large dataset and is used at inference time to make predictions on unseen images. In comparison to the previous approaches, the DIP framework (1b) uses a random input vector to reconstruct the image given at test time. Since the network impedance to noise is high compared to signal, the parametrization fits the signal first [5]. Thus, using an appropriate number of iterations, the framework is used for restoration and inverse tasks.

(a) Traditional paradigm - supervised (b) DIP training paradigm - unsupervised

Figure 1: Different deep learning training approaches in comparison: supervised vs. unsupervised (DIP). Individual images are taken from [31].

3.3 Model architecture

The network architecture is a encoder-decoder network with skip connections, such as U-Net [26]. This network can come in many different flavors, as shown by the work of Ulyanov et al. [31]. Their architecture is called SkipNet and is shown in Figure 2. The network has an downsampling path d_i and an upsampling path u_i, where i represents the depth of the block. Each block consists of several layers, such as convolutional layer, batch normalization, activation function and down- or upsampling layer. The filter size of a block is represented by $n_x[i]$ and the kernel size of the convolutional layers is called $k_x[i]$, where x can stand for downsampling d, upsampling u or skip connection s.

For each image and for each task, a slightly different model architecture and number of epochs are used in the experiments. This indicates, that each image has its own training strategy that works best and no "fits all" solution exists in practice.

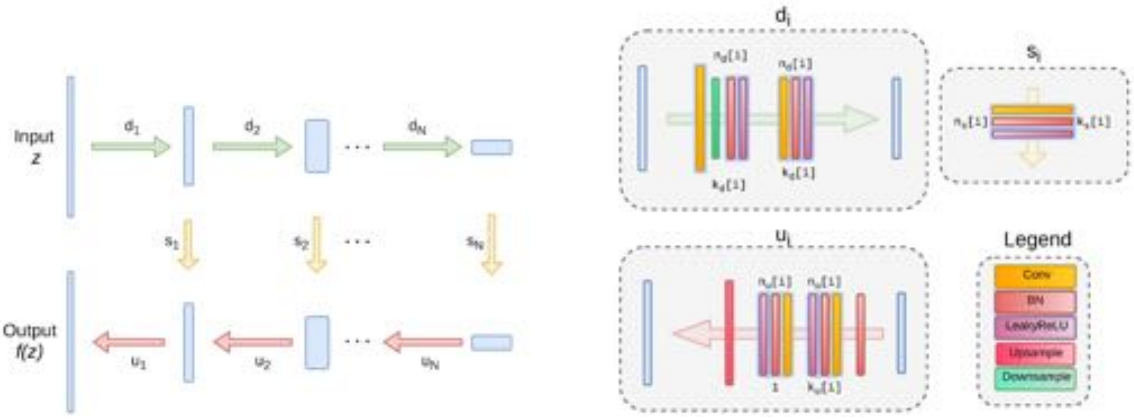


Figure 2: Architecture of SkipNet - the original DIP network. Best viewed in color. Graphic taken from Supplementary material of [31].

The activation function layer introduces a non-linearity to the model. The original activation function of SkipNet is Leaky ReLu [20] (3.3). Possible alternatives, that were already considered by other researchers [28, 27] are Relu [18], Swish [25], RSwish [28] and Mish [23] (see Figure 3).

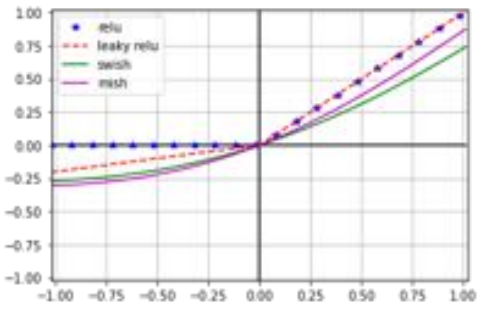


Figure 3: Visualization of activation functions.

Function	Equation
Relu	$\begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$
Leaky Relu	$\begin{cases} 0.02 \cdot x & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases}$
Swish	$\frac{x}{1+exp(-x)}$
Mish	$x \cdot tanh(ln(1 + exp(x)))$

Another part of the network that can be modified are the down- and upsampling methods. The downsampling in the decoder part of the network reduces the image dimensions in height and width. This can be achieved by means of average or maximum pooling layers or the stride built in convolutional layers. The upsampling restores the image dimension of the corresponding level in the network. Aside from bilinear or nearest upsampling, there are transposed convolutional layers, also called deconvolution. [6]

In comparison to the learning-based network architeture search by [6], some of above mentioned architecture modifications are tested with a manual hyperparameter search and results are presented in 5.

3.4 Training

The training workflow is sketched in Figure 4 for the first third training epochs. The input to the model training is a random noise input vector z and the corrupted image is used for the loss calculation. The model is initialized with random weights represented by θ_0. One training epoch consists of the following steps:

1. The current model state f_{θ_0} is applied on the random input vector z and the resulting prediction is used as the current model output $f_{\theta_0}(z)$.
2. The loss function between the model output and the corrupted image is calculated with the data term $E(f_{\theta_0}(z), \hat{x})$ from Equations (4 and 5), depending on the task. If the network prediction and the corrupted are not of same size, the output is down-sampled for the loss function calculation. A downsampling operator $d(.) : R^{C \times tH \times tW} \longrightarrow R^{C \times H \times W}$ with a lanczos kernel of size 3.
3. A gradient-based method is used for the optimization problem with respect to the model weights $\delta E / \delta \theta$.
4. The model weights are updated based on the optimizer $\theta_{k+1} = \theta_k - \alpha \cdot \frac{\delta E(f_\theta(z); \hat{x}}{\delta \theta}$, with leanring rate α.

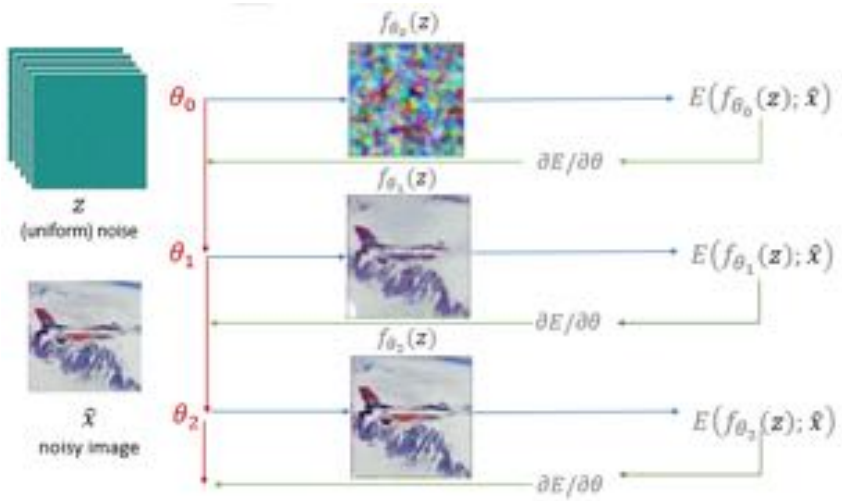


Figure 4: DIP training with denoising as example task.

4 Data

Two datasets used for the experiments in Section 5 are described in this section. All images are processed to have pixel values in the range $[0, 1]$.

4.1 Original data

The images from the paper "Deep Image Prior" by Ulyanov et al. [31] are considered as the original data (see Figure 5).

Figure 5: Original data from "Deep Image Prior", *F16* and *Zebra*. [31]

4.2 ISBI 2012

The dataset of the ISBI segmentation challenge 2012 [2] provides 2D electron microscopy (EM) images of cells. The challenge dataset consists of 30 sections for training (including validation dataset) and a separate test dataset of 60 sections. Some examples of the training dataset are shown in Figure 6 Additionally to the images, binary segmentation masks are provided. White pixels correspond to segmented objects, black pixels are the background (mostly membranes).

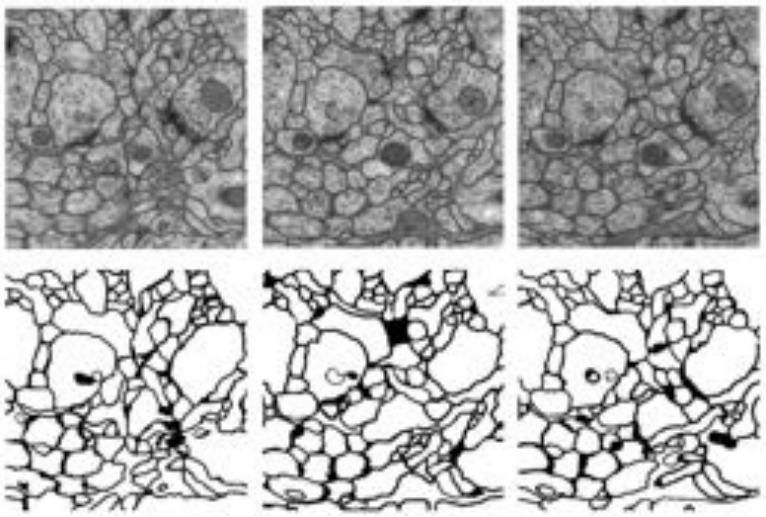


Figure 6: Data examples of the ISBI 2012 challenge datasets with the numbers 1-3. Top row: EM images, bottom row: segmentation mask. [2]

5 Experimental results

This section contains a description of the performance measures used and the results of experiments performed on the datasets described. Different modifications of the model architecture are tested as well as some modifications to the training setup. Tensorflow [1] is used for the implementation, which is available at `https://github.com/CarolineMagg/DeepImagePrior_Python` and is based on the implementation provided alongside the original paper at `https://github.com/DmitryUlyanov/deep-image-prior`. All experiments were performed on an NVIDIA RTX 3070 Laptop GPU 8GB and a Ryzen 7 5800H processor.

5.1 Performance measures

Mean squared error (MSE) (6) serves as loss function during training:

$$MSE(I,K) = \frac{1}{3mn}\sum_{j=0}^{m-1}\sum_{i=0}^{n-1}\sum_{c=0}^{2}(I_c(i,j) - K_c(i,j))^2. \tag{6}$$

I_c and K_c represent the cth channel of the input and the corrupted image, respectively. If the two images are the same, then the MSE is 0. Thus, the lower the value, the better.
The peak signal-to-noise ration (PSNR) (7) and the structural similarity index measure (SSIM) (8) are used as performance measurements during evaluation:

$$PSNR(I,K) = 10\log_{10}(\frac{Max(I)^2}{MSE(I,K)}) \tag{7}$$

$$SSIM(I,K) = \frac{(2\mu_I\mu_K + c_1)\cdot(2\sigma_{IK} + c_2)}{(\mu_I^2 + \mu_K^2 + c_1)\cdot(\sigma_I^2 + \sigma_K^2 + c_2)}. \tag{8}$$

$Max(I)$ is the maximal possible pixel value of the images. The average μ, the variance σ_K and σ_I and covariance σ_{IK} are calculated with respect to the images I or K. $c_1 = (k_1L)^2$, $c_2 = (k_2L)^2$ are calculated with L being the dynamic range of the pixel values and the default values $k_1 = 0.01$ and $k_2 = 0.03$. It is better for both values, the higher they are. The PSNR has a logarithmic shape and the SSIM is in the range of $[0, 1]$ with 1 if the two images are exactly the same.

5.2 Original data

Experiments on the original data were performed on a single image per task. In the code provided alongside the original paper, for each task and for each image an individual setting is proposed. Therefore, experiments for the original data are performed on a single image to get an image-specific comparison of different modifications.
To average the results and make a train run reproducible, the network and network input are initialized each time with a pre-defined random seed. After the training run, the trained model is applied to the initial random vector. The prediction is compared to the ground truth image and the corrupted image by means of PSNR,

SSIM and MSE. Additionally, the maximum and last value of the PSNR and SSIM during the training are documented. The best model version with respect to PSNR of network input and ground truth image (ie *val_PSNR*) is stored. This should prevent overfitting to the noise in the data.

5.2.1 Denoising

Denoising is tested on the image *F16*. The standard setting for this image is the following (this will be considered the baseline setting for denoising in this section):

- Optimizer: Adam with learning rate of 0.01
- Loss function: Mean Squared Error (MSE)
- Model architecture: SkipNet
- Feature map sizes: [128,128,128,128] for the decoder-encoder part and [4,4,4,4] for skip connections
- Model input size: $[512, 512, 32]$, ie 32 input channels
- Down- and upsampling mode: stride of convolutional layer and bilinear
- Activation function: Leaky Relu
- Number of epochs: 3000
- Gaussian noise: $\sigma = 25/255$

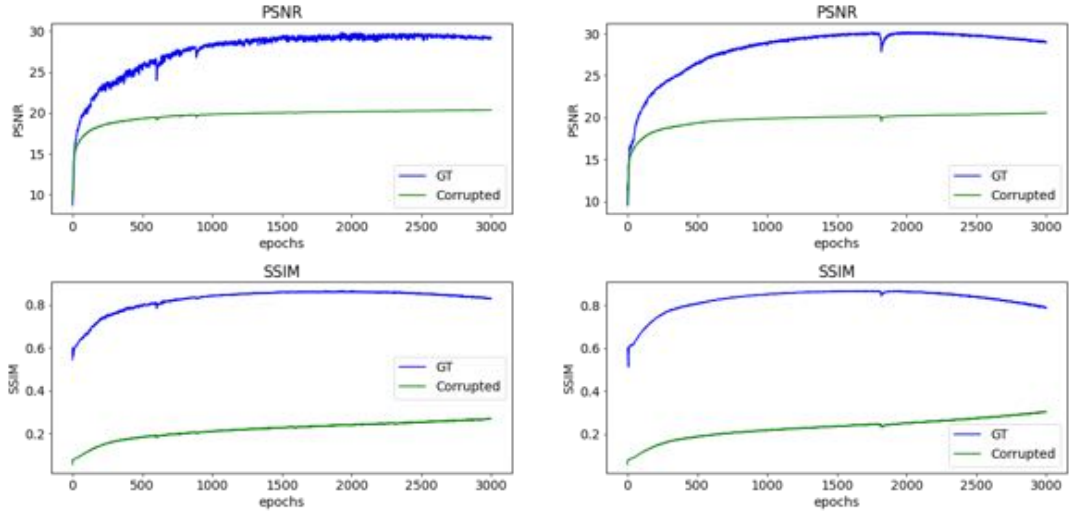
Different activation functions, number of feature maps and feature map sizes, input channel numbers, down- and upsampling methods are compared with the baseline setting. The performance for the ground truth and the corrupted image are shown in the Table 1, respectively. The average time for a training run was roughly 10 minutes.

The baseline settings achieve a PSNR of 30.006 and a SSIM of 0.861. Changing one setting in the model architecture results in 6 cases in a better PSNR and SSIM. Although, only in 4 cases, the PSNR as well as the SSIM are higher than for the baseline. Those settings are using Swish and Mish activation functions, increasing the model size to $[128, 128, 128, 128, 128]$ filters and increasing the capacity of skip connections from 4 to 8 filters. When combinations of these 4 better single setting changes are tested, only the combination Swish & 8 skip connection filters can improve PSNR and SSIM. This combination achieves the highest scores with 30.276 PSNR and 0.864 SSIM.

Both the values for last and maximal value of PSNR and SSIM and the training progress depicted in Figure 7 show that the baseline method reaches a plateau. The modifications with good results, such as Swish & 8 filters in the skip connection, show a smoother curve that is already declining at the end of the training. In Figure 8, visual results of selected settings for a single training run are presented.

Ground truth Settings	MSE ↓ eval	PSNR ↑ eval	SSIM ↑ eval	PSNR max	PSNR last	SSIM max	SSIM last	Time sec
Baseline	0.001	**30.006**	**0.861**	30.01	29.24	0.87	0.83	519.94
Activation function								
Relu	0.0011	29.593	0.844	29.59	28.86	0.85	0.81	503.74
Swish	0.00097	**30.143**	**0.862**	30.14	27.53	0.87	0.82	511.8
Mish	0.00098	**30.088**	**0.863**	30.09	28.81	0.86	0.79	528.46
Downsampling								
MaxPool	0.00122	29.17	0.85	29.17	28.33	0.85	0.84	513.11
AvgPool	0.00116	29.347	0.853	29.35	28.31	0.86	0.84	507.46
Upampling								
Nearest	0.00109	29.612	0.854	29.61	25.22	0.86	0.56	472.15
Feature maps (filters) and sizes								
3x128	0.00104	29.841	0.86	29.84	29.21	0.86	0.83	498.61
5x128	0.00097	**30.122**	**0.862**	30.12	29.53	0.86	0.84	530.74
4x32	0.00111	29.54	0.847	29.54	29.21	0.85	0.85	334.19
4x64	0.00097	**30.131**	0.857	30.13	29.9	0.86	0.86	387.9
$[16, 32, 64, 128]$	0.00107	29.713	0.848	29.71	29.51	0.85	0.85	310.08
$[16, 32, 64]$	0.00116	29.346	0.841	29.35	29.18	0.84	0.84	275.04
$[32, 64, 128]$	0.00096	**30.181**	0.853	30.18	29.9	0.86	0.85	317.72
Skip connections filters								
1 filter	0.00117	29.313	0.854	29.31	28.63	0.86	0.83	506.49
8 filters	0.00098	**30.1**	**0.864**	30.1	29.17	0.87	0.84	499.29
Input channels								
1 channel	0.00161	27.933	0.83	27.93	26.6	0.83	0.81	383.69
3 channels	0.00138	28.611	0.844	28.61	27.72	0.85	0.82	394.82
16 channels	0.0011	29.576	0.859	29.58	28.39	0.86	0.82	423.07
Combination of best 4 single setting changes								
5x128 & 8 skip	0.00105	29.813	0.86	29.81	28.91	0.86	0.83	523.83
Swish & 5x128	0.00106	29.747	0.858	29.75	28.15	0.86	0.82	589.0
Swish & 5x128 & 8 skip	0.001	29.992	**0.862**	29.99	28.96	0.86	0.81	555.22
Swish & 8 skip	0.00094	**30.276**	**0.864**	30.28	28.96	0.87	0.79	540.36
Mish & 5x128	0.00104	29.849	0.857	29.85	28.66	0.86	0.8	543.3
Mish & 5x128 & 8 skip	0.001	29.994	0.857	29.99	28.85	0.86	0.8	575.88
Mish & 8 skip	0.00096	**30.183**	0.861	30.18	27.63	0.86	0.77	547.66

Table 1: Comparison of the DIP denoising results with the original image *F16* (GT) for different settings. Except of the last block, single setting changes are made. The last block shows the results achieved by combining the best 4 changes. The first columns indicates the changes setting. All other settings are kept to the baseline setting.



(a) Baseline (b) Swish & 8 skip connection filters

Figure 7: PSNR and SSIM performance during training between network prediction and GT image. Best viewed in color.

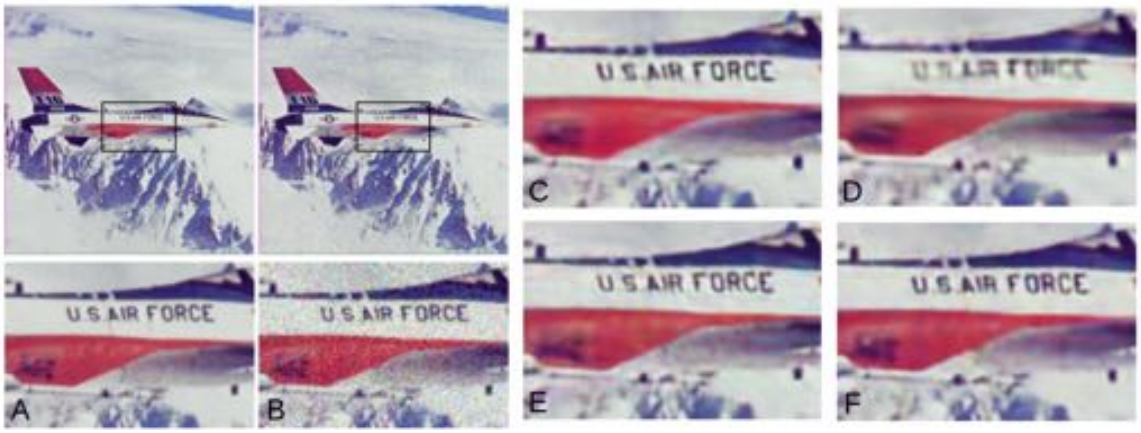


Figure 8: Visualization of GT (A) and noisy image (B) and denoising results of one training run for a crop-out region of the image with the settings: baseline (C), 1 channel input (D), Mish (E) and Swish & 8 skip connection filters (F).

5.2.2 Super Resolution

The super resolution is tested on the image *Zebra*. The standard setting for this image is the following (this will be considered the baseline setting for SR in this section):

- Optimizer: Adam with learning rate of 0.01
- Loss function: Mean Squared Error (MSE)
- Model architecture: SkipNet
- Feature map sizes: [128,128,128,128] for the decoder-encoder part and [4,4,4,4] for skip connections

- Model input size: $[512, 512, 32]$, ie 32 input channels
- Down- and upsampling mode: stride of convolutional layer and bilinear
- Activation function: Leaky Relu
- Number of epochs: 2000
- Downscaling factor: 4

The same setting combinations as for denoising are tested. The results are shown in Tables 2. The training runs take roughly 6 minutes on average.

In comparison to the denoising application, no modification exceeded the results of the baseline setting for a PSNR of 23.471 and a SSIM of 0.678. Although, the same combinations are again within the top 5. Swish activation function and 8 skip connection filters reach higher values in SSIM (0.679) and PSNR (23.486), respectively. Combining the both modifications result in a PSNR of 23.403 and a SSIM of 0.678.

Using DIP can boost the performance by some decimals compared to the standard upsampling methods with nearest neighbor and bicubic interpolation. Upsampling of the low-resolution image with nearest neighbor interpolation results in 21.01 PSNR and 0.603 SSIM. Using bicubic interpolation results in 23.122 PSNR and 0.678 SSIM.

Figure 9 shows results of the worst and best results aside from the baseline setting. Based on the results in the original paper, it was expected that the visual appearance of the result is better than it actually is.

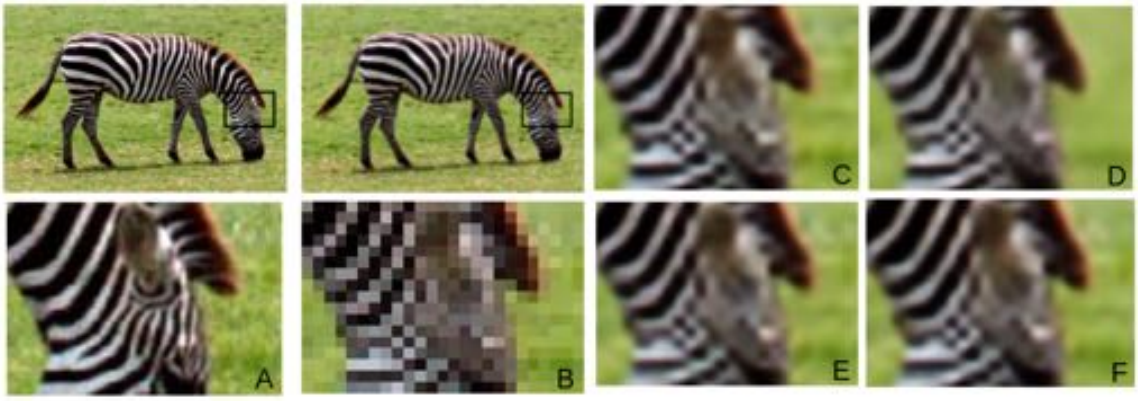


Figure 9: Visualization of high-resolution (A) and low-resolution image up-sampled with nearest-neighbors interpolation (B) and super resolution results of one training run for a crop-out region of the image with the settings: baseline (C), MaxPooling (D), Swish (E) and 8 skip connection filters (F). Please zoom in for more details.

GT	MSE ↓ eval	PSNR ↑ eval	SSIM ↑ eval	PSNR max	PSNR last	SSIM max	SSIM last	Time sec
Baseline	0.0045	**23.471**	**0.678**	23.47	23.33	0.68	0.68	306.74
Activation function								
Relu	0.00486	23.138	0.653	23.14	22.96	0.65	0.65	233.84
Swish	0.00451	23.457	**0.679**	23.46	23.4	0.68	0.68	249.64
Mish	0.00463	23.347	0.674	23.35	23.27	0.68	0.67	256.48
Downsampling								
MaxPool	0.00604	22.19	0.566	22.19	21.74	0.58	0.56	240.76
AvgPool	0.00579	22.374	0.613	22.37	21.64	0.62	0.61	232.28
Upampling								
Nearest	0.00631	22.02	0.578	22.02	21.55	0.6	0.59	225.58
Feature maps and sizes								
3x128	0.00456	23.412	0.678	23.41	23.26	0.68	0.68	219.73
5x128	0.00451	23.462	0.669	23.46	23.22	0.67	0.67	262.09
4x32	0.00529	22.766	0.607	22.77	22.67	0.61	0.61	222.39
4x64	0.00494	23.066	0.645	23.07	22.95	0.65	0.64	242.36
$[16, 32, 64, 128]$	0.00481	23.181	0.633	23.18	23.12	0.64	0.63	194.01
$[16, 32, 64]$	0.00528	22.772	0.593	22.77	22.59	0.6	0.59	152.56
$[32, 64, 128]$	0.00482	23.172	0.649	23.17	23.08	0.65	0.65	165.54
Skip connections								
1 filter	0.0048	23.186	0.657	23.19	22.91	0.66	0.65	246.74
8 filters	0.00448	**23.486**	0.675	23.49	23.3	0.68	0.68	246.07
Input channels								
1 channel	0.00541	22.673	0.618	22.67	22.22	0.62	0.6	207.5
3 channels	0.00524	22.805	0.63	22.81	20.88	0.64	0.58	210.88
16 channels	0.00484	23.15	0.664	23.15	22.91	0.67	0.66	218.84
Combination of best 2 single setting changes								
Swish & 8 skip	0.00457	23.403	0.678	23.4	22.14	0.68	0.64	295.83

Table 2: Comparison of the DIP SR results with the original image *Zebra* (GT) for different settings. The training settings are kept to the baseline setting, if not mentioned separately.

5.3 ISBI 2012

The training dataset of the challenge dataset ISBI 2012 with 24 images is used to make an evaluation for microscopy images over several images. The evaluation process stays the same as for the original data. However, the results of one training run are averaged over the dataset. The baseline settings are the same as described in Sections 5.2.1 and 5.2.2.

5.3.1 Denoising

Based on the results in Section 5.2.1, the best 5 aside from the baseline setting are tested and compared to each other. Table 3 collects results averaged over the 24 images. Except the larger network with 5 times 128 filters, all settings achieve better results than the baseline setting. The best setting is again Swish & 8 skip connection filters with PSNR 25.775 and SSIM 0.841.
Baseline results for the best reconstructed image (number 8) with PSNR 26.813 and SSIM 0.824 and the worst reconstruction (number 21) with PSNR 22.351 and SSIM 0.783 are shown in Figure 10.

GT	MSE ↓ eval	PSNR ↑ eval	SSIM ↑ eval	PSNR max	PSNR last	SSIM max	SSIM last	Time sec/img
Baseline	0.00344	24.909	0.819	24.91	24.55	0.82	0.82	475.06
Swish	0.00296	25.583	0.839	25.58	24.4	0.84	0.81	523.33
Swish & 8 skip	0.00285	25.775	0.841	25.78	24.23	0.85	0.8	527.45
8 skip	0.00325	25.165	0.829	25.16	24.69	0.83	0.82	475.67
5x128	0.00379	24.48	0.803	24.48	24.1	0.81	0.8	483.25
Mish	0.003	25.513	0.836	25.51	24.41	0.84	0.82	519.17

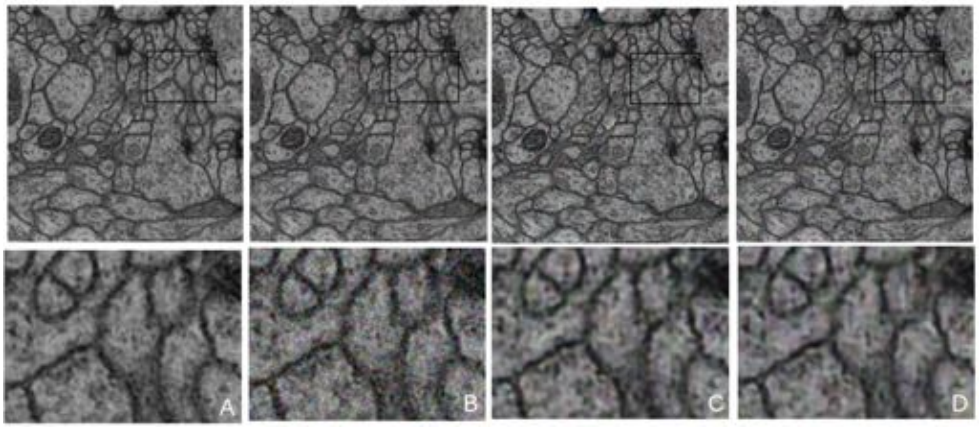
Table 3: Denoising: comparison of 6 different settings for the training dataset of the ISBI 2012 challenge.

5.3.2 Super Resolution

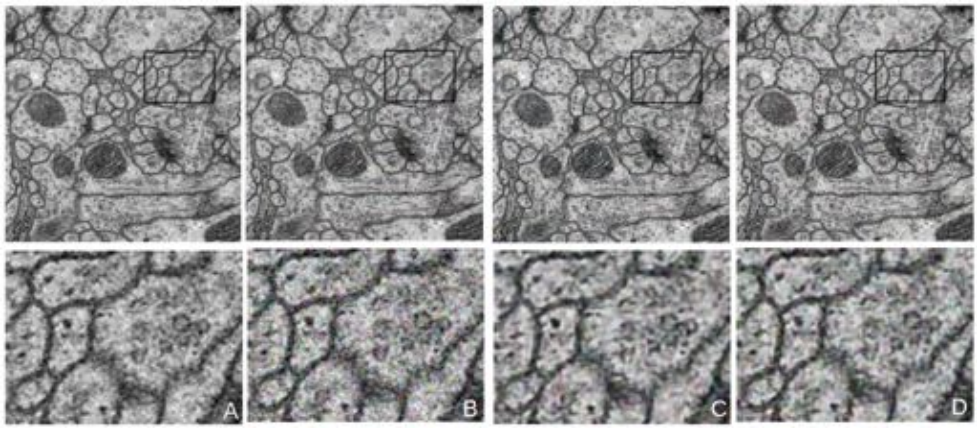
The super resolution results for the same settings as for denoising are collected in Table 3. The results are averaged over the dataset with 24 images. For super resolution the baseline setting achieves the best results with PSNR of 21.991 and SSIM of 0.542 for the medical dataset, in comparison to the single natural image.
Figure 11 shows the best high-resolution image (number 8: PSNR 25.049 and SSIM 0.647) and worst (number 21: PSNR 19.182 and SSIM 0.43) with respect to the baseline setting as a visual example.

GT	MSE ↓ eval	PSNR ↑ eval	SSIM ↑ eval	PSNR max	PSNR last	SSIM max	SSIM last	Time sec/img
Baseline	0.00678	21.991	0.542	21.99	21.87	0.55	0.53	403.18
Swish & 8 skip	0.00684	21.955	0.527	21.96	21.85	0.53	0.52	425.69
Swish	0.00688	21.926	0.53	21.93	21.77	0.54	0.52	420.57
8 skip	0.00679	21.985	0.539	21.98	21.86	0.54	0.53	395.99
5x128	0.00695	21.881	0.536	21.88	21.73	0.54	0.53	407.42
Mish	0.00682	21.962	0.531	21.96	21.42	0.54	0.49	434.66

Table 4: Super Resolution: Comparison of 6 different settings for the training dataset of the ISBI 2012 challenge.

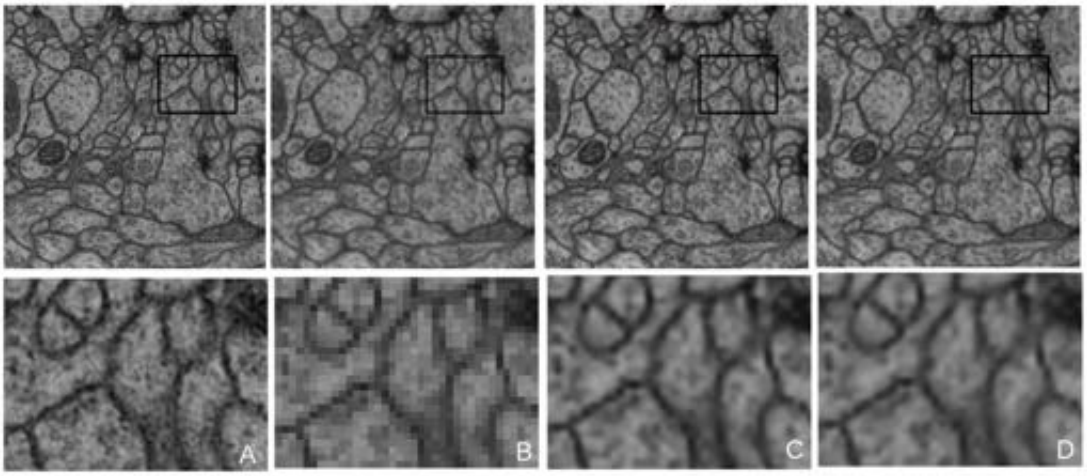


(a) Number 8 - best PSNR performance for baseline.

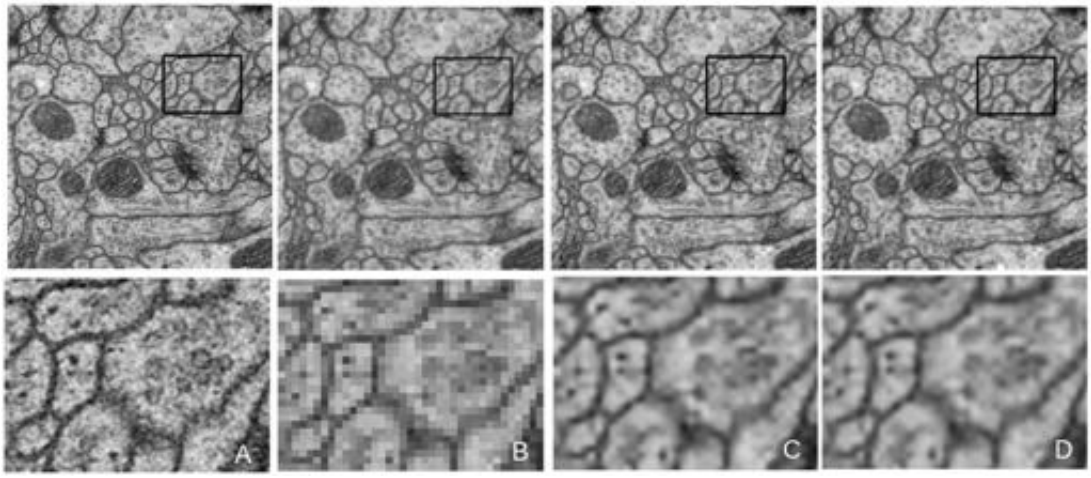


(b) Number 21 - worst PSNR performance for baseline.

Figure 10: Visualization of denoising for microscopy images from ISBI challenge dataset: GT (A), noisy image (B), baseline setting (C), Swish & 8 skip connection filters (D). Please zoom in for details.



(a) Number 8 - best PSNR performance for baseline.

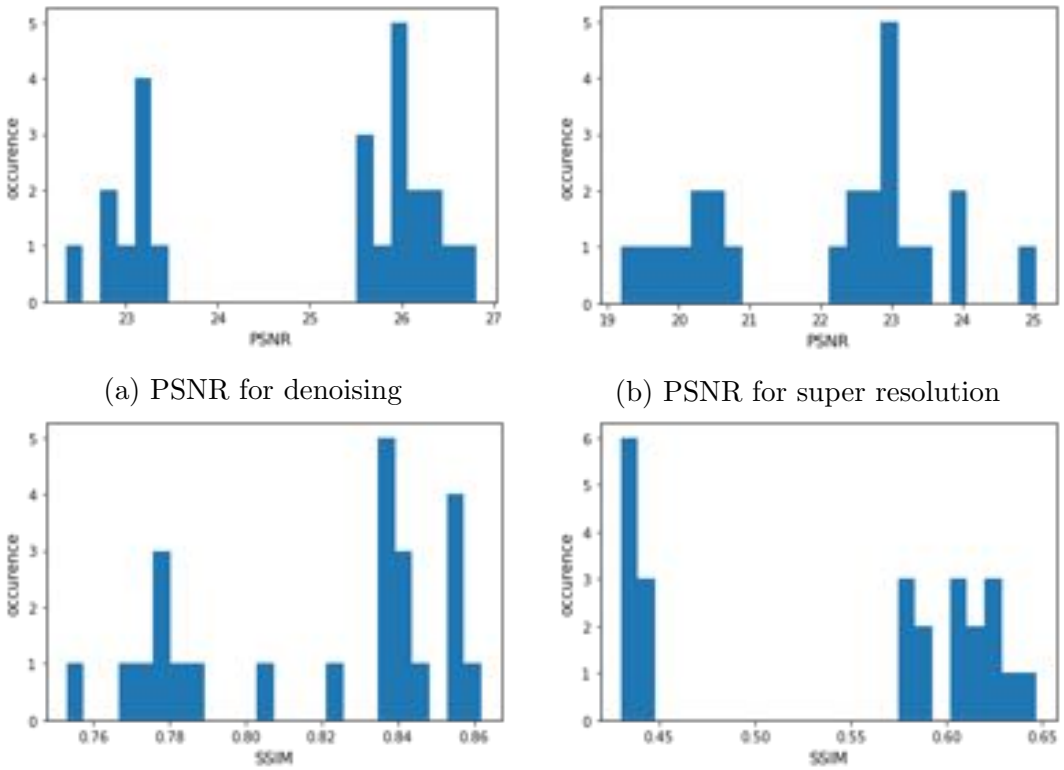


(b) Number 21 - worst PSNR performance for baseline.

Figure 11: Visualization of super resolution for microscopy images from ISBI challenge dataset: GT (A), noisy image (B), baseline setting (C), Swish & 8 skip connection filters (D). Please zoom in for details.

5.4 Overview best settings

The distribution of the PSNR and the SSIM for both tasks, denoising and super resolution, are shown in Figures 12. The performance measures show a wide range. After a visual inspection of some exemplary results, it seems that the noise component in some images is higher than in others and affects the results. Figure 13 visualizes the distribution of best settings in the ISBI 2012 challenge dataset for both tasks. Although the best candidates for denoising are limited, the same does not apply to super resolution. The reason for that needs to be further investigated. However a first hypothesis is that denoising is more stable in terms of network modifications and super resolution is more sensitive to smaller changes.



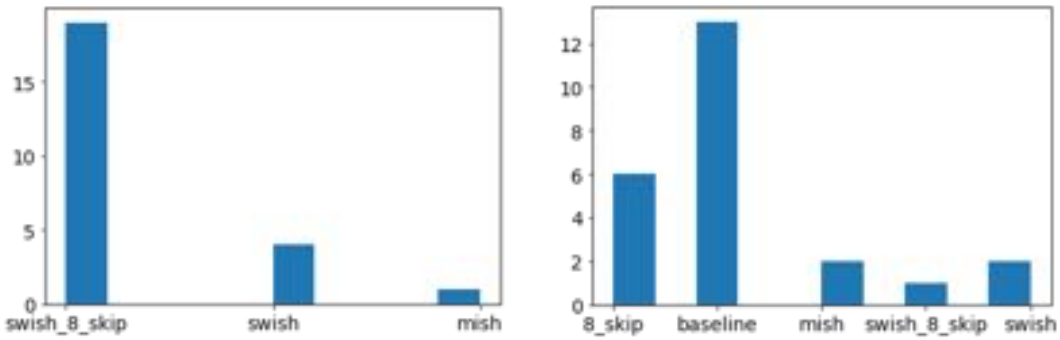
(a) PSNR for denoising

(b) PSNR for super resolution

(c) SSIM for denoising

(d) SSIM for super resolution

Figure 12: Distribution of PSNR and SSIM measure for denoising and super resolution.



(a) denoising

(b) super resolution

Figure 13: Bar plot of best settings for denosing and super resolution for ISBI 2012 challenge dataset.

6 Discussion

In this section, advantages and limitations of the DIP method are discussed.

6.1 Advantages of DIP

The big advantage of the DIP method is that no pre-training of a network with a large dataset is necessary compared to standard supervised deep learning approaches. This opens the possibility to apply the method to domains where large (labeled) datasets might be rare, such as the medical and biological domain.
The single-image based approach has the advantage, that the network is designed for the image, including the input shape that is adaptable to the input image size. With pre-trained networks containing dense or pooling layers, the image dimension needs to fit the network, not the other way around. The only trade-off that needs to be considered with using large input layers is the resulting network size and the GPU memory available.
Another bonus of DIP is that due to the fact that no prior training is used, the results don't show hallucinations [22]. Especially looking at generative or reconstruction approaches in the medical and biological domain, it is critical, that no anatomical or biological structures or artifacts are added to the image [22].

6.2 Limitations of DIP

As already discussed in the work by Ulyanov et al. [31], based on the number of training iterations, the results can get worse again. Since the number of iterations is never exceeding 3000, this effect is not observed in the experiment for this report. However, this raises the question for the optimal number of iterations. The user needs to stop the training at the right time otherwise the network will adapt to the noise in the corrupted image. Since the timing might be different for different images, this decision is critical and can be difficult to generalize. The solution for not overfitting to the noise in this work was using model checkpoints based on the PSNR. However, the best epoch is not known a-priori, which leads to a higher number of iterations as necessary.
Another drawback of the DIP method is the training time. Although, a time-consuming, dedicated training phase with a large dataset is not necessary, the training at inference time will take a couple of minutes depending on the hardware. The average time in the experiments in section 5 per image was between $6 - 10$ minutes depending on the network and training settings. Therefore, the method can not be applied to a large dataset on the fly. In addition, since training is applied at inference time, appropriate hardware, such as a GPU, is needed for inference. To overcome this limitation, parallelization and GPUS with large memory or TPU can be used.
Finding the optimal training and network settings for an image can be challenging. As already observed by others [22], there are differences between natural images and images from medical modalities. A good combination for natural images or single images in general do not result in optimal results for microscopy images or an entire dataset. In this work, the hyperparameter search was performed manu-

ally. NAS-DIP [6] provides an automatic framework to search for an appropriate network architecture. However, the NAS-DIP training is time-consuming and a separate train and test dataset are needed.

7 Conclusion & Future Work

This report investigated the method Deep Image Prior (DIP) by Ulyanov et al. An unsupervised method that can be used for inverse tasks, such as denoising and super resolution, and does not need a pre-training with a large dataset. Since the network is in the focus of the DIP method, different settings for the SkipNet architecture, such as activation function, the feature maps of the convolution layers, down- and upsampling methods, are tested and compared. In addition to analyzing different architecture settings, the method is applied to microscopy images. The results show that in general, the results from natural images can be translated to some extend to microscopy data. Testing different settings showed that the majority of the images achieve their best denoising result with one setting. However, looking at the best methods for each images indivdiually showed that there are exceptions. The preferable setting for super resolution comes from a larger selection which makes it harder to chose a good setting for a low-resolution images.
There are some limitations to the method, such as the long average training time per image at test time, finding the optimal time to stop the training, and the optimal network setting. However, the method shows an interesting approach for an unsupervised method. This opens the opportunity for applying noise reduction or generating high-resolution images to individual images. The unsupervised approach might help especially in domains, where large labeled datasets are rare, such as the medical and biological field.
In case the method can overcome the current limitations, DIP could also enrich the traditional supervised training workflow either with label generation or pre-labeling for inverse tasks or as a pre-processing step before supervised training. The latter would mean a combination of DIP with external training and could be beneficial also for high-level tasks, such as segmentation or classification. Improving the limitations of DIP would also result in improving Double-DIP, which might lead to a better usage for segmentation also for the medical and biological domain. Aside from improving DIP itself, another approach for the future would be to combine external and internal learning to make use of both paradigms and combine the benefits.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] I. Arganda-Carreras, S. C. Turaga, D. R. Berger, D. Cireşan, A. Giusti, L. M. Gambardella, J. Schmidhuber, D. Laptev, S. Dwivedi, J. M. Buhmann, T. Liu, M. Seyedhosseini, T. Tasdizen, L. Kamentsky, R. Burget, V. Uher, X. Tan, C. Sun, T. D. Pham, E. Bas, M. G. Uzunbas, A. Cardona, J. Schindelin, and H. S. Seung. Crowdsourcing the creation of image segmentation algorithms for connectomics. *Frontiers in Neuroanatomy*, 9, 2015.

[3] D. O. Baguer, J. Leuschner, and M. Schmidt. Computed tomography reconstruction using deep image prior and learned reconstruction methods. *Inverse Problems*, 36(9):094004, Sep 2020.

[4] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 60–65, 2005.

[5] P. Chakrabarty and S. Maji. The spectral bias of the deep image prior, 2019.

[6] Y.-C. Chen, C. Gao, E. Robb, and J.-B. Huang. NAS-DIP: Learning deep image prior with neural architecture search. In *European Conference on Computer Vision (ECCV)*, 2020.

[7] Z. Cheng, M. Gadelha, S. Maji, and D. Sheldon. A bayesian perspective on the deep image prior. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[8] J. Cui, K. Gong, N. Guo, and et al. PET image denoising using unsupervised deep learning. *European Journal of Nuclear Medicine and Molecular Imaging*, 46:2780â€“2789, 2019.

[9] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007.

[10] S. Dittmer, T. Kluth, P. Maass, and D. Otero Baguer. Regularization by architecture: A deep prior approach for inverse problems. *Journal of Mathematical Imaging and Vision*, 62(3):456â€“470, Oct 2019.

[11] W. Fan, H. Yu, T. Chen, and S. Ji. OCT image restoration using non-local deep image prior. *Electronics*, 9(5), 2020.

[12] Y. Gandelsman, A. Shocher, and M. Irani. "double-DIP": Unsupervised image decomposition via coupled deep-image-priors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[13] L. Gondara. Medical image denoising using convolutional denoising autoencoders. In *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pages 241–246, 2016.

[14] K. Gong, C. Catana, J. Qi, and Q. Li. PET image reconstruction using deep image prior. *IEEE Transactions on Medical Imaging*, 38(7):1655–1665, 2019.

[15] F. Hashimoto, H. Ohba, K. Ote, A. Teramoto, and H. Tsukada. Dynamic PET image denoising using deep convolutional neural networks without prior training datasets. *IEEE Access*, 7:96594–96603, 2019.

[16] S. Kaji and S. Kida. Overview of image-to-image translation by use of deep neural networks: denoising, super-resolution, modality conversion, and reconstruction in medical imaging. *Radiol Phys Technol*, 12:235–248, 2019.

[17] B. Karamata, K. Hassler, M. Laubscher, and T. Lasser. Speckle statistics in optical coherence tomography. *J. Opt. Soc. Am. A*, 22(4):593–596, Apr 2005.

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems*, volume 1, pages 1097–â€“1105. Curran Associates Inc., 2012.

[19] J. Liu, Y. Sun, X. Xu, and U. S. Kamilov. Image restoration using total variation regularized deep image prior. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7715–7719, 2019.

[20] A. L. Maas, A. Y. Hannun, and A. Y. Ng. Rectifier nonlinearities improveneural network acoustic models. In *ICML*, volume 30, 2013.

[21] G. Mataev, P. Milanfar, and M. Elad. DeepRED: Deep image prior powered by RED. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019.

[22] L. Max-Heinrich, T. Malte, and T. Ortmaier. Uncertainty estimation in medical image denoising with bayesian deep image prior. *Lecture Notes in Computer Science*, 12442, 2020.

[23] D. Misra. Mish: A self regularized non-monotonic activation function, 2020.

[24] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett. Deep learning techniques for inverse problems in imaging. *IEEE Journal on Selected Areas in Information Theory*, 1(1):39–56, 2020.

[25] P. Ramachandran, B. Zoph, and Q. V. Le. Searching for activation functions, 2017.

[26] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241. Springer International Publishing, 2015.

[27] R. Segawa and H. Hayashi. Performance Verification of Activation Functions in Denoising of Deep Image Prior. 2021.

[28] R. Segawa, H. Hayashi, and S. Fujii. Proposal of new activation function in deep image prior. *IEEJ Transactions on Electrical and Electronic Engineering*, 15(8):1248–1249, 2020.

[29] H. Sun, L. Peng, H. Zhang, Y. He, S. Cao, and L. Lu. Dynamic PET image denoising using deep image prior combined with regularization by denoising. *IEEE Access*, 9:52378–52392, 2021.

[30] P. Trampert, S. Schlabach, T. Dahmen, and P. Slusallek. Deep learning for sparse scanning electron microscopy. *Microscopy and Microanalysis*, 25(S2):158â€“159, 2019.

[31] D. Ulyanov, A. Vedaldi, and V. Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[32] D. Van Veen, A. Jalal, M. Soltanolkotabi, E. Price, S. Vishwanath, and A. G. Dimakis. Compressed sensing with deep image prior and learned regularization, 2020.

[33] T. Vu, A. DiSpirito, D. Li, and et al. Deep image prior for undersampling high-speed photoacoustic microscopy. *Photoacoustics*, 22:100266, 2021.

[34] T. D. G. X. e. a. Wang, N. A comprehensive survey to face hallucination. *Int J Comput Vis*, (106):9–30, 2014.

[35] T. Yokota, K. Kawai, M. Sakata, Y. Kimura, and H. Hontani. Dynamic PET image reconstruction using nonnegative matrix factorization incorporated with deep image prior. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 3126–3135, 2019.

[36] J. Yoo, K. H. Jin, H. Gupta, J. Yerly, M. Stuber, and M. Unser. Time-dependent deep image prior for dynamic MRI, 2021.

[37] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE Transactions on Image Processing*, 26(7):3142–3155, 2017.

[38] K. C. Zhou and R. Horstmeyer. Diffraction tomography with a deep image prior. *Opt. Express*, 28(9):12872–12896, Apr 2020.