Deep Image Prior Analyzed, finding natural images on the path from noise to corrupted?

With deep convolutional neural nets (ConvNets) being the state of the art in reverse image reconstruction, Ulyanov et al. [1] argue that the assumption that the excellent performance is obtained due to the ability of Conv-Nets to learn realistic image priors from data, is flawed. They aim do demonstrate that image reconstruction can be done using only the information of the single degraded input image.

Deep Image Prior is a very interesting topic to study. It shows remarkable results, creating very natural images, close to the original, by painting in large gaps in images, or from only 2% of the original pixels.

As part of the Deep Learning course at the Computer Vision Lab of the University of Delft, The Netherlands, we[2] have been analyzing the results and operation of Deep Image Prior.

We have tested inpainting and super resolution on different other images, analyzed the details of restoration and tried different network configurations. In this blog we mainly discuss the highlights of our analysis. The details can be found in our github project.

Deep Image Prior and other deep learning papers have been analyzed also by other students, please check out the results here: https://reproducedpapers.org.

How does Deep Image Prior work?

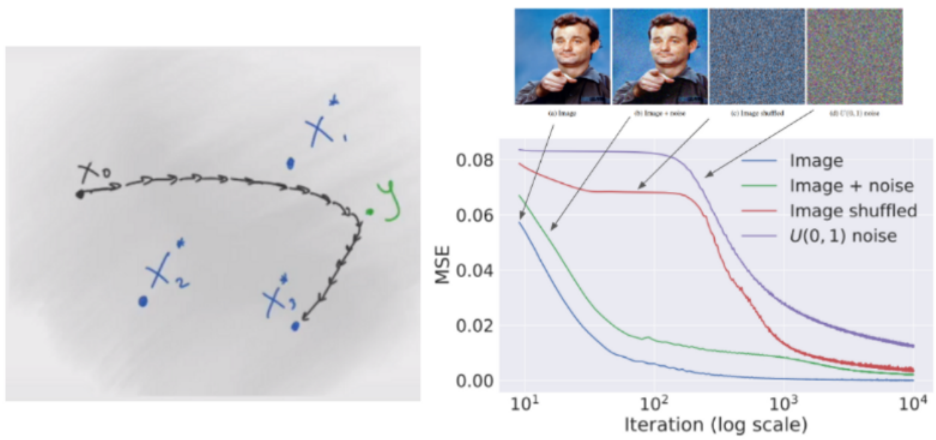
The base structure is a layered UNet type network, with multiple down and upstream steps and skip connections. Each consisting of a convolutional, batchnorm and activation layer. The input of the network is random noise, the target is the image which has been corrupted by applying a mask. The loss is computed by applying the same mask to the outputted image and then comparing with the corrupted image. This means that the loss function does not explicitly drive the repair of the noise/corruption (as it is re-applied before computing the loss). This is left over to the implicit behavior of the neural network.

Images copied from [1]

While trying to optimize towards the corrupted image the network produces very natural looking images close to the original un-corrupted image.

Why does Deep Image Prior work?

Mr. Ulyanov explains it very well in this presentation. The idea is that the neural network has the tendency to resist fitting to noise, it fits easier to structures in natural images.



Images copied from the presentation here

See the visualization above, on the path towards the noisy image (*X3*)the neural network passes the natural looking image (*Y*). See also the Bill Murray example, the MSE goes down much quicker with natural images than with more heavily corrupted images.

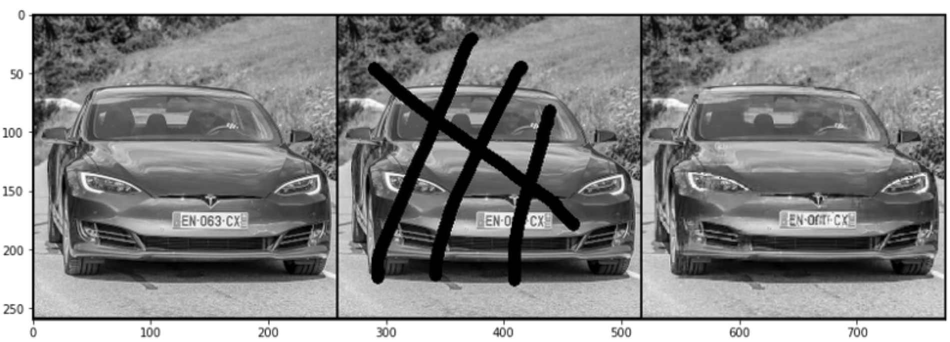
The important take-away is that Deep Image Prior does not propose a new restoration solution. As far as we can see, its message is more to share the special implicit behavior of these convolutional networks: to prefer natural over noisy structures.

Experimental Setup

The original authors performed all tasks on images from the GCF-BM3D dataset using varying hyperparameters which are explained in more depth in the original paper's supplementary. The code Ulyanov et al. have written to obtain their results in the original paper is fully provided and we use their code for reproducing their results. We will also use a custom made set of images to view the performance on a different dataset.

Inpainting, on different images and hyperparameters

For the inpainting reproduction we focused on two distinct aspects: hyper parameter settings and testing inpainting logic on other images. We report the results here, for more details see our github project.



For the hyperparameter settings:

Number of iterations: the article states that after many iterations the logic will probably overfit to the corrupted image. We ran with 15000 iterations, after the first ~3000 iterations there is no significant improvement in image quality. It seems that the network resets a couple of times within the optimisation process which is also mentioned by the article.

Learning rate (article: 0.001): with high learning rates (0.1), the network converges to a black image, while with very low learning rates (0.0001) the image gets increasingly blurrier, possible due to needing longer to converge. A low learning rate clearly blends the missing pixels better with the known pixels though.

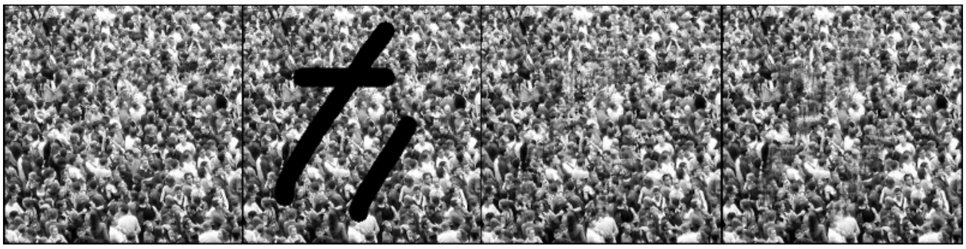
Input Depth (input channels, article: 32): with a higher input depth the filled in pixels are less noisy and blend in better with the rest of the image.

Network Input Type: the type used for the input image (noise or mesh grid), the noise gives much better results.

Input noise standard deviation: defines the additional noise added at each iteration, acts as a kind of regularization. As indicated by the article a value of 0.1 seems to work best.

We tried several other images (see above, the Tesla and below a grass field). Our tests showed that each image requires its specific set of hyper parameters, the result is very dependent on choice of hyperparameters picking.

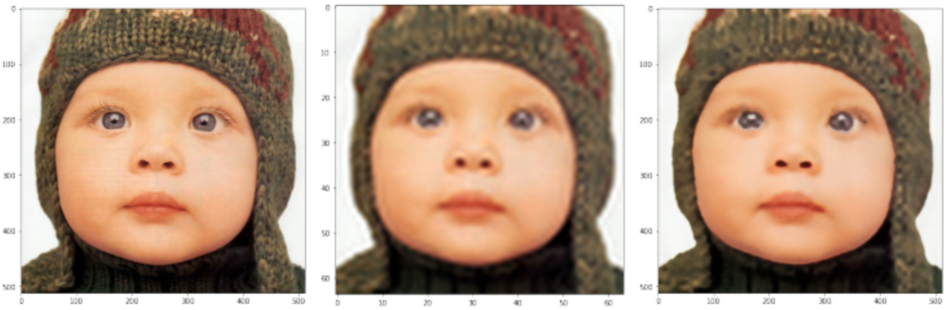
Results from running the network on a grass field with clouds, left image is the original, the second is the masked image, the third image is the result with input noise standard deviation of 0.05, a learning rate of 0.1, and an input depth of 2, the fourth image was created using an input noise standard deviation of 0.2, a learning rate of 0.001, and an input depth of 32 for the first image. This was with 5000 iterations. What is remarkable is that Deep Image Prior returns a more precise result (without the white shade).



Same parameter setting for the grass field above. What is also apparent is that Deep Image Prior has trouble creating a realistic image here. The crowd of people probably provides less 'natural' structure and is more 'noisy'.

Super resolution

Another application of deep image priors discussed in the Deep Image Prior paper [1] is super resolution. The aim of this task is to create a high resolution image from a low resolution image by upsampling it with a factor t.



Left: original image, middle: downsampled image, right: generated by Deep Image Prior

We tried to reproduce several results of Deep Image Prior, the results are shown below.

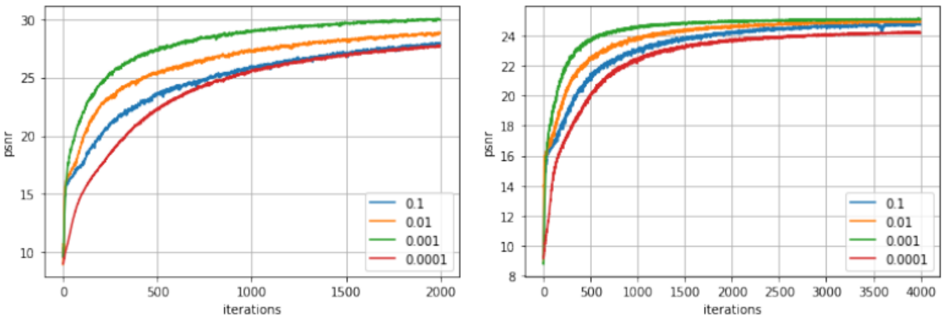
	Baby	Bird	Butterfly	Head	Woman
Ulyanov et al.	31.49	31.8	26.23	31.04	28.93
Ours	**29.53**	**28.86**	**24.41**	**28.21**	**26.77**

Super resolution results compared to [1] with factor 4

	Baby	Bird	Butterfly	Head	Woman
Ulyanov et al.	28.28	27.09	20.02	29.55	24.5
Ours	**26.95**	**24.79**	**18.95**	**27.08**	**23.15**

Super resolution results compared to [1] with factor 8

We could not get the same results, so we have tried different parameter settings. First we checked how different learning rates influence the performance.

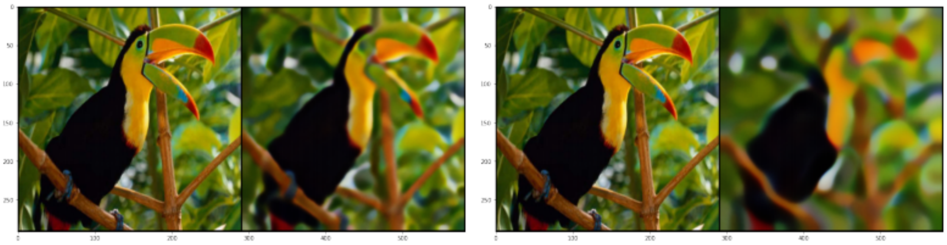


Result for different learning rates on left factor 4and right downsample factor 8

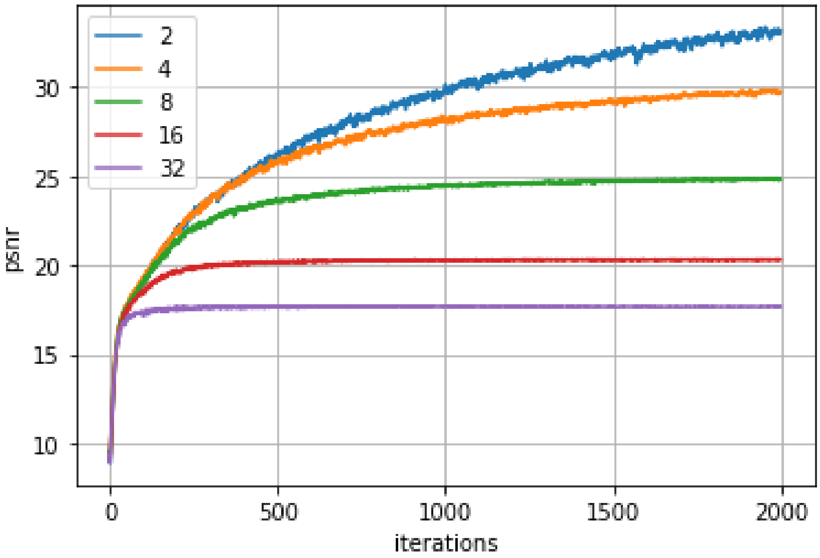
This shows that a learning rate of 0.001 gives better results than what the article uses: 0.01.

We also tried UNet instead of the original network (which has skip connections), however the original network showed much better results.

Another parameter we wanted to check is how far Deep Image Prior can get improving if we downsample with higher factors. The result is shown below in the chart and the bird images for factors 8 and 16. It shows that Deep Image Prior starts to show degrading performance above downsampling with more than factor 4.



Left: original and result after downsampling with factor 8, Right: same but downsampling with factor 16



PSNR development for different downsample factors

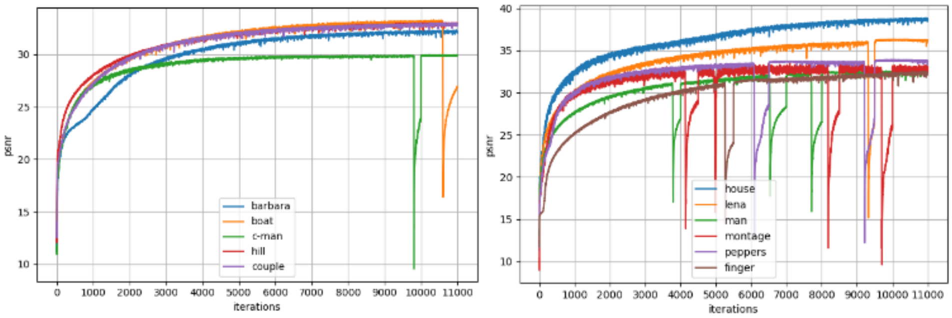
Restoration, reproduced

We worked on reproducing table 1 from [1] using the original source code. In addition we did a more detailed analysis of the performance during the optimization. This provides us with some interesting insights.

The table below shows the reproduction of table 1 of [1]. As the PSNR varies (see next image) we choose to display the maximum value after 11000 iterations.

	Barbara	Boat	House	Lena	Peppers	C.man	Couple	Finger	Hill	Man	Montage
Paper	32.22	33.06	39.16	36.16	33.05	29.8	32.52	32.84	32.77	32.20	34.54
Ours	**32.32**	**33.26**	**38.87**	**36.29**	**33.89**	**29.96**	**33.02**	**32.51**	**33.02**	**32.65**	**33.52**

As you can see we obtain sometimes better results, sometimes slightly less. Overall we think that table 1 in itself is reproducible. However, when looking at the detailed iterations there are some specific observations to make.



The images above show the PSNR per iteration for the 11 images of table 1 of [1]. All images quickly improve their PSNR, than flatten out, with sometimes sudden drops in the result.

When looking at the PSNR development over time, some thoughts on how and if this corresponds to the original article:

Yes: a clear tendency to stick to the natural image structure, the characteristic to stick to natural images seems even much stronger than described by the authors.

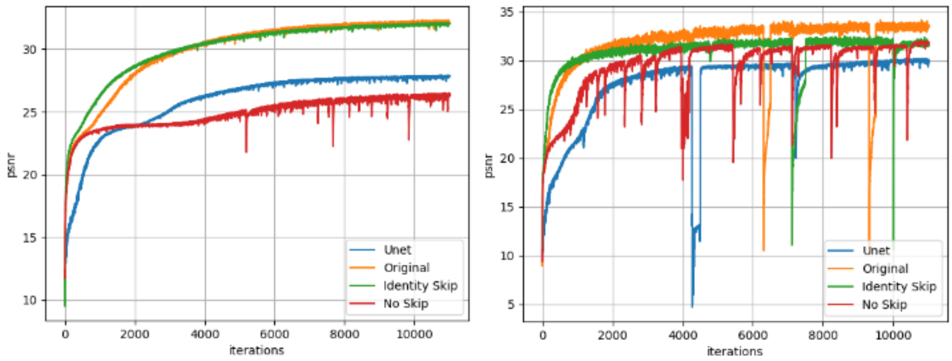
Yes: the logic will fit to the corrupted image. But this is fine as the loss function is computed after re-applying the mask on the output. So with this approach the logic can continue to run, generating natural fill-ins for the missing pixels.

Maybe Not: variations in PSNR make it difficult to reproduce the claim of [1] in table 1, that Deep Image Prior is better than other solutions.

Maybe Should: interesting to know why the PSNR sudden drops happen at all.

Restoration, other network configurations

Another interesting take was to see how important the skip connections are for the performance of the restoration. We tested 3 additional network configurations: UNet as used by the authors, the original network but without skip connections and the original network but with identity skip connections.



Different networks tested on images barbara and montage of [1]

The results are displayed above. This shows clearly (for the two example images) that the alternative network configurations do not perform better and also show more instability.

Conclusions

From the experiments run above we can conclude that indeed the results from the original Deep Image Prior paper can be replicated with their code and that the results are very dependend on the choice of hyperparameters. Hyperparameter tweaking is needed when trying this method on a different set of images.

Overall we see that that the hyper parameters for super resolution set by Ulyanov et al. perform near optimal. The only improvement that we found is to set the learning rate to `0.001` instead of `0.01`. For the best network, the skip network should be used over UNet and for small images like the bird image it is hard to reproduce a natural image when the factor gets higher than 8.

Another important conclusion is that indeed these convolutional networks have a strong tendency to prefer natural structures. This behavior seems even stronger than indicated. The sudden drops in the PSNR values are specific and it not sure why this happens.

The claim in table 1 of [1] that the Deep Image Prior performs better than the indicated alternative solution is not directly reproducable. Especially because the variations in the PSNR are considerable and sometimes larger than the published improvement in the article.

. . .

[1] Victor Lempitsky Dmitry Ulyanov, Andrea Vevaldi. Deep image prior.https://github.com/DmitryUlyanov/deep-image-prior

[2] We are three students studying for the Master Computer Science at the Technical University Delft, The Netherlands. Our names can be found in the README.md file in our github project.