On the first order optimization methods in Deep Image Prior

Pasquale Cascarano, Andrea Sebastiani
Department of Mathematics,
University of Bologna, Italy
{pasquale.cascarano2, andrea.sebastiani3}@unibo.it
Giorgia Franchini, Federica Porta
Department of Physics, Informatics and Mathematics
University of Modena and Reggio Emilia
Modena, Italy
{giorgia.franchini, federica.porta}@unimore.it

August 26, 2022

Abstract

Deep learning methods have state-of-the-art performances in many image restoration tasks. Their effectiveness is mostly related to the size of the dataset used for the training. Deep Image Prior (DIP) is an energy function framework which eliminates the dependency on the training set, by considering the structure of a neural network as an handcrafted prior offering high impedance to noise and low impedance to signal. In this paper, we analyze and compare the use of different optimization schemes inside the DIP framework for the denoising task.

Keywords: First order stochastic methods, Deep Image Prior, Convolutional Neural Network, Green Artificial Intelligence

1 Introduction

The acquisition process of a digital image usually leads to a corrupted data. The field of Image Restoration (IR) aims at recovering the unknown well-looking image from its observation. Mathematically, generic IR tasks are often recasted as the following linear inverse problem:

$$\text{given } v \in \mathbb{R}^m, \quad \text{find } u \in \mathbb{R}^n \quad \text{s.t.} \quad v = Au + \eta, \tag{1}$$

where $v \in \mathbb{R}^m$ is the observation, $A : \mathbb{R}^n \to \mathbb{R}^m$ is a linear forward operator defining the specific IR task and the vector $\eta \in \mathbb{R}^m$ is the noise corrupting the data. In particular we here assume η is an Additive White Gaussian Noise

(AWGN) component. A well-known drawback of imaging inverse problems is usually referred as ill-posedness, which makes not feasible the degradation inversion. Nevertheless, the regularization theory provide a wide range of successful methods which recast imaging inverse problems as well-posed energy-function minimization problems [1]. The keystone to succeed in the recovery process is the choice of the regularization which constrains the outcome to satisfy some prior information and, mathematically, attempts to stabilize the inversion process. This broad field is still capturing the attention of many researchers and a great effort is devoted to inspect several challenging aspects, ranging from the definition of suitable priors, to the development of optimization strategies for minimizing the obtained real-valued objective.

Nowadays, the availability of large amount of data has led to the development of data-driven methods [2] which define very effective priors reflecting the complexity of images. Among them we mention deep learning based methods representing the state-of-the-art in many IR tasks. The outstanding performances of these methods are mostly related to high parametrized models, such as Convolutional Neural Networks (CNNs) and their capability to manage very large datasets. Given several original-corrupted image example pairs, one could train a deep neural network to map a degraded image to its well-looking counterpart in a regression framework [3]. Once trained, the architecture can provide a non-explicit, but effective, prior even on images not belonging to the fixed training set. These supervised deep learning approaches are well-known to be data-hungry and this may limit their usage, thus researchers are currently attracted by unsupervised strategies.

The idea that the great performances of deep learning in IR tasks are only due to their capability to learn realistic image priors from data, has been surprisingly overcome. Indeed, nowadays it has been recognized that the ability in reflecting low-level image features cannot be explained only through the learning phase. In this regard, many interesting works have shown that the architecture of a CNN is strikingly capable to specify a structural prior even if it is not trained on an external dataset [4, 5]. More specifically, in [4] the authors introduce the Deep Image Prior (DIP) framework which can solve imaging inverse problems, such as denoising, inpainting and superresolution, without involving any end-to-end training phase. The approach requires an overparameterized CNN, randomly initialized, to be fitted to a single observation through a regularized by early stopping optimization process, whose objective is the likelihood given a specific degraded-image formation model. DIP has been deeply investigated and the researchers have mostly worked on its interpretation [2, 6, 7] and to boost its performances [8, 9, 10, 11, 12].

Motivation and contribution: So far, to the best of our knowledge, no attention has been devoted to the role of the optimizer in this framework and the current literature lacks of a detailed analysis with different optimization methods. A widely used strategy in the DIP framework is the AdaM algorithm [13], see for example [4, 8, 9]. In this paper we study the behaviour of DIP when different well-known first order optimizers are used. In particular we aim at investigating how the optimization methods employed to face the DIP minimization problem can affect the numerical results and if high impedance to noise and low impedance to signal properties are related to the choice of the optimizer.

2 Method

The widely used variational approach reformulates (1) as an unconstrained minimization problem whose objective is a weighted compromise and reads as below:

$$u^* \in \underset{u}{\operatorname{argmin}} \frac{1}{2}\|Au - v\|_2^2 + \lambda\phi(u). \tag{2}$$

The first L_2-norm based term usually reflects the consistency with the observation v whereas the second term $\phi : \mathbb{R}^n \to \mathbb{R}$ enforces prior information on the estimation u^* and is usually referred as regularization term. The nonnegative scalar λ is a trade-off parameter which balances the contribution of the two terms. The choice of the regularization term is essential to obtain a more reliable estimation u^* of the unknown u, however standard handcrafted terms do not well reflect image statistics.

Therefore, in the last decades, learning-based approaches have become popular. The methods belonging to this class make use of a training set $\mathcal{M} = \{(v^i, u^i) \mid i = 1, \ldots, M\}$, of degraded-clean example pairs, to find a function which mimics the inversion process. Mathematically, as in a regression framework, a function f_θ parametrized by the weights $\theta \in \Theta$, also called Deep Neural Network (DNN) architecture, is fixed as base model. Then, the following optimization problem is solved:

$$\theta^* \in \underset{\theta \in \Theta}{\operatorname{argmin}} \mathcal{T}(f_\theta, \mathcal{M}) := \sum_{i=1}^{M} \|f_\theta(v^i) - u^i\|_2^2, \tag{3}$$

where $\mathcal{T}$ is fixed loss function measuring the discrepancy between $f_\theta(v^i)$, that is the network applied to the degraded image v^i, and the clean image u^i.

This approach leads to state-of-the-art methods for many image restoration tasks. However, in some applications, building a training set $\mathcal{M}$ is expensive and impracticable for physical reasons. In those contexts, it is preferable to use the unsupervised learning approach which uses high-parametrized models as DNNs without relying on an external training set. The Deep Image Prior (DIP) framework, proposed by Ulyanov et al. in [4], is one of the most effective unsupervised approaches for image restoration tasks. This method tries to reconstruct the images, by avoiding the regularization term and substituting $u = f_\theta(z)$ in (2), where z is an input random variable and f_θ is a fixed generative CNN parametrized by its weights θ. The DIP framework is a combination of the following optimization problem:

$$\underset{\theta}{\operatorname{argmin}} \frac{1}{2}\|Af_\theta(z) - v\|_2^2, \tag{4}$$

an optimization scheme applied to (4) and an early-stopping procedure. The objective in (4) shifts the problem of finding a restored image from the image-pixel space to the space of CNN weights. More precisely, the framework looks for θ^*, an iterate of the chosen optimizer, applied to (4), such that the estimation of the desired u is computed as $u^* = f_{\theta^*}(z)$. The early-stopping procedure is required to prevent the optimization process overfits v. As mentioned before, (4) highlights that DIP is independent from a training set, in fact it exploits only the corrupted image v and a fixed CNN structure encoding information about the image statistics. The experiments carried on by the authors in [4], show that the parametrization by CNN weights offers high impedance to noise and low impedance to signal, thus inducing an implicit prior on the solution. In [4], the authors stress the importance of the choice of the CNN architecture, indeed different choices lead to different results. In particular, f_θ is usually defined as an autoencoder, a type of CNN used in unsupervised contexts. Differently from the analysis on the choice of the network architecture, we believe that an analogous study on the selection of the optimizer must be addressed. Furthermore, [4] shows that the noise injection on the input z during the optimization process induces an extra regularization, however the authors do not show the behaviour of DIP fixing the input z. In this paper we analyze how the optimizer choice and the input z definition affect the convergence of DIP to the desired reconstruction. In particular, we mainly focus on the denoising problem and we investigate the semi-convergence behaviour observed in [4, 7]. To improve the readability of the following section, we define $\mathcal{L}(\theta, z) := \frac{1}{2}\|Af_\theta(z) - v\|_2^2$ and by u_k^* the approximate solution at k-th step.

2.1 First order optimization methods

Before introducing proper optimization methods for solving problem (4), we need to specify how the input vector z is treated along the iterative process. Indeed, in [4] the authors allow to consider z as either a fixed randomly-initialized vector or a variable input perturbed at each iteration by an additive normal noise with zero mean and standard deviation σ_p. Numerically, it seems that the last choice is preferable in order to obtain better regularized solutions [14]. From an algorithmic perspective, setting z as a fixed or variable perturbed input leads to take into account different types of optimization methods for facing the DIP problem.

In the first case, the minimization problem (4) is completely deterministic and the more suitable class of algorithms for its handling is that of gradient descent methods whose generic iteration can be written as

$$\theta^{(k+1)} = \theta^{(k)} - \alpha_k \nabla_\theta \mathcal{L}(\theta^{(k)}, z), \tag{5}$$

where $\nabla_\theta \mathcal{L}(\cdot, z)$ is the gradient of the objective function and α_k is a positive parameter. The stationarity of the limit points of the sequence $\{\theta^{(k)}\}_{k\in\mathbb{N}}$ generated by the algorithm (5) is guaranteed under the hypothesis that the objective function is continuously differentiable with Lipschitz continuous gradient and α_k

is properly bounded by such Lipschitz parameter [15, Proposition 1.2.3]. The gradient descent methods are very attractive since they have a relatively simple implementation and a low computational cost per iteration. On the other hand, they can exhibit a poor convergence rate, especially when an high accurate solution is required. However, this drawback can be exploited in order to slow down the semi-convergence behaviour typical of the non regularized problem (4).

When the input z is perturbed by adding a random variable $Z \sim \mathcal{N}(0, \sigma_p^2)$, the DIP optimization problem can be considered as a stochastic optimization problem and rewritten as

$$\underset{\theta}{\operatorname{argmin}}\ \mathbb{E}\left(\frac{1}{2}\|Af_\theta(Z) - v\|_2^2\right) \equiv \underset{\theta}{\operatorname{argmin}}\ \mathbb{E}\big[\mathcal{L}(\theta, Z)\big], \tag{6}$$

where the expected value is taken with respect to the distribution of Z. We suppose to consider a random sample $Z^{[N]} = \{Z_1^{[N]}, \ldots, Z_N^{[N]}\}$ of size N of the variable Z. We remark again that, according to the DIP framework, to avoid overfitting every optimizers solving (6) must be early-stopped. In the following we refer to $\mathcal{L}(\theta^k, Z_{i_k}^{[N]})$ as the running loss of (6) at step k where $Z_{i_k}^{[N]}$ is randomly chosen from the sample data $Z^{[N]}$.

A very popular scheme to face (6) is the stochastic gradient descent (SGD) method, firstly proposed in [16]. Starting from a vector $\theta^{(0)}$, the basic SGD iteration for the minimization of (6) can be written as

$$\theta^{(k+1)} = \theta^{(k)} - \alpha_k \nabla_\theta \mathcal{L}(\theta^k, Z_{i_k}^{[N]}), \tag{7}$$

where α_k is a positive steplength, also known as learning rate. Under the assumption that the gradient of the objective function is L-Lipschitz continuous and some additional conditions on the first and second moments of the stochastic gradient, when α_k is bounded from above by a constant α_{max}, the expected sum of squared gradients of the objective function, corresponding to the SGD iterates, asymptotically converges to a value proportional to α_{max}. In other words, if the learning rate is sufficiently small, the SGD method generates iterates in the neighborhood of the stationary value for $k \to +\infty$. For a complete survey about the convergence properties of the SGD algorithm we refer the reader to [17]. As mentioned before, the parameter α_k affects the performance of the SGD algorithms: on one hand too small values for α_k leads to a very slow convergence rate, on the other hand too big ones do not ensure the convergence of the scheme. A good tuning of this parameter is a drawback of the SGD approach. To minimize (6), there exist other successful stochastic algorithms which are both not so dependent from a good choice of the learning rate and, in general, exhibit a faster convergence rate with respect to SGD. In particular we consider the Adaptive Moment estimation method [13], known as Adam, and the stochastic gradient descent method with momentum [18] which, hereafter, will be denoted by SGDmom. For problem (6), given $\alpha > 0$, $\hat{\epsilon}, \beta_1, \beta_2 \in (0, 1]$, $\theta^{(0)}$ and setting $m_0 = 0$, $v_0 = 0$, the Adam k-th iteration is provided by the following recurrence formulas:

$$
\begin{aligned}
m_k &= \beta_1 m_{k-1} + (1-\beta_1)\nabla_\theta \mathcal{L}(\theta, Z_{i_k}^{[N]}) \\
v_k &= \beta_2 v_{k-1} + (1-\beta_2)\left(\nabla_\theta \mathcal{L}(\theta, Z_{i_k}^{[N]})\right)^2 \\
\alpha_k &= \alpha \frac{\sqrt{1-\beta_2^k}}{1-\beta_1^k} \\
\theta^{(k+1)} &= \theta^{(k)} - \alpha_k \frac{m_k}{\sqrt{v_k}+\hat{\epsilon}}
\end{aligned}
\tag{8}
$$

where the gradient squaring is to be intended element-wise, while, given $\alpha > 0$, $\beta \in (0, 1]$, $x^{(0)}$ and setting $m_0 = 0$, the SGDmom k-th iteration consists in:

$$
\begin{aligned}
m_k &= \beta m_{k-1} + \nabla_\theta \mathcal{L}(\theta, Z_{i_k}^{[N]}) \\
\theta^{(k+1)} &= \theta^{(k)} - \alpha m_k
\end{aligned}
\tag{9}
$$

where, in both cases, $Z_{i_k}^{[N]}$ is randomly chosen from the sample data $Z^{[N]}$. The Adam algorithm is the reference algorithm for the solution of the optimization problem arising in the DIP framework, as reported by the authors in [4]. Moreover, such approach has been employed with both a fixed input z and a perturbed one. However we believe that a greater care in the choice of the optimization algorithm and its parameters is crucial in order to maximize the performance of the DIP methodology. To the best of our knowledge, a deep analysis on how the method employed to face both the minimization problems (4) and (6) can affect the DIP results is new. The results reported in the section devoted to the experiments show that the selection of a proper optimizer is not a secondary task and the Adam algorithm, classically employed, could not be the preferable choice.

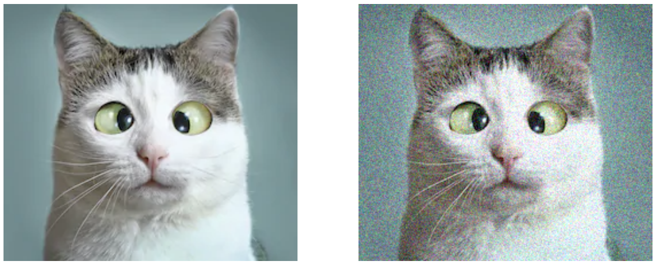
3 Experiments

We perform a comparative analysis among the different first-order algorithms described in Section 2.1 to tackle the optimization problems (4) and (6) with the aim of better understanding how a good reconstructed image is dependent on a suitable choice of the optimization method employed. We simulate a denoising problem, that is we define A equals to the identity operator in (4) and (6). Our experiments have been executed on different images, but here we report only the results obtained on the RGB image reported in Figure 1a for the sake of brevity. This image contains different important features that have to be preserved in the reconstruction tasks and we corrupt it by adding AWGN with standard deviation 20. The noisy version we use in the following experiments is depicted in Figure 1b. Regarding the choice of the CNN architecture, in this work we fix it as one of the networks proposed in [4], that is an autoencoder with five downsampling and five bilinear upsampling layers with convolutional skip connections.

3.1 Fixed input z

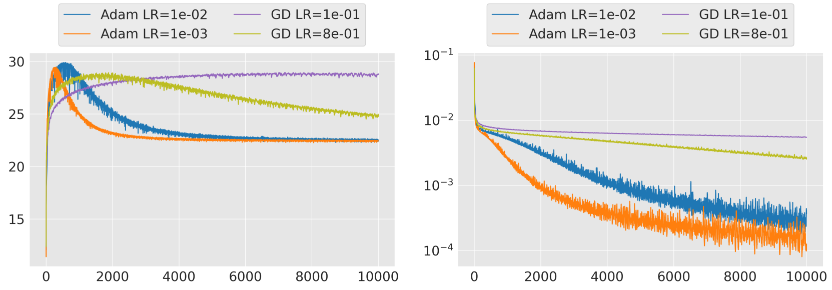
When the input z is fixed along the iterative procedure, the problem to minimize has the form (4). We investigate the behaviour of the GD method (5) compared with the Adam approach, even if we believe that this setting is not suitable for the employment of a stochastic algorithm. To perform the GD and Adam iterations we use the built-in optimizers in Pytorch. Dealing with deep neural networks, the estimation of the gradient Lipschitz constant is difficult, therefore we consider different constant values ($\alpha_k = \bar{\alpha}$) for the learning rate in the GD scheme. In order to understand a potential dependency of the Adam algorithm on the choice of the learning rate, also in this case, we take into account different learning rate values. The considered values of the learning rate are reported in each figure legend by using LR abbreviation. In Figure 2 we report the values of the PSNR (left panel) and the objective function (right panel) attained by the methods along the iterations.

Some considerations can be carried out from the curves plotted in Figure 2. First of all, we notice a very unstable behaviour of the Adam method: even if it first reaches the best PNSR, huge oscillations are present around this value. Moreover, from a pure optimization perspective, the Adam method is the fastest one as depicted in Figure 2b. However, such fast convergence rate leads to a considerable semi-convergence effect already in the early stage of the iterative process. As a consequence of the instability and the significant semi-convergence behaviour, it seems very hard to properly stop the Adam algorithm at a good reconstruction. On the other hand, the greater stability of the GD scheme allows to implement more reliable early stopping criteria. In particular, to early-stop the process we look at s consecutive values of the objective function along the iterations and we arrest the iterative process when the objective function does not decrease for more than s iterations. The rule has been implemented by using the patience option in Pytorch. In Table 1 we report the PSNR attained using two values 50 and 100 for the patience option both for GD (LR$= 0.1$) and Adam (LR$= 0.01$). It is evident that GD is more stable with respect to the choice of window size. We remark that the obtained results state that by fixing z along the iterations does not provide noise-free solutions, therefore a further


(a) Ground truth

(b) Noisy image

Figure 1: Ground truth and corrupted image used in the experiments.



(a) PSNR(u, u_k^*) (b) $\mathcal{L}(\theta, z)$

Figure 2: PSNR and objective function behaviour along the iterations of Adam and GD methods by considering a fixed input z.

regularization must be added.

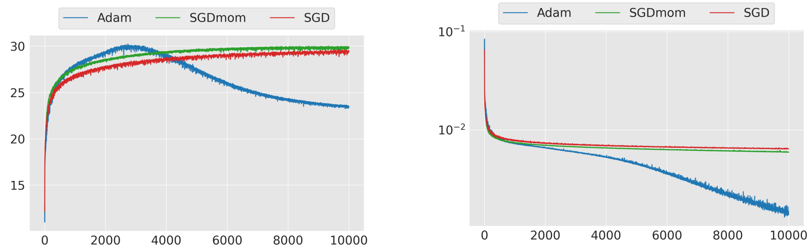
3.2 Perturbed input z

In dealing with problem (6) we consider the Adam, SGD and the SGDmom algorithms with different values of the learning rate. In Figure 3, the values of the PSNR (left) and running loss (right) generated by the considered methods at each iteration k are reported. From these figures, it is clear that the performance of these approaches are comparable in terms of the best PSNR reached, even if Adam needs a lower number of iterations than SGDmom and SGD. However, the Adam algorithm suffers from a more significant semi-convergence effect with respect to the other two schemes. Indeed SGDmom and SGD, regardless of the value assigned to the learning rate, present a stable trend when the best PNSR value is reached. Moreover, SGDmom seems to be the best compromise between Adam and SGD. We believe that the SGDmom method implicitly impede the fit of the noise.

In order to validate this thesis, we compare Adam and SGDmom in solving (6) with different target images v: 1) the natural image, depicted in Figure 1a, 2) the same image plus additive noise, proposed in Figure 1b, 3) a realization of white noise with standard deviation 20. In Figure 4 we report the running loss curves obtained by the two algorithms, and in the top right corner the images used as target. As regards the natural target (Figure 4a), the con-

	s =50		s =100	
Method	PSNR	itr	PSNR	itr
GD	27.18	997	27.78	1675
Adam	25.60	1787	23.24	3669

Table 1: PSNR and number of iterations performed by considering the patience stopping rule with GD (LR= 0.1) and Adam (LR= 0.01)



(a) PSNR(u, u_k^*)

(b) $\mathcal{L}(\theta^k, Z_{i_k}^{[N]})$

Figure 3: PSNR and running loss curves for the denoising task considering a random input at each iteration. Adam (LR= 0.01), SGDmom (LR= 0.1) and SGD (LR= 0.5)

vergence rate of Adam is faster than the one of SGDmom, which is consistent with the well-known faster behaviour of Adam with respect to other stochastic approaches. However, the running loss curve corresponding to the SGDmom is still decreasing. From Figures 4b and 4c it is quite evident that the decrease of the running loss obtained by Adam is constant, whereas the curve generated by the SGDmom is flat after a very limited number of iterations offering high impedance to noise. As a consequence, we believe that the SGDmom method is preferable in solving the DIP optimization problem, since it can guarantee performances comparable to those of Adam and, at the same, prevent from the fitting of the noise affecting the data.

For the sake of comparison, we report in Figure 5 the images obtained at iteration 2600, 4000 and 5400 using the three methods: SGD (LR=0.5), SGDmom (LR=0.1) and Adam (LR=0.01). As expected, the use of a random input at each iteration contributes to the regularization. This is evident from the Figures 5g and 5f. However, the choice of the optimization method is crucial: Adam worsens the image quality after the first thousands of iterations, reintroducing noise into the reconstruction. On the contrary, SGD and SGDmom have a desirable behaviour, in fact the image quality improves going on with the iterations. Analogously to the fixed input setting 4, the stable behaviour of SGD and SGDmom makes them more suitable for early-stopping strategies based on the running loss and Table 2 shows these methods are more robust to the choice of the parameter s in the aforementioned early-stopping criterion.

4 Conclusion

Although the DIP methodology has been proposed only very recently, it gained a large visibility in the literature for solving image restoration problems as a deep learning approach independent from the training of the network. Despite the studies performed on the CNN choice, no attention has been devoted to the selection of the optimization method to be exploited for the numerical solution

	s =50		s =100	
Method	PSNR	itr	PSNR	itr
SGD	28.86	1052	28.13	2925
SGDmom	27.69	1088	28.99	2838
Adam	29.42	2095	24.70	7076

Table 2: PSNR and number of iterations performed by considering the patience stopping rule with SGD (LR= 0.5), SGDmom (LR= 0.1) and Adam (LR= 0.01)

of the arising DIP minimization problem. In this paper we deeply investigate the class of standard first order algorithms. We empirically observed that a proper optimizer, combined with a noise injection during the training, is important to regularize the reconstruction. Furthermore, we propose an early-stopping criterion on the running loss which exploits the stability induced by the optimizer. In conclusion, we observe that the Adam scheme (always preferred for the DIP optimization task) is not always the right choice in terms of high impedance to noise and low impedance to signal properties.

acknowledgment

This work was also supported by the “Gruppo Nazionale per il Calcolo Scientifico (GNCS-INdAM)”.

The publication was created with the co-financing of the European Union-FSE-REACT-EU, PON Research and Innovation 2014-2020 DM1062/2021.

References

[1] Scherzer, O., Grasmair, M., Grossauer, H., Haltmeier, M., and Lenzen, F., 2009, “Variational methods in imaging,”.

[2] Arridge, S., Maass, P., Öktem, O., and Schönlieb, C.-B., 2019, “Solving inverse problems using data-driven models,” *Acta Numerica,* **28**, pp. 1–174.

[3] Jin, K. H., McCann, M. T., Froustey, E., and Unser, M., 2017, “Deep convolutional neural network for inverse problems in imaging,” *IEEE Transactions on Image Processing,* **26**(9), pp. 4509–4522.

[4] Ulyanov, D., Vedaldi, A., and Lempitsky, V., 2018, “Deep image prior,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 9446–9454.

[5] Heckel, R., and Hand, P., 2018, “Deep decoder: Concise image representations from untrained non-convolutional networks,” *arXiv preprint arXiv:1810.03982.*

[6] Dittmer, S., Kluth, T., Maass, P., and Baguer, D. O., 2020, “Regularization by architecture: A deep prior approach for inverse problems,” *Journal of Mathematical Imaging and Vision,* **62**(3), pp. 456–470.

[7] Cheng, Z., Gadelha, M., Maji, S., and Sheldon, D., 2019, “A bayesian perspective on the deep image prior,” In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5443–5451.

[8] Sagel, A., Roumy, A., and Guillemot, C., 2020, “Sub-dip: Optimization on a subspace with deep image prior regularization and application to superresolution,” In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, pp. 2513–2517.

[9] Mataev, G., Milanfar, P., and Elad, M., 2019, “Deepred: Deep image prior powered by red,” In Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 0–0.

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

[11] Cascarano, P., Sebastiani, A., Comes, M. C., Franchini, G., and Porta, F., 2021, “Combining weighted total variation and deep image prior for natural and medical image restoration via admm,” In 2021 21st International Conference on Computational Science and Its Applications (ICCSA), pp. 39–46.

[12] Cascarano, P., Comes, M. C., Mencattini, A., Parrini, M. C., Piccolomini, E. L., and Martinelli, E., 2021, “Recursive deep prior video: A super resolution algorithm for time-lapse microscopy of organ-on-chip experiments,” *Medical Image Analysis,* **72**, p. 102124.

[13] Kingma, D., and Ba, J., 2017, “Adam: A method for stochastic optimization,” *arXiv: 1412.6980 [cs.LG].*

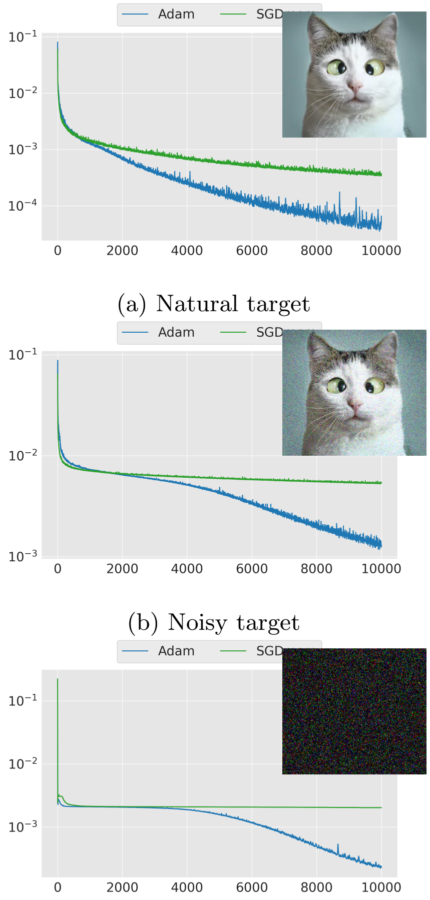
[14] Bishop, C. M., 1995, “Training with noise is equivalent to tikhonov regularization,” *Neural computation,* **7**(1), pp. 108–116.

[15] Bertsekas, D. P., 1999, *Nonlinear Programming*, 2nd ed. Athena Scientific.

[16] Robbins, H., and Monro, S., 1951, “A stochastic approximation method,” *The Annals of Mathematical Statistics,* **22**(3), pp. 400–407.

[17] Bottou, L., Curtis, F., and Nocedal, J., 2018, “Optimization methods for large-scale machine learning,” *SIAM Review,* **60**(2), pp. 223–311.

[18] Loizou, N., and Richtarik, P., 2018, “Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods,” *arXiv:1712:09677v2.*

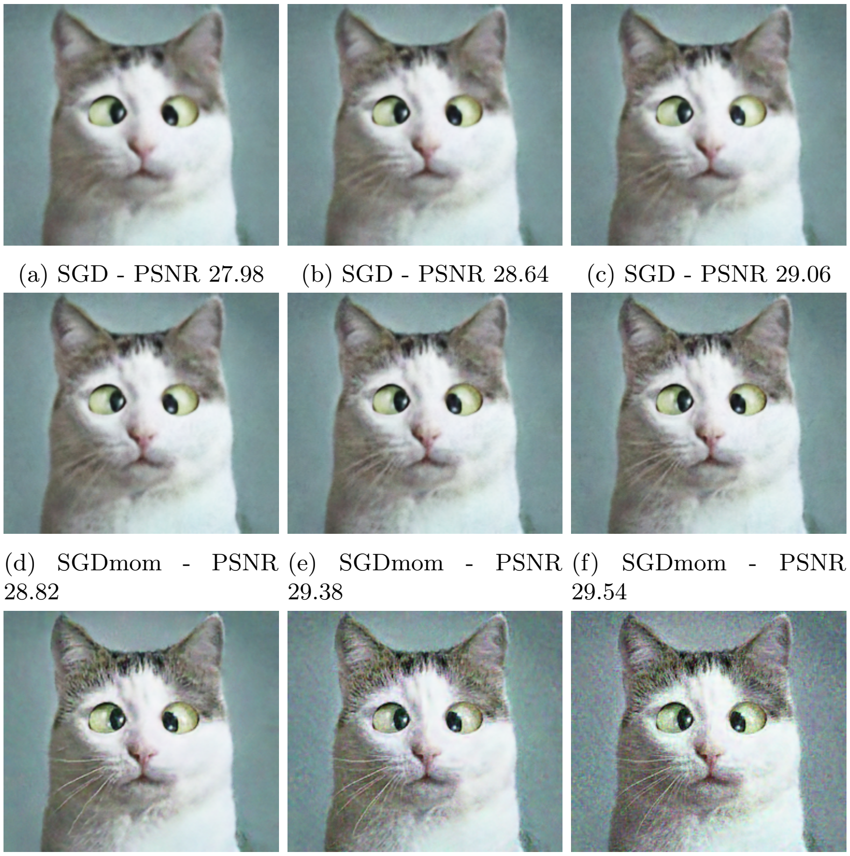


(a) Natural target

(b) Noisy target

(c) Only noise target

Figure 4: Running loss curves for the reconstruction tasks considering different optimizers. The target considered is described in the subpcation. Adam (LR= 0.01) and SGDmom (LR= 0.8)



(a) SGD - PSNR 27.98 (b) SGD - PSNR 28.64 (c) SGD - PSNR 29.06

(d) SGDmom - PSNR 28.82 (e) SGDmom - PSNR 29.38 (f) SGDmom - PSNR 29.54

(g) Adam - PSNR 29.22 (h) Adam - PSNR 28.86 (i) Adam - PSNR 26.80

Figure 5: Images obtained at a fixed number of iterations, for different methods. First column iteration 2600, second 4000 and third 5400. First row SGD (LR=0.5), second row SGDmom (LR=0.1) and last row Adam (LR=0.01). In the caption the PSNR of each reconstruction