



**HAL**  
open science

# Learning local regularization for variational image restoration

Jean Prost, Antoine Houdard, Andrés Almansa, Nicolas Papadakis

► **To cite this version:**

Jean Prost, Antoine Houdard, Andrés Almansa, Nicolas Papadakis. Learning local regularization for variational image restoration. International Conference on Scale Space and Variational Methods in Computer Vision (SSVM'21), May 2021, Cabourg, France. pp.358-370, 10.1007/978-3-030-75549-2\_29. hal-03139784

**HAL Id: hal-03139784**

**<https://hal.science/hal-03139784v1>**

Submitted on 12 Feb 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Learning local regularization for variational image restoration

Jean Prost<sup>1</sup>, Antoine Houdard<sup>1</sup>, Andrés Almansa<sup>2</sup>, and Nicolas Papadakis<sup>1</sup>

<sup>1</sup> Univ. Bordeaux, Bordeaux INP, CNRS, IMB, UMR 5251,F-33400 Talence, France  
jean.prost@math.u-bordeaux.fr

<sup>2</sup> Université de Paris, MAP5, CNRS, F-75006 Paris, France

**Abstract.** In this work, we propose a framework to learn a local regularization model for solving general image restoration problems. This regularizer is defined with a fully convolutional neural network that sees the image through a receptive field corresponding to small image patches. The regularizer is then learned as a *critic* between unpaired distributions of clean and degraded patches using a Wasserstein generative adversarial networks based energy. This yields a regularization function that can be incorporated in any image restoration problem. The efficiency of the framework is finally shown on denoising and deblurring applications.

## 1 Introduction

*Inverse problems and convex regularization.* Many image restoration tasks require to solve an inverse problem. This can be addressed with a variational formulation involving a data-fidelity term and a regularization term encouraging the solution to satisfy given properties or to belong to a space of possible solutions. Some of the most famous regularization terms used for image restoration are convex non-smooth terms like the total variation [20], or  $\ell^1$  minimization of transform-domain coefficients such as Wavelet frames [6,4] or local Fourier or DCT representations [26]. However these strategies tend to produce over-smoothed results, since they represent only a rough approximation of natural image statistics and geometry.

*CNN-based non-convex regularization.* Later-on more accurate natural image priors emerged in the form of non-convex regularization terms, such as patch-based Gaussian mixture models (to be discussed below) or convolutional neural networks (CNN). Most common CNN-based regularizers are, however, trained in a way that the prior or regularizer itself is only partially and implicitly known via its gradient [2,19,18] or proximal operator [24,16,27,12,21]. Such implicit CNN regularizers, and the associated optimization algorithms, lack convergence guarantees or do so under overly restrictive conditions on the regularizer, the regularization parameter or the kind of inverse problems they can solve [18,21].

To overcome these limitations a new breed of explicit CNN-based regularizers have been proposed, either in the form of the push-forward measure of a generative model [3], a variational autoencoder [7], or more directly as a discriminator network [15]. All these approaches are nevertheless limited to a particular class of image and do not generalize to images of arbitrary size.

*Patch-based non-convex regularization.* Learning prior information has also been widely studied from the patch point-of-view. The main idea is to learn the prior knowledge from patches, that are local sub-images of small size, instead of learning a prior from whole images. This allows to avoid the high-dimensional issues faced when working with full-size image distributions. These approaches rely on parametric models of the patch distribution such as Gaussian mixture models [29,10,23]. However such simple models can not accurately represent the complexity of the patch space.

In this work, we introduce an explicit non-convex regularization function encoded with a fully convolutional neural network that acts as a local regularizer. This prior knowledge on the patch distribution can be applied to a whole image without size limitation. We propose (i) to learn the convolutional regularizer as a discriminator between patches using the Wasserstein GAN framework [1] as in [15], and (ii) to integrate this regularizer in patch-based models such as [29].

### 1.1 Setup of the problem

The main goal of this paper is to perform image restoration by solving an inverse problem. That is, finding the underlying true image  $x^*$  from its perturbed observation  $y$  that we consider here to be of the form

$$y = Ax^* + \epsilon, \quad (1)$$

where  $\epsilon \sim \mathcal{N}(0, \sigma^2)$  is a Gaussian white noise and  $A$  is a degradation operator that can typically be the identity (pure denoising), a mask (inpainting) or a blurring kernel (deconvolution). These inverse problems can be addressed with a variational formulation involving a regularization term. This amounts to find an estimate  $\hat{x}$  of  $x^*$  of the form

$$\hat{x} \in \arg \min_x \frac{1}{2\sigma^2} \|Ax - y\|^2 + \lambda R(x), \quad (2)$$

where  $\|Ax - y\|^2$  is the data-fidelity term ensuring that the recovered image  $\hat{x}$  is close enough to the degraded observation  $y$ ,  $R(x)$  is the regularization term and  $\lambda \geq 0$  monitors the influence of both terms. In the case where  $R(x) = -\log(P_X(x)) + C$  is derived from a prior probability distribution  $P_X$  modeling the data  $x$ , then  $\hat{x}$  from (2) corresponds to the *maximum a posteriori* estimator.

The choice of the regularization function  $R$  has a strong impact on the final result. We propose to learn  $R$  through a local regularization functional  $r$  acting on patches. Denoting as  $\Omega_x = \{x_1, \dots, x_n\}$  the set of all patches of size  $p \times p$  from an image  $x$ , this function takes as input an image patch  $x_i$  and outputs a score  $r(x_i)$  that indicates how likely the patch is to be a clean one. As in [29], we define the global regularization functional as the average value of the local scores on the set of all patches of image  $x$ :

$$R(x) = \frac{1}{|\Omega_x|} \sum_{x_i \in \Omega_x} r(x_i). \quad (3)$$

Working with patches yields three main advantages. It first makes the learning phase simpler, as a patch model contains far less parameters than a full image model. Next, the number of images required for training is reduced, as a single image already provides several thousands of patches. Finally, contrary to [15], the regularization function can be applied on images of any size.

In practice, we consider  $r$  as a CNN with perceptual size equal to the patch size  $p \times p$  and taking values in  $\mathbf{R}$ . This representation is more general than Gaussian mixture models, and allows to encode complex distributions.

## 1.2 Contributions and outline.

We propose an image restoration method that relies on a regularization function learned on patches and applied to any image size. It gathers the advantages of previous CNN methods while avoiding the constraints of implicit plug & play priors (convergence guarantee) and of GAN or VAE priors (image size).

In addition, the regularization function is learned in an *unsupervised* manner, in the sense that it only relies on patch distributions of clean and degraded data and **it does not require paired data**. We can therefore deal with an unknown degradation model if a noisy dataset is available.

The organization of the paper is as follows. In Section 2, we propose an unsupervised framework for the learning of a compact convolutional neural network modeling the local patch regularity prior. We namely obtain the local regularization functional  $r$  as a critic trained to distinguish noisy patches from clean ones using the framework of Wasserstein generative adversarial models [1]. In section 3, we provide implementation details to make the work fully reproducible. We show in section 4 that the local functional  $r$  generalizes well to arbitrary levels of noise, i.e. noise level unseen during training. In Section 5, we demonstrate that the proposed framework is efficient for image denoising and deblurring.

## 2 Local regularization for image inverse problem

In this section we define our local image regularizer  $r_\theta$  as a convolutional neural network and we describe how we use and train it.

Patch-based methods have shown to be efficient tools for solving inverse problems in imaging [29]. Hence we aim at defining a regularization function  $r_\theta$  depending on parameters  $\theta \in \Theta$  that encode prior knowledge at a patch level. In the patch-based literature, such regularizers rely on statistical modeling of the distribution of clean patches and the model parameters are usually inferred with a maximum likelihood estimation [10]. This leads to two main limitations. First, it requires to have access to the probability density function of the prior distribution and consequently it does not properly represent the intrinsic low dimensional manifold of clean patches. Second, maximizing the likelihood of a complex model leads to non-convex problems that are difficult to solve in practice.

In order to tackle these issues, we propose to take advantage of having two data sets of clean and degraded patches –not necessarily paired– and consider  $r_\theta$  as a *critic* that tells us if a patch is more likely to be clean or degraded.

We first detail in section 2.1 how the local regularization function is integrated as a global regularizer on images in order to solve the variational problem (2). In section 2.2, we present the framework to learn the regularizer as a *critic* between two unpaired dataset of clean and degraded images.

## 2.1 Convolutional regularizers for variational problems

We define, for the variational problem (2), a regularization term  $R$  that takes into account local prior knowledge of the images. To do so, we propose to consider a class of functions  $r_\theta$  defined with a fully convolutional neural network with parameter  $\theta \in \Theta$ . We enforce the perceptual size of this network to be the patch size  $p \times p$ . That is, the successive convolutions operate on a window no larger than  $p \times p$  pixels. Using this architecture permits to compute the global regularizer  $R$  from (3) by directly applying  $r_\theta$  to the full image  $x$  and average the outputs. Once learned the local regularizer  $r_\theta^*$ , the variational problem to solve becomes

$$\min_x \frac{1}{2\sigma^2} \|Ax - y\|^2 + \frac{\lambda}{|\Omega_x|} \sum_i r_\theta^*(x_i). \quad (4)$$

We propose to find a local minimizer of (4) by performing an explicit gradient descent method. Let  $x^\ell$  the image at iteration  $\ell$ , a gradient step of step size  $\eta$  writes

$$x^{\ell+1} = x^\ell - \frac{\eta}{\sigma^2} A^*(Ax^\ell - y) - \frac{\eta\lambda}{|\Omega_x|} \sum_i \nabla r_\theta^*(x_i^\ell), \quad (5)$$

where  $A^*$  is the adjoint operator of  $A$ . Contrary to plug & play methods that rely on implicit schemes [24], this explicit scheme converges for differentiable regularization functions and adequate time steps.

We now describe how the framework for learning the local regularization function.

## 2.2 Adversarial Local Regularizer (ALR)

In order to train  $r_\theta$  as a critic between patch distributions, we consider the discriminator framework introduced for generative adversarial networks [8], without the generator network. Such approach nevertheless results in a critic  $r_\theta$  approximating the hard clustering between clean and degraded patches. It therefore induces steep gradients  $\nabla r_\theta$  that may lead to numerical instabilities during the minimization of problem (2).

As a consequence, we rather rely on the Wasserstein GAN [1] formulation that amounts to approximate the optimal transport cost between the distribution of clean patches  $\mathbb{P}_c$ , and a distribution of degraded patches  $\mathbb{P}_n$ . Relying on the dual formulation of the optimal transport [22], an optimal critic  $r_\theta^*$  is seen as a Kantorovitch potential and shall satisfy

$$r_\theta^* \in \arg \max_{\varphi \in \text{Lip}_1} \mathbb{E}_{z \sim \mathbb{P}_n} [\varphi(z)] - \mathbb{E}_{z \sim \mathbb{P}_c} [\varphi(z)]. \quad (6)$$

Under the assumption that the support of the clean patches distribution  $\mathcal{M}$  is compact [15], the solution of equation (6) corresponds to the distance function to the clean data manifold  $\mathcal{M}$ . Each iteration of the gradient descent on equation (2) thus brings our noisy data closer to the clean data.

In practice, imposing a neural network to be 1-Lipschitz is a difficult task and we therefore use the formulation proposed in [9] that encourages the gradient norm to be close to 1. This amounts to maximize the following quantity

$$D(\theta) = \mathbb{E}_{z \sim \mathbb{P}_n} [r_\theta(z)] - \mathbb{E}_{z \sim \mathbb{P}_c} [r_\theta(z)] - \mu \mathbb{E}_{z \sim \mathbb{P}_i} [(\|\nabla_z r_\theta(z)\|_2 - 1)^2] \quad (7)$$

where  $\mathbb{P}_i$  is the distribution of all lines connecting samples in  $\mathbb{P}_n$  and  $\mathbb{P}_c$ . In other words, the last term of (7) is a gradient penalty that makes the function 1-Lipschitz on the convex hull of the union of the support of  $\mathbb{P}_c$  and  $\mathbb{P}_n$ . By enforcing the gradient  $\nabla r_\theta$  to be of norm close to 1, vanishing gradient issues are also avoided when solving problem (2) with gradient descent approaches.

We illustrate the properties of the regularization functional with a synthetic example in Figure 1 containing random perturbations of clean data points located on a circle. The learned regularization function  $r_\theta(z)$  therefore approximates the distance function to the circle. The gradient  $\nabla r_\theta(z)$  thus indicates the direction to follow in order to transport  $z$  towards the a clean point within the circle.

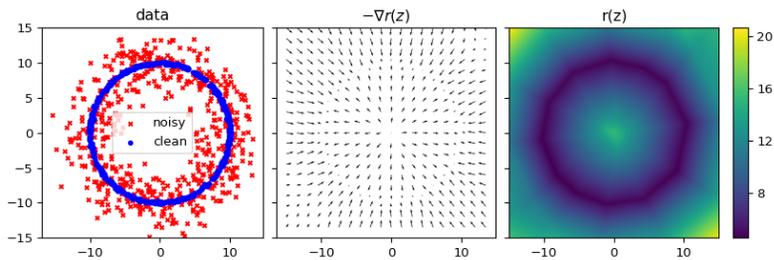


Fig. 1: Regularization functional  $r(z)$  learned on a synthetic 2d denoising problem with clean data (blue dots) on the circle and noisy ones (red crosses). The gradient penalty ensures that the gradient  $\nabla r$  is not flat close to the data manifold  $\mathcal{M}$ .

### 3 Practical considerations for image restoration

In this section, we provide implementation details to reproduce the proposed framework. After presenting the architecture of the regularization network  $r_\theta$ , we explain the training strategy and describe how image restoration is performed.

#### 3.1 Network architecture

The local regularization functional  $r_\theta$  is designed as a 6 layers convolutional network. Each layer is made of  $3 \times 3$  convolution operations followed by ReLU

activations [17]. This network has therefore a  $15 \times 15$  receptive field. No padding is used. Hence, when a patch of the size of the network receptive field is fed to the network  $r_\theta$ , the output is a scalar.

### 3.2 Training the regularization functional

The proposed regularization network is trained with patches matching the size of the receptive field of the network. We create the dataset  $\mathcal{D}_c$  of clean patches by extracting all  $15 \times 15$  patches from a 30000 image subset of the google landmarks dataset [25]. Similarly, we create the dataset  $\mathcal{D}_n$  of noisy patches by extracting all  $15 \times 15$  patches from another 30000 images subset of the landmarks dataset, to which we added an additive white Gaussian noise with standard deviation  $\sigma_{train}$ . Following [15], the local regularization network  $r_\theta$  is trained to minimize the criterion (7) with Algorithm 1. We use the Adam optimizer [13] with hyperparameters  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ , and an exponential learning rate decay, so that the learning rate  $\alpha$  begins at a value of  $10^{-3}$  for the first iteration, and ends up at  $10^{-4}$  for the last iteration. We use a batch size of  $m = 32$  and train the network for  $K = 10^5$  iterations. The gradient-penalty parameter is set to  $\mu = 5$ .

Training samples of clean and noisy patches  $z$ , with their final regularizer value  $r_\theta(z) \in \mathbb{R}$ , are shown in Figure 2. As can be observed from the functional values, there exists a slight ambiguity between texture patches ( $r_\theta(z) = -0.23$  for the last patch of top row) and noisy homogeneous patches ( $r_\theta(z) = -0.27$  for the first patches of bottom row). We nevertheless show in Figure 3 that the distributions of clean and noisy patches are globally well separated.

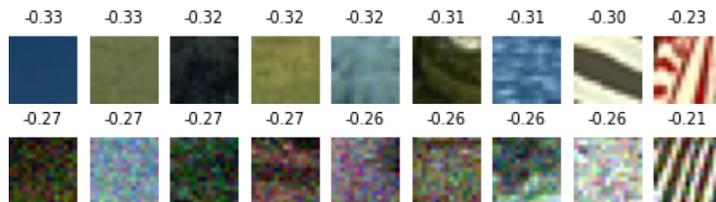


Fig. 2: Value of the local regularization functional  $r_\theta$  trained with  $\sigma_{train} = 0.1$  on clean (top row) and noisy (bottom row) patches ( $\sigma = 0.1$ ).

### 3.3 Solving the variational problem

Image restoration is realized by solving the variational problem (2). To do so, we search for the minimizing image  $x$  by performing 50 iterations of Adam [13], with the momentum parameter set to the default values  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ , and an exponential learning rate decay, with an initial learning rate of 0.1 and a final learning rate of 0.01 at the last iteration. We implement the method with the pytorch deep learning framework, so that the gradient of the global regularization functional  $R(x)$  can be easily computed using automatic differentiation.

---

**Algorithm 1** Learning the local regularization  $r_\theta$ 

---

**Input:** Datasets  $\mathcal{D}_c$  of clean patches and  $\mathcal{D}_n$  of noisy patches; gradient penalty  $\mu$ ; batch size  $m$ , number of iterations  $K$   
**Output:** regularization function  $r_\theta$   
**for**  $k = 1$  **to**  $K$  **do**  
    Sample minibatches of  $m$  clean patches  $\{z_j^c\}_{j=1}^m$  from  $\mathcal{D}_c$  and  $m$  noisy patches  $\{z_j^n\}_{j=1}^m$  from  $\mathcal{D}_n$  and a random number  $\alpha \in [0; 1]$   
    Define interpolated patches  $z_j^i = \alpha z_j^c + (1 - \alpha)z_j^n$   
    **for**  $j = 1$  **to**  $m$  **do**  
         $D_j(\theta) = r_\theta(z_j^n) - r_\theta(z_j^c) - \mu(\|\nabla_z r_\theta(z_j^i)\|_2 - 1)^2$   
    **end for**  
     $\theta \leftarrow Adam(\nabla_\theta \sum_{j=1}^m D_j(\theta))$   
**end for**

---

## 4 Generalization to unseen noise level

In this section, we study the robustness of the proposed regularization function to noise variations. The adversarial training of the regularization function, presented in the previous sections, requires to learn a different regularization function for every different noise level  $\sigma$ . We show how this limitation can be overcome.

We first analyze the behaviour of regularization functions trained on a single noise level  $\sigma_{train}$  and then used to denoise an image with a different noise level  $\sigma_{img}$ . Second, we propose to train the regularization functions with varying noise levels and demonstrate experimentally the superiority of this approach.

### 4.1 Robustness to unseen noise level

To study the ability of the local regularization function to generalize to noise levels unseen during training, we train 4 regularization functions on 4 different noise levels  $\sigma_{train} \in \{0.05, 0.1, 0.2, 0.4\}$ . We then evaluate the quality of the regularization of those networks on denoising tasks, for 5 different noise levels  $\sigma_{img} \in \{0.05, 0.1, 0.2, 0.3, 0.4\}$ . The 4 networks share the same architecture and the same training procedure as described in section 3.

While these regularizers have only been trained to distinguish between clean patches and noisy patches for a particular noise level, they generalize well to intermediate noise levels, in the sense that the regularizer value is an increasing function of the noise level of its input patch. Figure 3 illustrates this point for the noise level  $\sigma_{train} = 0.1$ . The overlap between the distribution for noise level  $\sigma = 0$  (top) and  $\sigma = 0.1$  (bottom) is small, showing the ability of the regularizer network to distinguish clean and noisy patches. Furthermore, the distribution for noise level  $\sigma = 0.05$  is located in between the distributions  $\sigma = 0$  and  $\sigma = 0.1$ , showing the ability of the network to generalize to intermediate noise levels but also to extrapolate to the noise level 0.15.

Next, we evaluate denoising quality by measuring the average PSNR on a validation set of 11 images. To that end, we solve problem (2) for  $A = \text{Id}$ . We denoise images with 5 noise levels  $\sigma_{img} \in \{0.05, 0.1, 0.2, 0.3, 0.4\}$ , and we respectively set the regularization parameter  $\lambda$  to  $\{0.15, 0.35, 0.6, 0.8, 1\}$ . The results displayed on table 1 demonstrate that the trained regularization functions generalize well to unseen noise level, as for all 5 levels of noise  $\sigma_{img}$ , the 4 regularization functions yield average PSNR values that are contained in an interval of size smaller than 1 dB. Furthermore, regularization networks trained on small noise levels  $\sigma_{train} \in \{0.05, 0.1\}$  generalize well to higher noise levels  $\sigma_{train}$  as they perform even better than networks trained on the specific noise level.

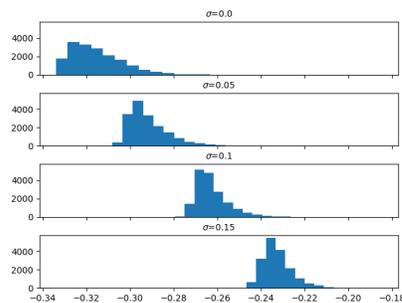


Fig. 3: Distribution of values  $r_\theta(x)$  for a regularizer trained on noise level  $\sigma_{train} = 0.1$ . It generalizes well to patches  $x$  with an intermediate (0.05) or extrapolated (0.15) noise levels.

We suggest that this is due to the fact that, when trained on a small noise level, the regularization function is forced to learn a tight boundary between the clean and the noisy distribution which favors denoising performance. However, for the highest noise level  $\sigma_{img} = 0.4$ , the regularization function trained on a small noise level  $\sigma_{train} = 0.05$  gives the worst results. As patches with very high noise levels are not seen during the training of the regularization function trained for  $\sigma_{train} = 0.05$ , we suggest that the gradient penalty is not enforced to 1 in this region of the patch space. Thus there is no guarantee that the gradient of the regularization function  $\nabla r_\theta$  is indeed directed towards the space of possible solutions. This prevents the optimization algorithm from finding a relevant local minimum of (2).

## 4.2 Robustness to noise variation during training

We evaluate how robust the regularization function is to noise level variation during training. To do so, we train a regularization function on a distribution containing patches with noise level  $\sigma_{train}$  uniformly sampled in the interval  $[0.05, 0.30]$ . We use the same network architecture and the same training procedure as in section 3.

We evaluate the effectiveness of this regularization function by measuring the average PSNR when this function is used for denoising. We compare the performance with the prior trained on a single noise level in the last row of table 1. Results show that the regularization function trained with a varying noise level has comparable performance with the regularization function trained on a single-noise level. Furthermore, for high noise level, the regularization function trained on a varying noise level significantly outperforms the regularization

Table 1: Average PSNR on AWGN denoising, in function of the image noise level  $\sigma_{img}$ , and the noise level the regularization network was trained on  $\sigma_{train}$ . For each image noise level, best result is displayed in bold, and second best result is underlined. Regularization networks trained on small noise level  $\sigma_{train}$  generalizes well to higher noise levels  $\sigma_{img}$ . The regularization network trained on varying level of noise  $\sigma_{train} \in [0.05; 0.3]$  performs better on high noise levels  $\sigma_{img}$ .

$\sigma_{train} \backslash \sigma_{img}$	0.05	0.1	0.2	0.3	0.4
0.05	<b>33.24</b>	<b>28.94</b>	<u>24.21</u>	<u>21.25</u>	18.91
0.1	<u>33.20</u>	28.82	24.17	<u>21.25</u>	<u>19.13</u>
0.2	32.42	28.23	23.80	21.03	18.96
0.4	33.01	28.23	23.84	21.03	18.96
[0.05; 0.3]	32.92	<u>28.90</u>	<b>24.91</b>	<b>22.58</b>	<b>20.84</b>

function trained on a single-noise level. This illustrates the fact that training the regularization function on varying noise level is actually beneficial.

We suggest that exposing the regularization function to various noise levels during training combines two advantages. It first learns a tight boundary around the clean patches distribution, as the networks trained on low noise levels. Second, the gradient-penalty is enforced even on highly noisy patches, as the networks trained on high noise levels.

## 5 Experiments

We evaluate the effectiveness of our learned regularization functional on two image restoration tasks, image denoising and image deblurring.

### 5.1 Denoising

We evaluate our method on additive white Gaussian noise denoising, which corresponds to solving (2) with  $A = I$ . We compare our method against two common patch-based denoising algorithms, BM3D [5] and EPLL [29], on 3 noise levels  $\sigma_{img} \in \{0.1, 0.2, 0.4\}$ . We use our model trained on varying noise level  $\sigma_{train} \in [0.05, 0.3]$ , with the regularization parameter  $\lambda$  respectively set to 0.15, 0.35 and 1. For BM3D, we use the implementation of [14] with default parameters, and for EPLL we use the implementation of [11] with default parameters and a prior GMM model learned on RGB patches.

The average PSNR and LPIPS [28] on the BSD68 dataset for the 3 methods are presented in Table 2, and examples of denoised images are shown on Figure 4. For the 3 noise levels, the adversarial local regularization denoising outperform EPLL and BM3D in terms of PSNR, while having comparable perceptual quality. This illustrates the ability of convolutional neural networks to be used as local regularizers when trained the right way.

Table 2: Comparisons in terms of PSNR (left) and LPIPS (right) of the patch-based denoising algorithms ALR, EPLL and BM3D, for white Gaussian noise. Results are averaged on the 68 images of the BSD68 dataset.

$\sigma$	PSNR			LPIPS		
	ALR	EPLL	BM3D	ALR	EPLL	BM3D
0.1	<b>28.85</b>	28.77	28.26	0.29	<b>0.28</b>	0.30
0.2	24.88	<b>24.92</b>	24.69	0.44	<b>0.42</b>	0.43
0.4	<b>21.58</b>	19.75	20.25	<b>0.57</b>	0.61	0.58

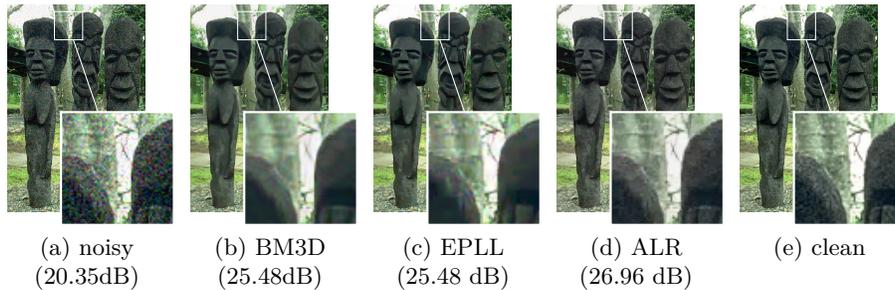


Fig. 4: Visual comparison of patch-based denoising methods for  $\sigma = 0.1$ .

## 5.2 Deblurring

To illustrate the adaptability of our local regularization function, we consider image deblurring. This corresponds to solving (2) with a linear degradation operator  $A$  taken as a convolution operation with a blur kernel  $k$ , that is  $y = k * x + \varepsilon$ . Figure 5 shows an example of image deblurring using our learned local regularization function. The image is blurred with a  $7 \times 7$  Gaussian kernel with standard deviation  $\sigma_k = 3$ , and an additive white Gaussian noise of standard deviation  $\sigma = 0.03$ .

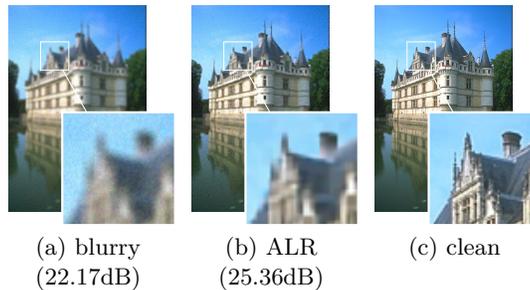


Fig. 5: Illustration of deblurring using a  $7 \times 7$  Gaussian kernel with standard deviation  $\sigma_k = 3$ .

## 6 Conclusion and Perspectives

We propose a new strategy to solve inverse problem in imaging using a convolutional neural network as a local regularization function. The local regularization network is trained to discriminate between clean and noisy patches, and the global regularization function is defined as the average value of the local function over the set of all image patches. Working with a local regularization function offers several advantages : it works with any image size, it requires less training data and has less parameters than a full size model. Furthermore, the fully convolutional architecture of the network makes it computationally efficient to compute the global regularization function and its gradient.

Experimental results on image denoising show that our method outperforms popular patch-based denoising algorithm such as EPLL and BM3D, illustrating the potential of convolutional networks to acts as regularization function for inverse imaging problems.

We believe that improving the training criterion of the regularization function could improves the performance of the regularization. Indeed, the training criterion of our local regularization network corresponds to the 1-Wasserstein distance. The regularizer thus grows linearly with the distance to the clean data manifold, whereas the data-fidelity term is quadratic. We suggest that these unbalanced terms make the variational problem difficult to solve, especially for high noise levels. We believe that learning a regularization term based on the 2-Wasserstein distance could help to overcome this limitation, as the learned regularization function would then grow with the square of the distance to the clean manifold.

*Acknowledgements.* This study has been carried out with financial support from the French Research Agency through the PostProdLEAP project (ANR-19-CE23-0027-01).

## References

1. Arjovsky, M., Chintala, S., Bottou, L.: Wasserstein generative adversarial networks. In: International conference on machine learning. pp. 214–223. PMLR (2017)
2. Bigdeli, S.A., Jin, M., Favaro, P., Zwicker, M.: Deep Mean-Shift Priors for Image Restoration. In: Adv. in Neural Information Proces. Systems 30. pp. 763–772 (2017)
3. Bora, A., Jalal, A., Price, E., Dimakis, A.G.: Compressed sensing using generative models. In: Int. Conference on Machine Learning. vol. 2, pp. 537–546 (2017)
4. Coifman, R.R., Donoho, D.L.: Translation-Invariant De-Noising. In: Wavelets and Statistics (Lect. Notes in Statistics, vol 103), chap. 5, pp. 125–150. Springer (1995)
5. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Trans. on Image Processing* **16**, 2080–95 (2007)
6. Donoho, D.L., Johnstone, J.M.: Ideal spatial adaptation by wavelet shrinkage. *Biometrika* **81**(3), 425–455 (1994)
7. González, M., Almansa, A., Delbracio, M., Musé, P., Tan, P.: Solving Inverse Problems by Joint Posterior Maximization with a VAE Prior (2019)
8. Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. arXiv preprint arXiv:1406.2661 (2014)

9. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.: Improved training of wasserstein gans. In: *Adv. in Neural Information Processing Systems*. vol. 30, pp. 5767–5777 (2017)
10. Houdard, A., Bouveyron, C., Delon, J.: High-dimensional mixture models for unsupervised image denoising (HDMI). *SIAM J. Imag. Sc.* **11**(4), 2815–2846 (2018)
11. Hurault, S., Ehret, T., Arias, P.: EPLL: An Image Denoising Method Using a Gaussian Mixture Model Learned on a Large Set of Patches. *Image Processing On Line* **8**, 465–489 (2018)
12. Kamilov, U.S., Mansour, H., Wohlberg, B.: A plug-and-play priors approach for solving nonlinear imaging inverse problems. *IEEE Signal Processing Letters* **24**(12), 1872–1876 (2017)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: *International Conference on Learning Representations* (2015)
14. Lebrun, M.: An Analysis and Implementation of the BM3D Image Denoising Method. *Image Processing On Line* **2**, 175–213 (2012)
15. Lunz, S., Öktem, O., Schönlieb, C.B.: Adversarial regularizers in inverse problems. In: *Adv. in Neural Information Processing Systems*. vol. 31, pp. 8507–8516 (2018)
16. Meinhardt, T., Moller, M., Hazirbas, C., Cremers, D.: Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In: *International Conference on Computer Vision*. pp. 1781–1790 (2017)
17. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: *International Conference on Machine Learning* (2010)
18. Reehorst, E.T., Schniter, P.: Regularization by Denoising: Clarifications and New Interpretations. *IEEE Trans. on Computational Imaging* **5**(1), 52–67 (mar 2019)
19. Romano, Y., Elad, M., Milanfar, P.: The Little Engine That Could: Regularization by Denoising (RED). *SIAM J. on Imaging Sciences* **10**(4), 1804–1844 (2017)
20. Rudin, L.I., Osher, S., Fatemi, E.: Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena* **60**(1-4), 259–268 (1992)
21. Ryu, E.K., Liu, J., Wang, S., Chen, X., Wang, Z., Yin, W.: Plug-and-play methods provably converge with properly trained denoisers. In: *International Conference on Machine Learning*. pp. 5546–5557 (2019)
22. Santambrogio, F.: Optimal transport for applied mathematicians. *Progress in Nonlinear Differential Equations and their applications* **87** (2015)
23. Teodoro, A.M., Bioucas-Dias, J.M., Figueiredo, M.A.T.: Scene-Adapted Plug-and-Play Algorithm with Guaranteed Convergence: Applications to Data Fusion in Imaging (2018)
24. Venkatakrisnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: *IEEE Global Conference on Signal and Information Processing*. pp. 945–948 (2013)
25. Weyand, T., Araujo, A., Cao, B., Sim, J.: Google landmarks dataset v2 – a large-scale benchmark for instance-level recognition and retrieval (2020)
26. Yu, G., Sapiro, G.: DCT image denoising: a simple and effective image denoising algorithm. *Image Processing On Line* **1** (2011)
27. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning Deep CNN Denoiser Prior for Image Restoration. In: *IEEE Conf. Comput. Vis. Pat. Recog.* pp. 2808–2817 (2017)
28. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric (2018)
29. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: *Int. Conference on Computer Vision*. pp. 479–486 (2011)