Image Denoising – Old and New

Michael Moeller and Daniel Cremers

Abstract Image Denoising is among the most fundamental problems in image processing, not only for the sake of improving the image quality, but also as the first proof-of-concept for the development of virtually any new regularization term for inverse problems in imaging. While variational methods have represented the state-of-the-art for several decades, they are recently being challenged by (deep) learning based approaches. In this chapter, we review some of the most successful variational approaches for image reconstruction and discuss their structural advantages and disadvantages in comparison to learning based approaches. Furthermore, we present a framework to incorporate deep learning approaches in inverse problem formulations so as to leverage the descriptive power of deep learning with the flexibility of inverse problems' solvers. Different algorithmic schemes are derived from replacing the regularizing subproblem of common optimization algorithms by neural networks trained on image denoising. We conclude from several experiments that such techniques are very promising but further studies are needed to understand to what extent and in which settings the power of the data-driven network transfers to a better overall performance.

1 Introduction

Fired by the continuously growing popularity of social media and communication applications the number of digital photos that is taken every day is rapidly increasing. While the hardware and with it the quality of the photographs is improving constantly, the demand for small imaging devices such as smartphones makes it

Daniel Cremers
Technical University of Munich, e-mail: daniel.cremers@in.tum.de

Michael Moeller
University of Siegen, e-mail: michael.moeller@uni-siegen.de

challenging to acquire high quality images in low light conditions. Thus, there is an urgent need to digitally remove the noise from such images while keeping the main characteristics of a realistic photograph.

Among the most powerful and well-studied methods for image denoising are energy minimization methods. One defines an energy or cost function E that depends on the noisy image f, and maps from a suitable space of candidate images to the real numbers in such a way, that a low number corresponds to an image with desirable properties, i.e. to a realistic and (ideally) noise-free image. Subsequently, one determines a denoised image $\hat{u}$ as the argument that minimizes E, i.e.,

$$\hat{u} \in \arg\min_{u} E(u). \tag{1}$$

In Section 2, we will provide a more systematic derivation of such variational approaches from the perspective of Bayesian inference. In Section 3, we will then summarize some of the most influential variational denoising methods, along with their underlying assumptions, advantages, and drawbacks.

An entirely different line of research that has become hugely popular and that has shown impressive performance over the last five years are data-driven learning-based methods: Whenever a sufficient amount of training data pairs of noisy and noise-free images (f^i, u^i) are available or can be simulated faithfully, one designs a parameterized function $G(f;\theta)$ and *learns* the parameters θ that lead to the best coincidence of $G(f^i;\theta)$ with u^i with respect to some predefined loss $\mathscr{L}$. To prevent overfitting, one often defines a regularization R on the weights θ and solves the energy minimization problem

$$\hat{\theta} \approx \arg\min_{\theta} \sum_{i} \mathscr{L}(G(f^i;\theta), u^i) + \alpha\, R(\theta). \tag{2}$$

Once the above (generally nonconvex) problem has been solved approximately (either by finding a critical point or by stopping early as an additional “regularization”), the inference simply passes new incoming noisy images f through the network, i.e. computes $G(f;\hat{\theta})$. We summarize some influential learning-based approaches to image denoising in Section 4.

Learning based strategies are a strong trend in the current literature and they have also been shown to compare favorably in several denoising works. Nevertheless, we are convinced that learning based strategies alone are not addressing the problem of image denoising exhaustively: Firstly, recent studies question the generalizability of learning based approaches to realistic types of noise [52]. More importantly, networks are very difficult to train. Solving (2) for a highly nested function G (often consisting of more than 20 layers) requires huge amounts of training data, sophisticated engineering and good initializations of the parameters θ as well as a considerable amount of manual fine tuning. Since the network architecture and weights may remain fixed during inference, it only works in the specific setting that it has been trained for. Finally, networks do not provide much control and guarantees about the output of the network. Although the training often leads to good results during inference, test images with characteristics different from the training data can easily lead to unpredictable behavior.

In Section 5 we will analyze these drawbacks in more detail. Moreover, we will analyze whether there is some potential in fusing concepts from energy minimization approaches with concepts from data-driven methods so as to combine the best of both worlds. To this end, we will present a framework for combining learning based approaches with variational methods. Indeed, preliminary numerical results indicate that the latter holds great promise in addressing some of the aforementioned challenges.

2 Denoising as Statistical Inference and MAP Estimation

A frequent motivation for energy based denoising methods of the form (1) are maximum a-posteriori probability (MAP) estimates: One aims at maximizing the conditional probability $p(u|f)$ of u being the true noise-free image, if one observed the noisy version f. According to the Bayesian formula, the posterior probability density can be written as

$$p(u|f) = \frac{p(f|u)p(u)}{p(f)}.$$

Instead of looking for the argument u that maximizes the above expression, by convention one equivalently minimizes its negative logarithm to obtain

$$\hat{u} \in \arg\min_u -\log(p(f|u)) - \log(p(u)). \tag{3}$$

The first term contains the probability of observing f given a true noise-free image u, and is referred to as the *data fidelity term*. For example, under the assumption of independent zero-mean Gaussian noise with standard deviation σ a spatially discrete formulation gives rise to

$$p(f|u) = \Pi_{\text{pixel } i} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(u_i - f_i)^2}{2\sigma^2}\right),$$

leading to the most commonly used ℓ^2-squared term for measuring data fidelity.[1] Many works have investigated the data fidelity terms arising from different distributions of the noise, see [66] for an example considering Poisson noise.

The quest for the right type of data fidelity term for denoising real photographs is, however, quite difficult and camera dependent: The raw sensor data undergoes several processing steps, such as white balancing, demosaicking, color correction / color space transformation, tonemapping, and possibly even compression. Depending on where in this processing chain the denoising is applied, different noise distributions have to be expected. In particular, color space transformations couple

[1] For a more detailed spatially continuous formulation, we refer to [21].

the noise over the color channels and demosaicking introduces a spatial correlation [60, 51]. The raw sensor data itself seems to follow a Poisson distribution and (for a reasonably high photon count) is well approximated by a Gaussian distribution with intensity dependent standard deviation – see e.g. [60].

3 Variational Image Denoising Methods

As suggested by the MAP estimate (3), typical energy minimization based techniques can be written as

$$E(u) \;=\; H_f(u) \;+\; R(u), \tag{4}$$

where the *data fidelity term* H_f corresponds to $-\log(p(f|u))$ in the MAP sense and measures how well the current estimate u fits to its noisy version f. The *regularization* R on the other hand corresponds to $-\log(p(u))$ in the MAP sense and penalizes oscillatory behavior of the noise. While the data fidelity term H_f can be motivated from the expected distribution of the noise in the data and can often be precalibrated by studying the sensor noise characteristics, the quest for a reasonable prior probability distribution p of natural images is significantly more challenging. In fact, the modeling of prior probabilities can be expected to benefit tremendously from suitable learning-based approaches such as deep neural networks – see Section 5.

3.1 Total Variation (TV) based Image Regularization

Even apart from the interpretation of MAP estimates, researchers have studied the properties of penalty functions R and their respective influence on the properties of the solution – often in a setting of ill-posed inverse problems in function spaces. Starting with penalties based on Tikhonov regularization, the advantageous properties of non-quadratic regularizations and non-linear filtering techniques in imaging have been studied from the 1980s on, see the references in [57] for some examples. The total variation (TV) regularization [61, 56] arguably is the most influential work in the field. For images $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$, it is defined as [30]

$$|u|_{TV} := \sup_{q \in C_0^\infty, |q(x)| \leq 1} \int_\Omega \mathrm{div}(q)(x)\; u(x)\; dx. \tag{5}$$

It has had an immense success in image denoising, because the functional is convex (enabling the efficient computation of optimal solutions), and because it applies to discontinuous functions u (enabling the preservation of sharp edges). For continuously differentiable functions u the total variation reduces to the integral over $|\nabla u(x)|$.

In parallel to the development of TV-based regularization methods, a tremendous amount of research has been conducted on image smoothing using (nonlinear) partial differential equations (PDEs) many of which arise as gradient flows of suitable regularization energies. For the sake of brevity, we will, however, not discuss these methods here.

3.2 Generalizations: Vectorial TV, Total Generalized Variation

A particularly interesting question for TV-based methods are suitable extensions to color images

$$u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}^d$$

with d color channels. Note that (assuming u to be differentiable) the Jacobian Ju is a $2 \times d$ matrix at each point x which raises the question in which norm $Ju(x)$ should be penalized for a suitable extension of the TV to color images. For non-differentiable functions u the analogue question is the quest for the most natural norm used to bound $q(x)$ in (5). Studies along these directions include the seminal work of Saprio and Ringach in [58], Blomgren and Chan [7], and a systematic study of different penalties of the Jacobian, e.g. [26]. Instead of using a penalty that strongly couples the color channels, some other lines of research consider

$$\int_\Omega \|\nabla C(u)(x)\| \; dx$$

for a suitable norm $\|\cdot\|$ and a linear operator C that changes the color space, e.g. [20], and possibly maps from three to more than three color-related channels, e.g. [2]. All studies agree that the alignment of edges of the RGB channels is of utmost importance to avoid visually disturbing color artifacts.

The success of total variation as a convex regularizer which can preserve discontinuities induced a quest for suitable higher order variants of the TV. To avoid the staircasing effect inherent to TV-based models, a second order derivative of the input image has to be considered in such a way, that the ability to reconstruct sharp edges is not lost. Higher-order TV models include the infimal-convolution regulariation [12] as well as the total generalized variation (TGV) [8]. The latter generalizes the total variation in (5) as follows:

$$\mathrm{TGV}_\alpha^k(u) = \sup\left\{ \int_\Omega u \,\mathrm{div}^k q \; dx \,\middle|\, v \in C_c^k(\Omega, \mathrm{Sym}^k(\mathbb{R}^d)), \|\mathrm{div}^l q\|_\infty \leq \alpha_l, l = 0,..,k-1 \right\},$$

where $\mathrm{Sym}^k(\mathbb{R}^d)$ denotes the space of symmetric tensors of order k with arguments in $\mathbb{R}^d$. Clearly through integration by parts the higher powers of the divergence operator correspond to higher order derivatives of the function u being penalized. Whereas the kernel of total variation merely contains the constant functions, the kernel of total generalized variation contains more interesting functions. Second order TGV, for example, contains the set of affine functions. Combined with the applicability to non-differentiable and discontinuous functions, this makes it well suited for denoising piecewise affine signals.

Similar to the extension of the total variation to color images, extensions of the total generalized variation to color images have been investigated e.g. in [47]. Note that in its discrete form, the second order TGV of a color image $u \in \mathbb{R}^{n_x \times n_y \times d}$ can be written as

$$TGV(u) = \inf_{w} \|D_1(u) - w\|_* + \|D_2(w)\|_+$$

where D_1 and D_2 are linear operators approximating suitable derivatives such that $D_1(u) \in \mathbb{R}^{n_x \times n_y \times d \times 2}$, and $D_2(w) \in \mathbb{R}^{n_x \times n_y \times d \times 2 \times 2}$. Thus, the TGV offers even more freedom in choosing different types of (tensor-based) norms $\|\cdot\|_*$ and $\|\cdot\|_+$ for different extensions to color images.

3.3 Nonconvex Regularizers

Given the success of total variation type regularizers in preserving sharp discontinuities, one may wonder if respective nonconvex generalizations may be even more suitable in preserving or even enhancing discontinuities.

More specifically, for a one-dimensional function which transitions monotonously between two values $a < b$, its total variation is exactly $b - a$, independent of how sharp this transition is. Discontinuities are hence associated with a finite penalty corresponding to the size of the step. An often undesired side effect of this property is the tendendcy of total variation to induce contrast loss.

In order to reduce this contrast loss, iterative techniques such as the Bregman iteration [50] can be considered. Similar ideas have also been investigated in [5], in which it was shown that an image's curvature is easier to reconstruct than the image itself, thus suggesting to use a two-step reconstruction procedure.

A different class of approaches, which can not only preserve but possibly even enhance discontinuities, penalize the gradient in a sublinear and therefore nonconvex manner. In the literature there exist numerous variants of this idea. Some of the most popular choices can be summarized in a regularization of the form

$$R(u) = \int_{\Omega} \psi(|\nabla u(x)|)\, dx \tag{6}$$

where typical choices of ψ include the linear one (absolute norm, i.e. total variation), the truncated linear, the truncated quadratic and (as the limiting case of the previous two) the Potts model (which penalizes any nonzero gradient with a constant value). See Figure 1 for a visualization. The truncated quadratic regularizer essentially corresponds to the Mumford-Shah model [48]. Such truncated regularizers are likely to preserve contrast because discontinuities are penalized with a constant cost ν independent of their size. This is indeed confirmed in the example in Figure 2.

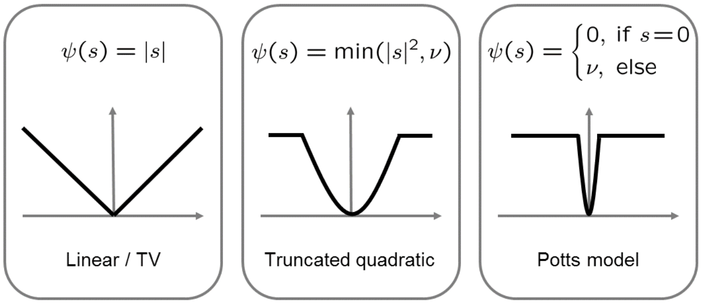


Fig. 1 Unified representation of various regularizers in the form (6) including the convex total variation (left) and the nonconvex truncated quadratic and (as its limiting case) the Potts model. The latter two regularizers essentially correspond to the weak membrane [6] or Mumford-Shah model [48].

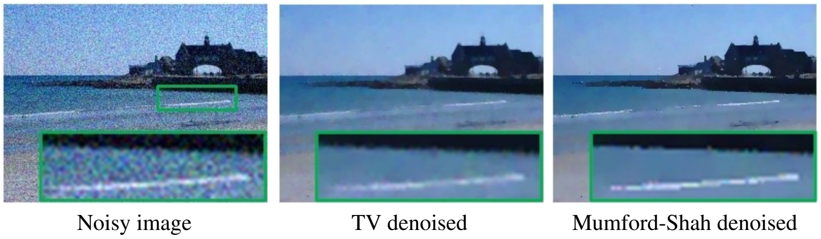


Fig. 2 While total variation regularization (center) induces a contrast loss, truncated regularizers like the Mumford-Shah model (right) better preserve discontinuities and contrast. The right image was computed using a convex relaxation of the vectorial Mumford-Shah model proposed in [64].

The Mumford-Shah model has been studied intensively in the applied mathematics literature because it is an interesting hybrid between a denoising and a segmentation approach. It is defined as follows:

$$E(u) = \int_{\Omega} \big(f(x) - u(x)\big)^2 dx + \lambda \int_{\Omega \setminus S_u} |\nabla u|^2 dx + \nu \mathscr{H}^1(S_u). \tag{7}$$

The aim is to approximate the input image $f : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$ in terms of a piecewise smooth function $u : \Omega \to \mathbb{R}$. The functional contains a data fidelity term and two regularity terms imposing smoothness of u in areas separated by the discontinuity set S_u and regularity of S_u in terms of its one-dimensional Hausdorff measure $\mathscr{H}^1(S_u)$. Related approaches were proposed in a spatially discrete setting by Geman and Geman [28] and by Blake and Zisserman [6]. The two regularizers in (7) clearly correspond to the truncated quadratic penalty above in the sense that energetically image locations where $\lambda|\nabla u|^2 > \nu$ will be associated with the discontinuity set S_u and hence are assigned a cost ν.

Sublinear penalties of the gradient norm are also more consistent with the statistics of natural images. Based on the observation that the regularizer is nothing but the negative logarithm of the prior – see Section 2 – one can study the statistics of gradient filter responses on natural images [35]. These statistics show heavy-tailed distributions which correspond to sublinear regularizers. An alternative representation of sublinear regularizers are the so-called TV-q models defined as:

$$\mathrm{TV}_q(u) := \int_\Omega |\nabla u(x)|^q \, dx, \tag{8}$$

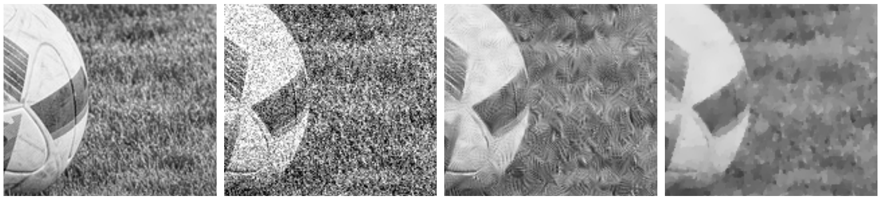
where for $q < 1$ the gradient is penalized sublinearly.

A challenging problem for the actual implementation of the aforementioned nonconvex variants of the total variation regularization is their optimization, in which one can only hope to determine local minimizers. While provably convergent methods typically have to rely on smoothing the nondifferentiable, nonconvex part, several works have shown very promising behavior of splitting techniques such as the alternating directions method of multipliers (ADMM), e.g. [16, 17, 27], or primal versions of primal-dual algorithms, e.g. [67, 65, 46]. We refer the interested reader to [71] to a recent summary on the convergence of the ADMM algorithm in a nonconvex setting.

3.4 Non-local Regularization

The most notable improvement – particularly for the problem of image denoising – was the development of non-local smoothing methods, starting with the non-local means (NLM) algorithm [10, 3]: Based on the idea that natural images are often self-similar one denoises images by first computing the similarity of pixels in a robust way, e.g. by comparing image patches, and subsequently determines the value of each denoised pixel by a weighted average based on pixel similarities. By considering the first step, i.e., the estimation of pixel similarities, as the formation of an image-dependent graph, regularization methods based on (different variants of) graph Laplacians were developed, see e.g. [39] for details. The extension of nonlocal methods to TV regularization was proposed in [29].

One of the most popular and powerful denoising algorithms is the block matching 3D (BM3D) algorithm [22], which is based on very similar assumptions as the above self-similarity methods, but sacrifices the interpretability in terms of a regularization function for a more sophisticated filtering strategy of patches that are considered to be similar. In particular, it estimates a first denoised version of an image to then recompute the similarity between pixels/patches, and denoises again. An interpretation in terms of a frame based regularization in a variational framework was given later in [24]. Further prominent extensions and improvements are based on learning the likelihood of natural image patches [76], and exploiting the low rank structure of similar image patches using weighted nuclear norm minimization [31].



(a) True image (b) Noisy image (c) NLM denoising (d) TV denoising

Fig. 3 Illustrating a failure of the self-similarity based NLM algorithm in a case where a faithful estimate of pixel correspondences is impossible: The NLM denoised image (c) contains strong artificial structures. While TV denoising is also unable to reconstruct the grass, it erases all high frequencies instead of hallucinating structures.

While the above methods are based on the assumption that every patch in an image has multiple similarly looking variants, the idea of *sparse representations* and *dictionary learning* relaxes this constraint. It merely demands that each patch can be represented as a linear combination of a few suitable patches from an overcomplete dictionary. The latter can not only be learned from a representative dataset, but even on the image to be reconstructed itself, with the K-SVD algorithm [1] being one of the most popular and powerful numerical methods for tackling the underlying nonconvex energy minimization problem. Hybrid self-similarity and dictionary learning techniques have been developed in [43], and a focus on dictionary learning for color image reconstruction has been set in [44].

3.5 A Discussion within Classical Denoising Methods

Before we discuss the extension of the partially data-driven model of dictionary learning to the mostly data-driven methods, let us compare the denoising methods we have recalled so far.

TV-type regularization methods are based on rather weak regularity assumptions and can therefore be applied to a wide range of different applications and types of images. Second order extensions such as TGV often improve upon TV while still depending on (weak) regularity assumptions only. The price for such improvements are an additional hyperparameter as well as a more complex minimization problem. Non-local methods such as NLM and BM3D rely on the reconstructed images to be self similar. While they often improve the results of local methods significantly, a faithful estimate of pixel/patch similarity is required. In settings of inverse problems where such an estimate is difficult to obtain, or in cases of strong noise in which similarity estimates become unreliable, such methods come at the risk of hallucinating self-similar structures as illustrated in Figure 3 – also see [9, Fig. 6].

Nonconvex variants of the above, e.g. TV^q or TGV^q models, can improve the results of their convex relatives – particularly in the presence of strong edges – but do come with the usual drawbacks of nonconvex optimization: No algorithm can guarantee not to get stuck in a bad local minimum. Similiarly, dictionary learning approaches such as the KSVD algorithm are based on nonconvex optimization problems and exploit a particular structure of the data, i.e. the ability to respresent each patch as a sparse linear combination of a few (learned) dictionary atoms.

In general the regularizing quality of the above approaches greatly improves with the strength of the assumptions that are made. This leads to self-similarity and dictionary learning based techniques clearly being the more powerful choice in usual practical settings of small or moderate noise and natural images. Strong assumptions can, however, influence the result in a very undesirable way if they do not hold, as we have illustrated in Figure 3. This makes the simpler (local) models possibly more attractive in applications where a structurally systematic error in the reconstruction can have dramatic consequences, e.g. in the field of medical imaging.

4 Learning Based Denoising Methods

In recent years researchers have had great success in replacing the implicit characterization of solutions as arguments that minimize a suitable energy function by explicit functions that directly map the input data to the desired solution. In the case of image denoising, such functions typically take the form

$$\begin{aligned} G: \mathbb{R}^{n\times m\times c} \times \mathbb{R}^k &\to \mathbb{R}^{n\times m\times c} \\ (f,\theta) &\mapsto G(f,\theta) \end{aligned} \tag{9}$$

where $f \in \mathbb{R}^{n\times m\times c}$ is the noisy input image, $G(f,\theta)$ is the denoising result and $\theta \in \mathbb{R}^k$ are weights that parametrize the function G. The latter are determined during *training*, which is the approximate solution of problem (2) for a suitable loss function $\mathscr{L}$, e.g. the ℓ^2-squared loss when optimizing for high peak signal-to-noise ratios (PSNRs). The pairs (f^i,u^i) of noisy and clean images used during training have to be representative for the setting the network is used in during inference, i.e., the types of images and the type of noise used for the training should originate from the same distribution as the test images.

The typical architecture of a *network* G is a nested function

$$G(f,\theta) = g_L(g_{L-1}(\dots g_2(g_1(f,\theta^1),\theta^2)\dots,\theta^{L-1}),\theta^L), \tag{10}$$

where each function g_i is referred to as a *layer*. The most common layers in basic architectures are parameterized affine *transfer functions* followed by a nonlinearity called *activation function*.

The specific architecture of G and its individual layers has evolved over the last years. The first networks to challenge the previous dominance of BM3D and KSVD type algorithms were *fully connected* using tangens hyperbolici as activation functions [11], e.g.,

$$\begin{aligned} g_i(x,\theta^i) &= \tanh(\theta^i[x;1]), \quad \forall i \in \{1,\dots,L-1\}, \\ g_L(x,\theta^L) &= \theta^L[x;1]. \end{aligned}$$

Small vectorized image patches of a noisy image are fed into the network. In each layer a 1 is attached to the input vector to allow for an offset, typically called *bias*. A crucial aspect of these powerful learning-based denoising approaches was a comparably large number of layers, relating to the overall trend of developing deep neural networks.

The work [73] proposed a sparse autoencoder architecture, also using fully connected layers and sigmoid activation functions. While [11] and [73] performed on par with BM3D and KSVD on removing Gaussian noise respectively, architectures based on convolutions, e.g. [37], or more recently convolutions with rectified linear units as activations, i.e.,

$$g_i(x,\theta^i) = \max(\theta^i_k * x + \theta^i_b, 0), \quad \forall i \in \{1,\dots,L-1\},$$

have shown promising results, e.g. in [74, 42]. Moreover, [74] proposed the idea of deep residual learning to the field of image denoising, i.e., the strategy of learning to output the estimated noise instead of the noise-free image itself.

Recent learning techniques such as [70, 41, 4] furthermore exploit the idea to filter image patches in (non-local) groups to mimic and improve upon the behavior of their designed relatives such as BM3D.

Besides a focus on more realistic types of noise (as pointed out in [52]), a promising direction for future denoising networks is to move from the (PSNR-optimizing) ℓ^2-squared loss function to perceptual [38], or GAN-based [40] loss functions that are able to reflect the subjective quality perception of the human visual system much more accurately.

Beyond the specific architecture and training of networks, further improvements can be made by tailoring denoising networks to specific classes of images determined by a prior classification network, see [53].

A drawback of most learning based approaches is that they are trained on a specific type of data, as well as a specific type and strength of noise. Thus, whenever one of these quantities changes, an expensive retraining is required. Although promising approaches for a more generic use of neural networks for varying strengths of Gaussian noise exist, see e.g. [72, 74], retaining a high quality solution over varying types of noise remains challenging.

In the next section we will discuss hybrid learning and energy minimization based approaches which represent a promising class of methods to not only adapt to different types of noise but even to different types of restoration problems easily.

5 Combining Learning and Variational Methods

5.1 Lacking Flexibility

Deep neural networks have proven to be extraordinary effective for a wide range of high and low level computer vision problems. Their effectiveness does, however, come at the costs of a complicated and expensive training procedure in which many different aspects such as different training algorithms, hyperparameters, initializations (e.g. [32]), dropout [63], dropconnect [69], batch-normalization [36], or the introduction of short cuts such as in ResNet [33], have to be considered to achieve good results. Moreover, networks often do not generalize well beyond the specific type of data they have been trained on: In the case of image denoising, for example, the authors of [52] showed that the classical BM3D algorithm yields better denoising results on real photograph than state-of-the-art deep learning techniques that were all trained on Gaussian noise.

While one might argue that the dominance of learning based approaches can be reestablished by training on more realistic datasets, several drawbacks remain:

1. Neural networks often do not generalize well beyond the specific setting they have been trained on. While approaches for training on a variety of different noise levels exist (e.g. in [72, 74]), networks typically cannot address arbitrary image restoration problem of reconstructing an image u from noisy data $f \approx Ku$ for a linear operator K, if the operator K was not already known during training time. Typically, every time the type of noise, the strength of noise, or the linear operator K changes, neural networks require additional training.
2. The separability of the data formation process from the regularization, and hence the negative log likelihood of the distribution of 'natural images', is lost in usual deep learning strategies despite the fact that learning-from-data seems to be the only way to realistically give a meaning to the term 'natural images' in the first place.
3. Even though a network might be trained on returning u^i for a given measurement $f^i = Ku^i + n^i$ for noise n^i drawn from a suitable distribution, there is no guarantee for the networks output $G(f, \hat{\theta})$ to follow the data formation $KG(f, \hat{\theta}) \approx f$ during inference, i.e., there is no *guarantee* for the output to be a reasonable explanation of the data.

On the other hand, one can constitute that

1. Variational methods have a plug-and-play nature in which one merely needs to adapt the data fidelity term H_f as the strength or type of noise or the linear operator K changes.
2. They clearly separate the data fidelity term from the regularization with each of the two being exchangeable.
3. The proximity to a given forward model can easily be guaranteed in variational methods by using suitable indicator functions for H_f.

4. Despite the above advantages, the expressive power of regularizations terms to measure how “natural” or “realistic” a given image is, is very limited. In particular, local (e.g. total variation based) or non-local smoothness properties do not capture the full complexity of textures and structures present in natural images. In fact, exploiting large data bases seems to be the most promising way for even defining what “natural images” are.

The complementary advantages of each method make possible ways to combine variational and learning based techniques an attractive field of research.

Considering the derivation of energy minimization methods from MAP estimates in Section 2, it seems natural to estimate $p(u)$ in (3) from training images. This, however, means estimating a probability distribution of natural images in a number-of-pixel dimensional space, which seems to be extremely ambitious even for moderately-sized images. In fact, the knowledge of such a distribution would allow to sample natural images – a task researchers currently try to tackle with generative adversarial networks (GANs), but still face many difficulties, e.g. for generating high resolution images. We refer the reader to [25, 49] for recent approaches that tackle inverse problems by estimating the distribution of natural images.

5.2 Learning the Regularizer

Researchers have already considered the general idea to learn the probabilty distribution of natural images more than a decade ago by settling for the probability distribution of separate patches, assuming a particular form of the underlying probability distribution, see the field of experts model by Roth and Black [55] for an example. While the latter actually tries to approximate the probability distribution of training data by combining a gradient ascent on the log-likelihood with a Monte Carlo simulation, the work [18] by Chen et al. proposes a different strategy: They show that an analysis-based sparsity regularization of the form

$$R(u,A) = \sum_{\text{patches } u_p} \; \sum_{\text{filters } A_i} \Phi(A_i * u_p) \tag{11}$$

is equivalent to the negative log-likelihood in the field of experts model. In the above, Φ denotes a robust penalty function such as $\log(1+z^2)$ and $A_i * u_p$ is the convolution of a filter A_i with the image patch u_p. For finding suitable filters A the authors, however, propose a bi-level optimization framework, which – in the case of image denoising – takes the form

$$\begin{aligned} &\min_A \left(\sum_{\text{training examples } (\hat{u}^j, f^j)} \|u^j(A) - \hat{u}^j\|_2^2 \right), \\ &\text{subject to } u^j(A) \in \arg\min_u \; \lambda \; \|u - f^j\|_2^2 + R(u,A), \end{aligned} \tag{12}$$

for pairs $(\hat{u}^j, f^j)$ of noise-free and noisy training images $\hat{u}^j$ and f^j, respectively.

Although the problem (12) is difficult to solve, the results in [18] are promising, the approach retains the interpretation of an energy minimization method, and the regularization can potentially generalize to arbitrary image restoration problems. Its limitation is, however, given by the manual choice and specific parametrization of R in (11).

5.3 Developing Network Architectures from Optimization Methods

For the sake of more freedom, the authors of [19] considered the minimization of energies like (4) for a parameterized regularization R with learnable weights. By using a gradient descent iteration, a discretization of a reaction-diffusion type of equation is obtained in which the authors, however, allow the parameterized regularization to change at each iteration of their scheme. Note that although this does not allow the interpretation as an energy minimization method anymore, it led to improved denoising performances.

Similarly, in [59] Schmidt and Roth construct a method based minimizing

$$\frac{1}{2}\|Ku-f\|_2^2 + \sum_{\text{filters } A_i} \sum_{\text{patches } u_p} \rho_i(A_i * u_p),$$

where ρ_i are suitable penalty functions to be learned. By considering a half-quadratic splitting that minimizes

$$E(u,z) = \frac{1}{2}\|Ku-f\|_2^2 + \sum_{\text{filters } A_i} \sum_{\text{patches } u_p} \rho_i(z_{i,p}) + \frac{\beta}{2}(z_{i,p} - A_i * u_p)^2, \tag{13}$$

for u and z in an alternating fashion, the update for u becomes a simple linear equation. The update for z reduces to what the authors call *shrinkage function* in [59], and which is called *proximal operators* in the optimization community. The proximal operator of a (typically proper, closed, convex) function $R : \mathbb{R}^n \to \mathbb{R} \cup \{\infty\}$ is defined as

$$\text{prox}_R(h) = \arg\min_v \frac{1}{2}\|v-h\|^2 + R(v). \tag{14}$$

In the case of minimizing for z in (13), all ρ_i are functions from $\mathbb{R}$ to $\mathbb{R}$, and so are prox_{ρ_i}. At this point, the idea of Schmidt and Roth is twofold:

1. They propose to parameterize and *learn* the proximal operators prox_{ρ_i} instead of the functions ρ_i, and even intentionally drop the constraint that the learned operators r_i must correspond to proximal operators. In fact, it is shown in [59] that the final r_i provably cannot be proximal operators anymore.
2. They allow the learned operators r_i to change in each iteration of the half-quadratic minimization.

By changing the operator in each step, the resulting algorithmic scheme does not resemble the structure of a minimization algorithm anymore. By omitting the monotonicity contraint which is necessary to even be able to identify an operator as the proximal operator of a function, not even a single iteration of the respective scheme can be interpreted as an energy minimization step. Nevertheless, as the training basically 'roles out' the algorithm for a fixed number of iterations, the paper naturally resemble a (deep) neural network whose architecture is motivated by the half-quadratic minimization method.

The methods from [59, 19] yield a nice motivation for the (otherwise somewhat handcrafted) architecture of a neural network, and both allow the extension from image denoising to more general linear inverse problems. Because both works, [59] and [19], do, however, have changing operators in each iterations and 'roll out' the iterations during training, they cannot be interpreted as an iteration yielding a (hopefully convergent) sequence of iterates as usual minimization algorithms. Moreover, the end-to-end training of the resulting algorithmic schemes still tailors the parameters to the specific setting (i.e. the specific operator K, type and strength of noise) they have been trained on.

5.4 Algorithmic Schemes based on Learned Proximal Operators

To avoid the aforementioned drawbacks recent research [75, 54, 15, 45] has considered fully decoupling the data formation process from learning a function that introduces the required regularity. All these approaches develop algorithmic schemes based on classical optimization methods and replace the proximal operator of the regularization by a neural network. The general idea originates from previous publications [23, 76, 68, 34] in which general algorithmic schemes were developed by replacing the proximal operator of the regularization by denoising algorithms such as BM3D or NLM. The premise that learning based approaches have the power to learn even more complex smoothness properties than the nonlocal similarities captured by NLM and BM3D subsequently motivated the introduction of neural networks. Let us review some of these ideas in more detail.

5.4.1 Deriving different schemes

As a motivation consider the problem of minimizing (4), i.e.

$$\min_u H_f(u) + R(u), \tag{15}$$

where the data term shall remain flexible as in usual variational methods, but the regularization shall be replaced by a data driven approach in order to benefit from the power of learning based strategies.

Following the idea of half-quadratic splitting we have already seen in (13), we could replace (15) by

$$\min_{u,z} H_f(u) \; + \; R(z) \; + \; \frac{\beta}{2}\|z-u\|^2, \tag{16}$$

which – under mild conditions – yields a minimizer of (15) for $\beta \to \infty$.

By applying alternating minimization to (16) one has to solve

$$u^{k+1} = \text{prox}_{\frac{1}{\beta}H_f}(z^k), \tag{17}$$

$$z^{k+1} = \text{prox}_{\frac{1}{\beta}R}(u^{k+1}). \tag{18}$$

As such an algorithm decouples the regularization from the data fidelity term, it is natural to replace the proximal operator of the regularization by a neural network. Based on the fact that the proximal operator of a regularization represents a denoising proceedure, or – in the extreme case – the projection onto a natural feasible set of natural images, researchers have trained respective networks to perform exactly these tasks, see [75, 15, 45]. In the above example of half quadratic splitting, the resulting algorithmic scheme becomes

$$u^{k+1} = \text{prox}_{\frac{1}{\beta}H_f}(z^k), \tag{19}$$

$$z^{k+1} = G(u^{k+1}; \hat{\theta}), \tag{20}$$

for a network G that has been trained on denoising or, more generally, on "making the image more realistic".

The above idea and derivation of the algorithmic scheme is of course not limited to the method of half-quadratic splitting, but actually applies to almost any minimization method for (16). Due to its flexibility in handling multiple terms the alternating direction method of multipliers (ADMM) and preconditioned variants thereof have mostly been used in this context, see [23, 76, 68, 34, 75, 54, 15, 45]. Since ADMM is known to not necessarily converge on nonconvex problems, this choice does not seem to be natural considering that approaches that replace a proximal operator by an arbitrary function are even beyond the setting of nonconvex minimization.

In Table 1 we provide an overview of a wide variety of different optimization methods and their corresponding algorithmic schemes that could be used in the very same fashion. Note that we not only considered replacing the proximal operator of the regularization with a neural network, but also its explicit counterpart – the explicit gradient descent step on R,

$$u - \tau \nabla R(u) \quad \to \quad G(u, \hat{\theta}).$$

This, for instance, leads to the algorithmic schemes of *proximal gradient 2* and *HQ splitting* to coincide despite originating from different optimization algorithms which do not even converge to the same minimizer in a suitable convex setting.

Method	Iteration	Algorithmic scheme
Gradient descent	$z^1 = u^k - 2\tau\nabla H_f(u^k),$ $z^2 = u^k - 2\tau\nabla R(u^k),$ $u^{k+1} = \frac{1}{2}(z^1 + z^2).$	$z^1 = u^k - 2\tau\nabla H_f(u^k),$ $z^2 = G(u^k;\hat{\theta}),$ $u^{k+1} = \frac{1}{2}(z^1 + z^2).$
Proximal Gradient 1	$z^k = u^k - \tau\nabla H_f(u^k),$ $u^{k+1} = \mathrm{prox}_{\tau R}(z^k).$	$z^k = u^k - \tau\nabla H_f(u^k),$ $u^{k+1} = G(z^k;\hat{\theta}).$
Proximal Gradient 2	$z^k = u^k - \tau\nabla R(u^k),$ $u^{k+1} = \mathrm{prox}_{\tau H_f}(z^k),$	$z^k = G(u^k;\hat{\theta}),$ $u^{k+1} = \mathrm{prox}_{\tau H_f}(u^k).$
HQ splitting	$u^{k+1} = \mathrm{prox}_{\frac{1}{\beta}H_f}(z^k),$ $z^{k+1} = \mathrm{prox}_{\frac{1}{\beta}R}(u^{k+1}).$	
ADMM	$u^{k+1} = \mathrm{prox}_{\frac{1}{\beta}H_f}(z^k - p^k),$ $z^{k+1} = \mathrm{prox}_{\frac{1}{\beta}R}(u^{k+1} + p^k),$ $p^{k+1} = p^k + u^{k+1} - z^{k+1}.$	$u^{k+1} = \mathrm{prox}_{\frac{1}{\beta}H_f}(z^k - p^k),$ $z^{k+1} = G(u^{k+1} + p^k;\hat{\theta}),$ $p^{k+1} = p^k + u^{k+1} - z^{k+1}.$
Primal-dual 1	$p^{k+1} = p^k + \beta\bar{u}^k$ $-\beta\mathrm{prox}_{\frac{1}{\beta}R}\left(\frac{p^k}{\beta} + \bar{u}^k\right),$ $u^{k+1} = \mathrm{prox}_{\tau H_f}(u^k - \tau p^{k+1}),$ $\bar{u}^{k+1} = u^{k+1} + (u^{k+1} - u^k).$	$p^{k+1} = p^k + \beta\bar{u}^k$ $-\beta G(\frac{p^k}{\beta} + \bar{u}^k;\hat{\theta}),$ $u^{k+1} = \mathrm{prox}_{\tau H_f}(u^k - \tau p^{k+1}),$ $\bar{u}^{k+1} = u^{k+1} + (u^{k+1} - u^k).$
Primal-dual 2	$z^{k+1} = z^k + \beta K\bar{u}^k$ $-\beta\mathrm{prox}_{\frac{1}{\beta}T_f}\left(\frac{z^k}{\beta} + K\bar{u}^k\right),$ $p^{k+1} = p^k + \beta\bar{u}^k,$ $-\beta\mathrm{prox}_{\frac{1}{\beta}R}\left(\frac{p^k}{\beta} + \bar{u}^k\right),$ $u^{k+1} = u^k - \tau K^T z^{k+1} - \tau p^{k+1},$ $\bar{u}^{k+1} = u^{k+1} + (u^{k+1} - u^k).$	$z^{k+1} = z^k + \beta K\bar{u}^k$ $-\beta\mathrm{prox}_{\frac{1}{\beta}T_f}\left(\frac{z^k}{\beta} + K\bar{u}^k\right),$ $p^{k+1} = p^k + \beta\bar{u}^k,$ $-\beta G(\frac{p^k}{\beta} + \bar{u}^k;\hat{\theta}),$ $u^{k+1} = u^k - \tau K^* z^{k+1} - \tau p^{k+1},$ $\bar{u}^{k+1} = u^{k+1} + (u^{k+1} - u^k).$

Table 1 Different algorithms for minimizing $H_f(u) + R(u)$ and the corresponding algorithmic schemes that replace explicit or implicit (proximal) gradient steps on the regularization by a neural network G. For the *Primal-dual 2* algorithm we assumed that $H_f = T_f \circ K$ for a linear operator K. Note that – even in a convex setting with some additional assumptions – the *HQ splitting* algorithm does not converge to a minimizer of $H_f(u) + R(u)$ but rather replaces R or H_f by the (smoother) Moreau envelope. The choice $\beta \to \infty$ can usually reestablish the convergence.

While the *HQ splitting* algorithm is unconditionally stable it converges to a minimizer of a smoothed version of the original energy (replacing R, respectively H_f, by its Moreau envelope). The *proximal gradient 2* algorithm on the other hand requires a step size $0 < \tau < \frac{2}{L}$ with L being the Lipschitz constant of ∇R to converge to a minimizer of $H_f + R$. This along with the long list of possible algorithmic schemes in Table 1, which could further be extended by the corresponding methods with inertia/momentum, raises the question which method should be used in practice. An exhaustive answer to this question (if it can be provided at all) requires a tremendous number of experiments involving different problems, different networks, data terms, parameters, and initializations, and goes beyond the scope of this paper. We do, however, provide some first experiments involving all algorithms in Section 6.

5.4.2 Hyperparameters of the algorithmic schemes

When comparing algorithmic schemes like *Proximal gradient 1* with their optimization algorithm counterpart, one observes that replacing the proximity operator with a neural network eliminates the step size parameter τ. The missing dependence of the 'regularization-step' on τ in the *Proximal gradient 1* scheme means that the step size τ merely rescales the data fidelity term: The resulting algorithmic scheme may always pick $\tau = 1$, i.e., eliminate the step size completely, and interpret any $\tau \neq 1$ as a part of H_f, see [45] for details. Interestingly, even simple choices like the function G being the identity may lead to divergent algorithmic schemes for large data fidelity parameters. This may motivate training a network function G on a rather small noise level such that even moderate data fidelity parameters can lead to a large emphasis on data fidelity over the course of the iteration. Note that - at least in the context of optimization - the aforementioned difficulties can be avoided by an implicit treatment of the data fidelity term as arising in the *Proximal Gradient 2* or *HQ splitting* algorithms.

The elimination of hyperparameters in the algorithmic schemes becomes even more interesting for the more sophisticated primal-dual and ADMM based schemes. Note that the parameter $1/\beta$ in the ADMM scheme also merely rescales the data fidelity term. As shown in [45], in the *primal-dual 1* scheme, we can define $\tilde{p} = p/\beta$ to arive at the update equations

$$\tilde{p}^{k+1} = \tilde{p}^k + \bar{u}^k - G(\tilde{p}^k + \bar{u}^k; \hat{\theta}), \tag{21}$$

$$u^{k+1} = \text{prox}_{\tau H_f}(u^k - \tau\beta\tilde{p}^{k+1}), \tag{22}$$

$$\bar{u}^{k+1} = u^{k+1} + (u^{k+1} - u^k). \tag{23}$$

In this case the parameter τ scales the data fidelity term, but the product of τ and β remains a factor for $\tilde{p}^{k+1}$ in the update of u^{k+1}. In the world of convex optimization, the product $\tau\beta$ has to remain smaller than the operator norm of the linear operator used in the primal-dual splitting, which – in our specific case of *primal-dual 1* – is the identity. Due to the equivalence of ADMM and the primal-dual algorithm in a convex setting, the largest value of the product $\beta\tau$ for which convergence can still be guaranteed is 1, which is also the choice we make in all our numerical experiments below. This allows to again eliminate τ completely as it merely rescales the data fidelity term H_f.

A similar computation allows to reduce the *primal-dual 2* scheme to a rescaling of T_f by $1/\beta$ and the product of $\tau\beta$. Since this splitting involves the linear operator of two stacked identities, the step size restriction in the convex setting would be $\tau\beta < 1/\sqrt{2}$. We chose $\tau\beta = 0.5$ in all our experiments.

5.4.3 Algorithm equivalence

A particularly interesting aspect in the above discussion is the equivalence of ADMM and primal-dual in the convex setting. Considering the ADMM scheme from Table 1, we notice that

$$z^k = u^k - p^k + p^{k-1}$$

such that the update in u^{k+1} can equivalently be written as

$$u^{k+1} = \text{prox}_{\frac{1}{\beta}H_f}(u^k - (2p^k - p^{k-1})).$$

By entirely eliminating the variable z from the update equations we arrive at the equivalent form

$$u^{k+1} = \text{prox}_{\frac{1}{\beta}H_f}(u^k - (2p^k - p^{k-1})), \tag{24}$$

$$p^{k+1} = p^k + u^{k+1} - G(p^k + u^{k+1}; \hat{\theta}). \tag{25}$$

In the convex setting, i.e., if G is the proximity operator of a proper, closed, convex function, Moreau's identity yields the commonly used form of the primal-dual algorithm as presented in [13]. Note that the equations (21)–(23) match those of (24) and (25) up to the extrapolation: While $2u^{k+1} - u^k$ appears in (21)–(23), the scheme (24)–(25) uses $2p^{k+1} - p^k$. In this sense, the *primal-dual* schemes of Table 1 represent algorithms arising from applying the convex ADMM optimization method to the dual optimization problem

$$\min_p (H_f)^*(p) + R^*(-p),$$

writing the algorihm in a primal-dual form, using Moreau's identity to obtain primal proximity operators only, and finally replacing one of the proximity operators by a neural network. While an algorithmic scheme motivated from a purely dual (and therefore inherently convex) point of view does not seem to have a clear intuition, our numerical experiments indicate that the two variants (21)–(23) and (24)–(25) perform quite similarly.

6 Numerical Experiments: Denoising by Denoising?

6.1 Different Noise Types and Algorithmic Schemes

So far, the literature on replacing proximal operators by neural networks, [75, 54, 15, 45], has focused on the linear inverse problems with a quadratic ℓ^2 norm as a data fidelity term, i.e.,

$$H_f(u) = \frac{\alpha}{2}\|Ku - f\|^2,$$

and ADMM or primal-dual type of algorithmic schemes. Interestingly, the behavior of such methods for image denoising with different types of noise, i.e., K being the identity and H_f being a penalty function different from the squared ℓ^2 norm, has received little attention despite the fact that adapting the type of penalty is known to be extremely important, particularly in the presence of outliers.

To investigate the behavior of the different algorithmic schemes presented in Table 1 we consider images with Gaussian and Salt-and-Pepper noise and use a Huber Loss

$$H^\nu_f(u) = \sum_{i,j} h^\nu(u_{ij} - f_{ij}), \qquad h^\nu(x) = \begin{cases} \frac{1}{2\nu}x^2 & \text{if } |x| \leq \nu, \\ |x| & \text{otherwise,} \end{cases}$$

as a data fideltiy term. The Huber loss has the advantage that it is differentiable with a L-Lipschitz continuous derivative for $L = \frac{1}{\nu}$, and, at the same time, also allows an efficient computation of its proximal operator, which is given by

$$\text{prox}_{\tau h}(y) = \begin{cases} y/(1+\frac{\tau}{\nu}) & \text{if } |y| \leq \nu + \tau \\ \text{sign}(y)(|y| - \tau) & \text{otherwise.} \end{cases}$$

In our experiments we evaluate the gradient descent (GD), proximal gradient 1 called forward-backward (FB) here, the half-quadratic splitting (HQ), the ADMM, the primal-dual 1 (PD1), and primal-dual 2 (PD2) (with K being the identity) schemes from Table 1 for denoising grayscale images using Matlabs built-in implementation of the DnCNN denoising network [74] as a proximal operator. For the sake of comparability, we also include the plain application of this denoising network (Net), and a total variation based denoising (TV) in our comparison. To each clean image, we add white Gaussian noise of standard deviation $\sigma = 0.05$ (for images with values in $[0,1]$), and additionally destroy 1% of the pixels using Salt-and-Pepper noise. While we are aware of the fact that this does not necessarily reflect a realistic data formation process for camera images, our goal here is to study to what extend each of the algorithmic schemes from Table 1 is able to adapt to different settings by changing the data fidelity term.

We fixed the smoothing parameter $\nu = 0.025$ for the Huber loss and then tuned the hyperparameters of TV and the algorithmic schemes on a validation image, where we found a data fidelity weight of 0.02 to be a good choice for all network-based algorithmic schemes. Note that this means that the factor in front of the Huber loss is smaller than $2/L$, where L is the Lipschitz constant of ∇H_f^ν. Clearly, the latter is important as the schemes that descent on H_f^ν in an explicit fashion typically require this condition even in a convex setting. Furthermore, we also met the requirements for 'convex convergence' in the primal dual schemes by choosing $\beta\tau = 1$ in the 'primal-dual 1' scheme, and $\beta\tau = 0.5$ for 'primal-dual 2'.

We keep all parameters fixed over a run on 7 different test images and show the resulting PSNR values for all algorithmic schemes in Table 2.

	cats	xmax	food	ball	car	monkey	pretzel	avg.
TV	27.53	24.12	29.46	24.89	27.27	28.00	30.57	27.41
Net	26.76	24.87	27.78	25.25	26.90	26.39	28.41	26.62
HQ	**28.97**	26.94	**29.97**	**26.97**	28.90	**29.42**	30.32	**28.79**
FB	28.86	26.73	29.84	26.81	28.97	29.29	**30.79**	28.76
GD	28.33	**26.96**	28.89	26.76	28.05	28.72	28.76	28.07
ADMM	28.86	26.73	29.84	26.81	28.97	29.29	**30.79**	28.76
PD1	28.86	26.73	29.84	26.81	28.97	29.29	**30.79**	28.76
PD2	28.85	26.76	29.83	26.81	**28.99**	29.30	**30.79**	28.76

Table 2 PSNR values for denoising images with Gaussian and Salt-and-Pepper noise obtained by applying a neural network trained on Gaussian noise (Net), total variation denoising (TV), and different algorithmic schemes with a neural network replacing the proximal operator of the regularization, and a Huber loss being used as a measure for data fidelity.

As we can see, algorithmic schemes are able to improve the results of plainly applying the network by more than 2db on average. Interestingly, the results among different algorithmic schemes vary very little with the gradient descent based algorithmic scheme being the only one that yields some deviation in terms of PSNR. While similar behavior of different algorithms is to be expected for convex optimization methods, it is quite remarkable that the algorithmic schemes behave similarly.

To investigate the robustness of the algorithmic schemes, we investigate their sensitivity with respect to the starting point. While we used a constant image whose mean coincides with the mean of the noisy image as a starting point for the results in 2, Table 3 shows the average PSNR values over the same test images when initializing with different images. As we can see the results remain remarkably stable with respect to different initializations.

In the above test we ran all algorithmic schemes for a fixed number of 100 iterations. An interesting question is, whether the algorithmic schemes actually converge or if they just behave somewhat nicely for a while, but do not yield any fixed points.

	TV	Net	HQ	FB	GD	ADMM	PD1	PD2
constant	27.41	26.62	28.79	28.76	28.07	28.76	28.76	28.76
random	27.40	26.65	28.81	28.77	28.10	28.77	28.77	28.80
noisy	27.41	26.64	28.79	28.76	28.06	28.76	28.76	28.77
different	27.40	26.68	28.79	28.75	27.98	28.76	28.76	28.70

Table 3 Average PSNR values each method achieved on the test set of 7 images used in Table 2 when initializing each method with a constant image (constant), with random numbers uniformly sampled in $[0,1]$ (random), with the noisy input image (noisy), or with Matlab's cameraman image, i.e., a different image (different). The final results of the algorithmic schemes remain remarkably stable and do not vary significantly more than the TV result (whose variations are merely due to different realizations of the noise).

6.1.1 Numerical convergence of algorithmic schemes

Several works in the literature investigate the question whether algorithmic schemes arising from the ADMM algorithm converge:

- The work [62] gives sufficient conditions under which a general denoiser, e.g. a neural network G, represents the proximal operator of some implicitly defined function. As G is assumed to be continuously differentiable and $\nabla G(u)$ has to be doubly stochastic for any u, the assumptions are, however, quite restrictive.
- The authors of [14] state a converge result of an ADMM based algorithmic scheme with adaptive penalty parameter under the assumption of a *bounded denoiser*. The adaptive scheme, however, possibly allows an exponential growth of the penalty parameter. While the latter safeguards the convergence the point it converges to might not be a fixed-point of the algorithmic scheme anymore.
- The work by Romano, Elad and Milanfar in [54] proposes a flexible way to incorporate denoiser G (such as neural networks) into different algorithmic frameworks by providing quite general conditions under which the function

$$R(u) = \frac{1}{2}\langle u, u - G(u)\rangle$$

  has a gradient $\nabla R(u) = u - G(u)$, such that it can easily be incorporated into existing optimization algorithms. While the assumption $G(\alpha u) = \alpha G(u)$ for all $\alpha \geq 0$ made in their work does hold for several denoisers, neural networks often have a bias in each layer which prevents the above homogeneity. We therefore investigate the question if the algorithmic schemes converge numerically for a state-of-the art denoising network which did not adapt its design to any particular convergence criteria.

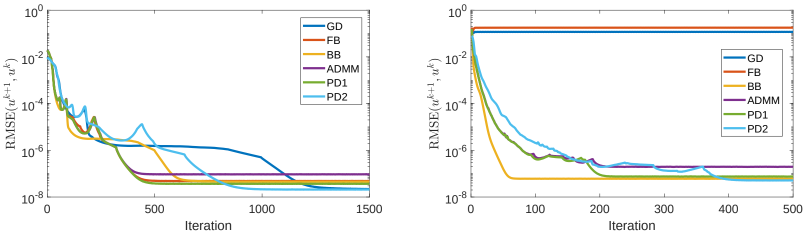
Figure 4 (a) shows the decay of the root mean square error (RMSE) of successive iterates

$$\mathrm{RMSE}(u^k, u^{k+1}) = \sqrt{\frac{1}{\text{number of pixels}} \sum_{i,j} (u^k_{ij} - u^{k+1}_{ij})^2}$$

for each of the algorithmic schemes from Table 2. As we can see, all algorithmic schemes converge to a reasonably small level (considering that all computations are done on a GPU in single precision).

We rerun the same test as above after multiplying the data fidelity term by a factor of 10 and illustrate the results in Figure 4 (b). As we discussed in Section 5.4.2 the data fidelity weight is directly connected to a step size of the algorithmic schemes. As expected based on the respective behavior in a convex optimization setting, methods that take explicit steps on the data fidelity term do not exhibit convergence anymore. Interestingly, the methods that evaluate the proximity operator of the data fidelity term still converge and seem to be quite independent of the magnitude of the data fidelity parameter.

While the numerical convergence behavior in our denoising test is closely related to the convergence behavior of the respective methods in the case of convex optimization, an analysis with sufficient conditions on the network to yield a provably convergent algorithm remains an interesting question of future research.



(a) Convergence with data fidelity 0.02

(b) Convergence with data fidelity 0.2

Fig. 4 Numerical convergence test of different algorithmic schemes. The schemes seem to behave similar to convex minimization techniques in the sense that they converge numerically if the convex stability criteria are satisfied.

6.2 *Handling Constraints*

Besides the lack of versatility of learning based approaches, a significant drawback is the lacking control over their output: For instance, once a network has been trained there is no parameter that allows to tune the amount of denoising. Moreover, although many types of constraints can be encouraged during training, there is no guarantee for the networks output to meet such constraints during testing. This is of utmost importance in any application where critical decisions depend on the networks output.

Interestingly, the framework of algorithmic schemes based on optimization algorithms allows to guarantee certain constraints by choosing the data fidelity term H_f to be the indicator function of the desired (convex) constrained set. As an example, consider the case where want to denoise an image under the constraint that each pixel may at most be altered by δ, i.e., we want our reconstruction u to meet $\|u-f\|_\infty \leq \delta$ for f being the noisy input image. Note that such constraints can easily be extended to a setting of inverse problem, e.g., requiring $\|Ku-f\|_2 \leq \delta$. The fact that indicator functions are not differentiable excludes the gradient descent, as well as the proximal gradient 1 algorithms. Moreover, the primal dual 2 scheme does not guarantee the output $u^{k_{\max}}$ to meet the constraint exactly unless it converged. We therefore return

$$\text{prox}_{H_f}\left(z^{k_{\max}} + u^{k_{\max}}\right),$$

which satisfies the constrain and is supposed to coincide with $u^{k_{\max}}$ upon convergence.

We simulate images with uniform noise and set our data fidelity term to be the indicator function of $\|u-f\|_\infty \leq \delta$, which has an easy-to-evaluate proximity operator. We run the algorithmic schemes HQ splitting, ADMM, primal-dual 1, and primal-dual 2, as well as TV denoising (as a baseline), and compare to the plain application of the denoising network.

The average PSNR values are shown in Table 4. Interestingly, the PSNR values do not differ significantly, and the algorithmic schemes may perform worse (HQ), or slightly better (PD2) than the plain application of the network. While these results would not justify the additional computational effort of the algorithmic schemes, note that the **Net** result violated the $\|u-f\|_\infty \leq \delta$ bound at about 25% of the pixels on average. Although the simple projection of the network's result would yield satisfactory results in this simple application, the constraint violation illustrates the lacking control of neural networks.

Finally, comparing the results of the network and the algorithmic schemes to plain TV denoising, we can see that TV denoising is (at least) on-par with the other methods. This yields the interesting conclusions that the advantages certain methods have as a denoiser do not necessarily carry over to other applications via the algorithmic schemes we presented in Table 1. In particular, an important question for future research is how networks (or general denoisers) can be designed in such a way that they work well in various different setting, in particular in such a way that they perform well with additional constraints on the output.

	TV	**Net**	**HQ**	**ADMM**	**PD1**	**PD2**
PSNR	32.77	32.66	31.99	32.54	32.67	32.80

Table 4 Average PSNR values each method achieved on the test set of 7 images with uniform noise and a suitable bound on $\|u-f\|_\infty$.

7 Conclusions

We have summarized some classical denoising methods including self-similarity based filtering and variational methods, and discussed various learning based methods that profit from a dataset of natural images. The framework of replacing proximal operators within optimization algorithms for energy minimization methods with denoising networks holds great promise in tackling various imaging tasks, using different data fidelities, and being able to adjust the amount of regularity without having to retrain the underlying neural network. Interestingly, the particular choice of algorithmic scheme had little influence on the final result in our numerical experiments and the convergence behavior of all algorithms was similar. Changing the algorithmic scheme from a penalty formulation to a constrained formulation changed the results quite significantly in the sense that the advantages of the neural network over TV regularization for image denoising did not transfer to the corresponding algorithmic scheme. Hence, an understanding of desirable properties of denoising algorithms for optimal results in the setting of algorithmic schemes remains an important question for future research.

References

1. M. Aharon, M. Elad, and A. Bruckstein. *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.
2. F. Aström and C. Schnörr. A geometric approach for color image regularization. *Computer Vision and Image Understanding*, 165:43 – 59, 2017.
3. Suyash P Awate and Ross T Whitaker. Higher-order image statistics for unsupervised, information-theoretic, adaptive, image filtering. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 44–51, 2005.
4. N.I. Cho B. Ahn. Block-matching convolutional neural network for image denoising, 2017. Preprint. Available online at https://arxiv.org/abs/1704.00524.
5. M. Bertalmo and S. Levine. Denoising an image by denoising its curvature image. *SIAM Journal on Imaging Sciences*, 7(1):187–211, 2014.
6. A. Blake and A. Zisserman. *Visual Reconstruction*. MIT Press, 1987.
7. P. Blomgren and T.F. Chan. Color tv: total variation methods for restoration of vector-valued images. *IEEE transactions on image processing*, 7(3):304–309, 1998.
8. K. Bredies, K. Kunisch, and T. Pock. Total generalized variation. *SIAM Journal on Imaging Sciences*, 3(3):492–526, 2010.
9. T. Brox and D. Cremers. Iterated nonlocal means for texture restoration. In F. Sgallari, A. Murli, and N. Paragios, editors, *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*, volume 4485 of *LNCS*, pages 13–24. Springer, May 2007.
10. A. Buades, B. Coll, and J. M. Morel. A non-local algorithm for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60–65, 2005.
11. H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, 2012.

12. A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167188, 1997.
13. A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision (JMIV)*, 2011.
14. S. H. Chan, X. Wang, and O. A. Elgendy. Plug-and-play admm for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2017.
15. J-H. Chang, C-L. Li, B. Poczos, B.V.K. Vijaya Kumar, and A.C. Sankaranarayanan. One network to solve them all — solving linear inverse problems using deep projection models. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.
16. R. Chartrand. Nonconvex splitting for regularized low-rank + sparse decomposition. *IEEE Transactions on Signal Processing*, 60(11):5810–5819, 2012.
17. R. Chartrand and B. Wohlberg. A nonconvex admm algorithm for group sparsity with sparse groups. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6009–6013, 2013.
18. Y. Chen, R. Ranftl, and T. Pock. Insights into analysis operator learning: From patch-based sparse models to higher order mrfs. *IEEE Transactions on Image Processing*, 23(3):1060–1072, 2014.
19. Y. Chen, W. Yu, and T. Pock. On learning optimized reaction diffusion processes for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.
20. L. Condat and S. Mosaddegh. Joint demosaicking and denoising by total variation minimization. In *IEEE International Conference on Image Processing (ICIP)*, pages 2781–2784, 2012.
21. D. Cremers, M. Rousson, and R. Deriche. A review of statistical approaches to level set segmentation: integrating color, texture, motion and shape. *International Journal of Computer Vision (IJCV)*, 72(2):195–215, 2007.
22. K. Dabov, A. Foi, and K. Egiazarian. Video denoising by sparse 3d transform-domain collaborative filtering. *European Signal Processing Conference (EUSIPCO)*, 2007.
23. A. Danielyan, V. Katkovnik, and K. Egiazarian. Image deblurring by augmented lagrangian with bm3d frame prior. 2010.
24. A. Danielyan, V. Katkovnik, and K. Egiazarian. Bm3d frames and variational image deblurring. *IEEE Transactions on Image Processing*, 21(4):1715–1728, 2012.
25. A. Dave, A.K. Vadathya, and K. Mitra. From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Image Processing (ICIP)*, 2017.
26. J. Duran, M. Moeller, C. Sbert, and D. Cremers. Collaborative total variation: A general framework for vectorial tv models. *SIAM Journal on Imaging Sciences*, 9(1):116–151, 2016.
27. X. Zhang E. Esser. Nonlocal path-based image inpainting through minimization of a sparsity promoting nonconvex functional, 2014. Preprint available at https://www.eoas.ubc.ca/ eesser/papers/nliPDHGM.pdf.
28. S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 6(6):721–741, 1984.
29. G. Gilboa and S. Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2009.
30. E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*. Birkhäuser, 1984.
31. S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2862–2869, 2014.
32. K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1026–1034, 2015.
33. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

34. F. Heide, M. Steinberger, Y.T. Tsai, M. Rouf, D. Pajk, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, J. Kautz, and K. Pulli. Flexisp: A flexible camera image processing framework. *ACM Special Interest Group on Computer Graphics (SIGGRAPH)*, 2014.
35. J. Huang and D. Mumford. Statistics of natural images and models. In *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference On.*, volume 1, pages 541–547. IEEE, 1999.
36. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, 2015.
37. V. Jain and S. Seung. Natural image denoising with convolutional networks. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 769–776. Curran Associates, Inc., 2009.
38. J. Johnson, A. Alahi, and F-F. Li. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision (ECCV)*, 2016.
39. A. Kheradmand and P. Milanfar. A general framework for regularized, similarity-based image restoration. *IEEE Transactions on Image Processing*, 23(12):5136–5151, 2014.
40. C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi. Photo-realistic single image super-resolution using a generative adversarial network, 2016. Preprint. Available online at https://arxiv.org/abs/1609.04802.
41. S. Lefkimmiatis. Non-local color image denoising with convolutional neural networks, 2017. Preprint. Available online at https://arxiv.org/abs/1611.06757.
42. P. Liu and R. Fang. Learning pixel-distribution prior with wider convolution for image denoising, 2017. Preprint. Available online at https://arxiv.org/abs/1707.09135.
43. J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman. Non-local sparse models for image restoration. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2272–2279, 2009.
44. J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008.
45. T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers. Learning proximal operators: Using denoising networks for regularizing inverse imaging problems. In *IEEE International Conference on Computer Vision (ICCV)*, 2017.
46. T. Möllenhoff, E. Strekalovskiy, M. Moeller, and D. Cremers. The primal-dual hybrid gradient method for semiconvex splittings. 8, 07 2014.
47. T. Möllenhoff, E. Strekalovskiy, M. Moeller, and D. Cremers. Low rank priors for color image regularization. In *Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR)*, 2015.
48. D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685, 1989.
49. A. Nguyen, J. Clune, Y. Bengio, A. Dosovitskiy, and J. Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
50. S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.
51. S. H. Park, H. S. Kim, S. Lansel, M. Parmar, and B. A. Wandell. A case for denoising before demosaicking color filter array data. In *Asilomar Conference on Signals, Systems and Computers*, pages 860–864, 2009.
52. T. Plötz and S. Roth. Benchmarking denoising algorithms with real photographs. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
53. T. Remez, O. Litany, R. Giryes, and A. M. Bronstein. Deep class-aware image denoising. In *2017 International Conference on Sampling Theory and Applications (SampTA)*, pages 138–142, 2017.
54. Y. Romano, M. Elad, and P. Milanfar. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

55. S. Roth and M. J. Black. Fields of experts: a framework for learning image priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 860–867, 2005.
56. L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D*, 1992.
57. P. Saint-Marc, J. S. Chen, and G. Medioni. Adaptive smoothing: a general tool for early vision. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 618–624, 1989.
58. G. Sapiro and D.L. Ringach. Anisotropic diffusion of multivalued images. In *Images, Wavelets and PDEs*, pages 134–140, 1996.
59. U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2774–2781, 2014.
60. T. Seybold, C. Keimel, M. Knopp, and W. Stechele. Towards an evaluation of denoising algorithms with respect to realistic camera noise. In *IEEE International Symposium on Multimedia*, pages 203–210, 2013.
61. D. Shulman and J-Y. Herve. Regularization of discontinuous flow fields. In *Visual Motion, 1989., Proceedings. Workshop on*, pages 81–86. IEEE, 1989.
62. S. Sreehari, S. V. Venkatakrishnan, B. Wohlberg, L.F. Drummy, J.P. Simmons, and C.A. Bouman. Plug-and-play priors for bright field electron tomography and sparse interpolation. *IEEE Transactions on Computational Imaging*, 2:408–423, 2016.
63. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958, 2014.
64. E. Strekalovskiy, A. Chambolle, and D. Cremers. A convex representation for the vectorial mumford-shah functional. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Providence, Rhode Island, June 2012.
65. E. Strekalovskiy and D. Cremers. Real-time minimization of the piecewise smooth mumford-shah functional. In *European Conference on Computer Vision (ECCV)*, 2014.
66. T.J. Asaki T. Le, R. Chartrand. A variational approach to reconstructing images corrupted by poisson noise. *Journal of Mathematical Imaging and Vision (JMIV)*, 27:257263, 2007.
67. T. Valkonen. A primaldual hybrid gradient method for nonlinear operators with applications to mri. *Inverse Problems*, 30(5), 2014.
68. S. Venkatakrishnan, C.A. Bouman, and B. Wohlberg. Plug-and-Play Priors for Model Based Reconstruction. *Global Conference on Signal and Information Processing (GlobalSIP)*, 2013.
69. L. Wan, M. Zeiler, S. Zhang, Y. Le Cun, and R. Fergus. Regularization of neural networks using dropconnect. In Sanjoy Dasgupta and David McAllester, editors, *International Conference on Machine Learning (ICML)*, volume 28 of *Proceedings of Machine Learning Research*, pages 1058–1066, Atlanta, Georgia, USA, 2013. PMLR.
70. R. Wang and D. Tao. Non-local auto-encoder with collaborative stabilization for image restoration. *IEEE Transactions on Image Processing*, 25(5):2117–2129, 2016.
71. Y. Wang, W. Yin, and J. Zeng. Global convergence of admm in nonconvex nonsmooth optimization. Preprint available at https://arxiv.org/abs/1511.06324, year=2017,.
72. Y. Q. Wang and J. M. Morel. Can a single image denoising neural network handle all levels of gaussian noise? *IEEE Signal Processing Letters*, 21(9):1150–1153, 2014.
73. J. Xie, L. Xu, and E. Chen. Image denoising and inpainting with deep neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 341–349. Curran Associates, Inc., 2012.
74. 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.
75. K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2808–2817, 2017.
76. D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Computer Vision (ICCV)*, pages 479–486, 2011.