Patch-Craft Self-Supervised Training for Correlated Image Denoising

Gregory Vaksman and Michael Elad
CS Department - The Technion
Haifa, Israel
grishav@campus.technion.ac.il, elad@cs.technion.ac.il

Abstract

Supervised neural networks are known to achieve excellent results in various image restoration tasks. However, such training requires datasets composed of pairs of corrupted images and their corresponding ground truth targets. Unfortunately, such data is not available in many applications. For the task of image denoising in which the noise statistics is unknown, several self-supervised training methods have been proposed for overcoming this difficulty. Some of these require knowledge of the noise model, while others assume that the contaminating noise is uncorrelated, both assumptions are too limiting for many practical needs. This work proposes a novel self-supervised training technique suitable for the removal of unknown correlated noise. The proposed approach neither requires knowledge of the noise model nor access to ground truth targets. The input to our algorithm consists of easily captured bursts of noisy shots. Our algorithm constructs artificial patch-craft images from these bursts by patch matching and stitching, and the obtained crafted images are used as targets for the training. Our method does not require registration of the images within the burst. We evaluate the proposed framework through extensive experiments with synthetic and real image noise.

1. Introduction

Supervised neural networks have proven themselves powerful, achieving impressive results in solving image restoration problems (e.g., [14–16, 18, 33, 41–43]). In the commonly deployed supervised training for such tasks, one needs a dataset consisting of pairs of corrupted and ground truth images. The degraded images are fed to the network input, while the ground truth counterparts are used as guiding targets. When the degradation model is known and easy to implement, one can construct such a dataset by applying the degradation to clean images. However, a problem arises when the degradation model is unknown. In such cases, while it is relatively easy to acquire distorted images, obtaining their ground truth counterparts can be challenging. For this reason, there is a need for self-supervised methods that use corrupted images only in the training phase. More on these methods is detailed in Section 2.

In this work we focus on the problem of image denoising with an unknown noise model. More specifically, we assume that the noise is additive, zero mean, but not necessarily Gaussian, and one that could be cross-channel and short-range spatially correlated[1]. We additionally assume that the noise is (mostly) independent of the image and nearly homogeneous, i.e., having low to moderate spatially variant statistics. Examples of such noise could be Gaussian correlated noise or real image noise in digital cameras. Several recent papers propose methods for self-supervised training under similar such challenging conditions. However, they all assume an uncorrelated noise or a noise with a known model, thus limiting their coverage of the need posed.

This work proposes a novel self-supervised training framework for addressing the problem of image denoising of an unknown correlated noise. The proposed algorithm gets as input bursts of shots, where each frame in the burst captures nearly the same scene, up to moderate movements of the camera and objects. Such sequences of images are easily captured in many digital cameras. Our algorithm uses one image from the burst as the input, utilizing the rest of the frames of the same burst for constructing (noisy) targets for training. For creating these target images, we harness the concept of patch-craft frames introduced in PaCNet [35]. Similar to PaCNet, we split the input shot into fully overlapping patches. For each patch, we find its nearest neighbor within the rest of the burst images. Note that, unlike PaCNet, we strictly omit the input shot from the neighbor search. We proceed by building m patch-craft

This research was partially supported by the Israel Science Foundation (ISF) under Grant 335/18 and the Council For Higher Education - Planning & Budgeting Committee.

[1] By short-term we refer to the case in which the auto-correlation function decays fast, implying that only nearby noise pixels may be highly correlated. The correlation range we consider is governed by the patch size in our algorithm - see Section 3 for more details.

images by stitching the found neighbor patches, where m is the patch size, and use these frames as denoising targets. The above can be easily extended by using more than one nearest neighbor per patch, this way enriching dramatically the number of patch-craft frames and their diversity.

The proposed technique for creating artificial target images is sensitive to the possibility of getting statistical dependency between the input and the target noise. To combat this flaw, we propose a method for statistical analysis of the target noise. This analysis suggests simple actions that reduce dependency between the target noise and the denoiser's input, leading to a significant boost in performance. We evaluate the proposed framework through extensive experiments with synthetic and real image noise, showing that the proposed framework outperforms leading self-supervised methods. To summarize, the contributions of this work are the following:

- We propose a novel self-supervised framework for training an image denoiser, where the noise may be cross-channel and short-range spatially correlated. Our approach relies simply on the availability of bursts of noisy images; the ground truth is unavailable, and the noise model is unknown.
- We suggest a method for statistical analysis of the target noise that leads to a boost in performance.
- We demonstrate superior denoising performance compared to leading alternative self-supervised denoising methods.

2. Related Work

This paper focuses on denoising of images when the noise model is unknown. However, there are various levels in this lack of knowledge, and accordingly, different levels of corresponding solutions. The most simple case is when the noise is known to be zero-mean Gaussian i.i.d (independent and identically distributed), and the only unknown parameter is the standard deviation σ. In this case, training a single model for handling a range of σ values can be an efficient and elegant solution [12, 19, 34, 41]. This approach is known as *blind denoising* [19]. Unfortunately, such a network is likely to perform very poorly when applied to images contaminated by a correlative or a non-Gaussian noise.

An approach known as *Noise2noise* [13] assumes that ground truth images are not available, but the training dataset consists of pairs of noisy images created by adding independent noise realizations to the same clean image. Noise2noise suggests to train on such image pairs, both being noisy but with independent noise realizations. Work reported in [17] adopts a similar yet different approach by utilizing pairs of noisy images to estimate a noise model, which is then employed for supervised training. These methods have been shown to be quite effective, however, the missing ingredient is the lack of an accessible way for acquiring such perfectly aligned noisy pairs, rendering these methods as challenging in real circumstances.

A more common assumption is that the noise model is known, but ground truth images are not available. Several works (e.g., [21, 23, 38]) utilize the idea of Noise2noise [13] for handling these cases as well. They propose, each in its own way, to create noisy image pairs. For instance, the work in [21, 38] suggests adding independent realizations of synthetic noise that follows the known model to the input images, then training a network using the noisier images as inputs and the original noisy ones as targets. Alternatively, [23] adds two different realizations of synthetic noise to the input images for training the denoising network.

A different idea proposed in [26, 27] is harnessing a variational auto-encoder (VAE) [7] to solve the denoising task. These techniques assume that the noise distribution is known, either as a formula or as a histogram. These algorithms construct a VAE that gets noisy images at the input and produces reconstructed ones at the output. In the training stage, they maximize the log-likelihood probability of the noisy image given the reconstructed one when the probability is calculated using the provided noise distribution formula or the histogram. In the inference stage, they use the VAE to generate many candidate outputs and then obtain the reconstructed image by approximating the MMSE or MAP estimate.

Another self-supervised technique suitable for these assumptions was introduced in [34] for lightweight architectures, and extended in [20] for more general networks. This technique, referred to as *noise resampling*, suggests the following: First, train an initial denoiser somehow and apply it to a set of corrupted images to obtain initial reconstructions. Then, for each reconstructed image, create its noisy counterpart by adding a new synthetic noise. Finally, retrain the network using pairs of re-corrupted and reconstructed images. When the noise model is unknown, noise-to-noise and noise resampling methods can be applied by assuming Gaussianity and estimating the parameter σ. However, such a strategy may lead to inferior performance when the noise is correlated or strongly deviates from Gaussianity.

A less strict assumption is that the noise model is unknown, but the noise is spatially independent. For such a case, several papers in recent literature have proposed to train networks that utilize the same noisy image both as input and output while applying various regularizations. For brevity of our discussion, we shall refer to these as *image2itself* techniques. For instance, Noise2void [8, 9] suggest using a blind-spot architecture in which the receptive field of each processed pixel excludes the pixel itself. Such a strategy constrains the network by avoiding to learn the trivial identity operation. The work reported

in [2, 10, 11, 28, 37] and other papers take this idea forward by suggesting more sophisticated blind-spot methods, sometimes combining them with additional regularization terms.

Blind-spotting is not the only regularization idea for these circumstances. For example, Neighbor2neighbor [5] proposes generating training pairs by random sub-sampling the same noisy image. The sub-sampling is conducted such that corresponding pixels of the same image pair are neighbors in the sampled image, thus having a very similar appearance. Alternatively, the works reported in [31, 44] use a regularizer based on the SURE [32] estimator as a replacement for the supervised targets.

Unfortunately, all these image-to-itself methods strongly rely on a spatial independence property of the noise, and therefore are doomed to overfit when the contaminating noise is correlated. In such a case, the network may confuse the noise for content, impairing the denoising performance. Worth noting is the exception in which HDN [26] shows an ability to recover microscopy images from structured noise. However, as natural images are considerably more diverse than microscopy ones, their approach may find a challenge when applied to general denoising tasks.

3. Proposed Framework – Preliminaries

This paper proposes a self-supervised framework for training denoisers using bursts of noisy images. Captured objects do not have to be static, yet we recommend avoiding bursts that contain sharp movements or severe lighting changes. One can obtain such sequences using burst mode in digital cameras or recording short videos.

We start by introducing some notations. Denote a noisy burst as $\{\mathbf{y}_1, \dots, \mathbf{y}_M\}$, where M is the burst length and $\mathbf{y}_i$ is the i'th image. The clean image and the input noise corresponding to $\mathbf{y}_i$ are denoted by $\mathbf{x}_i$ and $\mathbf{z}_i$, respectively, and thus $\mathbf{z}_i = \mathbf{y}_i - \mathbf{x}_i$. The symbol $f(\cdot)$ stands for a denoiser and $\hat{\mathbf{x}}_i$ for the reconstructed counterpart of $\mathbf{y}_i$, i.e., $\hat{\mathbf{x}}_i = f(\mathbf{y}_i)$. Finally, we denote the artificial target image corresponding to $\mathbf{y}_i$ by $\tilde{\mathbf{x}}_i$, and the target noise by $\mathbf{w}_i$, $\mathbf{w}_i = \tilde{\mathbf{x}}_i - \mathbf{x}_i$. The proposed framework is shown schematically in Figure 1. Our algorithm harnesses the concept of *patch-craft* frames introduced in [35]. Each input burst $\{\mathbf{y}_1, \dots, \mathbf{y}_M\}$ is split into two subsets: The first is $\mathbf{y}_i$, consisting of a single shot, and the the second, Γ, containing the rest of the images, $\Gamma = \{\mathbf{y}_1, \dots, \mathbf{y}_M\} \setminus \mathbf{y}_i$. The image $\mathbf{y}_i$ is used as a denoiser's input, while the set Γ is fed to the patch-craft block for creating an artificial target, $\tilde{\mathbf{x}}_i$.

The patch-craft block operates as follows. We start from splitting the input shot $\mathbf{y}_i$ to fully overlapping patches of size $n \times n$, boundary pixels handled by mirror padding. As a result, we get n^2 sets of non-overlapping patches that cover the full support of the image, $\{\Upsilon_{k,l}\}_{k,l=0}^{n-1}$, to which we refer by their offsets from the left upper corner. Two examples of such sets are shown in Figure 2. The offsets vary from $(0,0)$, i.e., no offset, to $(n-1, n-1)$. Each of the images $\{\Upsilon_{k,l}\}$ can be converted to a patch-craft image $\{\tilde{\mathbf{y}}_{k,l}\}$ by replacing each patch in $\Upsilon_{k,l}$ with its nearest neighbor from the set Γ and cutting out pixels corresponding to the padding. For finding the neighbors, we use an L_2 distance while the search in each image of Γ is restricted by a bounding box of size $B \times B$ centered at the patch location. Finally, at any iteration of the training, we randomly choose one of the n^2 available patch-craft images $\{\tilde{\mathbf{y}}_{k,l}\}$ to be a target $\tilde{\mathbf{x}}_i$. Figure 3 shows an example of a noisy image and one of the corresponding patch-craft images.

Few notes are in order: Since the neighbor patches come from different locations, all n^2 patch-craft images constructed from the same burst are similar but not identical. Each holds additional information enriching the training process. Furthermore, the proposed technique can be extended by finding k nearest neighbors per patch and choosing one of them (e.g. randomly) in the patch-craft construction. This extension may increase the number of possible patch-craft images, substantially enriching their diversity.

4. Proposed Framework - Analysis

Consider a denoiser $f_\theta(\cdot)$ parameterized by θ that gets a noisy image $\mathbf{y}$ and produces it's reconstructed image $\hat{\mathbf{x}}$, i.e., $\hat{\mathbf{x}} = f_\theta(\mathbf{y})$. The desired (ground truth) image is denoted by $\mathbf{x}$ and the input noise is $\mathbf{z}$, thus $\mathbf{y} = \mathbf{x} + \mathbf{z}$. In this section we discuss a training procedure in which instead of clean targets $\mathbf{x}$, one uses noisy ones $\tilde{\mathbf{x}}$ contaminated by a target noise $\mathbf{w}$, $\mathbf{w} = \tilde{\mathbf{x}} - \mathbf{x}$. Generally, training a denoiser is sensitive to the dependency between the input image $\mathbf{y}$ and the target noise $\mathbf{w}$. To illustrate this vulnerability, imagine the extreme case where $\mathbf{w}$ is equal to $\mathbf{z}$ for all pairs in the dataset. In such a case, the input and target images would be identical, and the denoiser would learn the useless identity operation. This example aligns with the intuition that lowering such dependency may result in better training and eventual denoising performance. In this section we propose a statistical analysis of the target noise and suggest a simple way to reduce it's dependency with the input noise.

4.1. Ideal Case

Results reported in Noise2Noise [13] suggest that if the target noise is independent of the network's input, using these targets for training a denoiser may be almost as effective as using the ground truth images. We formalize this hereafter, and start with few supporting notations.

Let $l = \frac{1}{2}\left\|f_\theta(\mathbf{y}) - \mathbf{x}\right\|_2^2$ be the supervised L_2 loss of the single image pair $\{f_\theta(\mathbf{y}), \mathbf{x}\}$. Denote by $\nabla_\theta l$ the gradient of l w.r.t. θ, $\nabla_\theta l = \nabla_\theta^T f_\theta(\mathbf{y})\left(f_\theta(\mathbf{y}) - \mathbf{x}\right)$. Then the supervised MSE loss and its full gradient are given by $L = \mathbb{E}[l]$ and $\nabla_\theta L = \nabla_\theta\left(\mathbb{E}[l]\right) = \mathbb{E}[\nabla_\theta l]$. Similarly, we denote the self-supervised L_2 loss of $\{f_\theta(\mathbf{y}), \tilde{\mathbf{x}}\}$ by

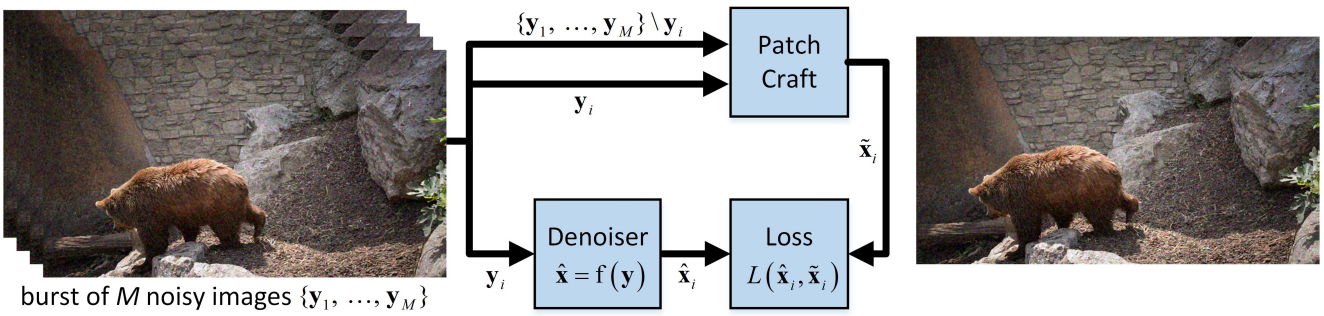
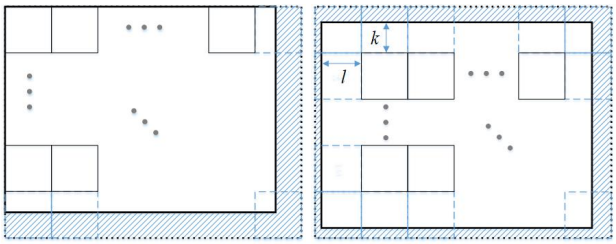


Figure 1. The proposed self-supervised training framework based on bursts of images and patch-craft created target images.



(a) $\Upsilon_{0,0}$ (b) $\Upsilon_{k,l}$

Figure 2. Examples of sets of non-overlapping patches. The solid part of the rectangle represents the input image support, while the dashed part stands for the boundary effects. Figure 2a shows the case of an offset $(0, 0)$, while Figure 2b refers to an offset (k, l).

(a) Noisy image (b) Patch-craft image

Figure 3. An example of a noisy shot and one of the corresponding patch-craft images.

$\tilde{l} = \frac{1}{2} \|f_\theta(\mathbf{y}) - \tilde{\mathbf{x}}\|_2^2$. Correspondingly, the gradient of $\tilde{l}$ is denoted by $\nabla_\theta \tilde{l}$, being $\nabla_\theta \tilde{l} = \nabla_\theta^T f_\theta(\mathbf{y}) \left(f_\theta(\mathbf{y}) - \tilde{\mathbf{x}}\right)$.

Lemma 1. *If the target noise* $\mathbf{w}$ *is independent of the image* $\mathbf{x}$ *and noise* $\mathbf{z}$*, and admits a zero-mean* $\mathbb{E}[\mathbf{w}] = \mathbf{0}$*, then* $\nabla_\theta \tilde{l}$ *is an unbiased estimator of* $\nabla_\theta L$*, i.e.,* $\mathbb{E}\left[\nabla_\theta \tilde{l}\right] = \nabla_\theta L$.

The proof of the Lemma is given in appendix A. The implication is that under appropriate assumptions, self-supervised training with noisy targets is equivalent to a variation of the SGD [29] algorithm with the regular supervised MSE loss. Thus, all guarantees and intuitions that are valid for a supervised training with SGD and an MSE loss are also correct for training with noisy targets.

Returning to the proposed scheme, a conclusion from Lemma 1 is that statistical independence between the input image $\mathbf{y}_i$ and the target noise $\mathbf{w}_i$ is desirable. Therefore, as a first and simple step for reducing this dependency, we omit $\mathbf{y}_i$ from the set Γ. A further method for reducing this dependency is discussed next.

4.2. Dependency Reduction

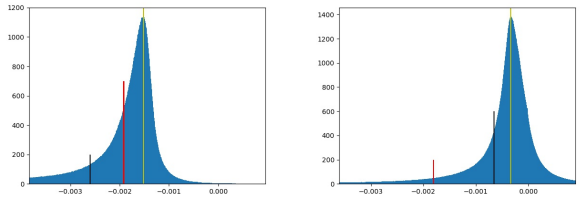
As mentioned above, the training procedure can be sensitive to a statistical dependency between the target noise and the network's input, and thus we seek ways to reduce it. We bring in this section the main points of the proposed method, leaving formal proofs and other details to appendix B.

As the proposed method may seem counter-intuitive at first glance, we start by building the reader's intuition gradually. Let us discuss the two most common types of dependencies that may be introduced by patch matching: (I) *overfitting input noise* and (II) *underfitting ground truth images*. Dependency of type (I) refers to cases when patch matching does a "too good" job, bringing target noise $\mathbf{w}$ that mimics the input noise, $\mathbf{z}$. This dependency is characterized by a positive correlation between $\mathbf{w}$ and $\mathbf{z}$.

As for type (II) dependency, it happens when the patch-craft, $\tilde{\mathbf{x}}$, and ground truth, $\mathbf{x}$, images tend to be dissimilar. It is less intuitive, but this dependency is manifested in a negative correlation between $\mathbf{w}$ and $\mathbf{x}$. Here is brief explanation of this phenomenon: Consider the following scalar covariances computed over pairs of images: $\sigma_{\tilde{\mathbf{x}},\tilde{\mathbf{x}}}$ $\sigma_{\mathbf{x},\tilde{\mathbf{x}}}$, $\sigma_{\mathbf{x},\mathbf{w}}$, and $\sigma_{\mathbf{x},\mathbf{x}}$. Clearly, $\sigma_{\mathbf{x},\tilde{\mathbf{x}}} = \sigma_{\mathbf{x},\mathbf{x}} + \sigma_{\mathbf{x},\mathbf{w}}$. Assuming that $\sigma_{\tilde{\mathbf{x}},\tilde{\mathbf{x}}} \approx \sigma_{\mathbf{x},\mathbf{x}}$, the dissimilarity between $\tilde{\mathbf{x}}$ and $\mathbf{x}$ reduces the value of $\sigma_{\mathbf{x},\tilde{\mathbf{x}}}$, which means that $\sigma_{\mathbf{x},\mathbf{w}}$ is necessarily negative (see more on this phenomenon in appendix B).

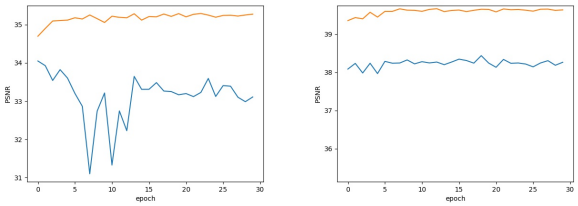
Let us look at an empirical covariance between $\mathbf{y}$ and $\mathbf{r}$, denoted by $s_{\mathbf{y},\mathbf{r}}$, where $\mathbf{r} = \tilde{\mathbf{x}} - \mathbf{y}$. This covariance is a scalar obtained for each possible image pair $\{\mathbf{y}, \mathbf{r}\}$. By assessing many such pairs, we get a histogram of these covariance values, which we analyze next. Observe that these covariance values are accessible, easily computed from the data we have. Here are few facts regarding $s_{\mathbf{y},\mathbf{r}}$:

- If $\mathbf{x}$, $\mathbf{z}$, and $\mathbf{w}$ are mutually independent, $s_{\mathbf{y},\mathbf{r}}$ converges in distribution to a Gaussian centered at $-\sigma_z^2$



(a) Correlated Gaussian noise (b) Real image noise

Figure 4. Examples of $s_{\mathbf{y},\mathbf{r}}$ histograms in experiments with correlated Gaussian and real image noise. The yellow bar is located at the histogram peak, while the black bar shows the location of the mean. The red bar indicates the location of s_{min}.



(a) Correlated Gaussian noise (b) Real image noise

Figure 5. Validation PSNR before and after dependency reduction. The blue line shows validation PSNR vs. epoch number during training on the full dataset, while the orange line indicates the PSNR after excluding image pairs for which $s_{\mathbf{y},\mathbf{r}} < s_{min}$.

and thus $\mathbb{E}\left[s_{\mathbf{y},\mathbf{r}}\right] = -\sigma_z^2$.

- Type (I) dependency implies that $\mathbf{z}$ and $\mathbf{w}$ are heavily correlated, thus $\mathbb{E}\left[s_{\mathbf{y},\mathbf{r}}\right] > -\sigma_z^2$. However, for large enough patch-sizes, and when discarding $\mathbf{y}_i$ from the set Γ, this behavior is expected to be rare and can be disregarded.

- We have seen that type (II) dependency leads to negative values of $\sigma_{\mathbf{x},\mathbf{w}}$. Thus, we get that $\mathbb{E}\left[s_{\mathbf{y},\mathbf{r}}\right] < -\sigma_z^2$.

A formal proof of these statements is given in appendix B.

Let us now return to the $s_{\mathbf{y},\mathbf{r}}$ histogram, while assuming that the dependency of type (I) is low. Figure 4 presents two examples of such histograms for two types of noise - more details on these noise realizations is given in the next section. One can easily spot the expression of type (II) dependencies in both – the longer left tail. Note that the histograms are cropped, and the tail is longer than shown in the figures (especially in Figure 4a). As expected, due to this dependency, the histogram mean is shifted left relative to its peak. To reduce this dependency, we cut the left tail by excluding from the training set all image pairs for which $s_{\mathbf{y},\mathbf{r}} < s_{min}$. The threshold s_{min} is set such that the mean of the resulting histogram coincides with its peak. As shown in Figure 5, the dependency reduction substantially boosts denoising performance.

5. Experimental Results

We turn to report the denoising performance of the proposed framework and its comparison with leading self-supervised methods. Our framework is referred to as Patch-Craft (PC). We consider two experiments, one with correlated Gaussian noise and the second with real-world noise. In both experiments, we train networks in an adaptation manner [34], beginning with bias-free networks [19] pre-trained for blind i.i.d. Gaussian denoising task, and retraining them using the proposed patch-craft method. We consider two different architectures for our scheme: DnCNN [41] and U-Net [30]. The first is similar to the one used in [6], while U-Net is taken from [19]. We use bias-free versions of both networks. We denote by PC-DnCNN and PC-UNet the networks retrained using the PC framework, where B-DnCNN and B-UNet stand for their initial versions trained for blind i.i.d. Gaussian denoising.

For comparison, we choose three latest state-of-the-art (SoTA) self-supervised training methods: Recorrupted-to-recorrupted (R2R) [23], Neighbor2Neighbor (N2N) [5], and Blind2Unblind (B2U) [36]. In addition, we show a comparison with BM3D [3], which gets as input parameter the standard deviation, σ, of the noise. Since the noise is not i.i.d. Gaussian, the actual standard deviation is not necessarily the optimal parameter for BM3D. Thus we apply BM3D with two configurations: a BM3D with the actual σ of the noise, and a grid search to find the best performing parameter. We call this configuration oracle BM3D (O-BM3D).

Our algorithm requires bursts of images for training. Therefore, we use datasets containing short video sequences in all experiments. Our analysis suggests that the patch size, n, should be big. Moreover, it should grow with the standard deviation of the noise, σ, and correlation range. Following this, we increase the value of n accordingly. For quantitative evaluation, we choose the commonly used PSNR and SSIM metrics. For more technical details regarding the training, we refer the reader to appendices H and I.

5.1. Correlated Gaussian Denoising

We start with additive correlated Gaussian noise, using the DAVIS dataset [25] at 480p resolution. We train the networks on 90 bursts of length 7 frames, each taken from a different training video sequence at an arbitrary location. In each training burst, the middle frame is used as the network input, whereas the rest 6 are utilized for building the patch-craft targets. For the test, we use frames taken from 30 test video sequences. From each sequence we take 3 non-consecutive frames at arbitrary locations. Each of the obtained 90 test frames is denoised as a single image.

The correlated noise is created by convolving an i.i.d. Gaussian noise with a rectangular flat kernel of size $k \times k$.

The competing methods are trained using the code packages and parameters supplied by the authors. For R2R, we choose $\alpha = 2$ among the three options listed in the original paper (0.5, 2, 20) since it leads to the best denoising results.

Table 1 summarizes the denoising performance for various σ and k values. Figure 6 and Figures 11, 12, and 13 in appendix I show visual comparisons between the denoised images. As can be seen from the table and figures, the current SoTA self-supervised methods with which we compare face difficulties [2] in train networks when the noise is correlated, when the difficulty increases with the correlation range and the intensity of the noise. The classical, signal processing oriented, O-BM3D method achieves relatively high PSNR (typically 1-3 dB below networks trained using our framework). However, as can be seen from the figures, in the case of moderate to severe noise, the visual quality of the O-BM3D outputs leaves much to be desired since the method tends to produce blurred images or leave a noticeable amount of low-frequency noise unfiltered. Not to mention that finding the optimal parameter σ for BM3D when the ground truth targets are unavailable may not be easy.

5.2. Real-World Noise Removal

Real-world noise refers to a particular sensor whose model is unknown, and it's distribution may vary with sensor parameters such as ISO, aperture, exposure time, etc.. Finding a dataset for this evaluation is a challenging task. To the best of our knowledge, there are no burst or video datasets with such noise that include ground truth images. For example, it is impossible to use the popular SIDD [1], DND [24], CC [22], and PolyU [39] datasets, as they only contain single images and not bursts.

For conducting a real-world image denoising experiment, we use the CRVD [40] dataset, which consists of 11 groups of noisy pictures taken in a photo laboratory and their ground truth counterparts. Each group captures a different scene, each scene is captured 7 times, there is some movement between each capture. Each of these groups of 7 images can be considered as an artificial video sequence. However, the movements of objects and changes in lighting captured in these artificial sequences are incomparably sharper than in a typical video or an image burst. Examples of such artificial video sequences are presented in figure 10 in Appendix Appendix I.

When applying the patch-craft framework, it is better to avoid sequences with sharp movements and severe lighting changes since the latter makes patch matching difficult. With that in mind, it is interesting that our framework achieves favorable results even when trained on a small and challenging dataset. Thus, among other things, this experiment indicates the robustness of the proposed method.

[2] B2U [36] training sometimes loses stability, getting extremely low PSNR/SSIM on images contaminated with spatially correlated noise.

Since the CRVD dataset is small, we augment it by replicating each sequence 7 times, where in each replica a different image is used as a middle frame. Then, similarly to the correlated Gaussian denoising experiment, we use the middle frames as network inputs, and use the 6 surrounding frames for building the patch-craft targets. We test the network using the same 77 CRVD frames by comparing their output with the ground truth images. Note that the network can not overfit the ground truth images, as they are not available during training.

Since the competing methods are not designed for training on CRVD, we adapt the CRVD dataset in such a way that each method trains in the conditions close to the ones described in its paper. R2R uses a set containing 400 images of size 180×180 pixels, augmenting it with scaling by 4 factors (1, 0.9, 0.8, 0.7). Thus, for training R2R, we use 4×400 random crops of CRVD images, each crop of size 180×180. Note that we disable the scaling augmentation since scaling may affect the noise statistics. Also, we choose $\alpha = 2$ which leads to the best denoising results. Unlike R2R, N2N and B2U methods train their networks using $44,328$ images from ImageNet [4]. For N2N and B2U we create $44,328$ random crops of the CRVD images, each of size 256×256.

Table 2 summarizes the denoising performance for different ISO values. A visual comparison of the denoised images is shown in Figure 6 and Figures 14, 15, and 16 in Appendix I. These results lead to a similar conclusion as in the correlated Gaussian denoising experiment: The current SoTA self-supervised methods that we compare with face difficulties in training when the contaminating noise is correlated, and this difficulty strengthens with ISO. For high ISO values, our framework outperforms the O-BM3D in terms of PSNR and SSIM, where the latter tends to leave a noticeable amount of low-frequency noise unfiltered.

6. Conclusion

Recent literature pays relatively little attention to the problem of correlated noise reduction in images, probably due to its toughness. Such methods can be used, for instance, for real-world noise reduction in sRGB color space[3], since such noise is usually correlated. This paper proposes a novel self-supervised framework for training a denoiser where the contaminating noise is spatially and cross-channel correlated. The proposed framework relies on the availability of bursts or short video sequences of noisy frames. Our method applies patch matching for building patch-craft images and employs them as training targets. We present a statistical analysis of the target noise that

[3] Note that some self-supervised learning methods, including R2R, B2U, and N2N, do succeed and show good denoising performance in *raw-RGB* color space, since the noise in this space has lower spatial and cross-channel correlations [22, 40]. However, this is not the case in sRGB.

σ	k	Noisy	R2R	N2N	B2U	BM3D	O-BM3D	B-DnCNN	B-UNet	PC-UNet	PC-DnCNN
5	2	34.15 0.852	38.88 0.960	35.20 0.886	29.30 0.720	38.28 0.951	39.69 0.969	37.83 0.945	36.73 0.923	39.27 0.967	39.57 0.969
	3	34.15 0.859	37.29 0.943	34.64 0.879	28.74 0.719	36.50 0.926	38.19 0.957	36.02 0.916	35.25 0.896	38.67 0.964	38.81 0.965
	4	34.16 0.868	36.22 0.930	34.48 0.885	30.46 0.765	35.83 0.920	37.13 0.948	35.33 0.908	34.83 0.894	38.06 0.961	38.31 0.964
10	2	28.13 0.639	34.55 0.902	29.55 0.707	23.65 0.441	33.25 0.867	35.37 0.927	32.82 0.850	31.57 0.799	35.89 0.937	36.10 0.939
	3	28.13 0.653	32.5 0.849	28.85 0.693	23.85 0.454	30.96 0.796	33.56 0.897	30.44 0.774	29.64 0.736	35.16 0.932	35.32 0.934
	4	28.13 0.670	31.21 0.818	28.67 0.705	23.45 0.433	30.16 0.782	32.3 0.872	29.58 0.756	29.07 0.730	34.69 0.931	34.79 0.932
15	2	24.61 0.489	31.81 0.828	26.27 0.567	22.31 0.374	30.28 0.776	32.99 0.886	29.72 0.747	28.56 0.688	33.77 0.907	33.96 0.909
	3	24.61 0.503	29.59 0.747	25.43 0.547	24.44 0.491	27.71 0.671	31.11 0.842	27.12 0.645	26.41 0.606	32.98 0.900	33.16 0.902
	4	24.61 0.521	28.26 0.709	25.26 0.562	22.26 0.398	26.83 0.653	29.79 0.806	26.22 0.623	25.75 0.596	32.4 0.897	32.57 0.899
20	2	22.11 0.387	30.1 0.765	23.82 0.46	20.53 0.304	28.17 0.691	31.41 0.851	27.48 0.655	26.41 0.594	32.28 0.876	32.43 0.879
	3	22.11 0.400	27.57 0.655	23.02 0.443	7.74 0.088	25.39 0.568	29.49 0.796	24.76 0.541	24.14 0.508	31.44 0.869	31.63 0.872
	4	22.11 0.417	25.93 0.599	22.91 0.461	22.33 0.417	24.48 0.548	28.15 0.752	23.84 0.52	23.41 0.496	30.78 0.863	30.97 0.866
Average		27.25 0.605	31.99 0.809	28.18 0.650	23.26 0.467	30.65 0.762	33.27 0.875	30.10 0.740	29.31 0.706	34.62 0.917	34.80 0.919

Table 1. Denoising performance with correlated Gaussian noise. PC-UNet and PC-DnCNN are trained using the proposed patch-craft framework. The best PSNR and SSIM results are marked Red. The second-best results are marked blue.

ISO	σ	Noisy	R2R	N2N	B2U	BM3D	O-BM3D	B-UNet	B-DnCNN	PC-UNet	PC-DnCNN
1600	3.3	37.67 0.925	39.58 0.962	37.71 0.925	36.88 0.915	38.61 0.946	41.12 0.979	37.71 0.926	37.71 0.926	41.25 0.981	41.33 0.981
3200	4.5	35.03 0.874	37.18 0.937	35.10 0.876	4.94 0.011	36.08 0.910	38.99 0.969	35.08 0.876	35.10 0.877	39.50 0.975	39.64 0.974
6400	6.3	32.10 0.794	34.67 0.892	32.19 0.798	4.98 0.002	33.20 0.851	36.57 0.956	32.14 0.795	32.17 0.797	37.18 0.964	37.32 0.965
12800	8.8	29.25 0.690	31.71 0.833	29.33 0.695	23.79 0.451	30.46 0.771	34.30 0.939	29.28 0.691	29.33 0.694	34.89 0.951	35.15 0.952
25600	13.1	25.77 0.506	28.26 0.701	25.86 0.512	20.11 0.326	27.10 0.620	31.35 0.910	25.80 0.507	25.85 0.511	32.45 0.933	32.38 0.932
Average		31.96 0.758	34.28 0.865	32.04 0.761	18.14 0.341	33.09 0.820	36.47 0.951	32.00 0.759	32.03 0.761	37.05 0.961	37.16 0.961

Table 2. Denoising performance with real-world image noise. PC-UNet and PC-DnCNN are trained with the proposed patch-craft framework (σ is the STD using ground truth images). The best PSNR and SSIM results are marked red, and the second-best marked blue.

leads to excluding of faulty image pairs from the training set, thereby boosting the obtained denoising performance. The proposed framework shows outstanding denoising results compared with the recent SoTA self-supervised training algorithms.[4]

[4]The code reproducing the results of this paper is available at https://github.com/grishavak/pcst.

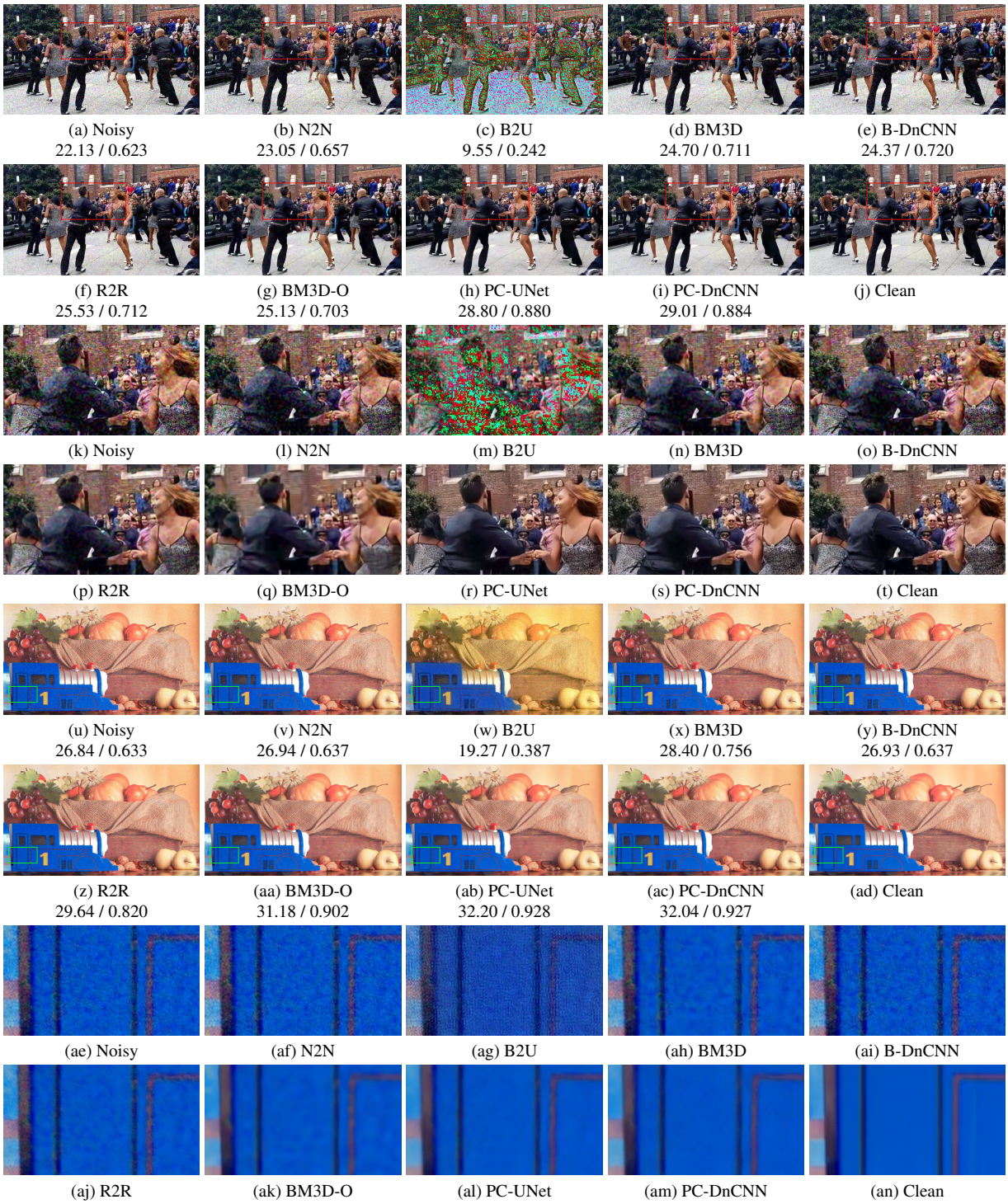


Figure 6. Denoising examples with two types of noise. The first four rows show frame 13 of the sequence *salsa* in the DAVIS dataset with correlated Gaussian noise with $\sigma = 20$ and $k = 3$. The last four rows present frame 4 of scene 10 in the CRVD dataset with IS0 25600. As can be seen, in both experiments, oracle BM3D leaves a substantial amount of low-frequency noise unfiltered. In addition, it produces blurred output for the DAVIS frame. Other algorithms, except ours (PC-UNet and PC-DnCNN), fail to remove the noise, while B2U loses stability during the training.

References

[1] Abdelrahman Abdelhamed, Stephen Lin, and Michael S Brown. A high-quality denoising dataset for smartphone cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1692–1700, 2018. 6

[2] Joshua Batson and Loic Royer. Noise2self: Blind denoising by self-supervision. In *International Conference on Machine Learning*, pages 524–533. PMLR, 2019. 3

[3] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007. 5

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[5] Tao Huang, Songjiang Li, Xu Jia, Huchuan Lu, and Jian zhuo Liu. Neighbor2neighbor: Self-supervised denoising from single noisy images. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14776–14785, 2021. 3, 5

[6] Zahra Kadkhodaie and Eero Simoncelli. Stochastic solutions for linear inverse problems using the prior implicit in a denoiser. *Advances in Neural Information Processing Systems*, 34:13242–13254, 2021. 5

[7] Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2014. 2

[8] Alexander Krull, Tim-Oliver Buchholz, and Florian Jug. Noise2void-learning denoising from single noisy images. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2129–2137, 2019. 2

[9] Alexander Krull, Tomáš Vičar, Mangal Prakash, Manan Lalit, and Florian Jug. Probabilistic noise2void: Unsupervised content-aware denoising. *Frontiers in Computer Science*, 2:5, 2020. 2

[10] Samuli Laine, Tero Karras, Jaakko Lehtinen, and Timo Aila. High-quality self-supervised deep image denoising. In *NeurIPS*, 2019. 3

[11] Wooseok Lee, Sanghyun Son, and Kyoung Mu Lee. Ap-bsn: Self-supervised denoising for real-world images via asymmetric pd and blind-spot network. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17704–17713, 2022. 3

[12] Stamatios Lefkimmiatis. Universal denoising networks : A novel cnn architecture for image denoising. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3204–3213, 2018. 2

[13] Jaakko Lehtinen, Jacob Munkberg, Jon Hasselgren, Samuli Laine, Tero Karras, Miika Aittala, and Timo Aila. Noise2noise: Learning image restoration without clean data. *ArXiv*, abs/1803.04189, 2018. 2, 3

[14] Jingyun Liang, Jie Cao, Guolei Sun, K. Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. *2021 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, pages 1833–1844, 2021. 1

[15] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. *arXiv preprint arXiv:1806.02919*, 2018. 1

[16] Pengju Liu, Hongzhi Zhang, Kai Zhang, Liang Lin, and Wangmeng Zuo. Multi-level wavelet-cnn for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 773–782, 2018. 1

[17] Ali Maleky, Shayan Kousha, M. S. Brown, and Marcus A. Brubaker. Noise2noiseflow: Realistic camera noise modeling without clean images. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 17611–17620, 2022. 2

[18] Xiao-Jiao Mao, Chunhua Shen, and Yubin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NIPS*, 2016. 1

[19] Sreyas Mohan, Zahra Kadkhodaie, Eero P. Simoncelli, and Carlos Fernandez-Granda. Robust and interpretable blind image denoising via bias-free convolutional neural networks. *ArXiv*, abs/1906.05478, 2020. 2, 5

[20] Sreyas Mohan, Joshua L Vincent, Ramon Manzorro, Peter Crozier, Carlos Fernandez-Granda, and Eero Simoncelli. Adaptive denoising via gaintuning. *Advances in Neural Information Processing Systems*, 34:23727–23740, 2021. 2

[21] Nick Moran, Dan Schmidt, Yu Zhong, and Patrick Coady. Noisier2noise: Learning to denoise from unpaired noisy data. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12061–12069, 2020. 2

[22] Seonghyeon Nam, Youngbae Hwang, Yasuyuki Matsushita, and Seon Joo Kim. A holistic approach to cross-channel image noise modeling and its application to image denoising. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1683–1691, 2016. 6

[23] T. Pang, Huan Zheng, Yuhui Quan, and Hui Ji. Recorrupted-to-recorrupted: Unsupervised deep learning for image denoising. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2043–2052, 2021. 2, 5

[24] Tobias Plotz and Stefan Roth. Benchmarking denoising algorithms with real photographs. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1586–1595, 2017. 6

[25] Jordi Pont-Tuset, Federico Perazzi, Sergi Caelles, Pablo Arbeláez, Alex Sorkine-Hornung, and Luc Van Gool. The 2017 davis challenge on video object segmentation. *arXiv preprint arXiv:1704.00675*, 2017. 5

[26] Mangal Prakash, Mauricio Delbracio, Peyman Milanfar, and Florian Jug. Interpretable unsupervised diversity denoising and artefact removal. In *International Conference on Learning Representations*, 2021. 2, 3

[27] Mangal Prakash, Alexander Krull, and Florian Jug. Fully unsupervised diversity denoising with convolutional variational autoencoders. In *ICLR*, 2021. 2

[28] Yuhui Quan, Mingqin Chen, T. Pang, and Hui Ji. Self2self with dropout: Learning self-supervised denoising from single image. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1887–1895, 2020. 3

[29] Herbert E. Robbins. A stochastic approximation method. *Annals of Mathematical Statistics*, 22:400–407, 2007. 4

[30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 5

[31] Shakarim Soltanayev and Se Young Chun. Training deep learning based denoisers without ground truth data. In *NeurIPS*, 2018. 3

[32] Charles M Stein. Estimation of the mean of a multivariate normal distribution. *The annals of Statistics*, pages 1135–1151, 1981. 3

[33] Ying Tai, Jian Yang, Xiaoming Liu, and Chunyan Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017. 1

[34] Gregory Vaksman, Michael Elad, and Peyman Milanfar. Lidia: Lightweight learned image denoising with instance adaptation. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2220–2229, 2020. 2, 5

[35] Gregory Vaksman, Michael Elad, and Peyman Milanfar. Patch craft: Video denoising by deep modeling and patch matching. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2137–2146, 2021. 1, 3

[36] Zejin Wang, Jiazheng Liu, Guoqing Li, and Hua Han. Blind2unblind: Self-supervised image denoising with visible blind spots. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2027–2036, 2022. 5, 6

[37] Yaochen Xie, Zhengyang Wang, and Shuiwang Ji. Noise2same: Optimizing a self-supervised bound for image denoising. *Advances in Neural Information Processing Systems*, 33:20320–20330, 2020. 3

[38] Jun Xu, Yuan Huang, Ming-Ming Cheng, Li Liu, Fan Zhu, Zhou Xu, and Ling Shao. Noisy-as-clean: Learning self-supervised denoising from corrupted image. *IEEE Transactions on Image Processing*, 29:9316–9329, 2020. 2

[39] Jun Xu, Hui Li, Zhetong Liang, David Zhang, and Lei Zhang. Real-world noisy image denoising: A new benchmark. *arXiv preprint arXiv:1804.02603*, 2018. 6

[40] Huanjing Yue, Cong Cao, Lei Liao, Ronghe Chu, and Jingyu Yang. Supervised raw video denoising with a benchmark dataset on dynamic scenes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2301–2310, 2020. 6

[41] Kai Zhang, Wangmeng Zuo, Yunjin Chen, Deyu Meng, and Lei Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017. 1, 2, 5

[42] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Ffdnet: Toward a fast and flexible solution for cnn-based image denoising. *IEEE Transactions on Image Processing*, 27(9):4608–4622, 2018. 1

[43] Yulun Zhang, Yapeng Tian, Yu Kong, Bineng Zhong, and Yun Fu. Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 1

[44] Magauiya Zhussip, Shakarim Soltanayev, and Se Young Chun. Extending stein's unbiased risk estimator to train deep denoisers with correlated pairs of noisy images. In *Neural Information Processing Systems*, 2019. 3