WINNet: Wavelet-inspired Invertible Network for Image Denoising
Jun-Jie Huang, and Pier Luigi Dragotti

Abstract—Image denoising aims to restore a clean image from an observed noisy image. The model-based image denoising approaches can achieve good generalization ability over different noise levels and are with high interpretability. Learning-based approaches are able to achieve better results, but usually with weaker generalization ability and interpretability. In this paper, we propose a wavelet-inspired invertible network (WINNet) to combine the merits of the wavelet-based approaches and learning-based approaches. The proposed WINNet consists of $K$-scale of lifting inspired invertible neural networks (LINNs) and sparsity-driven denoising networks together with a noise estimation network. The network architecture of LINNs is inspired by the lifting scheme in wavelets. LINNs are used to learn a non-linear redundant transform with perfect reconstruction property to facilitate noise removal. The denoising network implements a sparse coding process for denoising. The noise estimation network estimates the noise level from the input image which will be used to adaptively adjust the soft-thresholds in LINNs. The forward transform of LINNs produce a redundant multi-scale representation for denoising. The denoised image is reconstructed using the inverse transform of LINNs with the denoised detail channels and the original coarse channel. The simulation results show that the proposed WINNet method is highly interpretable and has strong generalization ability to unseen noise levels. It also achieves competitive results in the non-blind/blind image denoising and in image deblurring.

Index Terms—Image denoising, Wavelet transform, Invertible neural networks.

I. INTRODUCTION

Image denoising is a classical and fundamental inverse problem in image processing and computer vision. Image denoising algorithms aim to restore a noiseless image from noisy observations obtained by digital cameras. Given that the observations are inevitably noisy due to the random nature of the photon emission and sensing process, and the imperfection of the signal conversion process [1], [2], image denoising is an essential step for further image processing and computer vision applications. With the plug-and-play and the unfolding technique [3]–[7], image denoising algorithms have become even more important as they can also be used to solve other imaging applications due to its ability to provide sparse representation of images. It provides a versatile multi-resolution analysis with perfect reconstruction property and time-frequency localization property. Due to its properties, the wavelet transform has been applied for a wide range of image restoration problems, including image denoising [8], [10], image deconvolution [24], [25] and image inpainting [26], [27]. The wavelet transform is a fixed transform and, in some cases, learning a transformation better adapted to the data at hand may lead to more effective solutions. Dictionary learning tries to achieve that by learning a (redundant) sparseifying transform from training data. Elad and Aharon [12] proposed to perform image denoising with learned redundant dictionaries. Compared to fixed transforms, the combination of learned dictionaries and sparse coding leads to significantly improved denoising performance. However, both the analytical transforms and the learned dictionaries are linear transformations. A suitable non-linear transform with perfect reconstruction property has the potential to achieve better performances.

Besides the nice properties of the wavelet transform, the noise adaptive non-linear operator also contributes to the effectiveness of wavelet-based methods. Donoho and Johnstone [8] proposed soft-thresholding operator and applied it with the optimal “universal threshold” $T = \sqrt{2\sigma^2 \log N}$ (where $N$ is the number of samples) to the wavelet-domain coefficients to remove noise. Chang et al. [9] proposed a BayesShrink threshold $T = \hat{\sigma}^2 / \hat{\sigma}_X$ (where $\sigma$ and $\hat{\sigma}_X$ are the estimated standard deviation of noise and signal, respectively) for soft-thresholding with a Bayesian framework for wavelet-domain image denoising. Both the “universal threshold” and BayesShrink threshold are adaptively adjusted for different noise levels.

The learning-based methods [19]–[23] construct a denoising model by learning from noisy-clean image pairs. The availability of training data and the flexibility of network structure simplifies and enriches the algorithm design and boosts the performance, while the learned models have more restricted

\begin{equation}
y = x + n,
\end{equation}

where $x$ is the clean image and $n \sim N(0, \sigma^2)$ represents the measurement noise with variance $\sigma^2$.

Image denoising has a rich literature. Depending on how priors have been exploited, image denoising algorithms can be generally classified into model-based methods e.g., [8]–[18] and learning-based methods e.g., [19]–[23]. Currently, the learning-based methods achieve state-of-the-art performances, but the model-based methods are still highly competitive.

The model-based methods [8]–[18] use optimization strategies based on well-defined image priors or noise statistics which lead to algorithms with good interpretability and strong generalization ability. The typical priors used for image denoising include for instance image-domain smoothness, transform-domain sparsity [8]–[13], patch-domain non-local self-similarity [15], [16] and low-rank [17], [18].

The wavelet transform has been very effective in many imaging applications due to its ability to provide sparse representation of images. It provides a versatile multi-resolution analysis with perfect reconstruction property and time-frequency localization property. Due to its properties, the wavelet transform has been applied for a wide range of image restoration problems, including image denoising [8], [10], image deconvolution [24], [25] and image inpainting [26], [27]. The wavelet transform is a fixed transform and, in some cases, learning a transformation better adapted to the data at hand may lead to more effective solutions. Dictionary learning tries to achieve that by learning a (redundant) sparseifying transform from training data. Elad and Aharon [12] proposed to perform image denoising with learned redundant dictionaries. Compared to fixed transforms, the combination of learned dictionaries and sparse coding leads to significantly improved denoising performance. However, both the analytical transforms and the learned dictionaries are linear transformations. A suitable non-linear transform with perfect reconstruction property has the potential to achieve better performances.

I. INTRODUCTION

Image denoising is a classical and fundamental inverse problem in image processing and computer vision. Image denoising algorithms aim to restore a noiseless image from noisy observations obtained by digital cameras. Given that the observations are inevitably noisy due to the random nature of the photon emission and sensing process, and the imperfection of the signal conversion process [1], [2], image denoising is an essential step for further image processing and computer vision applications. With the plug-and-play and the unfolding technique [3]–[7], image denoising algorithms have become even more important as they can also be used to solve other image restoration problems by acting as a powerful prior. In this paper, we assume the noise is additive, white and Gaussian. The observed noisy image $y$ is expressed as:

\begin{equation}
y = x + n,
\end{equation}

where $x$ is the clean image and $n \sim N(0, \sigma^2)$ represents the measurement noise with variance $\sigma^2$.

Image denoising has a rich literature. Depending on how priors have been exploited, image denoising algorithms can be generally classified into model-based methods e.g., [8]–[18] and learning-based methods e.g., [19]–[23]. Currently, the learning-based methods achieve state-of-the-art performances, but the model-based methods are still highly competitive.

The model-based methods [8]–[18] use optimization strategies based on well-defined image priors or noise statistics which lead to algorithms with good interpretability and strong generalization ability. The typical priors used for image denoising include for instance image-domain smoothness, transform-domain sparsity [8]–[13], patch-domain non-local self-similarity [15], [16] and low-rank [17], [18].

The wavelet transform has been very effective in many imaging applications due to its ability to provide sparse representation of images. It provides a versatile multi-resolution analysis with perfect reconstruction property and time-frequency localization property. Due to its properties, the wavelet transform has been applied for a wide range of image restoration problems, including image denoising [8], [10], [11], image deconvolution [24], [25] and image inpainting [26], [27]. The wavelet transform is a fixed transform and, in some cases, learning a transformation better adapted to the data at hand may lead to more effective solutions. Dictionary learning tries to achieve that by learning a (redundant) sparseifying transform from training data. Elad and Aharon [12] proposed to perform image denoising with learned redundant dictionaries. Compared to fixed transforms, the combination of learned dictionaries and sparse coding leads to significantly improved denoising performance. However, both the analytical transforms and the learned dictionaries are linear transformations. A suitable non-linear transform with perfect reconstruction property has the potential to achieve better performances.

Besides the nice properties of the wavelet transform, the noise adaptive non-linear operator also contributes to the effectiveness of wavelet-based methods. Donoho and Johnstone [8] proposed soft-thresholding operator and applied it with the optimal “universal threshold” $T = \sqrt{2\sigma^2 \log N}$ (where $N$ is the number of samples) to the wavelet-domain coefficients to remove noise. Chang et al. [9] proposed a BayesShrink threshold $T = \hat{\sigma}^2 / \hat{\sigma}_X$ (where $\sigma$ and $\hat{\sigma}_X$ are the estimated standard deviation of noise and signal, respectively) for soft-thresholding with a Bayesian framework for wavelet-domain image denoising. Both the “universal threshold” and BayesShrink threshold are adaptively adjusted for different noise levels.

The learning-based methods [19]–[23] construct a denoising model by learning from noisy-clean image pairs. The availability of training data and the flexibility of network structure simplifies and enriches the algorithm design and boosts the performance, while the learned models have more restricted
generalization ability compared to the model-based methods and are usually treated as black-box systems with poor interpretability. Therefore, learning-based methods mainly aim to explore more efficient and effective architecture as well as strategies to improve the generalization ability of the learned denoising model. The denoising convolutional neural networks (DnCNN) method [21] learns a deep CNN model with batch normalization layers and a skip connection. By learning from training samples with different noise levels, DnCNN can perform blind image denoising when the noise level of the testing image is within the range of the training noise levels. The fast and flexible denoising network (FFDNet) [22] takes the noise level map and the noisy image as input to the model, and therefore can handle spatially varying noises. The convolutional blind denoising network (CBDNet) [23] consists of two convolution subnetworks: a noise estimation subnetwork and a non-blind denoising subnetwork. The noise estimation subnetwork learns to infer a noise level map from the noisy image, and the non-blind denoising subnetwork which has a U-Net like structure takes both the noise level map and input noisy image as input and estimates the clean image.

Here we propose to learn a (redundant) invertible sparsifying transform by leveraging the invertible neural network framework. In our design we want our non-linear transform to inherit some of the properties of the wavelet transform, including its sparsifying ability as well as its multi-scale property where wavelets at larger scales are more global that the wavelets at finer scales which are instead very local. We aim to combine the merits of the model-based and learning-based image denoising approaches. Specifically by following a strategy similar to wavelet domain thresholding, we aim to enhance the generalization ability and interpretability of the learning-based image denoising method.

We propose a novel wavelet-inspired invertible network (WINNet) for image denoising. The architecture is shown in Fig. 2 and is described in detail in Section III and IV. Rather than learning features with unconstrained CNNs, we propose to learn a non-linear wavelet-like transform with perfect reconstruction property using invertible neural networks with a structure inspired by the lifting scheme [28], [29]. We call these lifting inspired networks LINNs. With PR property, the learned LINNs can serve as a versatile transform. For denoising, a sparsity-driven denoising network is applied to the transform coefficients to remove the noise. The restored image can then be reconstructed using the inverse transform of the learned LINNs.

The proposed WINNet is made of several LINNs, one per scale. Moreover, to achieve good generalization ability, all the soft-thresholds in WINNet are set to be noise adaptive and can be adjusted according to the estimated noise level. In this way, the proposed denoising network achieves good generalization ability even to unseen noise. The noise level is estimated using a model-inspired noise estimation network which exploits low-rank patches on the input noisy image and estimates noise levels as the minimum singular value of the weighted patches.

The contribution of this paper is three-fold:

- We propose an invertible thresholding network for image denoising. It is designed based on the principles of wavelet-based methods, therefore leads to a model with high interpretability. The LINNs at various scales produce a non-linear redundant transform with perfect reconstruction property, and the denoising network implements a basis pursuit denoising process.
- The proposed method is able to achieve blind image denoising. The noise estimation network estimates noise from the input image. To adapt the learned WINNet to the test noise level, the soft-thresholds in network are adaptively adjusted with respect to the estimated noise level.
- The proposed WINNet is with high interpretability and strong generalization ability. It achieves performances comparable to those of state-of-the-art algorithms on non-blind/blind image denoising and image deblurring.

The rest of the paper is organized as follows: Section II discusses related works in image denoising and invertible neural networks. Section III provides an overview of WINNet. Section IV describes the architecture of LINNs which are the building blocks of WINNet, whereas Section V focuses on the denoising network, the noise estimation strategy and the training strategy. Section VI shows the numerical results, compares the proposed method with other image denoising methods and demonstrates the application of WINNet on image deblurring. Finally Section VII concludes the paper.

II. RELATED WORK

A. Learning Universal Image Denoising

Blind image denoising can be achieved by learning from training samples with varying noise levels [21]–[23], however, the learned deep models still have poor generalization ability on images with noise levels beyond the training levels. Hence, one may wish to learn a universal denoising model with good performances on arbitrary noise levels. Mohan et al. [30] found that the bias term in a denoising CNN will lead to a model bias to the training noise levels. Hence, they propose to remove all the bias terms in both ReLU activation functions and the batch normalization layers. The resultant bias-free CNN (BF-CNN) has a scale homogeneity property, i.e., \( f_{BF}(\alpha y) = \alpha f_{BF}(y) \) where \( f_{BF}(\cdot) \) denotes a bias-free CNN and \( \alpha > 0 \), and shows consistently better generalization ability than its counterpart with bias terms. Helou and Süssstrunk [31] proposed a blind universal image fusion denoiser (BUIFD) whose structure is derived from a Bayesian framework and consists of a noise level CNN, a prior CNN and a fusion network. The information of noisy image and image prior are fused based on signal-to-noise ratio (SNR). By explicitly incorporating the noise level, BUIFD shows stronger generalization ability than that of conventional denoising CNNs.

B. Lifting Scheme and Invertible Neural Networks

The lifting scheme introduced in [28], [29] proposes to construct a wavelet transform by first splitting the signal into an even and an odd part. A predictor is used to predict the
Our goal is to design an invertible neural network (INN) where the design choices are made to induce an invertible transform with properties similar to the wavelet transform. In Fig. 1(a) we show one level of a 1-D discrete wavelet transform which is obtained using a filter bank. Here \( y_1[n] \) is a “smooth” version of \( y \) and \( y_2[n] \) contains high-frequency information of \( y[n] \) and is sparse. We know that this decomposition can be achieved using the lifting scheme shown in Fig. 1(b).

The split operation divides the sequence into the even and odd components, “P” predicts the even using the odd leading to a sparse error sequence and “U” updates the odd sequence to make it smoother. The process is then repeated until \( y_1 \) and \( y_2 \) are obtained. “P” and “U” are fixed and depend on the filters \( H_0 \) and \( H_1 \). A two-level wavelet decomposition can also be implemented with the lifting scheme by applying it again on \( y_1[n] \). We note that the equivalent low-pass filter in the two-scale case is \( H_0(z)H_0(z^2) \) and is less local than \( H_0(z) \). It is therefore more adapted at catching more global proprieties of \( y \). The transforms discussed so far are non-redundant but the undecimated wavelet transform can be easily obtained by shifting-out the downsampler in Fig. 1(a) and by upsampling the filter coefficients by \( 2^{j-1} \), with \( j = 1, \ldots, J \), and \( J \) is the number of decomposition levels. The upsampling is essential to keep the multi-resolution property of the wavelet transform.

The architecture of our INN is inspired by the lifting scheme in the wavelet transform and the multi-resolution property of the wavelet transform. A WINNet with \( K \)-level lifting inspired neural network (LINN) is shown in Fig. 2 where we iterate the decomposition only on the coarse version of the image and replace the standard “P” and “U” operations with two convolutional neural networks. Through the use of a denoising strategy we enforce “P” to act as a “Predictor” and “U” as an “Update”. Moreover, at larger scales, we upsample the filters in the convolutional layers to mimic the algorithm à trous [45] used in wavelets theory to implement the wavelet transform. Finally we replace the split operator with a redundant invertible linear operator which leads to a redundant decomposition.

We achieve sparsity by applying a non-invertible denoising operation on the detail coefficients. The denoising operation forces the network to sparsify the detail coefficients so that only noise is removed. The network we use for denoising is also a sparsity-driven network. WINNet is then trained using noisy images as input and ground-truth noiseless images as output. By training the network with paired noisy and clean images, we force the network to sparsify the detail coefficients whilst preserving essential information in the coarse part. Besides leading to an effective and versatile non-linear invertible transform, this approach also leads to a very effective denoising algorithm. To make it more universal we also include a network to estimate the noise variance which in turn changes the soft-threshold in WINNet. In this way, our system adapts automatically to different unknown levels of noise. We also note that our scheme is highly interpretable as we know what each part of the network is doing. In what follows we describe in more detail each element of our network.

### III. Overview

Inspired by this scheme, the invertible neural networks (INNs) [32]–[35] was proposed for memory-efficient back-propagation model and used for constructing flow-based generative models. INNs are bijective function approximators and the intermediate feature representation of INNs can be used to perfectly reconstruct the input signal and the final representation.

### IV. Lifting Inspired Invertible Neural Networks

The lifting inspired invertible neural network (LINN) is used to replace one level of the wavelet transform (see Fig. 2 and Fig. 3). Compared to wavelet transforms and learned dictionaries, LINN can represent more complex features. Similar to the lifting scheme which splits the input signal and then alternates prediction and update [28], [29], LINN consists of splitting/merging operators and learnable predict and update networks.

Since LINN satisfies PR conditions, the obtained transform coefficients contain the same amount of information as the input image. Though noise is not removed in the transform coefficients, the non-linear transform will learn to separate signal and noise into the coarse and detail parts through the denoising operation applied on the detail part. The PR property of LINN also makes it a versatile transform for
image denoising with different noise levels and for other image restoration problems.

A. Splitting/Merging Operator

The splitting operator is used to separate the input image into two parts, and the merging operator performs the inverse of the splitting operator. The splitting/merging operator in current methods [32]–[38], [40], [46] keep the same number of input and output coefficients.

In this paper, we propose to learn LINNs with redundant representations in analogy with the redundant wavelet transform which provides better performance compared to the decimated wavelet transform in image restoration tasks [11]. Therefore, the splitting operator will lead to a redundant representation, and merging operator recovers the input image from the redundant representation.

For an input image $y^k \in \mathbb{R}^{1 \times N_1 \times N_2}$ at the $k$-th scale, the splitting operator $S$ is parameterized by a convolutional kernel $K_s \in \mathbb{R}^{c \times p \times p}$ where $c$ denotes the number of channels and $p$ denotes the spatial filter size. A convolution is performed to obtain multi-channel features $f^k \in \mathbb{R}^{c \times N_1 \times N_2}$:

$$f^k = K_s \otimes y^k,$$  

where $\otimes$ denotes the convolution operator.

The split operation then divides the first $h$ channels of $f^k$ into coarse part $c^k \in \mathbb{R}^{h \times N_1 \times N_2}$, and the remaining $c - h$ channels into detail part $d^k \in \mathbb{R}^{(c-h) \times N_1 \times N_2}$. In this paper, we set the number of coarse channels to $h = 1$, and we denote the splitting operator as:

$$S(y^k, K_s) = (c^k, d^k).$$

The merging operator $M$ represents the inverse of the splitting operator, and is parameterized by a convolutional kernel $K_m \in \mathbb{R}^{1 \times c \times p \times p}$. It first concatenates $c^k$ and $d^k$ and this results in $\hat{f}^k$. A convolution is performed to recover $y^k$:

$$y^k = K_m \otimes \hat{f}^k.$$  

The merging operator can then be denoted as:

$$M(\hat{f}^k, K_m) = y^k.$$  

To achieve invertibility, we can pick $K_s$ so that it can be reshaped to a tight frame with $c \geq p^2$. The merging convolutional kernel then simply corresponds to the transpose of the splitting convolutional kernel. This will ensure the splitting and merging operators are invertible. We can use orthogonal transforms of size $c \times c$ such as orthogonal wavelet transforms, Discrete Cosine Transforms (DCT) and other analytical transforms as the splitting and merging operators. The tight frame is then obtained by picking $p^2$ columns of the original orthogonal matrix.

Besides the analytical transforms, it is also possible to learn an orthogonal matrix with proper parameterization methods, for example, the Cayley transform [43] of a skew-symmetric matrix $\Theta - \Theta^T$ is an orthogonal matrix:

$$K = (I - (\Theta - \Theta^T)) (I + (\Theta - \Theta^T))^{-1},$$

where $I$ is the identity matrix. The corresponding tight frame $K_s$ can therefore be a sub-matrix (i.e., extracting $p^2$ columns) of the learned orthogonal matrix $K$ parameterized by learnable parameters $\Theta \in \mathbb{R}^{c \times c}$.

B. Network Architecture of LINN

At each scale, the Predict/Update networks are shared among the forward and inverse transform of LINN, but are connected with different signs and direction. Fig. 3(a) and Fig. 3(b) shows the schematics of the forward transform and the inverse transform of the $k$-th scale LINN.
respectively. In the forward transform, \((c^k, d^k)\) will be non-linearly transformed to a representation which is more suitable for denoising. After denoising, the denoised detail part and the original coarse part will be transformed back to the original domain using the inverse transform of LINN. In the forward transform (shown in Fig. 3(a)), a Predict network conditioned on the coarse part aims to predict the detail part to make the resultant residual of the detail part sparse. The Update network conditioned on the detail part is used to adjust the coarse part to make it a smoother version of the input image. The Predict and Update networks are applied alternatively to update the coarse and detail parts. Let us denote \((c^k_m, d^k_m) = (c^k, d^k)\). The \(m\)-th pair \((m \in [1, M])\) of update and predict operation can be expressed as:

\[
d^k_m = d^k_{m-1} - P^k_m(c^k_{m-1}) ,
\]

\[
e^k_m = e^k_{m-1} + U^k_m(d^k_m) ,
\]

where \(d^k_m\) and \(e^k_m\) denotes the updated detail part and coarse part using the \(m\)-th Predict network \(P^k_m(\cdot)\) and Update network \(U^k_m(\cdot)\), respectively.

In the inverse transform of the \(k\)-th scale LINN (shown in Fig. 3(b)), \((d^k_{m-1}, e^k_{m-1})\) for \(m \in [1, \cdots, M]\) can be estimated based on \((d^k_m, e^k_m)\) and \(P^k_m(\cdot)\) and \(U^k_m(\cdot)\) as follows:

\[
e^k_{m-1} = e^k_m - U^k_m(d^k_m) ,
\]

\[
d^k_{m-1} = d^k_m + P^k_m(e^k_{m-1}) .
\]

When no lossy operation is applied on \((c^5_{M}, d^5_{M})\), the inverse transform of LINN can, by construction, perfectly recover the inputs of the forward transform with any Predict and Update networks. In this paper, we enforce LINN to learn a sparsifying transformation through denoising.

C. Predict/Update Networks

The Predict and Update networks can be any functions and their properties will not affect the invertibility of LINN. In this paper, we use the same structure for each Predict/Update network which we therefore denote with PUNets. To achieve a better complexity and performance trade-off, we construct PUNets with depth-wise separable convolution (SepConv) layers \([49]\) and soft-thresholding non-linearity \([9]\). As shown in Fig. 4(c), a PUNet consists of an input convolutional layer \(\text{Conv}_i\) with a soft-thresholding, \(J\) residual blocks with SepConv layers, and an output convolutional layer \(\text{Conv}_o\).

Separable Convolution: A depth-wise separable convolution layer \([49]\) consists of a depth-wise convolution layer and a \(1 \times 1\) convolution layer.

Soft-thresholding: The soft-thresholding operator is used as the non-linearity in PUNets. The soft-thresholding operator can be expressed as:

\[
T_{\lambda}(x) = \text{sgn}(x) \max(|x| - \lambda, 0) .
\]

To ensure \(\lambda\) are non-negative, a Softplus function\(^1\) is applied on each learned thresholds.

Multi-scale Property: Multi-resolution signal decomposition is an essential property of the wavelet transform. For an input image, the wavelet transform provides a multi-scale analysis which captures the information at different scales. In order to mimic the wavelet transform, we iterate the LINN-based decomposition on the coarse part. Moreover, we apply

\(^1\)\text{Softplus}(x) = \frac{1}{\beta} \log(1 + \exp(\beta \cdot x))\)
à trous convolution with dilation rate $2^{k-1}$ for the $k$-th scale LINN even though the redundancy factor in our setting is not two. As a result, the larger scale LINN will have a bigger receptive field. Fig. 5 shows the exemplars of dilated $3 \times 3$ filter at two different levels.

V. BLIND IMAGE DENOISING

The denoising network aims to denoise the detail part $d^k_M$ by removing the noise features while retaining the image content. The denoising networks are the only non-invertible components in WINNet, and the noisy information can only be removed there. The denoising networks also force LINNs to be a sparsifying transformation which is desirable for many image processing tasks.

A. Sparsity-Driven Denoising Network

Any existing image denoiser can be applied here, however, as our objective is to achieve an image denoising algorithm with high interpretability, we propose to use a sparsity-driven denoising architecture.

Let us assume that the learned forward transform by LINN is a sparsifying transform and the small number of large coefficients correspond to the image contents. (The forward transformed images by LINN will be shown in Section VI Simulation Results to validate this assumption). The denoising process can then be modeled as a basis pursuit denoising problem. To enrich the representation ability of the sparsity-driven denoising network, we over-parameterize with a convolutional dictionary as:

$$
g = \arg \min_g \frac{1}{2} \||d^k_M - \sum_{i=1}^{N} D_i \otimes g_i||_2^2 + \sum_{i=1}^{N} \lambda_i \||g_i||_1, \quad \text{(12)}$$

where $g = \{g_1, \cdots, g_N\}$ are the sparse features, $D = \{D_1, \cdots, D_N\}$ is the over-parameterized convolutional dictionary, and $\lambda = \{\lambda_1, \cdots, \lambda_N\}$ are the regularization parameters.

The above $l_1$-norm minimization problem can be solved using Iterative Shrinkage-Thresholding Algorithm (ISTA) [50]:

$$
g_t = T_{\lambda/\mu} \left( g_{t-1} + \frac{1}{\mu} D^T \otimes (d^k_M - D \otimes g_{t-1}) \right), \quad \text{(13)}$$

where $\mu$ is the step size.

The LISTA [51] algorithm parameterizes the unknown dictionaries in ISTA as learnable parameters. We apply a convolutional LISTA (CLISTA) as our sparsity-driven denoising network. The schematic of the CLISTA denoising network is shown in Fig. 6. There are $T$ layers of soft-thresholding operations in CLISTA. The convolutional dictionaries $W_s$ and $W_a$ are shared across layers. For the $t$-th layer, there is a soft-thresholding operator with learnable soft-thresholds:

$$
g_t = T_{\lambda_t} \left( g_{t-1} + W_a \otimes (d^k_M - W_s \otimes g_{t-1}) \right), \quad \text{(14)}$$

where $W_a \in \mathbb{R}^{N \times (c-h) \times r \times r}$ is the analysis convolutional dictionary, $W_s \in \mathbb{R}^{(c-h) \times N \times r \times r}$ is the synthesis convolutional dictionary, and $\lambda_t \in \mathbb{R}^N$ is the soft-threshold vector.

For the first layer, $g_0$ is initialized as $W_a \otimes d^k_M$. At the last layer, the synthesis convolution layer $W_s$ is used to reconstruct the denoised detail channels from the estimated sparse feature $g_T$, i.e., $d^k_M = W_s \otimes g_T$.

B. Noise Adaptive Network

To achieve robust image denoising, we propose to adapt the soft-thresholds in PUNets and CLISTA to the noise level $\sigma$ in the image. For images with larger noise levels, the soft-thresholds will be larger leading to more effective denoising. This is similar to the model-based image denoising methods in [8]–[18] where the soft-thresholds scale with the noise levels. For noisy image super-resolution problem, DeepAM [52] also shows that better generalization can be achieved by adjusting soft-thresholds with respect to noise level.

Different from BF-CNN which does not have bias terms and has restricted expressive power, our PUNets and CLISTA denoising network have the dynamically adjusted soft-thresholds with respect to the noise level. This gives us good generalization ability, flexibility as well as improved expressive power.

Spectral Norm Loss: When learning from noisy-clean image pairs of a single noise level $\sigma_N$, the parameters of PUNets are learned to optimize the performance of this training noise level. When the test noise level $\sigma_T < \sigma_N$, the rescaled soft-thresholds could make the PUNets change from a contractive function to an expansive function. We therefore use a spectral norm loss to regularize the property of the PUNets:

$$
\mathcal{L}_s = \frac{1}{K \cdot M \cdot J} \sum_{k=1}^{K} \sum_{m=1}^{M} \sum_{j=1}^{J} \|P^k_{m,j}\|S + \|U^k_{m,j}\|S, \quad \text{(15)}
$$

where $\| \cdot \|S$ denotes the spectral norm of a matrix, and $P^k_{m,j}$ and $U^k_{m,j}$ are the effective linear filter of the $j$-th residual block on the $m$-th lifting step at the $k$-th scale when soft-thresholds are set to zeros.

The linear loss $\mathcal{L}_s$ is used to regularize the Lipschitz constant of the PUNets when $\sigma_T \to 0$, i.e., assuming that soft-thresholding operators in PUNets are identity functions. It can
improve the generalization ability of WINNets to unseen small noise levels and can also stabilize the training of deep/multi-scale models by constraining the expansiveness of the PUNets.

**Orthogonal Loss:** For CLISTA denoising network, the soft-thresholds will also be adjusted with respect to noise level. We would like to enforce $d_M^i = d_M^i$ when $\sigma_T = 0$. That is, the CLISTA denoising network should be close to an identity function when there is no noise. This can be achieved when $W_s$ and $W_a$ are orthogonal. Therefore, an orthogonal loss is imposed between the synthesis and the analysis convolutional dictionary:

$$L_o = \|W_s \otimes W_a - \delta\|_F^2,$$  \hspace{1cm} (16)

where $\delta \in \mathbb{R}^{(c - h) \times (c-h) \times (2r-1) \times (2r-1)}$ is the Kronecker delta function which takes value 1 at coordinate $(i, i, r, r)$ $\forall i \in [1, \cdots, c-h]$ otherwise 0.

### C. Noise Estimation Network

Since the PUNets are conditioned on the image noise level, we propose a simple model-inspired noise estimation network in order to achieve blind image denoising. The proposed noise estimation network is inspired by [53] here, authors use an iterative algorithm to select low-rank patches from the input noisy image and estimate the noise level as the smallest eigenvalue of the covariance matrix of the selected low-rank patches. Compared to CNNs used to infer noise levels, e.g., [25], [31], this model-based method can provide more interpretable results and robust generalization ability.

We propose a learnable noise estimation network (NENet) which is based on the principles of [53] and uses backpropagation through singular value decomposition (SVD) in PyTorch. The network is shown in Fig. 7. The input image is divided into $P \in \mathbb{R}^{s \times s \times N_p}$ patches where each column represents a vectorized $s \times s$ patch and $N_p$ is the number of patches. Instead of iteratively selecting low-rank patches, we propose to use a selection network (SENet) $S : \mathbb{R}^{s^2} \rightarrow \mathbb{R}^1$ to estimate a scalar weight $w_i \in [0, 1]$ for each patch. The SENet performs a soft selection and will learn to assign large weights to low-rank patches and vice versa. Given the estimated weights and the patches, the noise variance is estimated as the minimum singular value of the weighted covariance matrix of the patch matrix:

$$\hat{\sigma}^2 = \frac{1}{\sum_{i=1}^{N_p} w_i} \sigma_{\text{min}},$$  \hspace{1cm} (17)

where $\sigma_{\text{min}}$ denotes the smallest singular value of the weighted covariance matrix $P \text{diag}(w) P^T$ with $w = [w_1, \cdots, w_{N_p}]^T$.

The SENet is composed of blocks with 1-D convolution layer and ReLU non-linearity without bias term, followed by a global averaging pooling and a Sigmoid function to map the weight in the range of $[0, 1]$. The parameters of SENet can be learned using backpropagation. For the noise estimation network, the MSE noise loss is used:

$$L_n = \frac{1}{2N} \sum_{i=1}^{N} (\sigma_i - \hat{\sigma}_i)^2,$$  \hspace{1cm} (18)

where $\sigma_i$ and $\hat{\sigma}_i$ denotes the ground-truth noise level and the estimated noise level, respectively.

### D. Training Details

WINNet can learn from noisy-clean image pairs of a single noise level or a range of noise levels as in [21]–[23]. The average mean squared error (MSE) between the restored image and the clean image is used as the reconstruction loss:

$$L_r = \frac{1}{2N} \sum_{i=1}^{N} \|x_i - \hat{x}_i\|_2^2,$$  \hspace{1cm} (19)

where $x_i$ and $\hat{x}_i$ denotes the input clean image and the denoised image, respectively.

The overall training objective of our WINNet is:

$$L_{\text{all}} = L_r + \lambda_1 L_s + \lambda_2 L_o,$$  \hspace{1cm} (20)

where $\lambda_1$ and $\lambda_2$ are regularization parameters.

### VI. Experimental Results

In this section, we perform experiments to show the properties and validate the effectiveness of the proposed WINNet. We will first introduce the experimental settings, visualize the learned transform and the sparsity-driven denoising network results, and finally perform comparisons with other methods.

#### A. Implementation Details

We follow the training and evaluation settings in [21] to perform experiments. Before training and testing, the clean images are normalized to $[0, 255]$. The noisy images are obtained by adding additive white Gaussian noise to the clean image with respect to Eq. (1) with variance $\sigma^2$.

1) **WINNet Configuration:** For default setting, the convolution kernel $K_s \in \mathbb{R}^{c \times 1 \times p \times p}$ of the splitting operator is reshaped from a DCT transform matrix $T \in \mathbb{R}^{p^2 \times p^2}$ (i.e. the $i$-th row of $T$ is the reshaped $i$-th $p \times p$ filter in $K_s$). We set $c = p^2$ and $p = 4$. The convolution kernel for the merging operator can be constructed in a similar manner. Since by default $p = 4$, there will be 1 coarse channel and 15 detail channels.

The number of update and predict network pairs is set to $M = 4$, and the number of residual blocks in each PUNet is set to $J = 4$. The number of feature channels in PUNet is set to 32 and the spatial filter size in SepConv layers is set to $q = 5$. For CLISTA denoising network, there are $T = 3$ layers with 64 channels. The support of the spatial filter for $W_a$ and $W_s$ is set to $r = 3$.
2) Training and Testing settings: The 400 training images from the Berkeley segmentation dataset (BSD) [54] of size 180 × 180 are used for training. The training patch size is 40 × 40, and the number of patches for training is 2 × 10^5.

For training with a single noise level, three noise levels are considered, i.e., σ_N = 15, 25 and 50. For blind image denoising scenario, the training noise level σ_N is uniformly drawn from [0, 55]. The regularization parameter for spectral norm loss and orthogonal loss is set to λ_1 = 0.1 and λ_2 = 10, respectively. The spectral norm loss is evaluated once every 10 iterations. As discussed in Section V-B, when training blind image denoising models, all the soft-thresholds will be rescaled with respect to the training noise level.

The weights of the convolution layers are initialized using the Kaiming initialization method [55]. The stochastic gradient descent with Adam optimizer [56] is used for training with initial learning rate lr = 1 × 10^{-3} and β = (0.9, 0.999). The total number of training epochs is set to 50 and the learning rate decays from 1 × 10^{-3} to 1 × 10^{-4} at the 30-th epoch. The batch size N is set to 32.

The testing images include the 12 images from Set12 [21], and the 68 natural images from the BSD68 [54]. PSNR is used as the evaluation metric.

B. Visualizing the Intermediate Results

In this section, we will visualize the output produced by components of a WINNet with K = 2 levels which is trained on data with noise level σ_N = 25.

1) Elementary Building Blocks Learned by WINNet: The proposed invertible network is inspired by the lifting scheme [28], [29] in wavelets, therefore, it would be meaningful to visualize the learned elementary atoms.

The basis functions or elementary atoms of the wavelet transform can be visualized by setting to zero all the wavelet coefficients with the exception of one coefficient and then by reconstructing the corresponding signal with the synthesis filter bank. In analogy with the wavelet case, we set all the feature maps in (c_k^M, ˆd_k^M) to zero with the exception of the center pixel on one of the feature maps, and then reconstruct the image using the backward pass of LINN. As the PUNets represent highly non-linear functions and may have different responses for signals with varying amplitudes, the elementary atoms is visualized using different non-zero values. Since our transform is highly non-linear, it is incorrect to call the reconstructed functions “basis functions” or elementary atoms, however, to keep the intuition that what we are visualizing is a variation of the basis functions related to linear transforms, with a slight abuse, we keep calling them “elementary atoms”.

Fig. 8 shows the atoms corresponding to different channels at level 1 and level 2 in WINNet. For better visualization, the maximum absolute value in each visualized basis function has been rescaled by a factor which is shown on the lower left corner of each figure.
Fig. 9. Visualization of the coarse and detail channels. (a) and (f) show the coarse component at two different scales. (b) - (e) and (g) - (j) show the detail channels before and after denoising by CLISTA denoising network in WINNet. Each sub-figure shows the feature map of a detail channel and the corresponding histogram. \( d_{M}^{j} \) denotes the \( j \)-th channel of the detail part.

Fig. 10. Visualization of the selected patches for noise level estimation and the generalization ability of the NENet.

The amplitude increases from 0.01 to 1.25. For the detail channels, the shape of the elementary atoms only slightly changes when we increase the amplitude of the input pixel. Fig. 8 (e) - (h) show the atoms of 4 representative channels from level 2 LINN. Different from what we observed at level 1, the basis functions in level 2 have larger support, and the shapes change from concentrated to spread.

Both level 1 and level 2 functions have non-linear responses to the input amplitude, while the atoms of the level 2 have much larger support compared to that of the level 1. The different support size of the functions at level 1 and level 2 is possibly due to their different functionality in WINNet, and is also a consequence of the dilation of the filter support discussed before and depicted in Fig. 5.

2) Noisy and Denoised Feature Maps: The CLISTA denoising network is the only non-invertible component in WINNet, and is responsible for removing the feature components corresponding to noise. By visualizing the feature maps before and after the denoising network, we can further gain insights into the workings of the proposed WINNet.

In Fig. 9 we show the feature maps of an exemplar Butterfly image from Set12. Each sub-figure shows the feature map of a channel and the corresponding histogram. Fig. 9 (a) (b) (d) show the feature map of 3 exemplar channels output from the level 1 LINN in WINNet. We can see that the coarse channel feature map looks like a natural image but is still with artifacts and the detail channel feature maps before denoising contain both noise and some image contents and the histogram is spread. Fig. 9 (j) shows the region of the selected low-rank patches by the selection network (SENet) for different noise levels, respectively. We can see that SENet tends to select different regions for noise estimation. For low noise level, SENet carefully selects smooth regions, while for high noise level SENet avoids highly textured regions and selects more patches. The results are consistent with that in [53] which uses a heuristic
and iterative algorithm to select low-rank patches. Fig. 10 (c) further shows the noise estimation results for images with noise level $\sigma_T \in [0, 100]$. We can find that the proposed NENet is able to provide highly accurate noise estimation for images with $\sigma_T \in [0, 100]$ though the training data is only with noise level $\sigma_N \in [0, 55]$.  

C. Comparison with Other Methods

We compare the proposed WINNet with several state-of-the-art image denoising algorithms including the model-based methods: BM3D [16], WNNM [17], and EPLL [57], and the learning-based methods: DnCNN [21], FFDNet [22], BUIFD [31], BF-CNN [30]. The model-based methods can handle images with arbitrary noise levels. DnCNN [21] and BUIFD [31] can perform blind image denoising when the test noise level is within the range of the training noise levels, BF-CNN [30] can handle unseen noise levels, while the FFDNet [22] requires as input a noise map of the test noise level.

1) Non-blind Image Denoising: Table I shows the comparison results of different non-blind image denoising methods evaluated on three noise levels ($i.e., \sigma = 15, 25, 50$). All the learning-based methods learn from training samples with the correct noise level. From the table, we can see that the proposed WINNets achieve better performance than the model-based methods $i.e.,$ BM3D [16], WNNM [17], and EPLL [57] by a large margin, and our proposed WINNets also outperforms the deep learning based methods including TNRD [20], DnCNN-S [21], BF-CNN [30] and FFDNet [22]. We also note that the model size of WINNet (1-scale) is only around 30% of the model size of DnCNN, and WINNet (1-scale) achieves comparable or better results than DnCNN-S. With one more level decomposition, WINNet (2-scale) further improves the WINNet (1-scale).

Without the bias terms, BF-CNN [30] has stronger generalization ability to testing images with unseen noise levels than DnCNN-S ($i.e.,$ we will show this in Fig. 11), while in Table I the performance of BF-CNN is around 0.15 dB lower than that of the DnCNN-S. This suggests that the generalization ability is improved at the cost of reduced expressiveness.

WINNet has strong generalization ability to images with unseen noise levels as well. Although WINNet only sees training image pairs with a fixed noise level $\sigma_N,$ its parameters can be adjusted to adapt to unseen noise levels. When the testing noise level $\sigma_T \geq \sigma_N,$ all the soft-thresholds in PUNets and in CLISTA denoising networks are rescaled by a factor $\sigma_T/\sigma_N.$ When $\sigma_T < \sigma_N,$ only the soft-thresholds in CLISTA denoising networks are rescaled by a factor $\sigma_T/\sigma_N.$

Fig. 11 shows the performance of the proposed WINNet, BF-CNN and DnCNN-S which are trained on noise level $\sigma_N = 25,$ while are tested on testing noise levels $\sigma_T \in [5, 195].$ We can see that the performance of DnCNN remains similar when $\sigma_T \leq \sigma_N$ and quickly deteriorates otherwise. For BF-CNN and WINNet, they can well generalize to testing images with noise levels $\sigma_T \neq \sigma_N.$ When $\sigma_T \geq \sigma_N,$ WINNet achieves improved gain compared to BF-CNN. When $\sigma_T = 15,$ the performance of WINNet is inferior to BF-CNN, but when $\sigma_T = 5,$ WINNet achieves around 0.8 dB higher PSNR than BF-CNN.

2) Blind Image Denoising: For blind image denoising, the comparison methods include DnCNN-B [21], BUIFD [31], and BF-CNN [30]. The depth of DnCNN-B and BF-CNN are

| Dataset   | Methods        | Model Size | $\sigma = 15$ | $\sigma = 25$ | $\sigma = 50$ |
|-----------|----------------|------------|---------------|---------------|---------------|
| BSD68     | BM3D [16]      | -          | 31.07         | 28.57         | 25.63         |
|           | WNNM [17]      | -          | 31.37         | 28.83         | 25.87         |
|           | EPLL [57]      | -          | 31.21         | 28.68         | 25.67         |
|           | TNRD [20]      | $27 \times 10^3$ | 31.42 | 28.92 | 25.97         |
|           | DnCNN-S [21]   | $556 \times 10^3$ | **31.71** | 29.20 | 26.22         |
|           | BF-CNN [30]    | $556 \times 10^3$ | 31.58 | 29.08 | 26.12         |
|           | FFDNet [22]    | $556 \times 10^3$ | 32.63 | 29.19 | 26.29         |
|           | WINNet (1-scale) | $173 \times 10^3$ | 31.69 | 29.22 | 26.29         |
|           | WINNet (2-scale) | $347 \times 10^3$ | 31.70 | **29.24** | **26.31**     |
| Set12     | BM3D [16]      | -          | 32.37         | 29.97         | 26.72         |
|           | WNNM [17]      | -          | 32.70         | 30.26         | 27.05         |
|           | EPLL [57]      | -          | 32.14         | 29.69         | 26.47         |
|           | TNRD [20]      | $27 \times 10^3$ | 32.50 | 30.06 | 26.81         |
|           | DnCNN-S [21]   | $556 \times 10^3$ | **32.83** | 30.40 | 27.16         |
|           | BF-CNN [30]    | $556 \times 10^3$ | 32.67 | 30.25 | 27.03         |
|           | FFDNet [22]    | $556 \times 10^3$ | 32.75 | 30.43 | **27.32**     |
|           | WINNet (1-scale) | $173 \times 10^3$ | 32.82 | 30.43 | 27.25         |
|           | WINNet (2-scale) | $347 \times 10^3$ | **32.83** | **30.47** | **27.29**     |
The average PSNR (dB) of different blind image denoising methods trained on BSD400 dataset with noise level $\sigma \in [0, 55]$ and tested on BSD68 and Set12 dataset with noise level $\sigma \in [5, 145]$. (The best and the second best result in each column is in bold and with underline, respectively.)

| Dataset | Methods   | $\sigma = 5$ | $\sigma = 25$ | $\sigma = 45$ | $\sigma = 65$ | $\sigma = 85$ | $\sigma = 105$ | $\sigma = 125$ | $\sigma = 145$ |
|---------|-----------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|
| BSD68   | DnCNN-B [21] | 37.75        | 29.18        | 26.52        | 23.00        | 16.07        | 13.19        | 11.68        | 10.79        |
|         | BF-CNN [30]  | 37.41        | 29.11        | 26.58        | 25.12        | 24.10        | 23.33        | 22.70        | 22.18        |
| Set12   | WINNet (1-scale) | 37.82        | 29.13        | 26.66        | 25.23        | 24.23        | 23.46        | 22.81        | 22.23        |

The training data for all methods is noise with $\sigma \in [5, 145]$. For WINNet, we use the 1-scale model for comparison which has around 173,000 parameters. The number of parameters for NENet is only around 6 x $10^5$. The training data for all methods is BSD400 with AWGN $\sigma_N \in [0, 55]$.

Table II shows the testing results of different methods evaluated with images with noise level $\sigma \in [5, 145]$. We can see that the performance of DnCNN-B [21] is highly competitive when the testing noise level is within the range of the training noise level, while quickly deteriorates otherwise. BUlFD [31] consists of a noise level CNN, a prior CNN and a fusion network which are all based on DnCNN architecture. Its total number of parameters is around 1186 x $10^5$. With explicit noise level learning, BUlFD shows stronger generalization ability towards unseen noise levels compared to DnCNN-B. By removing all bias terms in DnCNN, BF-CNN [30] is able to well generalize beyond the training noise levels, however, it is slightly less effective when $\sigma \in [0, 55]$ compared to DnCNN. With the exception of $\sigma = 25$, WINNet consistently outperforms all the other methods.

Fig. 12 further shows the noisy image with different noise levels and the blind image denoising results by the proposed WINNet. We can see that the proposed WINNet is able to achieve robust denoising not only within the training noise level range (marked in green) but also beyond the training noise levels.

### D. Application on Image Deblurring

With the plug-and-play technique, image denoisers can be applied to solve general image restoration problems [19]–[23], for example, image deblurring. In this case, the goal is to recover a sharp image $x$ from the blurred and noisy observation $y = k \otimes x + n$ where $k$ is the blurring kernel and $n \sim \mathcal{N}(0, \sigma^2)$ represents the measurement noise with variance $\sigma^2$. The image deblurring task can be formulated as the following optimization problem:

$$x = \arg \min_x \frac{1}{2\sigma^2} \|y - k \otimes x\|^2_2 + \lambda \Phi(x),$$

(21)

where $\Phi(\cdot)$ is a prior term and $\lambda$ is the regularization parameter. With half-quadratic splitting [6], the image deblurring problem can be solved by iteratively optimizing two sub-problems:

$$x_k = \arg \min_x \frac{1}{2\sigma^2} \|y - k \otimes x\|^2_2 + \frac{\lambda\sigma^2}{\beta^2} \|x - z_{k-1}\|^2_2,$$

(22)

$$z_k = \arg \min_z \frac{1}{2\beta^2} \|z - x_k\|^2_2 + \Phi(z),$$

(23)

where $\beta = \sqrt{\lambda/\mu}$ is a hyper-parameter and can be interpreted as the noise level if the $z$ sub-problem is treated as Gaussian denoising on $x_k$.

In [6], a CNN-based Gaussian denoiser is used to solve the $z$ sub-problem. The hyper-parameter $\lambda$ is set to be fixed during iterations, while $\mu$ is set to exponentially decay from a large value to the given noise level $\sigma$ with a fixed iteration.
Cameraman process of the proposed method on image performance. Fig. 13 shows an exemplar image deblurring that the proposed method is able to achieve highly competitive with the proposed WINNet. IRCNN [6] and the proposed method evaluated on Set12 image deblurring algorithm with the proposed WINNet. Robust NENet and WINNet, we can achieve image deblurring. The training kernel is the first kernel of size 19×19 from [58] and noise level 2.55 (1%). We can see that the proposed method converges after 8 iterations and the PSNR of $x_k$ and $z_k$ consistently improves and finally reaches a similar result.

### VII. Conclusions

In this paper, we have proposed a wavelet-inspired invertible network (WINNet). It consists of $K$ levels of lifting inspired invertible neural network (LINN) and sparsity-driven denoising networks. LINNs are designed to mimic the nice properties of wavelet transform and are used as a non-linear redundant transform with perfect reconstruction property. For image denoising task, the sparsity-driven denoising network is used to remove the noise in the detail parts of the transform coefficients and the denoising network can be adjusted to adapt to unseen noise levels. Together with a model-inspired noise estimation network, the proposed blind WINNet can achieve robust blind image denoising results beyond the training noise levels. The flexibility of WINNet has also been demonstrated on the image deblurring task.

### Table III

| kernel | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|--------|---|---|---|---|---|---|---|---|
| EPLL   | 31.47 | 31.00 | 31.23 | 29.80 | 32.47 | 32.28 | 31.12 | 30.53 |
| IRCNN  | 32.26 | 31.65 | 30.87 | 31.76 | 31.82 | 31.99 | 31.10 | 31.07 |
| Proposed | 32.37 | 32.04 | 32.03 | 31.52 | 32.47 | 33.17 | 32.02 | 31.87 |

Algorithm 1: Plug-and-Play image deblurring with blind WINNet.

1. **Input:** Input image $y$, kernel $k$, parameter $\lambda$;
2. **Initialize:** $z_0 = y$, $\beta_0 = \text{NENet}(z_0)$, $\beta_1 = 10 \times \beta_0$, $k = 1$;
3. **while** $\beta_k > \beta_0$ **do**
   4. $x_k = \arg\min_x \|y - k \otimes x\|_2^2 + \frac{\lambda k^2}{\beta_k} \|x - z_{k-1}\|_2^2$;
   5. $\beta_{k+1} = \text{NENet}(x_k)$;
   6. $z_k = \text{WINNet}(x_k, 2\beta_{k+1})$;
   7. $k = k + 1$;
4. **end**
5. **Output:** Deblurred image $x = z_{k-1}$.

Table III shows the average PSNR (dB) of different image deblurring methods evaluated on Set12 with kernels from [58] and noise level 2.55 (1%).

Number. Since the proposed NENet is an effective noise level estimator and WINNet can denoise images with noise beyond training noise levels, we propose to use WINNet for image deblurring. The $x$ sub-problem can be solved with closed-form solution with the estimated noise level $\beta_k$ by NENet. The $z$ sub-problem can be solved using the proposed blind WINNet with noise level $2\beta_k$ (perform denoising with a stronger strength to ensure convergence). With the proposed robust NENet and WINNet, we can achieve image deblurring without accessing the noise levels and using the pre-defined regularization parameters; $\lambda$ is the only free parameter and is set to 0.23 as in [6]. Algorithm 1 illustrates the plug-and-play image deblurring algorithm with the proposed WINNet.

References

[1] T. Brooks, B. Mildenhall, T. Xue, J. Chen, D. Sharlet, and J. T. Barron, “Unprocessing images for learned raw denoising,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11 036–11 045.
[2] Y. Wang, H. Huang, Q. Xu, J. Liu, Y. Liu, and J. Wang, “Practical deep raw image denoising on mobile devices,” in *European Conference on Computer Vision*. Springer, 2020, pp. 1–16.
[3] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, “Plug-and-play priors for model based reconstruction,” in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 945–948.
[4] C. A. Metzler, A. Maleki, and R. G. Baraniuk, “BM3D-AMP: A new image recovery algorithm based on BM3D denoising,” in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 3116–3120.
[53] X. Liu, M. Tanaka, and M. Okutomi, “Single-image noise level estimation for blind denoising,” *IEEE transactions on image processing*, vol. 22, no. 12, pp. 5226–5237, 2013.

[54] S. Roth and M. J. Black, “Fields of experts,” *International Journal of Computer Vision*, vol. 82, no. 2, p. 205, 2009.

[55] K. He, X. Zhang, S. Ren, and J. Sun, “Delving deep into rectifiers: Surpassing human-level performance on imagenet classification,” in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.

[56] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *CoRR*, vol. abs/1412.6, 2014.

[57] D. Zoran and Y. Weiss, “From learning models of natural image patches to whole image restoration,” in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 479–486.

[58] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, “Understanding and evaluating blind deconvolution algorithms,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 1964–1971.