

Learning Deep CNN Denoiser Prior for Image Restoration

Kai Zhang^{1,2}, Wangmeng Zuo¹, Shuhang Gu², Lei Zhang²

¹School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

²Dept. of Computing, The Hong Kong Polytechnic University, Hong Kong, China

cskaizhang@gmail.com, wmzuo@hit.edu.cn, shuhanggu@gmail.com, cslzhang@comp.polyu.edu.hk

Abstract

Model-based optimization methods and discriminative learning methods have been the two dominant strategies for solving various inverse problems in low-level vision. Typically, those two kinds of methods have their respective merits and drawbacks, e.g., model-based optimization methods are flexible for handling different inverse problems but are usually time-consuming with sophisticated priors for the purpose of good performance; in the meanwhile, discriminative learning methods have fast testing speed but their application range is greatly restricted by the specialized task. Recent works have revealed that, with the aid of variable splitting techniques, denoiser prior can be plugged in as a modular part of model-based optimization methods to solve other inverse problems (e.g., deblurring). Such an integration induces considerable advantage when the denoiser is obtained via discriminative learning. However, the study of integration with fast discriminative denoiser prior is still lacking. To this end, this paper aims to train a set of fast and effective CNN (convolutional neural network) denoisers and integrate them into model-based optimization method to solve other inverse problems. Experimental results demonstrate that the learned set of denoisers not only achieve promising Gaussian denoising results but also can be used as prior to deliver good performance for various low-level vision applications.

1. Introduction

Image restoration (IR) has been a long-standing problem for its highly practical value in various low-level vision applications [1, 9, 47]. In general, the purpose of image restoration is to recover the latent clean image \mathbf{x} from its degraded observation $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{v}$, where \mathbf{H} is a degradation matrix, \mathbf{v} is additive white Gaussian noise of standard deviation σ . By specifying different degradation matrices, one can correspondingly get different IR tasks. Three classical IR tasks would be image denoising when \mathbf{H} is an identity matrix, image deblurring when \mathbf{H} is a blurring operator,

image super-resolution when \mathbf{H} is a composite operator of blurring and down-sampling.

Since IR is an ill-posed inverse problem, the prior which is also called regularization needs to be adopted to constraint the solution space [50, 66]. From a Bayesian perspective, the solution $\hat{\mathbf{x}}$ can be obtained by solving a Maximum A Posteriori (MAP) problem,

$$\hat{\mathbf{x}} = \arg \max_{\mathbf{x}} \log p(\mathbf{y}|\mathbf{x}) + \log p(\mathbf{x}) \quad (1)$$

where $\log p(\mathbf{y}|\mathbf{x})$ represents the log-likelihood of observation \mathbf{y} , $\log p(\mathbf{x})$ delivers the prior of \mathbf{x} and is independent of \mathbf{y} . More formally, Eqn. (1) can be reformulated as

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda \Phi(\mathbf{x}) \quad (2)$$

where the solution minimizes an energy function composed of a fidelity term $\frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$, a regularization term $\Phi(\mathbf{x})$ and a trade-off parameter λ . The fidelity term guarantees the solution accords with the degradation process, while the regularization term enforces desired property of the output.

Generally, the methods to solve Eqn. (2) can be divided into two main categories, *i.e.*, model-based optimization methods and discriminative learning methods. The model-based optimization methods aim to directly solve Eqn. (2) with some optimization algorithms which usually involve a time-consuming iterative inference. On the contrary, discriminative learning methods try to learn the prior parameters Θ and a compact inference through an optimization of a loss function on a training set containing degraded-clean image pairs [2, 13, 51, 55, 57]. The objective is generally given by

$$\min_{\Theta} \ell(\hat{\mathbf{x}}, \mathbf{x}) \quad s.t. \quad \hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda \Phi(\mathbf{x}; \Theta) \quad (3)$$

Because the inference is guided by the MAP estimation, we refer to such methods as MAP inference guided discriminative learning methods. By replacing the MAP inference with a predefined nonlinear function $\hat{\mathbf{x}} = f(\mathbf{y}, \mathbf{H}; \Theta)$, one

can treat the plain discriminative learning methods as general case of Eqn. (3). It can be seen that one obvious difference between model-based optimization method and discriminative learning method is that, the former is flexible to handle various IR tasks by specifying degradation matrix \mathbf{H} , whereas the later needs to use the training data with certain degradation matrices to learn the model. As a consequence, different from model-based optimization methods which have flexibility to handle different IR tasks, discriminative learning methods are usually restricted by specialized tasks. For example, model-based optimization methods such as NCSR [22] are flexible to handle denoising, super-resolution and deblurring, whereas discriminative learning methods MLP [8], SRCNN [21], DCNN [62] are designed for those three tasks, respectively. Even for a specific task such as denoising, model-based optimization methods (*e.g.*, BM3D [17] and WNNM [29]) can handle different noise levels, whereas discriminative learning method of [34] separately train a different model for each level.

With the sacrifice of flexibility, however, discriminative learning methods can not only enjoy a fast testing speed but also tend to deliver promising performance due to the joint optimization and end-to-end training. On the contrary, model-based optimization methods are usually time-consuming with sophisticated priors for the purpose of good performance [27]. As a result, those two kinds of methods have their respective merits and drawbacks, and thus it would be attractive to investigate their integration which leverages their respective merits. Fortunately, with the aid of variable splitting techniques, such as alternating direction method of multipliers (ADMM) method [5] and half-quadratic splitting (HQS) method [28], it is possible to deal with fidelity term and regularization term separately [44], and particularly, the regularization term only corresponds to a denoising subproblem [18, 31, 61]. Consequently, this enables an integration of any discriminative denoisers into model-based optimization methods. However, to the best of our knowledge, the study of integration with discriminative denoiser is still lacking.

This paper aims to train a set of fast and effective discriminative denoisers and integrate them into model-based optimization methods to solve other inverse problems. Rather than learning MAP inference guided discriminative models, we instead adopt plain convolutional neural networks (CNN) to learn the denoisers, so as to take advantage of recent progress in CNN as well as the merit of GPU computation. Particularly, several CNN techniques, including Rectifier Linear Units (ReLU) [37], batch normalization [32], Adam [36], dilated convolution [63] are adopted into the network design or training. As well as providing good performance for image denoising, the learned set of denoisers are plugged in a model-based optimization method to tackle various inverse problems.

The contribution of this work is summarized as follows:

- We trained a set of fast and effective CNN denoisers. With variable splitting technique, the powerful denoisers can bring strong image prior into model-based optimization methods.
- The learned set of CNN denoisers are plugged in as a modular part of model-based optimization methods to tackle other inverse problems. Extensive experiments on classical IR problems, including deblurring and super-resolution, have demonstrated the merits of integrating flexible model-based optimization methods and fast CNN-based discriminative learning methods.

2. Background

2.1. Image Restoration with Denoiser Prior

There have been several attempts to incorporate denoiser prior into model-based optimization methods to tackle with other inverse problems. In [19], the authors used Nash equilibrium to derive an iterative decoupled deblurring BM3D (IDDBM3D) method for image deblurring. In [24], a similar method which is equipped with CBM3D denoiser prior was proposed for single image super-resolution (SISR). By iteratively updating a back-projection step and a CBM3D denoising step, the method has an encouraging performance for its PSNR improvement over SRCNN [21]. In [18], the augmented Lagrangian method was adopted to fuse the BM3D denoiser into an image deblurring scheme. With a similar iterative scheme to [19], a plug-and-play priors framework based on ADMM method was proposed in [61]. Here we note that, prior to [61], a similar idea of plug-and-play is also mentioned in [66] where a half quadratic splitting (HQS) method was proposed for image denoising, deblurring and inpainting. In [31], the authors used an alternative to ADMM and HQS, *i.e.*, the primal-dual algorithm [11], to decouple fidelity term and regularization term. Some of the other related work can be found in [6, 12, 48, 49, 54, 58]. All the above methods have shown that the decouple of the fidelity term and regularization term can enable a wide variety of existing denoising models to solve different image restoration tasks.

We can see that the denoiser prior can be plugged in an iterative scheme via various ways. The common idea behind those ways is to decouple the fidelity term and regularization term. For this reason, their iterative schemes generally involve a fidelity term related subproblem and a denoising subproblem. In the next subsection, we will use HQS method as an example due to its simplicity. It should be noted that although the HQS can be viewed as a general way to handle different image restoration tasks, one can also incorporate the denoiser prior into other convenient and proper optimization methods for a specific application.

2.2. Half Quadratic Splitting (HQS) Method

Basically, to plug the denoiser prior into the optimization procedure of Eqn. (2), the variable splitting technique is usually adopted to decouple the fidelity term and regularization term. In half quadratic splitting method, by introducing an auxiliary variable \mathbf{z} , Eqn. (2) can be reformulated as a constrained optimization problem which is given by

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}} \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda\Phi(\mathbf{z}) \quad s.t. \quad \mathbf{z} = \mathbf{x} \quad (4)$$

Then, HQS method tries to minimize the following cost function

$$\mathcal{L}_{\mu}(\mathbf{x}, \mathbf{z}) = \frac{1}{2} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \lambda\Phi(\mathbf{z}) + \frac{\mu}{2} \|\mathbf{z} - \mathbf{x}\|^2 \quad (5)$$

where μ is a penalty parameter which varies iteratively in a non-descending order. Eqn. (5) can be solved via the following iterative scheme,

$$\begin{cases} \mathbf{x}_{k+1} = \arg \min_{\mathbf{x}} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 + \mu \|\mathbf{x} - \mathbf{z}_k\|^2 & (6a) \\ \mathbf{z}_{k+1} = \arg \min_{\mathbf{z}} \frac{\mu}{2} \|\mathbf{z} - \mathbf{x}_{k+1}\|^2 + \lambda\Phi(\mathbf{z}) & (6b) \end{cases}$$

As one can see, the fidelity term and regularization term are decoupled into two individual subproblems. Specifically, the fidelity term is associated with a quadratic regularized least-squares problem (*i.e.*, Eqn. (6a)) which has various fast solutions for different degradation matrices. A direct solution is given by

$$\mathbf{x}_{k+1} = (\mathbf{H}^T \mathbf{H} + \mu \mathbf{I})^{-1} (\mathbf{H}^T \mathbf{y} + \mu \mathbf{z}_k) \quad (7)$$

The regularization term is involved in Eqn. (6b) which can be rewritten as

$$\mathbf{z}_{k+1} = \arg \min_{\mathbf{z}} \frac{1}{2(\sqrt{\lambda/\mu})^2} \|\mathbf{x}_{k+1} - \mathbf{z}\|^2 + \Phi(\mathbf{z}) \quad (8)$$

According to Bayesian probability, Eqn. (8) corresponds to denoising the image \mathbf{x}_{k+1} by a Gaussian denoiser with noise level $\sqrt{\lambda/\mu}$. As a consequence, any Gaussian denoisers can be acted as a modular part to solve Eqn. (2). To address this, we rewrite Eqn. (8) by following

$$\mathbf{z}_{k+1} = \text{Denoiser}(\mathbf{x}_{k+1}, \sqrt{\lambda/\mu}) \quad (9)$$

It is worth noting that, according to Eqns. (8) and (9), the image prior $\Phi(\cdot)$ can be implicitly replaced by a denoiser prior. Such a promising property actually offers several advantages. First, it enables to use any gray or color denoisers to solve a variety of inverse problems. Second, the explicit image prior $\Phi(\cdot)$ can be unknown in solving Eqn. (2). Third, several complementary denoisers which exploit different image priors can be jointly utilized to solve one specific problem. Note that this property can be also employed in other optimization methods (*e.g.*, iterative shrinkage/thresholding algorithms ISTA [4, 14] and FISTA [3]) as long as there involves a denoising subproblem.

3. Learning Deep CNN Denoiser Prior

3.1. Why Choose CNN Denoiser?

As the regularization term of Eqn. (2) plays a vital role in restoration performance, the choice of denoiser priors thus would be pretty important in Eqn. (9). Existing denoiser priors that have been adopted in model-based optimization methods to solve other inverse problems include total variation (TV) [10, 43], Gaussian mixture models (GMM) [66], K-SVD [25], non-local means [7] and BM3D [17]. Such denoiser priors have their respective drawbacks. For example, TV can create watercolor-like artifacts; K-SVD denoiser prior suffers high computational burden; non-local means and BM3D denoiser priors may over-smooth the irregular structures if the image does not exhibit self-similarity property. Thus, strong denoiser prior which can be implemented efficiently is highly demanded.

Regardless of the speed and performance, color image prior or denoiser is also a key factor that needs to be taken into account. This is because most of the images acquired by modern cameras or transmitted in internet are in RGB format. Due to the correlation between different color channels, it has been acknowledged that jointly handling the color channels tends to produce better performance than independently dealing with each color channel [26]. However, existing methods mainly focus on modeling gray image prior and there are only a few works concentrating on modeling color image prior (see, *e.g.*, [16, 41, 46]). Perhaps the most successful color image prior modeling method is CBM3D [16]. It first decorrelates the image into a luminance-chrominance color space by a hand-designed linear transform and then applies the gray BM3D method in each transformed color channels. While CBM3D is promising for color image denoising, it has been pointed out that the resulting transformed luminance-chrominance color channels still remain some correlation [42] and it is preferable to jointly handle RGB channels. Consequently, instead of utilizing the hand-designed pipeline, using discriminative learning methods to automatically reveal the underlying color image prior would be a good alternative.

By considering the speed, performance and discriminative color image prior modeling, we choose deep CNN to learn the discriminative denoisers. The reasons of using CNN are four-fold. First, the inference of CNN is very efficient due to the parallel computation ability of GPU. Second, CNN exhibits powerful prior modeling capacity with deep architecture. Third, CNN exploits the external prior which is complementary to the internal prior of many existing denoisers such as BM3D. In other words, a combination with BM3D is expected to improve the performance. Fourth, great progress in training and designing CNN have been made during the past few years and we can take advantage of those progress to facilitate discriminative learning.

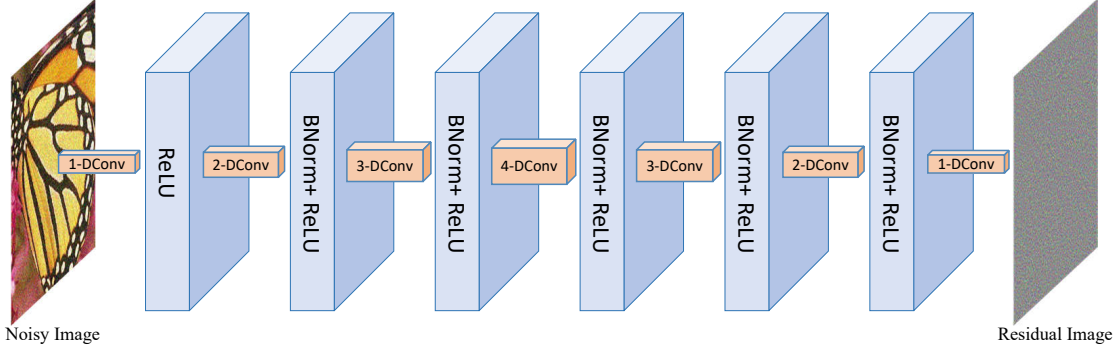


Figure 1. The architecture of the proposed denoiser network. Note that “ s -DConv” denotes s -dilated convolution [63], here $s = 1, 2, 3$ and 4; “BNorm” represents batch normalization [32]; “ReLU” is the rectified linear units ($\max(\cdot, 0)$).

3.2. The Proposed CNN Denoiser

The architecture of the proposed CNN denoiser is illustrated in Figure 1. It consists of seven layers with three different blocks, *i.e.*, “Dilated Convolution+ReLU” block in the first layer, five “Dilated Convolution+Batch Normalization+ReLU” blocks in the middle layers, and “Dilated Convolution” block in the last layer. The dilation factors of (3×3) dilated convolutions from first layer to the last layer are set to 1, 2, 3, 4, 3, 2 and 1, respectively. The number of feature maps in each middle layer is set to 64. In the following, we will give some important details in our network design and training.

Using Dilated Filter to Enlarge Receptive Field. It has been widely acknowledged that the context information facilitates the reconstruction of the corrupted pixel in image denoising. In CNN, to capture the context information, it successively enlarges the receptive field through the forward convolution operations. Generally, there are two basic ways to enlarge the receptive field of CNN, *i.e.*, increasing the filter size and increasing the depth. However, increasing the filter size would not only introduce more parameters but also increase the computational burden [53]. Thus, using 3×3 filter with a large depth is popularized in existing CNN network design [30, 35, 56]. In this paper, we instead use the recent proposed dilated convolution to make a tradeoff between the size of receptive field and network depth. Dilated convolution is known for its expansion capacity of the receptive field while keeping the merits of traditional 3×3 convolution. A dilated filter with dilation factor s can be simply interpreted as a sparse filter of size $(2s+1) \times (2s+1)$ where only 9 entries of fixed positions can be non-zeros. Hence, the equivalent receptive field of each layer is 3, 5, 7, 9, 7, 5 and 3. Consequently, it can be easily obtained that the receptive field of the proposed network is 33×33 . If the traditional 3×3 convolution filter is used, the network will either have a receptive field of size 15×15 with the same network depth (*i.e.*, 7) or have a depth of 16 with the same receptive field (*i.e.*, 33×33). To show the advantage of our

design over the above two cases, we have trained three different models on noise level 25 with same training settings. It turns out that our designed model can have an average P-SNR of 29.15dB on BSD68 dataset [50], which is much better than 28.94dB of 7 layers network with traditional 3×3 convolution filter and very close to 29.20dB of 16 layers network.

Using Batch Normalization and Residual Learning to Accelerate Training.

While advanced gradient optimization algorithms can accelerate training and improve the performance, the architecture design is also an important factor. Batch normalization and residual learning which are two of the most influential architecture design techniques have been widely adopted in recent CNN architecture designs. In particular, it has been pointed out that the combination of batch normalization and residual learning is particularly helpful for Gaussian denoising since they are beneficial to each other. To be specific, it not only enables fast and stable training but also tends to result in better denoising performance [65]. In this paper, such strategy is adopted and we empirically find it also can enable fast transfer from one model to another with different noise level.

Using Training Samples with Small Size to Help Avoid Boundary Artifacts.

Due to the characteristic of convolution, the denoised image of CNN may introduce annoying boundary artifacts without proper handling. There are two common ways to tackle with this, *i.e.*, symmetrical padding and zero padding. We adopt the zero padding strategy and wish the designed CNN has the capacity to model image boundary. Note that the dilated convolution with dilation factor 4 in the fourth layer pads 4 zeros in the boundaries of each feature map. We empirically find that using training samples with small size can help avoid boundary artifacts. The main reason lies in the fact that, rather than using training patches of large size, cropping them into small patches can enable CNN to see more boundary information. For example, by cropping an image patch of size 70×70 into four small non-overlap patches of size 35×35 , the boundary in-

formation would be largely augmented. We also have tested the performance by using patches of large size, we empirically find this does not improve the performance. However, if the size of the training patch is smaller than the receptive field, the performance would decrease.

Learning Specific Denoiser Model with Small Interval Noise Levels. Since the iterative optimization framework requires various denoiser models with different noise levels, a practical issue on how to train the discriminative models thus should be taken into consideration. Various studies have shown that if the exact solutions of subproblems (*i.e.*, Eqn. (6a) and Eqn. (6b)) are difficult or time-consuming to optimize, then using an inexact but fast subproblem solution may accelerate the convergence [39, 66]. In this respect, there is no need to learn many discriminative denoiser models for each noise level. On the other hand, although Eqn. (9) is a denoiser, it has a different goal from the traditional Gaussian denoising. The goal of traditional Gaussian denoising is to recover the latent clean image, however, the denoiser here just acts its own role regardless of the noise type and noise level of the image to be denoised. Therefore, the ideal discriminative denoiser in Eqn. (9) should be trained by current noise level. As a result, there is tradeoff to set the number of denoisers. In this paper, we trained a set of denoisers on noise level range [0, 50] and divided it by a step size of 2 for each model, resulting in a set of 25 denoisers for each gray and color image prior modelling. Due to the iterative scheme, it turns out the noise level range of [0, 50] is enough to handle various image restoration problems. Especially noteworthy is the number of the denoisers which is much less than that of learning different models for different degradations.

4. Experiments

The Matlab source code of the proposed method can be downloaded at <https://github.com/cszn/ircnn>.

4.1. Image Denoising

It is widely acknowledged that convolutional neural networks generally benefit from the availability of large training data. Hence, instead of training on a small dataset consisting of 400 Berkeley segmentation dataset (BSD) images of size 180×180 [13], we collect a large dataset which includes 400 BSD images, 400 selected images from validation set of ImageNet database [20] and 4,744 images of Waterloo Exploration Database [40]. We empirically find using large dataset does not improve the PSNR results of BSD68 dataset [50] but can slightly improve the performance of other testing images. We crop the images into small patches of size 35×35 and select $N=256 \times 4,000$ patches for training. As for the generation of corresponding noisy patches, we achieve this by adding additive Gaussian noise to the

Table 1. The average PSNR(dB) results of different methods on (gray) BSD68 dataset.

Methods	BM3D	WNNM	TNRD	MLP	Proposed
$\sigma = 15$	31.07	31.37	31.42	-	31.63
$\sigma = 25$	28.57	28.83	28.92	28.96	29.15
$\sigma = 50$	25.62	25.87	25.97	26.03	26.19

Table 2. The average PSNR(dB) results of CBM3D and proposed CNN denoiser on (color) BSD68 dataset.

Noise Level	5	15	25	35	50
CBM3D	40.24	33.52	30.71	28.89	27.38
Proposed	40.36	33.86	31.16	29.50	27.86

clean patches during training. Since the residual learning strategy is adopted, we use the following loss function,

$$\ell(\Theta) = \frac{1}{2N} \sum_{i=1}^N \|f(\mathbf{y}_i; \Theta) - (\mathbf{y}_i - \mathbf{x}_i)\|_F^2 \quad (10)$$

where $\{(\mathbf{y}_i, \mathbf{x}_i)\}_{i=1}^N$ represents N noisy-clean patch pairs. To optimize the network parameters Θ , the Adam solver [36] is adopted. The step size is started from $1e-3$ and then fixed to $1e-4$ when the training error stops decreasing. The training was terminated if the training error was fixed in five sequential epochs. For the other hyperparameters of Adam, we use their default setting. The mini-batch size is set to 256. Rotation or/and flip based data augmentation is used during mini-batch learning. The denoiser models are trained in Matlab (R2015b) environment with MatConvNet package [60] and an Nvidia Titan X GPU. To reduce the whole training time, once a model is obtained, we initialize the adjacent denoiser with this model. It takes about three days to train the set of denoiser models.

We compared the proposed denoiser with several state-of-the-art denoising methods, including two model-based optimization methods (*i.e.*, BM3D [17] and WNNM [29]), two discriminative learning methods (*i.e.*, MLP [8] and TNRD [13]). The gray image denoising results of different methods on BSD68 dataset are shown in Table 1. It can be seen that WNNM, MLP and TNRD can outperform BM3D by about 0.3dB in PSNR. However, the proposed CNN denoiser can have a PSNR gain of about 0.2dB over those three methods. Table 2 shows the color image denoising results of benchmark CBM3D and our proposed CNN denoiser, it can be seen that the proposed denoiser consistently outperforms CBM3D by a large margin. Such a promising result can be attributed to the powerful color image prior modeling capacity of CNN.

For the run time, we compared with BM3D and TNRD due to their potential value in practical applications. Since the proposed denoiser and TNRD support parallel computation on GPU, we also give the GPU run time. To make a further comparison with TNRD under similar PSNR perfor-

mance, we additionally provide the run time of the proposed denoiser where each middle layer has 24 feature maps. We use the Nvidia cuDNN-v5 deep learning library to accelerate the GPU computation and the memory transfer time between CPU and GPU is not considered. Table 3 shows the run times of different methods for denoising images of size 256×256 , 512×512 and 1024×1024 with noise level 25. We can see that the proposed denoiser is very competitive in both CPU and GPU implementation. It is worth emphasizing that the proposed denoiser with 24 feature maps of each layer has a comparable PSNR of 28.94dB to TNRD but delivers a faster speed. Such a good compromise between speed and performance over TNRD is properly attributed to the following three reasons. First, the adopted 3×3 convolution and ReLU nonlinearity are simple yet effective and efficient. Second, in contrast to the stage-wise architecture of TNRD which essentially has a bottleneck in each immediate output layer, ours encourages a fluent information flow among different layers, thus having larger model capacity. Third, batch normalization which is beneficial to Gaussian denoising is adopted. According to the above discussions, we can conclude that the proposed denoiser is a strong competitor against BM3D and TNRD.

Table 3. Run time (in seconds) of different methods on images of size 256×256 , 512×512 and 1024×1024 with noise level 25.

Size	Device	BM3D	TNRD	Proposed ₂₄	Proposed ₆₄
256×256	CPU	0.66	0.47	0.10	0.310
	GPU	-	0.010	0.006	0.012
512×512	CPU	2.91	1.33	0.39	1.24
	GPU	-	0.032	0.016	0.038
1024×1024	CPU	11.89	4.61	1.60	4.65
	GPU	-	0.116	0.059	0.146

4.2. Image Deblurring

As a common setting, the blurry images are synthesized by first applying a blur kernel and then adding additive Gaussian noise with noise level σ . In addition, we assume the convolution is carried out with circular boundary conditions. Thus, an efficient implementation of Eqn. (7) by using Fast Fourier Transform (FFT) can be employed. To make a thorough evaluation, we consider three blur kernels, including a commonly-used Gaussian kernel with standard deviation 1.6 and the first two of the eight real blur kernels from [38]. As shown in Table 4, we also consider Gaussian

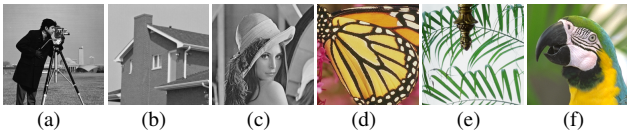


Figure 2. Six testing images for image deblurring. (a) *Cameraman*; (b) *House*; (c) *Lena*; (d) *Monarch*; (e) *Leaves*; (f) *Parrots*.

noise with different noise levels. For the compared methods, we choose one discriminative method named MLP [52] and three model based optimization methods, including IDDBM3D [19], NCSR [22] and EPLL. Among the testing images, apart from three classical gray images as shown in Figure 2, three color images are also included such that we can test the performance of learned color denoiser prior. In the meanwhile, we note that the above methods are designed for gray image deblurring. Specially, NCSR tackles the color input by first transforming it into YCbCr space and then conducting the main algorithm in the luminance component. In the following experiments, we simply plug the color denoisers into the HQS framework, whereas we separately handle each color channel for IDDBM3D and MLP. Note that MLP trained a specific model for the Gaussian blur kernel with noise level 2.

Once the denoisers are provided, the subsequent crucial issue would be parameter setting. From Eqns. (6), we can note that there involve two parameters, λ and μ , to tune. Generally, for a certain degradation, λ is correlated with σ^2 and keeps fixed during iterations, while μ controls noise level of denoiser. Since the HQS framework is denoiser-based, we instead set the noise level of denoiser in each iteration to implicitly determine μ . Note that the noise level of denoiser $\sqrt{\lambda/\mu}$ should be set from large to small. In our experimental settings, it is decayed exponentially from 49 to a value in $[1, 15]$ depending on the noise level. The number of iterations is set to 30 as we find it is large enough to obtain a satisfying performance.

The PSNR results of different methods are shown in Table 4. As one can see, the proposed CNN denoiser prior based optimization method achieves very promising PSNR results. Figure 3 illustrates deblurred *Leaves* image by different methods. We can see that IDDBM3D, NCSR and MLP tend to smooth the edges and generate color artifacts. In contrast, the proposed method can recover image sharpness and naturalness.

Table 4. Deblurring results of different methods.

Methods	σ	<i>C.man</i>	<i>House</i>	<i>Lena</i>	<i>Monar.</i>	<i>Leaves</i>	<i>Parrots</i>
Gaussian blur with standard deviation 1.6							
IDDBM3D	2	27.08	32.41	30.28	27.02	26.95	30.15
NCSR		27.99	33.38	30.99	28.32	27.50	30.42
MLP		27.84	33.43	31.10	28.87	28.91	31.24
Proposed		28.12	33.80	31.17	30.00	29.78	32.07
Kernel 1 (19×19) [38]							
EPLL	2.55	29.43	31.48	31.68	28.75	27.34	30.89
Proposed		32.07	35.17	33.88	33.62	33.92	35.49
EPLL	7.65	25.33	28.19	27.37	22.67	21.67	26.08
Proposed		28.11	32.03	29.51	29.20	29.07	31.63
Kernel 2 (17×17) [38]							
EPLL	2.55	29.67	32.26	31.00	27.53	26.75	30.44
Proposed		31.69	35.04	33.53	33.13	33.51	35.17
EPLL	7.65	24.85	28.08	27.03	21.60	21.09	25.77
Proposed		27.70	31.94	29.27	28.73	28.63	31.35

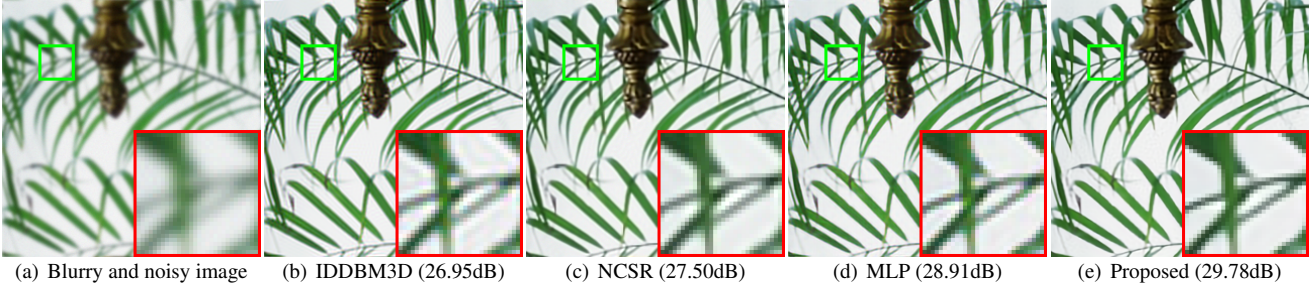


Figure 3. Image deblurring performance comparison for *Leaves* image (the blur kernel is Gaussian kernel with standard deviation 1.6, the noise level σ is 2).

4.3. Single Image Super-Resolution

In general, the low-resolution (LR) image can be modeled by a blurring and subsequent down-sampling operation on a high-resolution one. The existing super-resolution models, however, mainly focus on modeling image prior and are trained for specific degradation process. This makes the learned model deteriorates seriously when the blur kernel adopted in training deviates from the real one [23, 64]. Instead, our model can handle any blur kernels without re-training. Thus, in order to thoroughly evaluate the flexibility of the CNN denoiser prior based optimization method as well as the effectiveness of the CNN denoisers, following [45], this paper considers three typical image degradation settings for SISR, *i.e.*, bicubic downsampling (default setting of Matlab function *imresize*) with two scale factors 2 and 3 [15, 21] and blurring by Gaussian kernel of size 7×7 with standard deviation 1.6 followed by downsampling with scale factor 3 [22, 45].

Inspired by the method proposed in [24] which iteratively updates a back-projection [33] step and a denoising step for SISR, we use the following back-projection iteration to solve Eqn. (6a),

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha(\mathbf{y} - \mathbf{x}_k \downarrow_{sf}) \uparrow_{bicubic}^{sf} \quad (11)$$

where \downarrow_{sf} denotes the degradation operator with downscaling factor sf , $\uparrow_{bicubic}^{sf}$ represents bicubic interpolation operator with upscaling factor sf , and α is the step size. It is worthy noting that the iterative regularization step of methods such as NCSR and WNNM actually corresponds to solving Eqn. (6a). From this viewpoint, those methods are optimized under HQS framework. Here, note that only the bicubic downsampling is considered in [24], whereas Eqn. (11) is extended to deal with different blur kernels. To obtain a fast convergence, we repeat Eqn. (11) five times before applying the denoising step. The number of main iterations is set to 30, the step size α is fixed to 1.75 and the noise levels of denoiser are decayed exponentially from $12 \times sf$ to sf .

The proposed deep CNN denoiser prior based SISR method is compared with five state-of-the-art methods, in-

cluding two CNN-based discriminative learning methods (*i.e.*, SRCNN [21] and VDSR [35]), one statistical prediction model based discriminative learning method [45] which we refer to as SPMSR, one model based optimization method (*i.e.*, NCSR [22]) and one denoiser prior based method (*i.e.*, SRBM3D [24]). Except for SRBM3D, all the existing methods conducted their main algorithms on Y channel (*i.e.*, luminance) of transformed YCbCr space. In order to evaluate the proposed color denoiser prior, we also conduct experiments on the original RGB channels and thus the PSNR results of super-resolved RGB images of different methods are also given. Since the source code of SRBM3D is not available, we also compare two methods which replace the proposed CNN denoiser with BM3D/CBM3D denoiser. Those two methods are denoted by SRBM3D_C and SRBM3D_G, respectively.

Table 5 shows the average PSNR(dB) results of different methods for SISR on Set5 and Set14 [59]. Note that SRCNN and VDSR are trained with bicubic blur kernel, thus it is unfair to use their models to super-resolve the low-resolution image with Gaussian kernel. As a matter of fact, we give their performances to demonstrate the limitations of such discriminative learning methods. From Table 5, we can have several observations. First, although SRCNN and VDSR achieve promising results to tackle the case with bicubic kernel, their performance deteriorates seriously when the low-resolution image are not generated by bicubic kernel (see Figure 4). On the other hand, with the accurate blur kernel, even NCSR and SPMSR outperform SRCNN and VDSR for Gaussian blur kernel. In contrast, the proposed methods (denoted by Proposed_G and Proposed_C) can handle all the cases well. Second, the proposed methods have a better PSNR result than SRBM3D_C and SRBM3D_G which indicates good denoiser prior facilitates to solve super-resolution problem. Third, both of the gray and color CNN denoiser prior based optimization methods can produce promising results. As an example for the testing speed comparison, our method can super-resolve the *Butterfly* image in 0.5 second on GPU and 12 seconds on CPU, whereas NCSR spends 198 seconds on CPU.

Table 5. Average PSNR(dB) results of different methods for single image super-resolution on Set5 and Set14.

Dataset	Scale	Kernel	Channel	SRCNN	VDSR	NCSR	SPMSR	SRBM3D	SRBM3D _G	SRBM3D _C	Proposed _G	Proposed _C
Set5	2	Bicubic	Y	36.65	37.56	-	36.11	37.10	36.34	36.25	37.43	37.22
			RGB	34.45	35.16	-	33.94	-	34.11	34.22	34.22	35.05
	3	Bicubic	Y	32.75	33.67	-	32.31	33.30	32.62	32.54	33.39	33.18
			RGB	30.72	31.50	-	30.32	-	30.57	30.69	31.26	31.25
	3	Gaussian	Y	30.42	30.54	33.02	32.27	-	32.66	32.59	33.38	33.17
			RGB	28.50	28.62	30.00	30.02	-	30.31	30.74	30.92	31.21
Set14	2	Bicubic	Y	32.43	33.02	-	31.96	32.80	32.09	32.25	32.88	32.79
			RGB	30.43	30.90	-	30.05	-	30.15	30.32	30.79	30.78
	3	Bicubic	Y	29.27	29.77	-	28.93	29.60	29.11	29.27	29.61	29.50
			RGB	27.44	27.85	-	27.17	-	27.32	27.47	27.72	27.67
	3	Gaussian	Y	27.71	27.80	29.26	28.89	-	29.18	29.39	29.63	29.55
			RGB	26.02	26.11	26.98	27.01	-	27.24	27.60	27.59	27.70

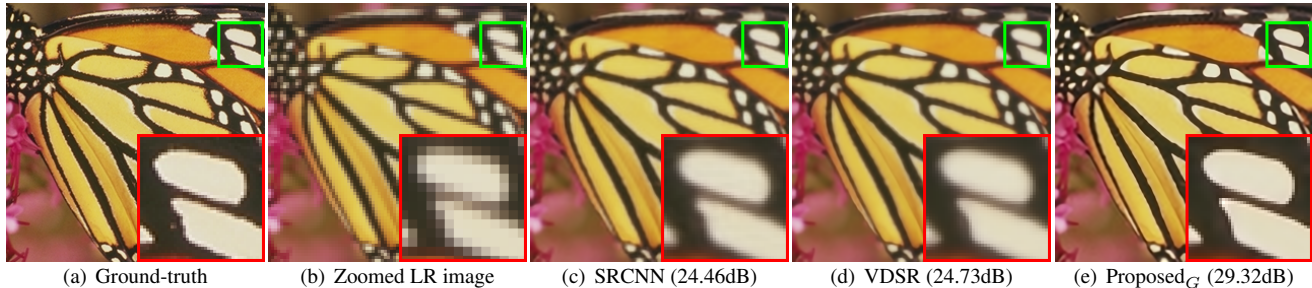


Figure 4. Single image super-resolution performance comparison for *Butterfly* image from Set5 (the blur kernel is 7×7 Gaussian kernel with standard deviation 1.6, the scale factor is 3). Note that the comparison with SRCNN and VDSR is unfair. The proposed deep CNN denoiser prior based optimization method can super-resolve the LR image by tuning the blur kernel and scale factor without training, whereas SRCNN and VDSR need additional training to deal with such cases. As a result, this figure is mainly used to show the flexibility advantage of the proposed deep CNN denoiser prior based optimization method over discriminative learning methods.

5. Conclusion

In this paper, we have designed and trained a set of fast and effective CNN denoisers for image denoising. Specially, with the aid of variable splitting technique, we have plugged the learned denoiser prior into a model-based optimization method of HQS to solve the image deblurring and super-resolution problems. Extensive experimental results have demonstrated that the integration of model-based optimization method and discriminative CNN denoiser results in a flexible, fast and effective framework for various image restoration tasks. On the one hand, different from conventional model-based optimization methods which are usually time-consuming with sophisticated image priors for the purpose of achieving good results, the proposed deep CNN denoiser prior based optimization method can be implemented effectively due to the plug-in of fast CNN denoisers. On the other hand, different from discriminative learning methods which are specialized for certain image restoration tasks, the proposed deep CNN denoiser prior based optimization method is flexible in handling various tasks while can produce very favorable results. In summary, this work highlights the potential benefits of integrating flexible model-based optimization methods and fast discriminative learning methods. In addition, this work has shown that learning expressive CNN denoiser prior is a good alternative to model image prior.

While we have demonstrated various merits of plugging powerful CNN denoiser into model-based optimization methods, there also remain room for further study. Some research directions are listed as follows. First, it will be interesting to investigate how to reduce the number of the discriminative CNN denoisers and the number of whole iterations. Second, extending the proposed CNN denoiser based HQS framework to other inverse problems such as inpainting and blind deblurring would be also interesting. Third, utilizing multiple priors which are complementary to improve performance is certainly one promising direction. Finally, and perhaps most interestingly, since the HQS framework can be treated as a MAP inference, this work also provides some insights into designing CNN architecture for task-specific discriminative learning. Meanwhile, one should be aware that CNN has its own design flexibility and the best CNN architecture is not necessarily inspired by MAP inference.

6. Acknowledgements

This work is supported by HK RGC General Research Fund (PolyU 5313/13E) and National Natural Science Foundation of China (grant no. 61672446, 61671182). We gratefully acknowledge the support from NVIDIA Corporation for providing us the Titan X GPU used in this research.

References

- [1] H. C. Andrews and B. R. Hunt. Digital image restoration. *Prentice-Hall Signal Processing Series, Englewood Cliffs: Prentice-Hall, 1977*, 1, 1977. **1**
- [2] A. Barbu. Training an active random field for real-time image denoising. *IEEE Transactions on Image Processing*, 18(11):2451–2462, 2009. **1**
- [3] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009. **3**
- [4] J. M. Bioucas-Dias and M. A. Figueiredo. A new TwIST: Two-step iterative shrinkage/thresholding algorithms for image restoration. *IEEE Transactions on Image Processing*, 16(12):2992–3004, 2007. **3**
- [5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning*, 3(1):1–122, 2011. **2**
- [6] A. Brifman, Y. Romano, and M. Elad. Turning a denoiser into a super-resolver using plug and play priors. In *IEEE International Conference on Image Processing*, pages 1404–1408, 2016. **2**
- [7] A. Buades, B. Coll, and J.-M. Morel. A non-local algorithm for image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 60–65, 2005. **3**
- [8] H. C. Burger, C. J. Schuler, and S. Harmeling. Image denoising: Can plain neural networks compete with BM3D? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2392–2399, 2012. **2, 5**
- [9] P. Campisi and K. Egiazarian. *Blind image deconvolution: theory and applications*. CRC press, 2016. **1**
- [10] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004. **3**
- [11] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. **2**
- [12] S. H. Chan, X. Wang, and O. A. Elgandy. Plug-and-Play ADMM for image restoration: Fixed-point convergence and applications. *IEEE Transactions on Computational Imaging*, 3(1):84–98, 2017. **2**
- [13] Y. Chen and T. Pock. Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 2016. **1, 5**
- [14] P. L. Combettes and V. R. Wajs. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005. **3**
- [15] Z. Cui, H. Chang, S. Shan, B. Zhong, and X. Chen. Deep network cascade for image super-resolution. In *European Conference on Computer Vision*, pages 49–64, 2014. **7**
- [16] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Color image denoising via sparse 3D collaborative filtering with grouping constraint in luminance-chrominance space. In *IEEE International Conference on Image Processing*, volume 1, pages I–313, 2007. **3**
- [17] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian. Image denoising by sparse 3-D transform-domain collaborative filtering. *IEEE Transactions on Image Processing*, 16(8):2080–2095, 2007. **2, 3, 5**
- [18] A. Danielyan, V. Katkovnik, and K. Egiazarian. Image deblurring by augmented lagrangian with BM3D frame prior. In *Workshop on Information Theoretic Methods in Science and Engineering*, pages 16–18, 2010. **2**
- [19] A. Danielyan, V. Katkovnik, and K. Egiazarian. BM3D frames and variational image deblurring. *IEEE Transactions on Image Processing*, 21(4):1715–1728, 2012. **2, 6**
- [20] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. **5**
- [21] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on Pattern Analysis and Machine Intelligence*, 38(2):295–307, 2016. **2, 7**
- [22] W. Dong, L. Zhang, G. Shi, and X. Li. Nonlocally centralized sparse representation for image restoration. *IEEE Transactions on Image Processing*, 22(4):1620–1630, 2013. **2, 6, 7**
- [23] N. Efrat, D. Glasner, A. Apartsin, B. Nadler, and A. Levin. Accurate blur models vs. image priors in single image super-resolution. In *IEEE International Conference on Computer Vision*, pages 2832–2839, 2013. **7**
- [24] K. Egiazarian and V. Katkovnik. Single image super-resolution via BM3D sparse coding. In *European Signal Processing Conference*, pages 2849–2853, 2015. **2, 7**
- [25] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Transactions on Image processing*, 15(12):3736–3745, 2006. **3**
- [26] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape adaptive DCT denoising with structure preservation in luminance-chrominance space. In *International Workshop on Video Processing and Quality Metrics for Consumer Electronics*, 2006. **3**
- [27] Q. Gao and S. Roth. How well do filter-based MRFs model natural images? In *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*, pages 62–72, 2012. **2**
- [28] D. Geman and C. Yang. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7):932–946, 1995. **2**
- [29] S. Gu, L. Zhang, W. Zuo, and X. Feng. Weighted nuclear norm minimization with application to image denoising. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014. **2, 5**
- [30] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. **4**
- [31] F. Heide, M. Steinberger, Y.-T. Tsai, M. Rouf, D. Pajak, D. Reddy, O. Gallo, J. Liu, W. Heidrich, K. Egiazarian, et al. Flexisp: A flexible camera image processing framework. *ACM Transactions on Graphics*, 33(6):231, 2014. **2**
- [32] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning*, pages 448–

- 456, 2015. 2, 4
- [33] M. Irani and S. Peleg. Motion analysis for image enhancement: Resolution, occlusion, and transparency. *Journal of Visual Communication and Image Representation*, 4(4):324–335, 1993. 7
- [34] V. Jain and S. Seung. Natural image denoising with convolutional networks. In *Advances in Neural Information Processing Systems*, pages 769–776, 2009. 2
- [35] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1646–1654, 2016. 4, 7
- [36] D. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference for Learning Representations*, 2015. 2, 5
- [37] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 2
- [38] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1964–1971, 2009. 6
- [39] Z. Lin, M. Chen, and Y. Ma. The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices. *arXiv preprint arXiv:1009.5055*, 2010. 5
- [40] K. Ma, Z. Duanmu, Q. Wu, Z. Wang, H. Yong, H. Li, and L. Zhang. Waterloo exploration database: New challenges for image quality assessment models. *IEEE Transactions on Image Processing*, 26(2):1004–1016, 2017. 5
- [41] J. Mairal, M. Elad, and G. Sapiro. Sparse representation for color image restoration. *IEEE Transactions on Image Processing*, 17(1):53–69, 2008. 3
- [42] T. Miyata. Inter-channel relation based vectorial total variation for color image recovery. In *IEEE International Conference on Image Processing*, pages 2251–2255, 2015. 3
- [43] S. Osher, M. Burger, D. Goldfarb, J. Xu, and W. Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005. 3
- [44] N. Parikh, S. P. Boyd, et al. Proximal algorithms. *Foundations and Trends in optimization*, 1(3):127–239, 2014. 2
- [45] T. Peleg and M. Elad. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE Transactions on Image Processing*, 23(6):2569–2582, 2014. 7
- [46] A. Rajwade, A. Rangarajan, and A. Banerjee. Image denoising using the higher order singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(4):849–862, 2013. 3
- [47] W. H. Richardson. Bayesian-based iterative method of image restoration. *JOSA*, 62(1):55–59, 1972. 1
- [48] Y. Romano, M. Elad, and P. Milanfar. The little engine that could regularization by denoising (RED). *arXiv preprint arXiv:1611.02862*, 2016. 2
- [49] A. Rond, R. Giryes, and M. Elad. Poisson inverse problems by the plug-and-play scheme. *Journal of Visual Communication and Image Representation*, 41:96–108, 2016. 2
- [50] S. Roth and M. J. Black. Fields of experts. *International Journal of Computer Vision*, 82(2):205–229, 2009. 1, 4, 5
- [51] U. Schmidt and S. Roth. Shrinkage fields for effective image restoration. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2774–2781, 2014. 1
- [52] C. J. Schuler, H. Christopher Burger, S. Harmeling, and B. Scholkopf. A machine learning approach for non-blind image deconvolution. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1067–1074, 2013. 6
- [53] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference for Learning Representations*, 2015. 4
- [54] S. Sreehari, S. Venkatakrishnan, B. Wohlberg, L. F. Drummy, J. P. Simmons, and C. A. Bouman. Plug-and-play priors for bright field electron tomography and sparse interpolation. *arXiv preprint arXiv:1512.07331*, 2015. 2
- [55] J. Sun and M. F. Tappen. Separable markov random field model and its applications in low level vision. *IEEE Transactions on Image Processing*, 22(1):402–407, 2013. 1
- [56] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition*, June 2015. 4
- [57] M. F. Tappen. Utilizing variational optimization to learn markov random fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. 1
- [58] A. M. Teodoro, J. M. Bioucas-Dias, and M. A. Figueiredo. Image restoration and reconstruction using variable splitting and class-adapted image priors. In *IEEE International Conference on Image Processing*, pages 3518–3522, 2016. 2
- [59] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian Conference on Computer Vision*, pages 111–126, 2014. 7
- [60] A. Vedaldi and K. Lenc. MatConvNet: Convolutional neural networks for matlab. In *ACM Conference on Multimedia Conference*, pages 689–692, 2015. 5
- [61] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg. Plug-and-play priors for model based reconstruction. In *IEEE Global Conference on Signal and Information Processing*, pages 945–948, 2013. 2
- [62] L. Xu, J. S. Ren, C. Liu, and J. Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, pages 1790–1798, 2014. 2
- [63] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 2, 4
- [64] K. Zhang, X. Zhou, H. Zhang, and W. Zuo. Revisiting single image super-resolution under internet environment: blur kernels and reconstruction algorithms. In *Pacific Rim Conference on Multimedia*, pages 677–687, 2015. 7
- [65] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising. *IEEE Transactions on Image Processing*, 2017. 4
- [66] D. Zoran and Y. Weiss. From learning models of natural image patches to whole image restoration. In *IEEE International Conference on Computer Vision*, pages 479–486, 2011. 1, 2, 3, 5