Stacking Networks Dynamically for Image Restoration Based on the Plug-and-Play Framework

Haixin Wang[1,2], Tianhao Zhang[1], Muzhi Yu[1], Jinan Sun[1(✉)], Wei Ye[1], Chen Wang[1], and Shikun Zhang[1]

[1] Peking University, Beijing, China
{wang.hx,tianhao_z,muzhi.yu,sjn,wye,wangchen,zhangsk}@pku.edu.cn
[2] University of Science and Technology Beijing, Beijing, China

Abstract. Recently, stacked networks show powerful performance in Image Restoration, such as challenging motion deblurring problems. However, the number of stacking levels is a hyper-parameter fine-tuned manually, making the stacking levels static during training without theoretical explanations for optimal settings. To address this challenge, we leverage the iterative process of the traditional plug-and-play method to provide a dynamic stacked network for Image Restoration. Specifically, a new degradation model with a novel update scheme is designed to integrate the deep neural network as the prior within the plug-and-play model. Compared with static stacked networks, our models are stacked dynamically during training via iterations, guided by a solid mathematical explanation. Theoretical proof on the convergence of the dynamic stacking process is provided. Experiments on the noise dataset BSD68, Set12, and motion blur dataset GoPro demonstrate that our framework outperforms the state-of-the-art in terms of PSNR and SSIM score without extra training process.

Keywords: Low-level Vision; Image Restoration; Plug-and-Play

1 Introduction

Image Restoration (IR) is a classic yet hot task in low-level vision for its high application value. It aims to recover the clean image x from its corrupted observation y. Classic degradation model is $y = Ax + n$, where A is a degradation matrix referred to as the identity matrix in image denoising or the blurring matrix in image deblurring. n is often regarded as additive white Gaussian noise.

Solutions to this ill-posed inverse model include two main categories: model-based and learning-based. Model-based methods estimate A and n in the degradation model by a series of constraints and regularizations, and then iteratively solve for the latent clean image supported with strong mathematical theory. But they rely heavily on fixed and handcrafted priors that certainly are insufficient in characterizing clean images. Learning-based methods gradually show superiority to learn the regression between the corrupted input image and the latent

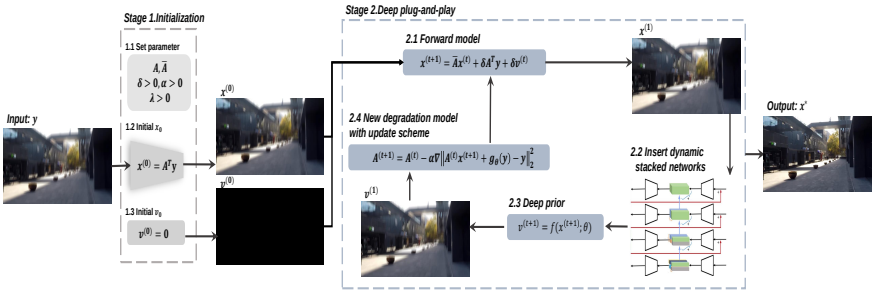

Fig. 1: The architecture of our proposed deep plug-and-play framework with dynamic stacked networks.

clean image directly. Meanwhile, new tricks, like the network stacking diagram, are borrowed to further improve performance. However, networks are with too many static hyper-parameters, and the learning performance depends seriously on carefully tuning of them. These facts make the training very tricky, let alone hard process and GPU limit, but also theoretical analysis difficult.

To address this challenge, we propose to stack networks dynamically for IR based on the plug-and-play framework. First, by plugging the pre-trained deep prior into our framework, we can iteratively reuse the prior knowledge like stacking deep networks without increasing parameter size. More importantly, a new degradation model $y = A^{(t)}x+n$ with update scheme is designed, and solved for the theoretical optima guided by strong explanations on the convergence. Thus, our dynamic stacking diagram not only leverages the iterations to the maximum extent, but also plays the role of shifting focus for better performance level-by-level in training stacked networks.

In this paper, we solve for the latent clean image in image denoising and extend the idea to complex motion deblurring tasks. The framework consisting a new degradation model and pre-trained deep prior is formalized as

$$x^* = x^{(t+1)} = \arg\min_{x^{(t)}} ||y - A^{(t)}x^{(t)}||^2 + \lambda f(x^{(t)}; \theta) \tag{1}$$

where y is the noisy or blurry image need to be restored, x is the latent clean solution, and θ is the parameter of plugged deep denoiser or deblurrer. Unlike the basic degradation model, A is no longer a specific degradation matrix fixed in a specific task. We note that previous works' results are superior level-by-level which means more stacked sub-models after training will focus on more blurry details spatially. But simple iterations cannot shift the focus of pre-trained networks on more blurry cues. Based on these observations, we propose to update A in each iteration to shift focus on more corrupted areas. Thus, in Eq. (1), $||y - A^{(t)}x||^2$ represents the fidelity term, and $\lambda f(x; \theta)$ is the regularization term known as the prior.

Fig. 1 shows the architecture of our novel dynamic stacked networks based on plug-and-play framework. The key point is that our dynamic diagram can converge to the optima with a firm theoretical foundation instead of tuning stacking levels of static stacked networks. Our deep plug-and-play framework, contributing to the leverage of the new degradation model with an update scheme, applies to image denoising and complex motion deblurring problems successfully first of the time to our best knowledge. We conduct extensive experiments to demonstrate the superior results of our framework, compared with the static networks who set the state-of-the-art on famous noise dataset BSD68, Set12, and motion blur dataset GoPro. Both objective evaluating metrics PSNR, SSIM and visualization in the following section help to prove that we effectively improve the performance in both image denoising and motion deblurring tasks.

Our contributions are summarized as follows:

1) We propose a dynamic stacked networks for IR based on the plug-and-play framework to solve the new degradation model. Compared with static stacking diagrams, our framework leverages the iteration process to dynamically reuse the prior knowledge.
2) Theoretical analysis is provided to show that our framework with a new degradation model inside is able to solve for the optima with fast convergence by means of iterating the output dynamically.
3) To our best knowledge, our framework exploits both the merits of model-based and learning-based methods for image denoising and non-uniform blind motion deblurring for the first time.
4) Experiments on datasets BSD68, Set12, and GoPro have proven that our framework outperforms existing methods on PSNR, SSIM, and visualization.

The remainder of the paper is organized as follows. In Section 2, we give an overview of the related work. In Section 3, we provide a detailed description of the proposed method. Finally, in Section 4, we perform an evaluation of our framework on image denoising and motion deblurring tasks, and compare it to the state-of-the-art. Meanwhile, mathematical explanations are provided. Section 5 concludes this paper.

2 Related Work

2.1 Plug-and-Play Methods

The plug-and-play method was first introduced to solve IR tasks in [8,27,35]. Its core idea is to decouple the fidelity term and regularization term in the energy function by splitting techniques, as well as to replace the prior associated sub-problem by any off-the-shelf Gaussian denoiser. For its flexibility and good performance, a set of work have been done mainly in three aspects: 1) various priors, including conventional priors such as the well-known BM3D [7], Gaussian mixture model [35] and the state-of-the-art CNN denoiser such as [34] as well as their combination [11]; 2) various variable splitting algorithms, such as

half-quadratic splitting (HQS) algorithm [1], alternating direction method of multipliers (ADMM) algorithm [3] and primal-dual algorithm [19]; 3) theoretical analysis on the convergence from the aspect of fixed point [5,17]. In [31], priors have been proved not limited to Gaussian denoiser. In this paper, priors can transfer to be image deblurrer.

2.2 Image Denoising

Image denoising is a classic low-level vision task. DNN has been exploited since 2009, which has developed the solution in two aspects: 1) learn the clean target. MLP [4] has been adopted to learn the mapping from noise patch to clean pixel. In [6], a trainable nonlinear reaction diffusion (TNRD) model has been proposed and it can be expressed as a feed-forward deep network by unfolding a fixed number of gradient descent inference steps. Santhanam et al. [23] introduce a recursively branched deconvolutional network (RBDN), where pooling/unpooling is adopted to obtain and aggregate multi-context; 2) learn the noise. Residual learning with batch normalization (DnCNN) was first proposed by Zhang et al. [33] which outperforms other methods. A set of frameworks take advantage of DnCNN as a denoising network for various applications [28]. In [14], only noisy inputs are exploited to train in the network with L2 loss function which outputs the mean of all results. By observing the noisy inputs twice during training on a big enough dataset, the network can estimate the noise distribution in an unsupervised manner. Inspired by the success of DNN-based method, many work [16,20,34] attempted to integrate the conventional method like BM3D, wavelet transformation, including plug-and-play with DNN.

2.3 Non-uniform Blind Deblurring

The goal of non-uniform blind image deblurring is to remove the undesired blur caused by camera motion and scene dynamics [24,32]. Conventional methods used to employ a variety of constraints or regularizations to approximate the motion blur filters, involving an expensive non-convex non-linear optimization. Moreover, the commonly used assumption of spatially-uniform blur kernel is overly restrictive, resulting in a poor deblurring of complex blur patterns.

CNN-based methods have shown a powerful ability to deal with the complex motion blur in a time-efficient manner. They are developed in two main respects: 1) Learning the blur kernel. [29] proposed a deconvolutional CNN which removes blur in a non-blind setting by recovering a sharp image given the estimated blur kernel. Their network uses separable kernels which can be decomposed into a small set of filters. [25] estimated and removed a non-uniform motion blur from an image by learning the regression between image patches and their corresponding kernels. 2) Learning the sharp Image. In [32], Recurrent Neural Network (RNN) is applied as a deconvolutional decoder on feature maps extracted by the first CNN module. Another CNN module learns weights for each layer of RNN. The last CNN module reconstructs images from deblurred feature maps.

3 Proposed Methods

3.1 Overview

As shown in Fig. 1, the pre-trained stacked network is plugged into the framework dynamically in Stage 2.2 $\sim$ 2.3. Different from existing deep plug-and-play methods [34], we solve our new degradation model by transforming the forward model into a single step of gradient descent in Stage 2.1, which not only provides fast convergence but also keeps remarkable performance with theoretical support. Ahead of the implementation, the degradation matrix $A^{(t)}$ and other parameters are initialized in Stage 1. Furthermore, an optimization target is set in Stage 2.4, which will be solved to update the variable $A^{(t)}$ in $y = A^{(t)}x + n$ in each iteration adaptively to adjust the focus of deblurring networks.

3.2 Deep Plug-and-Play

In IR, A in the basic model is reasonably equal to the blur kernel. However, the blur kernel is hard to know in complex application situations. Therefore, our new degradation model is designed, in which we no longer need to know or to estimate the blur kernel but initialize it as an identity matrix before the adaptive updating scheme. Based on the new degradation model, we can utilize the deep prior as denoiser for denoising tasks. Furthermore, we transfer the deep plug-and-play framework to complex deblurring problems by the other deep prior. Basically, to plug the deep prior into the optimization procedure of Eqn. (1), the variable splitting technique is usually adopted to decouple the fidelity term and regularization term. In Half Quadratic Splitting (HQS) method, by introducing an auxiliary variable v, Eqn. (1) can be reformulated as a constrained optimization problem which is given by

$$(x^*, v^*) \leftarrow \arg\min_{x,v} \frac{1}{2}||y - A^{(t)}x||^2 + \lambda f(v;\theta) \quad s.t. \;\; x = v \tag{2}$$

Then, standard optimization algorithms are able to be used to solve the problem. The equally constrained optimization problem can be converted into a non-constrained optimization problem,

$$\mathcal{L}(x, v) = \frac{1}{2}||y - A^{(t)}x||^2 + \lambda f(v;\theta) + \frac{\mu}{2}||v - x||^2 \tag{3}$$

where μ is a penalty parameter which varies iteratively in a non-descending order. Eqn. (3) can be solved via the following iterative scheme,

$$\begin{cases} x^{(t+1)} = \arg\min_{x} ||y - A^{(t)}x||^2 + \mu||x - v^{(t)}||^2 & (4) \\ v^{(t+1)} = \arg\min_{v} \frac{\mu}{2}||v - x^{(t+1)}||^2 + \lambda f(v;\theta) & (5) \end{cases}$$

The first step only depends on the choice of a forward model, while the second step only depends on the choice of prior and can be interpreted as a denoising operation [27]. However, it is not limited in denoiser and can be extended to be deblurrer in our paper. Typically, Eqn. (4) is a quadratic optimization problem that can be solved in closed-form, as $x^{(t+1)} = W^{-1}x_t$, where W is a matrix related to the degradation matrix A. It is time-consuming to compute like this while Fast Fourier Transformation (FFT) is often applied as a feasible implementation [34]. However, FFT methods still cannot solve for answers efficiently. In our framework, we propose to take advantage of iterative classic conjugate gradient (CG) algorithm, which is a common optimization algorithm. More briefly, we only compute with a single step of gradient descent for an inexact solution,

$$\begin{aligned} x^{(t+1)} &= x^{(t)} - \delta[{A^{(t)}}^T(A^{(t)}x^{(t)} - y) + \mu(x^{(t)} - v^{(t)})] \\ &= [(1 - \delta\mu)I - \delta{A^{(t)}}^T A^{(t)}]x^{(t)} + \delta{A^{(t)}}^T y + \delta v^{(t)} \end{aligned} \tag{6}$$

where δ is the step size. It is proven that this single descent step is sufficient for convergence which follows the idea of [9] shown in section 4. Eqn. (5) is considered to be a task-dependent denoiser or deblurrer in our framework. In this paper, inspired by the success of deep learning-based methods for IR tasks, we plug in the pre-trained deep neural network model to replace the proximity operator of conventional priors.

$$v^{(t+1)} = f(x^{(t+1)}; \theta) \tag{7}$$

Eqn. (7) is just the solution of Eqn. (5). After several alternating iterations, it is expected that the final reconstructed image attains the high-quality restoration.

3.3 Deep Prior

Denoiser In order to exploit the merits of learning-based methods, we need to specify the denoiser network according to Eqn. (7). Inspired by [10], we only need to modify most of the existing learning-based denoisers. The pretrained prior we adopt is the variational denoising network (VDN), which efficiently approximates the true posterior with the latent variables. The framework includes two subnets standing for noise estimation and noise removal separately.

The weights of VDN are initialized according to [12]. In each epoch, we randomly crop $N = 64 \times 5000$ patches with size 128×128 from the images for training. The Adam algorithm [13] is adopted to optimize the network parameters through minimizing the proposed negative lower bound objective. The initial learning rate is set as 2e-4 and linearly decayed in half every 10 epochs to 1e-6.

Deblurrer We base our deep deblurrer on DMPHN [31] including its stacked versions which is the state-of-the-art. It processes images of different scales by dividing into different numbers of image patches from the coarsest to the finniest level. This end-to-end network can be stacked as a part with more stacks. Since

the network is determined by the stacking level, this static stacking diagram is hard to find the optimal form in two-fold: 1) It has to stack many filters since their weights are fixed and spatially invariant; 2) A geometrically uniform receptive field without adaption is sub-optimal for the real-world scene. Therefore, limit to the GPU memory and long training time, there exists no performance of deeper stacked networks. After we plugging pre-trained shallow stacked models into our framework, they can explore the solution effectively dynamically.

Inside the DMPHN, there is an encoder and decoder of each layer. The encoder consists of 15 convolutional layers, 6 residual links, and 6 ReLU units. The layers of decoder and encoder are identical except that two convolutional layers are replaced by deconvolutional layers to generate images. The input of each layer is the blurry image divided into specific image patches. The output of both encoder and decoder from a lower level (corresponds to the finer grid) will be added to the upper level (one level above) so that the top level contains all information inferred in the finer levels.

3.4 Adaptive Update Scheme

Previous work has shown that the performance of stacked networks is superior level-by-level, which means more stacked sub-models will focus on more blurry details spatially. Since pre-trained networks focus on several severe blurriness are constructed from coarse-to-fine, the restoration image tends to induce undesired local blurriness after a few simple iterations especially when the motion blur is distributed all over the image. Since simple iterations cannot shift the focus of networks on more blurry cues, we design the adaptive update scheme in our degradation model to shift the focus adaptively. To be specific, A will be used as the adaptive optimized variable to complete our design. We find that networks trained under MSE loss all try to do one thing essentially: learning the mapping from corrupted images y to latent clean images x with a network $g_\theta(\cdot)$.

$$\begin{aligned} & \arg\min_x ||g_\theta(y) - x||^2 \\ = \ & \arg\min_x ||g_\theta(Ax + n) - x||^2 \end{aligned} \tag{8}$$

A precondition that cannot be ignored is that deep plug-and-play framework should follow the basic degradation model $y = Ax + n$ with unknown variables in IR tasks. That means the network aims to complete the fitting task $g_\theta(y) = A^{-1}[(Ax+n)-n]$ by training on big data. However, the network cannot fit to the uncertain targets and may be trained as $g_\theta(y) = H^{-1}[(Ax+n)-\varepsilon] = H^{-1}Ax-\eta$ with deblurring ability to a certain extent. When we suppose $A = I$ in a learning-based method where H^{-1} may not be A^{-1}, the final solution must be over-blurred by an operation that is uncertain. That means that simple iterations with pre-trained deblurrers cannot improve the deblurring performance for over-blur.

Therefore, we propose a new degradation model $y = A^{(t)}x + n$ with update scheme to battle the operation H^{-1} fixed in the pre-trained deep prior which may cause undesired blurriness. Our purpose is to let $H^{-1} = A^{-1}$ and $\eta = 0$.

This idea is equal to make the basic degradation model $y = Ax + n$ reasonable in each iteration. So we set the optimization target after implementing Eqn. (7).

$$\begin{aligned} & \arg\min_{A^{(t)}} ||A^{(t)}x^{(t)} + n - y||^2 \\ = \ & \arg\min_{A^{(t)}} ||A^{(t)}x^{(t)} + g_\theta(y) - y||^2 \end{aligned} \tag{9}$$

where noise n in the optimization target is estimated by the pre-trained deep denoiser. This optimization target will update $A^{(t)}$ in each iteration between Eqn. (5) and Eqn. (4) after the end of an epoch. This step of adaption can shift the focus of the deep deblurrer and pass the adjusted matrix $A^{(t)}$ to the forward model.

As for the solution, we tackle this optimization target in the similar way of the forward model. Hence, the matrix $A^{(t)}$ will be updated in a single step of gradient descent

$$A^{(t+1)} = A^{(t)} - \alpha {x^{(t+1)}}^T A^{(t)} x^{(t+1)} - \alpha(g_\theta(y) - y) \tag{10}$$

where α is the step size of gradient descent, and $g_\theta(\cdot)$ is the pre-trained deep prior to estimate the noise. We train a common deep denoiser DnCNN [33] as the $g_\theta(\cdot)$ due to its success. According to the deductive optimization target Eqn. (10), we make $A^{(t)}$ adaptively fit to the degradation model and consequently focus on more corrupted areas.

4 Experiments

Our deep priors are based on their released version without retraining with a single NVIDIA Titan RTX GPU. In the alternating iterations between Eqn.(4) and Eqn.(5), we need to tune μ and set λ to make the performances satisfying. In addition, the step size of gradient descent δ in the forward model Eqn.(4) and α in the update step are also needed to be set previously. Actually, the step size varies with different datasets and different deep priors for different convergence speed. Although setting such parameters has been considered as a non-trivial task [21], the parameters of our framework are easy to be obtained with the following principles . Firstly, λ is fixed associated with noise level in denoising, we can instead multiply noise level by a scalar λ and therefore ignore the λ in Eqn. (5). And the noise level in deblurring is 0. Secondly, the step size is tuned from 1e-3 to 1e-5 for a total of 2 to 4 iterations.

4.1 Image Denoising

Datasets. The denoising prior is trained on the famous BSD500. We test our framework on two widely used datasets BSD68 and Set12. BSD68 [22] contains of 68 nature images subtracted from Berkeley segmentation dataset. The common data augmentation operations such as flip and rotations are implemented to it.

Table 1: Average PSNR(dB)/SSIM results of the competing methods for image denoising with noise levels $\sigma = 15$, 25 and 50 on datasets Set12 and BSD68. Outperforming results are noted red bold while the second best results are blue bold.

Dataset	σ	DnCNN [33]		IRCNN [34]		MWCNN [16]		NLRN [15]		DPDD [9]		VDN [30]		Proposed	
Set12	15	32.86	0.903	32.77	0.901	33.15	0.909	33.16	0.907	32.91	0.889	**33.33**	0.912	**33.44**	0.912
	25	30.44	0.862	30.38	0.860	30.79	0.871	30.80	0.869	30.54	0.811	**30.90**	**0.875**	**31.04**	**0.875**
	50	27.18	0.783	27.14	0.780	27.74	0.806	27.64	0.798	27.50	0.739	**28.00**	**0.816**	**28.27**	**0.816**
BSD68	15	31.73	0.891	31.63	0.888	31.86	0.895	31.88	0.893	**32.29**	0.888	32.22	**0.917**	**32.42**	**0.918**
	25	29.23	0.828	29.15	0.825	29.41	0.836	29.41	0.833	29.88	0.827	**30.03**	0.851	**30.17**	0.851
	50	26.23	0.719	26.19	0.717	26.53	0.737	26.47	0.730	27.02	0.754	**27.18**	0.754	**27.64**	0.754

Another famous dataset we exploit contains 12 famous gray images for Image Processing tasks. Note that all those images are widely used for the evaluation of Gaussian denoising methods and they are not included in the training dataset. To evaluate our framework, we consider three common noise levels of additive white Gaussian noise with $\sigma = 15$, 25, 50. Noise is randomly added to the image before testing.

Baselines. We compare the proposed framework with six methods, including four learning-based methods (*i.e.*,DnCNN [33], MWCNN [16], NLRN [15], VDN [30]) as well as two methods combing conventional models and deep priors (*i.e.*, DPDD [9], IRCNN [34]). DnCNN consists of 17 layers of ResNet for learning the noise of the degraded images; NLRN integrates non-local self-similarity in natural images as an effective prior into existing deep networks for end-to-end training to capture deep feature correlation between each location and its neighborhood. A similar method is applied in MWCNN for transforming the convolutional neural networks in view of multi-level wavelet. VDN proposes a new variational inference method and integrates both noise estimation and image denoising into a unique Bayesian framework for blind denoising. IRCNN and DPDD both combine the deep neural networks and conventional models, while the former is based on a model and the latter transforms the model into several layers in a network. Our proposed framework is fairly compared to them.

Denoising Results. The PSNR and SSIM results of different methods for image denoising on the dataset BSD68 and Set12 are shown in Table 1, from which we have several observations. Firstly, VDN which set the state-of-the-art currently shows great lift with about 1.2 $\sim$ 2.1 dB on PSNR and 0.06 $\sim$ 0.11 on SSIM on Set12, about 0.2 $\sim$ 1.9 dB on PSNR and 0.08 $\sim$ 0.15 on SSIM on BSD68, which may be attributed to simultaneously implementing both noise estimation and blind image denoising tasks in a unique Bayesian framework. Secondly, IRCNN and DPDD are both superior to the methods they are based on which inspire more work to focus on deep plug-and-play methods including us. Last, our proposed framework outperforms existing best method, VDN, 0.1 $\sim$ 0.27 dB on PSNR on Set12 and 0.18 $\sim$ 0.5 dB on BSD68 but no obvious lift on SSIM due to no difference on variance. Our framework is based on deep priors but surpasses them for the unique combination of both the advantages of model- and learning-based method with self-adaption.

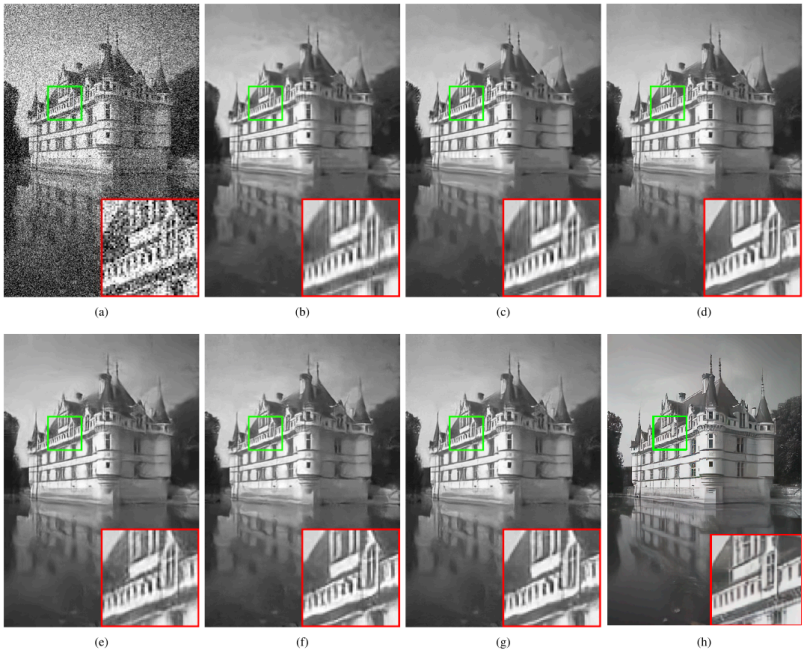

Fig. 2: Denoising results of one image from BSD68 with noise level 50. (a) Noisy, 14.76dB. (b) BM3D, 26.21dB. (c) WNNM, 26.51dB. (d) DnCNN-B, 26.92dB. (e) MLP, 26.54dB. (f) TNRD, 26.59dB. (g) DnCNN-S, 26.90dB. (h) **Proposed**, **28.47dB**

Fig.2 shows the visual comparison of different methods for Gaussian denoising with $\sigma = 50$ on dataset BSD68. Conventional model methods like BM3D, WNNM cannot restore the latent image as clean as learning-based methods such as MLP, TNRD and DnCNN. DnCNN_S produces better results than similar methods. However, all these methods cause the local blurriness with error, especially the edges of the castle in Fig.2. Our proposed method shows powerful abilities to overcome the tendency of blurriness and reduce the local error, so that the edge of the castle can be seen clearly. As we all know, PSNR evaluates the distance of all the pixels which stands for the error between two images. Our proposed framework performs the best naturally for improving the quality of reconstruction.

4.2 Image Deblurring

Datasets. GoPro dataset [18] consists of 3214 pairs of blurred and clean images extracted from 33 sequences captured at 720 × 1280 resolution. The blurred images are generated by averaging varying number (7–13) of successive latent frames to produce varied blur. For a fair comparison, we follow the protocol in [18], which uses 2103 image pairs for training and the remaining 1111 pairs for testing. To make the fair comparison with DMPHN itself, we follow the principle

Table 2: Quantitative analysis of our framework on the GoPro dataset compared with baselines. PSNR and SSIM are common evaluating metrics for image restoration tasks. Each of our proposed frameworks are based on the deep deblurreres marked in the notation. Outperforming results in the specific stacking diagram are noted red bold while the second best results are blue bold.

Models	PSNR(dB)	SSIM	Size(MB)
Sun et al. [25]	24.64	0.8429	54.1
Nah et al.[18]	29.23	0.9162	303.6
Zhang et al.[32]	29.19	0.9306	37.1
Tao et al.[26]	30.10	0.9323	33.6
DMPHN(1-2)	29.77	0.9286	14.5
DMPHN(1-2-4)	30.21	0.9345	21.7
Proposed(DMPHN(1-2-4))	**30.32**	**0.9358**	21.7
DMPHN(1-2-4-8-16)	29.87	0.9305	36.2
DMPHN(1-2-4-8)	30.25	0.9351	29.0
Proposed(DMPHN(1-2-4-8))	**30.40**	**0.9400**	29.0
Stack(2)-DMPHN	30.71	0.9403	43.4
Proposed(Stack(2)-DMPHN)	30.92	0.9478	43.4
Stack(3)-DMPHN	31.16	0.9451	65.1
Proposed(Stack(3)-DMPHN)	**31.32**	**0.9510**	65.1
Stack(4)-DMPHN	31.20	0.9453	86.8
Proposed(Stack(4)-DMPHN)	**31.44**	**0.9530**	86.8
VMPHN	30.90	0.9419	43.4
Stack(2)-VMPHN	**31.50**	**0.9483**	86.4
Proposed(Stack(2)-VMPHN)	**31.72**	**0.9567**	86.4

that randomly crop images to 256 $\times$ 256 pixel size. The batch size is set to 6 during training and the Adam solver is used to train the model for 3000 epochs. The initial learning rate is set to 0.0001 and the decay rate is 0.1. Then we normalize image to range the [0, 1] and subtract 0.5. Finally, we plug the deep prior into our framework.

Baselines. We compare our proposed framework with 5 competing motion deblurring methods. [25] proposed to deal with the complex motion blur using CNN in an early time by learning the regression between 30 $\times$ 30 image patches and their corresponding kernels. To exploit the deblurring cues at different processing levels, the "coarse-to-fine" scheme has been extended to deep CNN scenarios by a multi-scale network architecture and a scale-recurrent architecture. For multi-scale architecture, [18] exploited a multi-scale CNN to restore sharp images in an end-to-end fashion from images whose blur is caused by various factors. A multi-scale loss function is employed to mimic the coarse-to-fine pipeline in conventional deblurring approaches. For RNN architecture, proposed by [32], a network consisting of three deep CNNs and one RNN, is a prominent example. The RNN is applied as a deconvolutional decoder on feature maps extracted by the first CNN module. As for [31], it successfully improves the deblurring performance by using localization deblurring cues via a fine-to-coarse hierarchical representation.

Deblurring Results. The PSNR and SSIM results of competing methods for non-uniform blind motion deblurring on the dataset GoPro are shown in

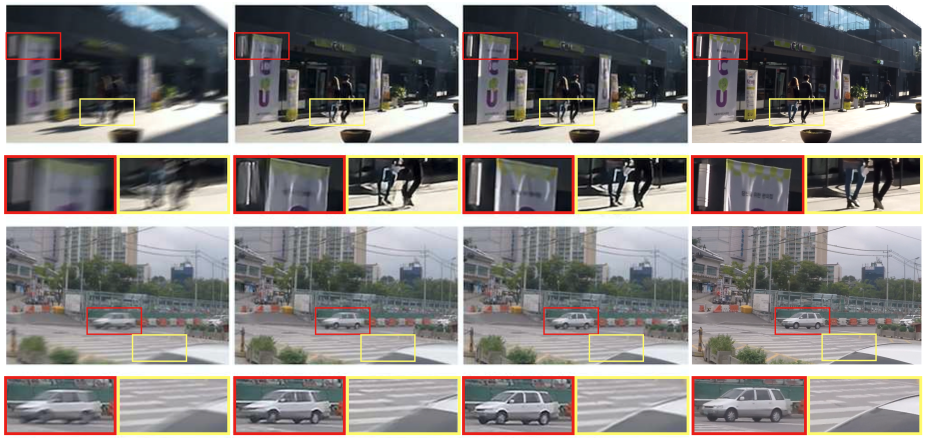

Fig. 3: Deblurring performance on the blurry images from the GoPro dataset. The first column contains the original blurry images, the second column is the result of [26], the third column is the result of [31]. Our results are presented in the last column which achieve the best performance across different scenes.

Table 2, from which we have several observations. Firstly, recent studies on non-uniform blind motion deblurring focus on deep end-to-end networks all reach relatively excellent performance. Secondly, our proposed framework outperforms the based deep network [31] in all of the stacking diagrams about $0.15 \sim 0.24$ dB on PSNR without extra parameters. Furthermore, we find that our framework based on shallow stacked networks tend to outperform deeper stacked networks themselves (*i.e.*Our proposed Stack(3)-DMPHN outperforms Stack(4)-DMPHN 0.12 dB on PSNR) which shows that we successfully design the deep plug-an-play deblurring framework with dynamic stacked networks to explore more optimal clean images. Thirdly, since we can improve deblurring performance simply with whatever pre-trained deep deblurrers both on PSNR and SSIM, deep plug-and-play framework show the feasibility to combine both the advantages of model-based and learning-based IR methods.

Besides, stacked variant Stack(4)-DMPHN (including our framework based on deep deblurrers) outperformed shallower model DMPHN by 1 % PSNR, VMPHN outperformed DMPHN by 0.7% PSNR while stacked variant Stack(2)-VMPHN outperformed shallower DMPHN by 1.3% PSNR. SSIM scores also indicate the same trend. Part of the results are visualized in Fig. 3.

4.3 Analysis of Convergence

To evaluate the superiority of convergence, we compare our dynamic stacked networks with three static stacked networks with the same parameters in complex motion deblurring. In Fig. 4, we represent the deblurring performance of our dynamic diagram varying with iteration numbers in solid lines and the performance of simply iterating static stacked networks in dotted lines. From the

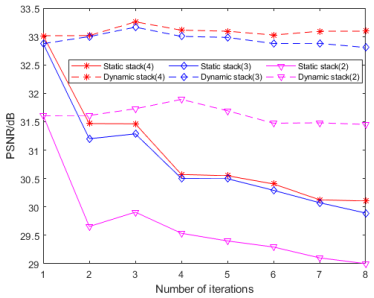

Fig. 4: Comparison between our dynamic stacking diagrams with iterating static pre-trained stacked networks simply. One can see that our framework reaches optimal solution within 4 iterations.

(a) **Input** (b) **I1(31.89)** (c) **I2(31.94)** (d) **I3(32.06)** (e) **I4(32.16)** (f) **I5(31.98)**

Fig. 5: Outputs and PSNR of different iterations of our proposed framework based on Stack(4)-DMPHN. From left to right and up to bottom are the images of different iterations denoted by I with concrete epochs.

results we have several observations: 1) Compared to static ones whose performance decrease gradually, our dynamic stacking diagram can leverage the prior to improve performance by iterations effectively; 2) Our dynamic stacking diagram is able to solve for the optimal result with fast convergence, superior to conventional models, which need large numbers of iterations and lack the reasonable stop criterion [5], as well as existing deep plug-and-play methods, which need at least 15 iterations based on Fast Fourier Transform (FFT) [34]. The visualization of the iteration process in our dynamic stacking diagram is shown in Fig. 5, which is an instance to prove that our frameworks attain the cleaner image gradually within 4 iterations indicated by PSNR. As far as we are concerned, two main reasons contribute to fast convergence: 1) We exploit the pre-trained stacked networks as the deep prior, which goes through large numbers of modulation on parameters to explore the latent clean images similar to the process of iterations in conventional models; 2) We solve the forward model with a single step of gradient descent, which improves the computation efficiency and still attains the good results. However, the drawback of the fast convergence is that our framework may cause over-fitting due to the gradient descent. That's why we can see the decrease in performance with more iterations.

4.4 Mathematical Explanations

Compared to the static stacking diagram, dynamic stacking levels are determined by the iteration number based on deep plug-and-play model. Furthermore, the iteration convergence is supported by theoretical explanations below. Since $\nabla_x \mathcal{L}(x, v)$ according to Eqn. (3) is Lipschitz continuous and our forward model is a single step of gradient descent, we have the property

$$\mathcal{L}(x^{(t)}, v^{(t)}) - \mathcal{L}(x^{(t+1)}, v^{(t)}) \geq C_1 ||x^{(t)} - x^{(t+1)}||_2^2 \tag{11}$$

where C_1 is a positive constant related to Lipschitz constant and step size. According to [2], our deep prior can be regarded as a approximately orthogonal projection of the blurry input y to the manifold of clean images. Therefore, we have

$$\mathcal{L}(x^{(t+1)}, v^{(t)}) - \mathcal{L}(x^{(t+1)}, v^{(t+1)}) \geq C_2 ||\tilde{\nabla}_v \mathcal{L}(x^{(t+1)}, v^{(t)})||_2^2 \tag{12}$$

where $C_2 > 0$ and $\tilde{\nabla}_v \mathcal{L}(x^{(t+1)}, v^{(t)})$ is a continuous limiting subgradient of $\mathcal{L}$. Therefore, by adding Eqn. (11) and Eqn. (12), the sequence $(x^{(t)}, v^{(t)})$ is proved to be bounded and has a convergent subsequence to (x^*, v^*). Then, telescopic summing over $t = 0, 1, ...$ and by monotonicity and boundedness of $\mathcal{L}(x^{(t)}, v^{(t)})$, we have the summability property

$$\lim_{t \to \infty} ||x^{(t)} - x^{(t+1)}||_2 = 0 \tag{13}$$

$$\lim_{t \to \infty} ||\tilde{\nabla}_v \mathcal{L}(x^{(t+1)}, v^{(t)}||_2 = 0 \tag{14}$$

Thus, $\nabla_x \mathcal{L}(x^*, v^*)$ and $\nabla_v \mathcal{L}(x^*, v^*)$ all equal to 0 which makes (x^*, v^*) a stationary point standing for the optima. Above all, our framework can find for the optimal clean restoration image.

5 Conclusion

In this paper, we have stacked networks dynamically for IR based on the plug-and-play framework. Different from static stacked networks, our framework not only shows the performance improvement but also finds for the optimal solution with solid theoretic support. In addition, we have designed a new degradation model with a novel update scheme to better integrate the model-based and learning-based methods. We have also transformed the forward model into a single step of gradient descent effectively for faster convergence. Simply based on pre-trained networks, our framework can remove noise and complex motion blur beyond networks themselves. Experiments on the noise dataset BSD68, Set12, and motion blur dataset GoPro have proven the effectiveness of our framework. In the future, more research on the architecture of deep end-to-end networks will boost the use of our framework.

Acknowledgement

This work was supported in part by grants from the National Natural Science Foundation of China (NSFC, No. 61973007, 61633002).

References

1. Afonso, M.V., Bioucas-Dias, J.M., Figueiredo, M.A.: Fast image recovery using variable splitting and constrained optimization. IEEE transactions on image processing **19**(9), 2345–2356 (2010)
2. Alain, G., Bengio, Y.: What regularized auto-encoders learn from the data-generating distribution. The Journal of Machine Learning Research **15**(1), 3563–3593 (2014)
3. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Foundations and Trends® in Machine learning **3**(1), 1–122 (2011)
4. Burger, H.C., Schuler, C.J., Harmeling, S.: Image denoising: Can plain neural networks compete with bm3d? In: 2012 IEEE conference on computer vision and pattern recognition. pp. 2392–2399. IEEE (2012)
5. Chan, S.H., Wang, X., Elgendy, O.A.: Plug-and-play ADMM for image restoration: Fixed-point convergence and applications. IEEE Transactions on Computational Imaging **3**(1), 84–98 (2016)
6. Chen, Y., Pock, T.: Trainable nonlinear reaction diffusion:A flexible framework for fast and effective image restoration. IEEE transactions on pattern analysis and machine intelligence **39**(6), 1256–1272 (2016)
7. Dabov, K., Foi, A., Katkovnik, V., Egiazarian, K.: Image denoising by sparse 3-D transform-domain collaborative filtering. IEEE Transactions on image processing **16**(8), 2080–2095 (2007)
8. Danielyan, A., Katkovnik, V., Egiazarian, K.: Image deblurring by augmented Lagrangian with BM3D frame prior. In: Workshop on Information Theoretic Methods in Science and Engineering (WITMSE), Tampere, Finland. pp. 16–18 (2010)
9. Dong, W., Wang, P., Yin, W., Shi, G., Wu, F., Lu, X.: Denoising prior driven deep neural network for image restoration. IEEE transactions on pattern analysis and machine intelligence **41**(10), 2305–2318 (2018)
10. Gharbi, M., Chaurasia, G., Paris, S., Durand, F.: Deep joint demosaicking and denoising. ACM Transactions on Graphics (TOG) **35**(6), 191 (2016)
11. Gu, S., Timofte, R., Van Gool, L.: Integrating local and non-local denoiser priors for image restoration. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 2923–2928. IEEE (2018)
12. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: Proceedings of the IEEE international conference on computer vision. pp. 1026–1034 (2015)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Lehtinen, J., Munkberg, J., Hasselgren, J., Laine, S., Karras, T., Aittala, M., Aila, T.: Noise2noise: Learning image restoration without clean data. arXiv preprint arXiv:1803.04189 (2018)
15. Liu, D., Wen, B., Fan, Y., Loy, C.C., Huang, T.S.: Non-local recurrent network for image restoration. In: Advances in Neural Information Processing Systems. pp. 1673–1682 (2018)
16. Liu, P., Zhang, H., Zhang, K., Lin, L., Zuo, W.: Multi-level wavelet-cnn for image restoration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 773–782 (2018)
17. Liu, R., Fan, X., Cheng, S., Wang, X., Luo, Z.: Proximal alternating direction network: A globally converged deep unrolling framework. In: Thirty-Second AAAI Conference on Artificial Intelligence (2018)

18. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3883–3891 (2017)
19. Ono, S.: Primal-dual plug-and-play image restoration. IEEE Signal Processing Letters **24**(8), 1108–1112 (2017)
20. Plötz, T., Roth, S.: Neural nearest neighbors networks. In: Advances in Neural Information Processing Systems. pp. 1087–1098 (2018)
21. Reehorst, E.T., Schniter, P.: Regularization by denoising: Clarifications and new interpretations. IEEE Transactions on Computational Imaging **5**(1), 52–67 (2018)
22. Roth, S., Black, M.J.: Fields of experts. International Journal of Computer Vision **82**(2), 205 (2009)
23. Santhanam, V., Morariu, V.I., Davis, L.S.: Generalized deep image to image regression. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5609–5619 (2017)
24. Sellent, A., Rother, C., Roth, S.: Stereo video deblurring. In: European Conference on Computer Vision. pp. 558–575. Springer (2016)
25. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 769–777 (2015)
26. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8174–8182 (2018)
27. Venkatakrishnan, S.V., Bouman, C.A., Wohlberg, B.: Plug-and-play priors for model based reconstruction. In: 2013 IEEE Global Conference on Signal and Information Processing. pp. 945–948. IEEE (2013)
28. Wang, S., Wen, B., Wu, J., Tao, D., Wang, Z.: Segmentation-aware image denoising without knowing true segmentation. arXiv preprint arXiv:1905.08965 (2019)
29. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: Advances in neural information processing systems. pp. 1790–1798 (2014)
30. Yue, Z., Yong, H., Zhao, Q., Zhang, L., Meng, D.: Variational denoising network: Toward blind noise modeling and removal. arXiv preprint arXiv:1908.11314 (2019)
31. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep Stacked Hierarchical Multi-patch Network for Image Deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5978–5986 (2019)
32. Zhang, J., Pan, J., Ren, J., Song, Y., Bao, L., Lau, R.W., Yang, M.H.: Dynamic scene deblurring using spatially variant recurrent neural networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2521–2529 (2018)
33. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Transactions on Image Processing **26**(7), 3142–3155 (2017)
34. Zhang, K., Zuo, W., Gu, S., Zhang, L.: Learning deep CNN denoiser prior for image restoration. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 3929–3938 (2017)
35. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: 2011 International Conference on Computer Vision. pp. 479–486. IEEE (2011)