Non-Blind Image Deblurring via Deep Learning in Complex Field

Yuhui Quan, Peikang Lin, Yong Xu*, Yuesong Nan and Hui Ji

***Abstract*—Non-blind image deblurring is about recovering the latent clear image from a blurry one generated by a known blur kernel, which is an often-seen yet challenging inverse problem in imaging. Its key is how to robustly suppress noise magnification during the inversion process. Recent approaches made breakthrough by exploiting convolutional neural network (CNN)-based denoising priors in the image domain or gradient domain, which allows using a CNN for noise suppression. The performance of these approaches is highly dependent on the effectiveness of the denoising CNN in removing magnified noise whose distribution is unknown and varies at different iterations of the deblurring process for different images. In this paper, we introduce a CNN-based image prior defined in the Gabor domain. The prior not only utilizes the optimal space-frequency resolution and strong orientation selectivity of Gabor transform, but also enables using complex-valued representations in intermediate processing for better denoising. A complex-valued CNN is developed to exploit the benefits of the complex-valued representations, with better generalization to handle unknown noises over the real-valued ones. Combining our Gabor-domain complex-valued CNN-based prior with an unrolling scheme, we propose a deep-learning-based approach to non-blind image deblurring. Extensive experiments have demonstrated the superior performance of the proposed approach over the state-of-the-art ones.**

***Index Terms*—Image deblurring, Gabor transform, Complex-valued CNN, Deep Learning, Inverse problem**

I. Introduction

IMAGE blurring is one prime cause of poor image quality, which removes important image details such as edges. Let $\boldsymbol{Y}, \boldsymbol{X}$ denote an observed blurry image and its latent clear version respectively. In many scenarios, the formation of a blurry image can be formulated as a convolution process:

$$\boldsymbol{Y} = \boldsymbol{X} * \boldsymbol{K} + \boldsymbol{N}, \tag{1}$$

where $\boldsymbol{K}, \boldsymbol{N}, *$ denote the blur kernel, measurement noise and convolution operation respectively. Image deblurring is about estimating (deconvolving) $\boldsymbol{X}$ from $\boldsymbol{Y}$. Such a task is referred to as non-blind/blind deblurring when $\boldsymbol{K}$ is given/unknown. In this paper, we focus on the non-blind image deblurring, which is not only a critical technique in blind image deblurring, but also a vital tool in many other image restoration tasks, *e.g.* super-resolution. Non-blind image deblurring also has practical values in the scenarios where hardware-assistant modules for blur kernel estimation are available.

Owing to its ill-posedness, the non-blind image deblurring is a challenging problem even with the blur kernel given. Recovering the high-frequency components diminished or killed by the blurring process via pseudo inverse inevitably magnifies the noise. Particularly, when the noise strength exceeds a certain level, the magnified noise may dominate the image. Many existing approaches impose certain image priors to regularize the inversion process for suppressing the noise magnification. In general, given that $\boldsymbol{N}$ is the additive white Gaussian noise (AWGN) with standard deviation (s.t.d.) σ, these approaches consider the following model for deblurring:

$$\min_{\boldsymbol{X}} \ \frac{1}{\sigma^2}\|\boldsymbol{Y} - \boldsymbol{X} * \boldsymbol{K}\|_{\mathrm{F}}^2 + \lambda\phi(\boldsymbol{X}), \tag{2}$$

where $\phi(\cdot)$ is the functional *w.r.t.* certain image prior and $\lambda \in \mathbb{R}$ determines the strength of the prior.

In recent years, unrolling-based deep learning approaches (*e.g.* [1]–[7]) have shown promising performance in non-blind image deblurring. In principle, these approaches define $\phi(\cdot)$ as the prior learned by one or more convolutional neural networks (CNNs) and unroll the optimization process of model (2) into a trainable neural network (NN). There are mainly two sub-processes alternating in an unrolling-based approach: at the t^{th} iteration,

- **(a) Inversion:** Estimating the latent image $\boldsymbol{X}^{(t)}$ using both $\boldsymbol{Y}$ and the output from previous denoising process;
- **(b) Denoising:** Removing the noises (artifacts) from $\boldsymbol{X}^{(t)}$ output by the previous inversion process, using CNNs.

The main difference among existing unrolling-based approaches lies in the different designs of the two sub-processes.

The performance of an unrolling-based approach is highly dependent on the denoising capability of its denoising module. Nonetheless, the denoising task arising from the unrolled optimization process is much more difficult than the traditional ones that focus on the noise with known distribution (*e.g.* AWGN). One main challenge comes from that the noise produced at each iteration of an unrolled process often has a complex yet unknown distribution. The noise magnified by the inversion is unlikely to be AWGN, even if the noise $\boldsymbol{N}$ in the input image is AWGN. Furthermore, such noise may also have quite discrepant characteristics for different input

Yuhui Quan, Peikang Lin and Yong Xu are with School of Computer Science and Engineering at South China University of Technology, Guangzhou 510006, China. (email: csyhquan@scut.edu.cn; csalanlin@mail.scut.edu.cn; yxu@scut.edu.cn)

Yuesong Nan and Hui Ji are with Department of Mathematics at National University of Singapore, Singapore 119076. (email: nanyuesong@u.nus.edu; matjh@nus.edu.sg)

Asterisk indicates the corresponding author.

This work was supported in part by National Natural Science Foundation of China under Grants 61872151 and 62072188, in part by Science and Technology Program of Guangdong Province under Grant 2019A050510010, in part by Science and Technology Program of Guangzhou under Grant 201802010055 and in part by Singapore MOE Academic Research Funds R-146-000-315-114 and MOE2017-T2-2-156.

images, different blur kernels and different measurement noise strengths. In many scenarios, there is a lack of sufficient data for training a CNN to remove such complex and variable intermediate noise. A training dataset which covers all kinds of image patterns, all possible kernels and all possible noise strengths is indeed very challenging to construct.

Early unrolling-based approaches (*e.g.* [3]) employ a CNN pre-trained on AWGN removal (*e.g.* [8]) as the denoising module. However, the CNN trained using AWGN cannot be well adapted to the noise generated in the unrolled optimization, and the resulting performance is limited. Many state-of-the-art (SOTA) approaches (*e.g.* [6], [7]) train the denoising CNN together with the inversion process by the end-to-end manner. One key in these approaches is how to design an effective CNN with strong expressibility as well as good generalizability, for overcoming the deficiency of training data.

In addition to the denoising sub-process, the inversion sub-process also plays an important role in an unrolling-based approach. The performance of the inversion sub-process not only relies on the quality of the denoising results, but also is related to the selection of the prior-related regularization functional $\phi(\cdot)$ that determines the detailed form of inversion.

A. Basic Idea and Motivations

To improve both the denoising and inversion sub-processes, in this work, we introduce a CNN-based denoising prior defined in the Gabor domain and investigate the advantages of complex-valued (CV) representation over the real-valued (RV) one for non-blind image deblurring. Concretely, we define the functional $\phi(\cdot)$ in the Gabor domain and develop a complex-valued CNN (CV-CNN) for denoising. The benefits of such a scheme are discussed as follows.

1) Benefits from Gabor-domain prior: Gabor transform is known for its optimality on joint time-frequency resolution and strong orientation selectivity. Many previous studies (*e.g.* [9]) have shown that Gabor-based regularizations lead to noticeable improvement over classic regularizers such as total variation (TV) and the ones using local discrete cosine transform or real wavelets. Gabor transform also has deep connections to visual perception, *e.g.* both simple cells and complex cells can be well modeled by Gabor wavelets [10].

2) Benefits from CV representations: Recall that the Gabor coefficients are complex numbers. Complex numbers are known to have richer representational capacity than the real ones. It is observed in many studies that the properties of natural images can be easily captured using complex numbers, *e.g.*, the local phase of an image often offers detailed information of objects on shapes, edges and orientations, and it is sufficient to use phase to restore the main information in the image [11]. Therefore, the Gabor-domain CV representation has the potential to improve the denoising process.

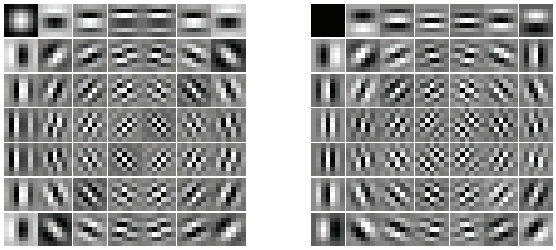
3) Benefits of CV-CNN in terms of generalizability: Using complex numbers in CNNs also has many benefits. Many works suggest that owing to the redundancy in CV representations, CV-CNNs could have better generalizability [12] and noise-robust memory mechanisms [13]. Indeed, a CV-CNN is not a simple double-dimension real-valued CNN (RV-CNN). The interactions between real parts and imaginary parts in the CV operations of a CV-CNN encode implicit connectivity with regularization effects. As a demonstration, we compare DnCNN [8], a classic denoising CNN, with its CV counterpart in terms of removing real-world noise on the DND dataset [14]. Recall that the main ingredients of DnCNN are convolutional (Conv) layers, batch normalization (BN) and rectified linear unit (ReLU), all of which deal with real numbers. The CV counterpart of DnCNN is constructed by replacing all its Conv layers with the CV ones, with adjustment on the NN's width (*i.e.* number of channels at each layer) to ensure the same size of two CNNs. The ReLU/BN are run on the real part and imaginary part separately. These two versions of DnCNNs are trained with the same training method used in [8], including the optimizer, initialization scheme and training data. The AWGN with strength randomly chosen from $[0, 55]$ is added to the images of the BSD500 dataset [15] for generating noisy/clean image pairs for training. The results on the DND dataset which contains 50 images with real noise of unknown strength are listed in Table I. It can be seen that CV-DnCNN performs much better with the same model size. The different characteristics between AWGN and real-world noise, together with the superior performance of CV-DnCNN over DnCNN, suggest that CV-CNNs have potentially better generalizability in handling complex and unknown noise, *e.g.* the noise generated in the inversion sub-process of deblurring.

TABLE I: Denoising results of DnCNN and its CV counterpart on DND Dataset [14].

Method	DnCNN	CV-DnCNN
PSNR(dB)/SSIM	32.90/0.82	34.41/0.86

4) Benefits of CV-CNN in terms of compactness: CV operations enable effective size reduction on CNN models. Firstly, the response of a CV filter contains two feature maps (*i.e.* real part and imaginary part) whereas they are viewed as one channel. Thus, we can halve the model by using CV expression. Consider a 64ch-to-64ch ('ch' for channels) RV Conv layer with a 3×3 filter. Its parameter number is $64 \times 64 \times 3 \times 3$. Then consider a 32ch-to-32ch CV Conv layer. Its parameter number is $32 \times 32 \times 2 \times 3 \times 3$ where "2" is for real and imaginary parts of complex number. Therefore, we can have a 2X model reduction by using the CV expression. Secondly, we can use the tensor product of two 1D CV filters to simulate 2D filters with different orientations. See Fig. 1 for an example. The 1D tensor product decomposition enables compact representation in the block of 2D convolution. All such advantages not only lead to model reduction (additional 1.5X reduction in the above example), but also helps avoiding overfitting, particularly when the training data is insufficient, which is often the case for for deblurring.

To summarize, there are many appealing features of using Gabor-domain representation and CV-CNN based processing for constructing the inversion sub-process and the denoising sub-process. Such features inspired us to develop a Gabor-domain CV-CNN-based unrolled optimization approach for non-blind image deblurring.



(a) Real part (b) Imaginary part

Fig. 1: 2D filters generated by the tensor products of all pairs of seven 1D Gabor filters in $\mathbb{C}^7$. The figure is quoted from [9].

B. Contributions

Motivated by the benefits of Gabor representation and CV-CNN, in this paper, this paper studied how such complex-valued representations and CV-CNN can benefit the development of more powerful deep learning method for non-blind image deblurring. The proposed method is built upon a Gabor-filter-based inversion process and a CV-CNN-based denoising prior. Extensive experiments on benchmark datasets have demonstrated the effectiveness of the proposed approach and its superior performance over existing ones. Our contributions are three-fold:

- First, the Gabor-domain denoising prior is exploited in the unrolled optimization model. Compared to the CNN-based denoising prior defined in image domain or gradient domain, ours can exploit the strong orientation selectivity and excellent space-frequency analysis capability of Gabor transform for better inversion and denoising. In addition, we develop a channel attention module for better inversion with the estimated Gabor coefficients of different channels.
- Second, we develop a CV-CNN for the denoising module, which comprises compact non-separable convolution blocks implemented by successive 1D convolutions of CV filters. We also extend the recently-proposed h-swish activation unit [16] to the CV version, which allows sophisticated interactions between real parts and imaginary parts in the CV-CNN. Our CV-CNN has the merits of compactness and better generalization for unseen noises.
- Almost all the CNNs used in existing image deblurring approaches and other image recovery tasks are RV-CNNs. In this paper, we investigate the advantages of CV-CNNs over RV-CNNs for image deblurring and noise removal. Considering noise removal is a critical part in many image inverse problems, our experimental results indeed show the potentials of CV-CNNs for image recovery.

II. Related Work

In this section, we perform a literature review on non-blind image deblurring, with a particular focus on the deep-learning-based methods. A short review on blind image deblurring is also given. In addition, we have a brief review on the development of CV-CNNs.

A. Model-Based Non-Blind Image Deblurring

There is abundant literature on developing regularization models for non-blind image deblurring. What image prior is used and how its is used outline the main differences among these regularization-model-based approaches. We only give a brief review as follows. **(a) Sparsity Prior:** Assuming natural images have sparse local intensity variations, many classic approaches developed ℓ_1 regularization models based on TV (*e.g.* [17]) or wavelet (*e.g.* [18]). In [19], the $\ell_p (p \in (0, 1])$ regularization is used to better model the heavy-tailed statistical distribution on image gradients. **(b) Nonlocal Self-Similarity (NSS) Prior:** The so-called non-local approaches consider the repeating of similar image patches. IDDBM3D [20] used a non-local denoising framelet system to exploit the NSS of patches in the variational deblurring model. NCSR [21] constructed a sparse-coding-based deblurring model based on NSS of the input image. LNLFrame [22] incorporated NSS into wavelet frame for deblurring. **(c) Learned Natural Image Prior:** Instead of using pre-defined image priors, some approaches attempted to learn image priors from a set of natural images. FoE [23] learned a parametric random field of image gradients. EPLL [24] learned a mixture of Gaussians on clean image patches.

B. Deep Learning for Non-blind Image Deblurring

With the revolution of deep learning in recent years, there have been great progresses on the learning-based approaches for non-blind image deblurring. In practice, directly learning a mapping from blurry images to the clear ones using CNNs of common structures is neither effective nor efficient for non-blind image deblurring. The reason is, the knowledge on blurry image formation and blur kernel is not utilized in such a straightforward way, and the training process would waste its effort on teaching the CNN such knowledge.

To exploit the knowledge about the blurring process, some approaches treated the deconvolution problem as denoising a deblurred image with artifacts. Schuler *et al.* [25] proposed to first deblur the input image with regularized inverse filtering and then use a multi-layer perceptron (MLP) to remove the artifacts of the deblurred image. Xu *et al.* [26] concatenated a deconvolutional CNN and a denoising CNN, both of which are learnable. The deconvolutional CNN was designed and initialized so as to simulate the pseudo-inverse filtering which is factorized into the weighted sum of separable 1D convolutions. These two approaches need to train their CNNs on individual blur kernel and noise strength, and thus their practicability is limited. Ren *et al.* [27] constructed the deconvolutional layers based on a low-rank approximation to a large number of blur kernels, which enables training a universal CNN that can work on different kernels.

In recent years, the majority of deep learning approaches for non-blind image deblurring are concentrated on unrolling the iteration scheme of regularization approaches. As discussed in Section I, these approaches usually decompose each iteration step into an inversion (deconvolution) process and a denoising process. The denoising process with a predefined image prior is replaced by the one using the prior modeled by a CNN. Such

unrolling-based approaches allow being trained with images degraded by different blur kernels and thus can deal with unseen blur kernels. An inspiring work can be traced back to CSF [28] which learns filters (convolution) and shrinkage (activation) functions in the unrolled optimization of half-quadratic splitting (HQS). Such an approach can be viewed as using a shallow CNN with specific structure for the denoising sub-process at each iteration.

Later, there are some studies along this line. Kruse *et al*. [2] extended CSF by replacing the filter learning and shrinkage learning with the denoising CNN learning. Zhang *et al*. [1] used HQS-based unrolling and trained a set of denoising CNNs on horizontal/vertical image gradients. Zhang *et al*. [29] also used HQS for the unrolling and trained a denoising CNN with residual learning. To improve the efficiency, Dong *et al*. [7] proposed to solve the inversion sub-problem from HQS using single-step gradient descent. The resulting iterative process is unfolded into a CNN which is composed of multiple denoising modules interleaved with back-projection modules. Intead of employing HQS, Meinhardt *et al*. [3] used the primal-dual hybrid gradient approach for the unrolling and replaced the proximal operator by a denoising CNN. In all above approaches except Kruse *et al*. [2], the models are trained on different noise strengths individually. As a result, they are not good at dealing with varying noise strengths. Jin *et al*. [4] unrolled the Bayesian maximum a posteriori (MAP), which enables estimating the noise strength during the deblurring process. Also unrolling MAP, Bigdeli *et al*. [5] proposed to learn a denoising autoencoder and showed that their approach corresponds to learning a Gaussian smoothed version of natural image distribution. Gong *et al*. [30] unrolled the gradient descent optimizer and incorporated an NN into it, which leads to a recurrent gradient descent network with good generalization for image deblurring. Nan *et al*. [31] unrolled a wavelet-based regularization model via variational expectation maximization. Nan and Ji [32] unrolls a total least squared estimator for improving the robustness of the network to kernel error. Rather than unroll to an iterative optimization algorithm, Gilton *et al*. [33] proposed to approximate the inversion-based solution of a regularized least squares deblurring model using a truncated Neumann series. The regularizer is defined by a CNN, based on which a novel deep model is constructed.

All the aforementioned approaches use RV-CNNs in image domain, gradient domain or RV transform domain. In comparison, our approach defines the denoising prior using CV-CNN in Gabor domain, leading to performance improvement.

C. Blind Image Deblurring

There is also plenty of recent studies focusing on blind image deblurring. Since blind image deblurring is much different from non-blind deblurring, we only have a brief review on this line. Blind uniform deblurring often assumes the blur kernel is also unknown but the noise is not significant. Compared to the nonblind ones, these blind uniform deblurring methods aim at estimating the blur kernel; see *e.g*. [18], [34]–[36]. Once the kernel is determined, the clear image can be restored by calling some non-blind deblurring method. Blind non-uniform deblurring assumes the blurring effect is spatially-varying which cannot be modeled by convolution. Most existing work focuses on non-uniform motion deblurring; see *e.g*. [37], [38].

D. Complex-Valued Convolutional Networks

Recently, an increasing number of studies (*e.g*. [39]–[43]) have paid attention to CV-CNNs. Most of these studies focus on the applications of CV-CNNs to recognition. One related work on image recovery is found in [44], which used a CV MLP to identify the blur kernel for image deconvolution, and the deblurring is done by a traditional approach using the identified kernel. In comparison to this approach, ours is for image deblurring with a given kernel.

III. Main Body

This section is devoted to describing the proposed deep NN for non-blind image deblurring. We start from an unrolled optimization, based on which a deep NN framework is developed. Then, the details of each block and training loss of the framework are presented. Through this section, unless specified, we use uppercase hollow letters for sets, uppercase calligraphic letters for operators, uppercase boldfaced letters for matrices, lowercase boldfaced letters for vectors, and normal letters for scalars. Let $\mathbb{R}, \mathbb{C}$ denote the set of real numbers and complex numbers respectively. Let $\Re(\cdot), \Im(\cdot)$ denote the real part and the imaginary part of a complex number respectively. Let $\mathbb{E}\{\cdot\}$ denotes the expected value.

A. Unrolled Optimization

The proposed approach unrolls an optimization model built upon a Gabor-domain CNN-based denoising prior. Recall the model of (2) is based on a prior ϕ about the clean image. The prior usually is imposed on high-frequency image components, as they are the main parts lost in the blurring process and to be recovered. Existing approaches often define ϕ in the gradient domain, *e.g*. $\phi(\boldsymbol{X}) = \|\Delta \boldsymbol{X}\|_1$ for the sparsity prior and $\phi(\boldsymbol{X}) = \|\Delta \boldsymbol{X}\|_p (0 < p < 1)$ for the hyper-Laplacian prior. We consider a finer measurement on high frequencies. Instead of only using first-order derivatives along the x, y axes, we use a high-pass Gabor filter bank which enjoys optimal space-frequency resolution and strong orientation selectivity.

Let $\{\boldsymbol{D}_j\}_{j=1}^J$ be a set of 2D Gabor filters. In this paper, we use eight high-pass Gabor filters [9]: $\{\boldsymbol{d}_{k_1}\boldsymbol{d}_{k_2}^\top : 1 \le k_1, k_2 \le 3, k_1 \cdot k_2 \ne 1\}$, which are generated by three 1D Gabor filters:

$$\boldsymbol{d}_1 = [\frac{\sqrt{2}}{6} + 0i,\ \frac{\sqrt{2}}{3} + 0i,\ \frac{\sqrt{2}}{6} + 0i], \tag{3}$$

$$\boldsymbol{d}_2 = [-\frac{\sqrt{2}}{8} + \frac{1}{2\sqrt{6}}i,\ \frac{\sqrt{2}}{4} + 0i,\ -\frac{\sqrt{2}}{8} - \frac{1}{2\sqrt{6}}i], \tag{4}$$

$$\boldsymbol{d}_3 = [-\frac{\sqrt{2}}{8} - \frac{1}{2\sqrt{6}}i,\ \frac{\sqrt{2}}{4} - 0i,\ -\frac{\sqrt{2}}{8} + \frac{1}{2\sqrt{6}}i], \tag{5}$$

where $i = \sqrt{-1}$. Note that the low-pass filter $\boldsymbol{d}_1\boldsymbol{d}_1^\top$ is not used. Then we consider the following optimization problem:

$$\min_{\boldsymbol{X}}\ \|\boldsymbol{Y} - \boldsymbol{X} * \boldsymbol{K}\|_{\mathrm{F}}^2 + \sum_{j=1}^{J} \lambda_j \psi(\boldsymbol{D}_j * \boldsymbol{X}), \tag{6}$$

where $\psi(\cdot)$ denotes some prior-inducing function.

We adopt HQS for solving (6). By introducing a set of auxiliary variables $\{\boldsymbol{A}_j\}_{j=1}^J$ which is CV, the solver alternatively updates $\boldsymbol{X}$ and $\{\boldsymbol{A}_j\}_{j=1}^J$ as follows:

$$\begin{aligned}\boldsymbol{X}^{(t)} &\in \operatorname*{argmin}_{\boldsymbol{X}} \|\boldsymbol{Y}-\boldsymbol{X}*\boldsymbol{K}\|_{\mathrm{F}}^2+\sum_{j=1}^J \beta_j^{(t)}\|\boldsymbol{A}_j^{(t)}-\boldsymbol{D}_j*\boldsymbol{X}\|_{\mathrm{F}}^2,\\ \boldsymbol{A}_j^{(t+1)} &\in \operatorname*{argmin}_{\boldsymbol{A}_j} \|\boldsymbol{A}_j-\boldsymbol{D}_j*\boldsymbol{X}^{(t)}\|_{\mathrm{F}}^2+\alpha_j^{(t)}\psi(\boldsymbol{A}_j),\forall j. \end{aligned} \quad (7)$$

where $\{\beta_j\}_{j=1}^J, \{\alpha_j\}_{j=1}^J$ are two auxiliary parameter sequences. In above, the auxiliary variables $\{\boldsymbol{A}_j^{(t)}\}_j$ can be viewed as the estimation of high frequencies of $\boldsymbol{X}$. In other words, there are two iterative steps:

$$\begin{aligned}&\textbf{Inversion (INV)}: \boldsymbol{Y}, \{\boldsymbol{A}_j^{(t)}\}_j \longrightarrow \boldsymbol{X}^{(t)},\\ &\textbf{Denoising (DN)}: \boldsymbol{X}^{(t)} \longrightarrow \{\boldsymbol{A}_j^{(t+1)}\}_j.\end{aligned} \quad (8)$$

The inversion step is about deconvolving $\boldsymbol{Y}$ regularized by estimated high frequencies of $\boldsymbol{X}^{(t)}$. The denoising step aims at using the prior ψ to remove noise (artifacts) from high-frequencies of $\boldsymbol{X}^{(t)}$.

B. Overall Structure

We develop a deblurring NN by turning the above inversion step and denoising step in (8) into two learnable modules: INV module for inversion and DN module for denoising. See Fig. 2 for an illustration of our NN's architecture. By sequentially and alternatively concatenating INVs and DNs, we construct a CNN for non-blind image deblurring, as shown in Fig. 2(a). Our CNN accepts a blurry image $\boldsymbol{Y}$ and the corresponding blur kernel $\boldsymbol{K}$ as input, and outputs a deblurred image. There are $T+1$ blocks in the NN. The first block is an INV module that outputs an initial estimate $\boldsymbol{X}^{(0)}$ with the given initial $\{\boldsymbol{A}_j^{(0)}\}_j$:

$$\boldsymbol{X}^{(0)} = \mathrm{INV}(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{A}^{(0)}). \quad \text{[Block \#0]} \quad (9)$$

Each of the latter T blocks contains one DN and one INV:

$$\boldsymbol{A}^{(t)} = \mathrm{DN}(\boldsymbol{X}^{(0)}, \cdots, \boldsymbol{X}^{(t-1)}), \quad (10)$$

$$\boldsymbol{X}^{(t)} = \mathrm{INV}(\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{A}^{(t)}), \quad \text{[Block \#t]} \quad (11)$$

for $t = 1, \cdots, T$. The output of the last block, *i.e.* $\boldsymbol{X}^{(T)}$, is used as the final deblurred image of the CNN. Note that weight sharing is not used across the blocks.

C. Inversion Module

There are two problems in the inversion process: (a) how to setup the auxiliary parameter sequence $\{\beta_j^{(t)}\}_{j=1}^J$ and (b) how to solve $\boldsymbol{X}^{(t)}$. The latter has the explicit solution:

$$\widehat{\boldsymbol{X}}^{(t)} = \frac{\widehat{\boldsymbol{K}}^* \odot \widehat{\boldsymbol{Y}} + \sum_{j=1}^J \beta_j^{(t)} \widehat{\boldsymbol{D}}_j^* \odot \widehat{\boldsymbol{A}_j^{(t)}}}{\widehat{\boldsymbol{K}}^* \odot \widehat{\boldsymbol{K}} + \sum_{j=1}^J \beta_j^{(t)} \widehat{\boldsymbol{D}}_j^* \odot \widehat{\boldsymbol{D}}_j}, \quad (12)$$

where $\widehat{(\cdot)}, (\cdot)^*, \frac{(\cdot)}{(\cdot)}$ denote the 2D discrete Fourier transform, element-wise complex conjugate, and element-wise division respectively. Note that we take the real part after the 2D inverse discrete Fourier transform to ensure the final result $\boldsymbol{X}^{(t)}$ is RV.

Regarding $\{\beta_j^{(t)}\}_{j=1}^J$, they can be viewed as the attention weights applied to the feature map (*i.e.* $\boldsymbol{A}_j$) of each Gabor channel respectively. It is a challenging task to have an explicit formula for determining the optimal values of $\beta_j^{(t)}$. In the next, we consider a simplified case by assuming that the residuals $\boldsymbol{A}_j^{(t)} - \boldsymbol{D}_j * \boldsymbol{X} = \boldsymbol{N}_j^{(t)}$ independently follow normal distribution with zero mean and varying variances.

Statement 1. *Assuming that* $\boldsymbol{Y} - \boldsymbol{X} * \boldsymbol{K} = \boldsymbol{N} \sim \mathcal{N}(0, \sigma^2)$, *and* $\boldsymbol{A}_j^{(t)} - \boldsymbol{D}_j * \boldsymbol{X} = \boldsymbol{N}_j^{(t)} \sim \mathcal{N}(0, (\sigma_j^{(t)})^2), \forall j$. *The optimal regularization parameter sequence* $\{\beta_j^{(t)}\}_{j=1}^J$ *is related to the residual magnitude sequence* $\{\gamma_j^{(t)} = \|\boldsymbol{R}_j^{(t)}\|_{\mathrm{F}}^2\}_{j=1}^J$ *where*

$$\boldsymbol{R}_j^{(t)} = \boldsymbol{D}_j * \boldsymbol{Y} - \boldsymbol{K} * \boldsymbol{A}^{(t)}, \quad \forall j, \quad (13)$$

as follows:

$$\frac{1}{N}\|\boldsymbol{R}_j^{(t)}\|_F^2 \approx \sigma^2(c_1 + c_{2,j}/\beta_j^{(t)}) \quad (14)$$

where $c_1, c_{2,j} > 0$.

Proof. See Appendix for the proof. □

Remark 1. Although the formula (14) provides an explicit definition of the parameters $\{\beta_j^{(t)}\}_{j=1}^J$, it cannot be directly used in the method as the statistical assumption on $\boldsymbol{A}_j^{(t)} - \boldsymbol{D}_j * \boldsymbol{X} = \boldsymbol{N}_j^{(t)}$ is over-simplified. Nevertheless, Statement 1 implies that the measurements $\{\|\boldsymbol{D}_j * \boldsymbol{Y} - \boldsymbol{K} * \boldsymbol{A}^{(t)}\|_F^2\}_j$ can be used for predicting the parameters $\{\beta_j^{(t)}\}_{j=1}^J$.

Based on the remark, we construct a so-called channel attention (CA) module

$$\textbf{CA:} \quad \boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{A}^{(t)} \to [\beta_j^{(t)}]_{j=1}^J \in \mathbb{R}^J, \quad (15)$$

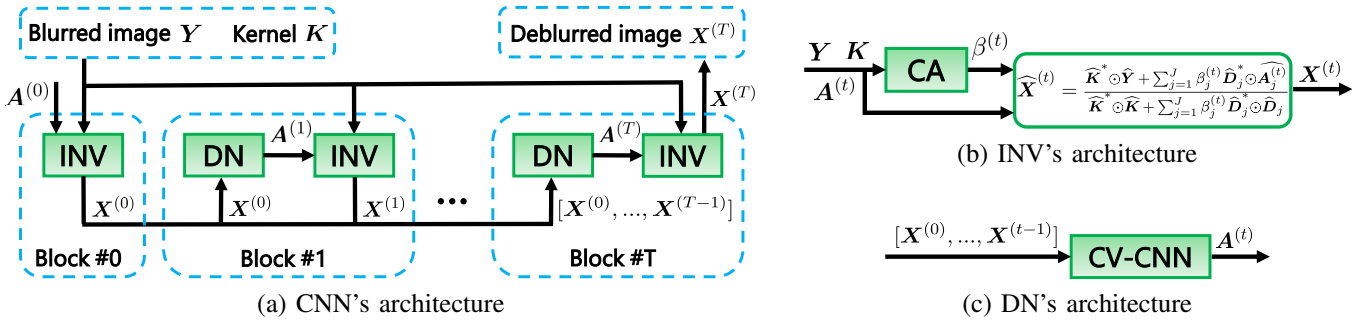
which estimates the weights of Gabor channels $\{\beta_j^{(t)}\}_{j=1}^J$ using the residual magnitudes $\{\boldsymbol{R}_j^{(t)}\}_{j=1}^J$ defined in (13). The CA module is implemented with an MLP with ReLU activation units, whose structure is illustrated in Fig. 3. Given $\boldsymbol{Y}, \boldsymbol{K}, \boldsymbol{A}^{(t)}$, the CA module first calculates the residual $\boldsymbol{R}_j^{(t)}$ for all j using Then it passes $\{\|\boldsymbol{R}_j^{(t)}\|_{\mathrm{F}}^2\}_{j=1}^J$ to a 3-layer MLP. The output of MLP is defined as the Gabor channel weights $\{\beta_j^{(t)}\}_{j=1}^J$.

Based on all above, the INV module is implemented by combining the learnable CA module that output weights of Gabor channels and a non-learnable module that calculates Eq. (12). See Fig. 2(b) for an illustration of INV's structure.

D. Denoising Module

The DN module first transforms $\boldsymbol{X}^{(t)}$ to the Gabor domain and then outputs the J-channel denoised Gabor coefficients. For better performance, we reuse the previous estimates $\boldsymbol{X}^{(t-1)}, \cdots, \boldsymbol{X}^{(0)}$ as additional input which contain extra information that might be helpful for denoising. Each previous estimate is also transformed to Gabor domain first. All these Gabor coefficient maps are fed to a CV-CNN which can deal with and exploit the Gabor-domain CV representations. See Fig. 2(c) for an illustration of DN's structure.

The structure of the CV-CNN used in our approach is illustrated in Fig. 4(a). The CV-CNN at the t-th block calculates the



(a) CNN's architecture (b) INV's architecture (c) DN's architecture

Fig. 2: Diagram of proposed CNN for non-blind image deblurring.

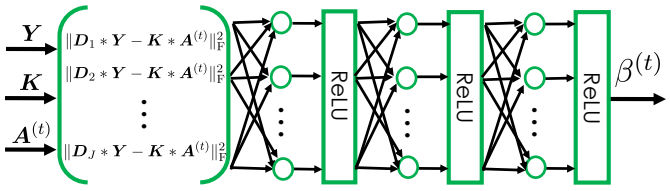


Fig. 3: Diagram of structure of CA.

Gabor coefficient maps $\{\boldsymbol{M}_j^{(0)}\}_{j=1}^J, \cdots, \{\boldsymbol{M}_j^{(t-1)}\}_{j=1}^J$ (CV) of $\boldsymbol{X}^{(0)}, \cdots, \boldsymbol{X}^{(t-1)}$:

$$\boldsymbol{M}_j^{(t_0)} = \boldsymbol{D}_j * \boldsymbol{X}^{(t_0)}, \quad \forall t_0. \tag{16}$$

Then it stacks all these maps as input and outputs the denoised high frequencies $\boldsymbol{A}^{(t)}$ which is CV. The input is passed to 14 sequentially-connected convolutional units. Each convolutional unit except the first and last ones contains a complex-valued Conv (CV-Conv) layer followed by a BN and a complex-valued h-swish (CV-HS) activation. The first convolutional unit does not contain BN while the last one only contains a CV-Conv layer. The CV-Conv layer is implemented by replacing all the RV convolution operations in a Conv layer with the CV ones. For a CV feature map $\boldsymbol{F}$ and a CV kernel $\boldsymbol{G}$, the CV-Conv is defined by

$$\begin{aligned}\boldsymbol{F} * \boldsymbol{G} = &(\Re(\boldsymbol{F}) * \Re(\boldsymbol{G}) - \Im(\boldsymbol{F}) * \Im(\boldsymbol{G})) \\ &+ (\Im(\boldsymbol{F}) * \Re(\boldsymbol{G}) + \Re(\boldsymbol{F}) * \Im(\boldsymbol{G}))i,\end{aligned} \tag{17}$$

It can be seen that the real part and the imaginary part interact with each other in CV-Conv. On all CV-Conv layers, the 2D CV-Conv is implemented by two consecutive 1D CV-Convs:

$$\boldsymbol{F} * \boldsymbol{G} = \boldsymbol{F} * \boldsymbol{g}_1 * \boldsymbol{g}_2, \tag{18}$$

where $\boldsymbol{g}_1, \boldsymbol{g}_2$ are a horizontal 1D CV kernel and a vertical 1D CV kernel respectively. See Fig. 4(b) for the illustration of the implementation of the CV-Conv layer. As discussed in Section I-A, such a factorization can reduce the model size while keeping the rich expressibility of the NN.

The CV-HS is implemented by extending the h-swish function (RV) [16]. We use the h-swish function for two reasons: (a) it is shown in [16] that h-swish brings possible improvement over other often-seen activation functions such as ReLU, with negligible computational latency cost; (b) its simple CV extension involves explicit interactions between real part and imaginary part. The extension is given as follows:

$$\begin{aligned}&\text{CV-HS}(x) = \\ &\left(\Re(x)\frac{\text{ReLU6}(\Re(x)+3)}{6} - \Im(x)\frac{\text{ReLU6}(\Im(x)+3)}{6}\right) \\ &+\left(\Re(x)\frac{\text{ReLU6}(\Im(x)+3)}{6} + \Im(x)\frac{\text{ReLU6}(\Re(x)+3)}{6}\right)i,\end{aligned} \tag{19}$$

for any $x \in \mathbb{C}$, where ReLU6 denotes the clipped ReLU with maximal output clipped to 6. It can be seen that the CV-HS introduces the interactions between real part and imaginary part. This enables sophisticated nonlinear activation on the phase of input, improving the expressibility of our CV-CNN. The BN is applied to the real part and imaginary part respectively. All CV-Conv layers have 32 output channels.

The middle 8 convolutional units are implemented as 4 residual blocks (RBs) for better performance and faster convergence, whose structure is illustrated in Fig. 4(c). To enlarge the receptive field for further improvement and higher computational efficiency, a CV-Conv layer with stride of 2 is used before/after each RB for downscaling/upscaling respectively. A skip connection is also added from the input of the first RB to the output of the last RB for preserving the image details.

E. Loss Function

Let $\{(\boldsymbol{X}_h, \boldsymbol{Y}_h)\}_{h=1}^H$ denote a set of training image pairs where $\boldsymbol{X}_h$ is a clear image and $\boldsymbol{Y}_h$ is the corresponding blurred image. Let $\boldsymbol{X}_h^{(t)}$ denote the output of the t-th block in our CNN. The loss function for training our CNN model is as follows:

$$L := \frac{1}{H}\sum_{h=1}^{H}\left(\|\boldsymbol{X}_h^{(T)} - \boldsymbol{X}_h\|_{\text{F}}^2 + \mu\sum_{t=1}^{T-1}\|\boldsymbol{X}_h^{(t)} - \boldsymbol{X}_h\|_{\text{F}}^2\right), \tag{20}$$

where $\mu > 0$ is the weight to balance the losses of final result and intermediate results. Note that the output of the first block is not supervised in the loss.

IV. Experiments

This section is for experimental evaluation. The proposed approach is evaluated on several benchmark datasets with different noise settings: AWGN with known strengths, AWGN with varying strengths, and Poisson noise. A robustness test considering kernel error is also given. Finally, ablation studies and parameter influence tests are conducted for more analysis.

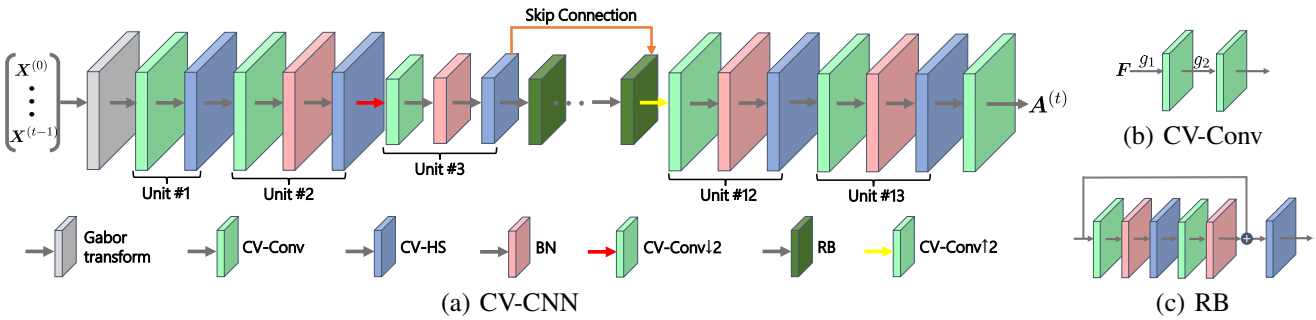


Fig. 4: Diagram of CV-CNN.

A. Experimental Setting

1) Data Preparation: The schemes for generating the training and test data for performance evaluation are as follows:

- **Training data.** For training, 1473 clear images of size 256×256 are generated by randomly cropping the images from the BSDS500 dataset [15] and used as the truth images. On each truth image, we generate 10 blurry images by individually applying 10 different blur kernels randomly selected from the 192 synthetic motion blur kernels used in [45]. All the resulting 14730 blurred images are then corrupted with the AWGN of certain strength. Since the pixel values on the noisy blurry images may exceed the range $[0, 256)$ and may be not integers, we apply pixel-wise 8-bit quantization that rounds each pixel value to an integer and clips them to $[0, 256)$. Different settings on the strength of AWGN are used in different experiments, which will be given latter.
- **Test data.** Three image datasets are used for test: (a) Sun *et al.*'s dataset [46] used in [29]; (b) Levin *et al.*'s dataset [47] used in [2], [4]; and (c) Set18 that contains 18 classic images shown in in supplementary materials. The 8 real motion-blur kernels used in [48], which differ from those for training, are applied to each test image respectively, followed by noise corruption and 8-bit quantization for simulating real scenarios of image storage. Different noise corruption schemes will be used in different experiments. In the robustness test, the blur kernels are generated by running existing kernel estimation methods on input images.

See Table II for the characteristics of the training and test datasets used in the experiments.

TABLE II: Characteristics of training data and test data.

Dataset	BSD500	Sun *et al.*'s	Levin *et al.*'s	Set18
# clear im.	1473	80	4	18
# blur im.	14730	640	32	144
# kernels	192	8		
fixed σ	$1\%, 3\%, 5\%$	$1\%, 3\%, 5\%$		
varying σ	$[0.5\%, 4.5\%]$	$1\% \cup 2\% \cup 3\% \cup 4\%$		
image size	256×256	$\approx$ 900×700	255×255	256×256 512×512

2) Implementation Details: The proposed approach is implemented using Pytorch with CUDA acceleration. We set $\mu = 0.8$ in the loss function and $T = 3$ in the model through all the experiments. Our model is trained with the end-to-end manner using Adam, with initial learning rate of 1×10^{-3}, batch size of 8, and 30 epochs. As for initialization, we set $\boldsymbol{A}_j^{(0)} = \mathbf{0}$ for all j and initialize the weights of our CV-CNN using the Xavier uniform initializer, while the weights/biases of the CA module are all initialized to 1/0. Run on a desktop PC with an NVIDIA Titan-RTX GPU, our training process takes around 98 minutes an epoch.

3) Baselines: The approaches selected for the performance comparison include the representative ones in regularization-model-based non-learning approaches and the recent deep-learning-based approaches. The details on these approaches will be given in each experiment. Throughout all experiments, the PSNR and SSIM are used as the quantitative metrics.

B. Performance Evaluation on Deblurring in The Presence of AWGN With Known Noise Strength

There are some existing non-blind image deblurring approaches that assume the noise is AWGN with known strength. These approaches train their CNNs using the data with the same noise strength as test data. To evaluate the performance of our approach in this setting, we follow the scheme of [29], where a model is trained by fixing the strength of AWGN in training data to $\sigma = 1\%, 3\%, 5\%$ respectively. We use Ours to denote our model trained in this setting. The test images are also corrupted by the AWGN with strength $\sigma = 1\%, 3\%, 5\%$ respectively. Each trained model is tested on the corresponding AWGN strength. The following methods are selected for comparison: EPLL [24], IDDBM3D [20], CSF [28], FCNN [1], IRCNN [29], DMSP [5], DPDNN [7] and VEM [31]. The results of these approaches are quoted from existing literature whenever possible, or produced by their published codes under our same scheme with efforts made on parameter tuning.

The quantitative results are listed in Table III for comparison. Our approach consistently outperforms all others by a large margin on all noise strengths across all datasets, in terms of PSNR. In Fig. 6(a), we also compare the NN-based approaches in terms of overall average PSNR on all datasets versus model size measured by number of parameters. The model size of our approach is comparable to FCNN but significantly less than DMSP, DPDNN and VEM. While having a larger model size, our approach outperforms IRCNN noticeably. The quantitative improvement of our method is also consistent with the improvement on visual quality. See Fig. 5

TABLE III: Average PSNR(dB)/SSIM($\times 10^{-4}$) of deblurred images in image deblurring in the presence of AWGN with known strengths. The best results are marked in **blue** and the second best ones are marked in green.

Dataset	σ	EPLL	IDDBM3D	CSF	FCNN	IRCNN	DMSP	DPDNN	VEM	Ours
Sun *et al.*'s	1%	30.53/8732	32.24/8811	31.04/8633	32.17/8795	30.91/8219	31.79/8639	31.23/8511	32.73/8952	**33.10/9022**
	3%	27.46/7514	28.74/7830	27.84/7345	28.95/7840	27.93/7401	28.26/7715	29.07/8023	29.41/8055	**29.54/8094**
	5%	26.08/6903	27.30/7305	26.53/6630	27.57/7329	27.21/7377	27.54/7440	27.91/7574	28.04/7502	**28.07/7503**
Levin *et al.*'s	1%	32.03/9198	33.75/9244	29.85/8776	32.25/9108	32.66/8746	32.62/9107	31.14/8912	34.31/9382	**35.44/9467**
	3%	28.31/8360	29.26/8500	27.28/7800	29.30/8573	29.15/8220	29.30/8433	28.93/8540	30.50/8798	**30.85/8829**
	5%	27.15/7510	27.33/7938	26.25/7208	27.63/8114	27.56/8035	27.75/8172	27.54/8148	28.52/8348	**28.80/8381**
Set18	1%	29.35/8429	31.65/8565	29.79/8491	31.08/8562	31.93/8217	31.35/8329	30.52/8398	32.07/8790	**32.79/8849**
	3%	26.65/7562	27.77/7782	26.82/7346	27.90/7803	28.76/7540	29.31/7895	28.49/7900	28.99/8025	**29.40/8044**
	5%	25.73/7178	26.15/7316	25.49/6827	26.48/7290	27.21/7299	27.23/7503	27.38/7520	27.54/7561	**27.86/7563**

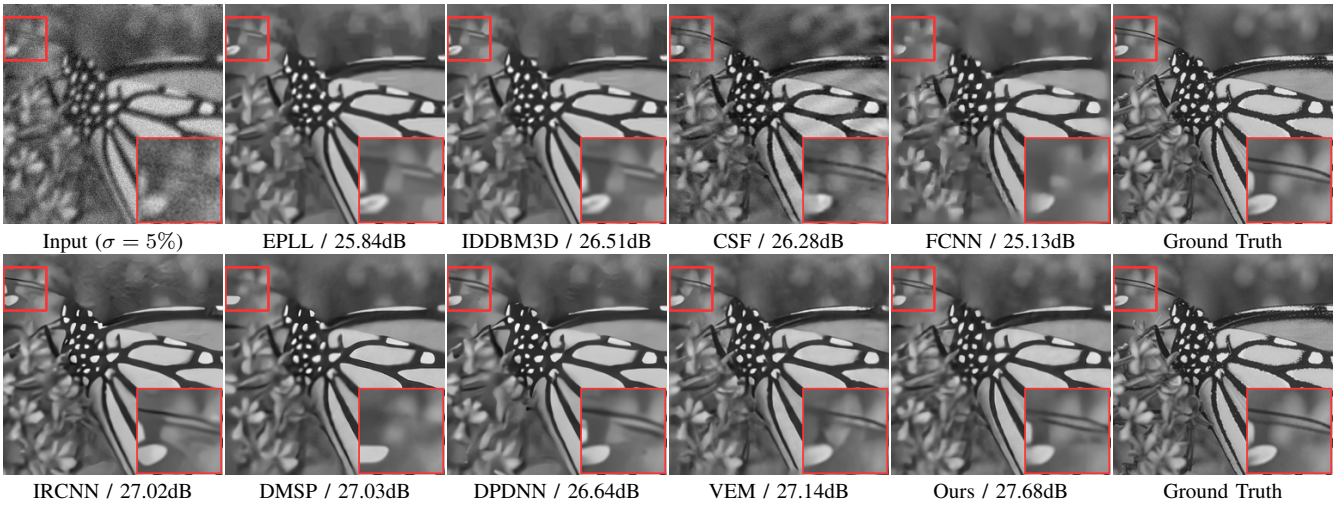
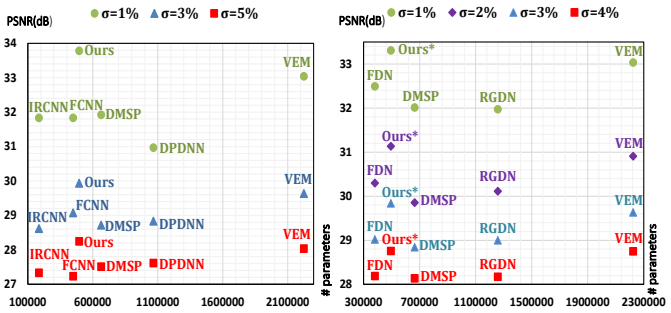


Fig. 5: Visual comparison of deblurring results on image 'Butterfly' in the presence of AWGN with fixed strength.



(a) Known AWGN strength (b) Unknown AWGN strength

Fig. 6: Scatter plots of overall PSNR versus number of parameters on different deep models in two AWGN settings.

for the images deblurred by different approaches on a blurry image 'Butterfly'. It can be found that our result contains more details than other approaches. For instance, the antennae of the butterfly is clearer in our deblurred image than in other approaches. Also, compared to others, the patterns on the butterfly's body and wing recovered by our approach are clearer with less artifacts. The superior performance of our approach is mainly attributed to the better image characterization provided by the Gabor-domain prior for inversion, as well as the power of the CV-CNN for denoising.

C. Performance Evaluation on Deblurring in The Presence of AWGN With Unknown Noise Strength

In practice, the noise strength is usually unknown and an approach blind to the noise strength is more appealing. This setting aims at evaluating the performance of our approach in handling the AWGN with strength varying within a certain range. We follow the scheme of [4], where a universal model is trained with the strength of AWGN randomly chosen from $[0.5\%, 4.5\%]$. We use Ours* to denote our model trained in this setting. The test images are corrupted by AWGN with $\sigma = 1\%, 2\%, 3\%, 4\%$ respectively. The trained model is then tested on all the test images. The following approaches are selected for comparison: IDDBM3D [20], FDN [2], GradNet [4], EPLL+NA [4], DMSP [5], RGDN [30] and VEM [31]. The results of these approaches are quoted from existing literature whenever possible, or produced by their published codes in our setting with efforts made on parameter tuning.

The quantitative results are listed in Table IV for comparison. In Fig. 6(b), we also compare the NN-based approaches in terms of overall average PSNR on all datasets versus model size measured by number of parameters. Overall, our approach performs better than other compared ones in terms of PSNR, with a moderate model size. The VEM method performs slightly better than ours in some settings, but its model size is much larger than that of our model. See Fig. 7 for the

TABLE IV: Average PSNR(dB)/SSIM($\times 10^{-4}$) of deblurred images in image deblurring in the presence of AWGN with unknown strengths. The best results are marked in **blue** and the second best ones are marked in green.

Dataset	σ	IDDBM3D	FDN	EPLL-NA	GradNet7S	DMSP	RGDN	VEM	Ours*
Sun *et al.*'s	1%	32.24/8811	32.30/8837	32.18/8820	31.75/8730	31.76/8637	31.29/8926	32.73/8952	**32.75/8983**
	2%	29.95/8234	30.12/8206	30.08/8260	29.31/7980	29.62/8040	29.66/8385	30.57/8412	**30.61/8430**
	3%	28.74/7830	28.97/7785	28.77/7750	28.04/7500	28.68/7743	28.72/8034	29.41/**8055**	**29.45**/8041
	4%	27.93/7483	28.21/7480	27.81/7360	27.81/7330	28.06/7508	28.04/7760	**28.65/7812**	28.49/7634
Levin *et al.*'s	1%	33.75/9244	33.65/9300	32.16/9230	31.43/9120	32.61/9101	33.47/9378	34.31/9382	**34.86/9429**
	2%	30.96/8809	31.18/8924	30.25/8880	28.88/8410	30.40/8648	31.19/9004	32.02/9063	**32.37/9092**
	3%	29.26/8497	29.79/8641	28.96/8560	27.55/7970	29.31/8432	29.75/8712	30.50/8798	**30.78/8811**
	4%	28.17/8120	28.84/8391	27.85/8240	26.96/7830	28.52/8269	28.67/8433	29.42/**8563**	**29.45**/8474
Set18	1%	31.65/8565	31.54/8618	31.41/8559	-/-	31.68/8488	31.16/8729	32.07/8790	**32.32/8803**
	2%	29.18/8105	29.38/8122	29.60/8190	-/-	29.55/8170	29.50/8280	30.14/**8342**	**30.43**/8327
	3%	27.77/7782	28.31/7745	28.61/7915	-/-	28.54/7852	28.54/7970	28.99/**8025**	**29.29**/7992
	4%	26.85/7459	27.52/7503	27.46/7478	-/-	27.84/7500	27.81/7666	28.18/**7778**	**28.34**/7630

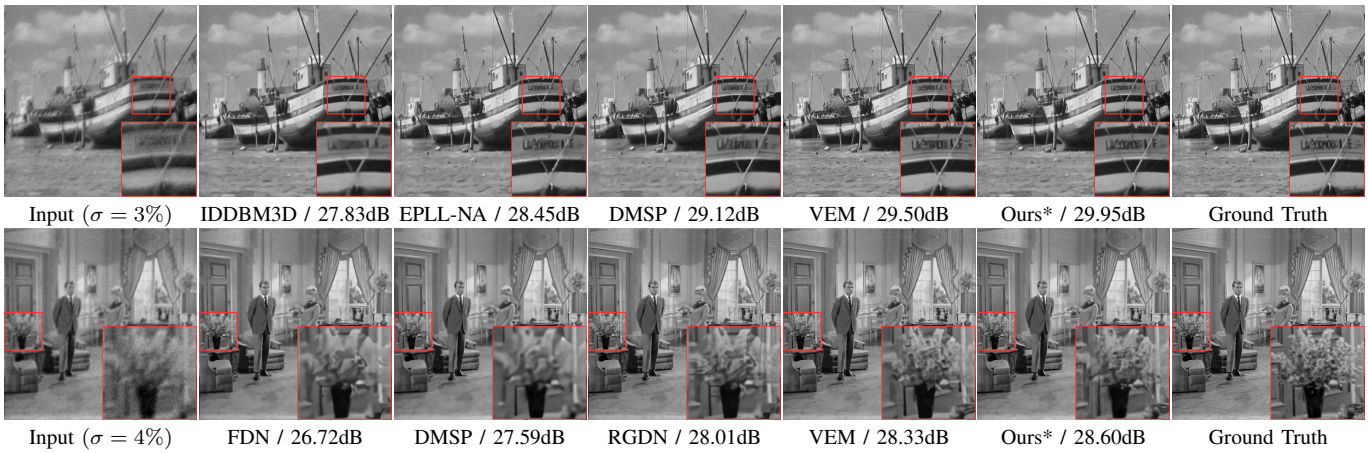


Fig. 7: Visual comparison of deblurring results on images 'Boat' & 'Couple' in the presence of AWGN with unknown strength.

deblurred images of different approaches on two blurry images 'Boat' and 'Couple', where the improvement of our approach over others can also be observed from the visual quality of the deblurred images. Regarding 'Boat', the characters on the boat in our deblurred image are the clearest to be identified. Regarding 'Couple', the flowers in our deblurred image are clearer with more details than those of other approaches. One source of the superior performance of our approach in noise-strength-blind deblurring comes from the good generalizability of CV-CNN when handling unseen noise distributions.

Remark 2. The method with the closest performance to ours is VEM. The visual results from VEM contain more details but also more artifacts. In comparison, our results have less artifacts. The main reason for such a visual difference comes from the different implementations on the inversion process and different schemes for handing varying noise level. Our network adopts Gabor filters for regularizing the inversion process, while VEM uses RV wavelet filters. Since Gabor filters have better orientation selectivity which lead to better treatment on oriented edges [9], our network produces less artifacts than VEM. The reason why VEM keeps more details is due to its under-estimated noise level by its noise level estimator [32]. As a result, the denoising in VEM for removing artifacts is done less aggressively than that in our network, leading to more details but also more artifacts. Roughly, our method prefers artifact reduction while VEM prefers detail preserving. In terms of quantitative metrics, our results are overall better than that from VEM.

D. Test on Deblurring in The Presence of Poisson Noise

We also evaluate the effectiveness of our model trained on AWGN in handling unseen noise types (*i.e.* non-Gaussian noise). With this purpose, the Poisson noise instead of AWGN is added to the blurry images in the test datasets. Two kinds of implementations are used for simulating Poisson noise: (a) The common sampling-based scheme with noise strength characterized by the peak values. The peak value is set to $1, 2, 4$ respectively; (b) The scheme from [49], that is, $\boldsymbol{Y}^2 = |\boldsymbol{Z}|^2 + \boldsymbol{N}$ with $\boldsymbol{N} \sim \mathcal{N}(0, \alpha^2 \text{Diag}(|\boldsymbol{Z}|^2))$, where $\boldsymbol{Z} = \boldsymbol{X} * \boldsymbol{K}$ and $\text{Diag}(|\boldsymbol{Z}|^2)$ is a diagonal matrix with diagonal elements $|\boldsymbol{Z}|^2$. The scalar α controls the noise strength and is set to $2.55, 12.75, 25.5$ respectively. The methods for comparison include the ones specifically designed for deblurring images with Poisson noise: RW2L [50] and VST-BM3D [51], as well as three deep learning methods: FDN [2], DMSP [5] and VEM [31]. Instead of being trained on Poisson noise, all the compared deep models are directly called from those trained

TABLE V: Average PSNR(dB) of deblurred images in image deblurring in the presence of Poisson noise with different strengths. Best results are marked in **blue** and the second best ones are marked in green.

Dataset	Peaks	RWL2	VSTBM3D	FDN	DMSP	VEM	Ours*	α	RWL2	VSTBM3D	FDN	DMSP	VEM	Ours*
Sun *et al.*'s	1	27.23	27.33	27.78	27.63	27.91	**28.03**	1%	27.57	27.61	33.15	33.06	**34.81**	34.45
	2	27.43	27.57	28.91	29.05	29.00	**29.48**	5%	27.34	27.51	28.93	29.03	29.88	**29.97**
	4	27.55	27.60	29.89	29.77	30.01	**30.49**	10%	26.51	24.02	26.72	26.68	**27.84**	27.39
Levin *et al.*'s	1	27.60	28.12	27.94	29.11	27.08	**29.52**	1%	28.23	29.57	31.67	34.45	36.22	**36.95**
	2	27.93	29.15	28.80	30.53	28.84	**30.97**	5%	27.90	28.17	28.69	30.98	31.07	**31.36**
	4	28.09	29.41	29.78	31.67	30.81	**32.50**	10%	26.52	21.31	26.62	**28.17**	28.03	27.48
Set18	1	26.69	26.65	27.32	27.25	27.32	**27.88**	1%	27.25	27.10	32.72	32.98	**33.79**	33.69
	2	27.00	27.00	28.15	28.32	28.44	**29.23**	5%	27.09	27.01	28.73	29.10	29.48	**29.80**
	4	27.13	27.07	29.60	29.87	29.54	**30.22**	10%	26.39	23.97	26.88	26.86	**27.37**	27.10

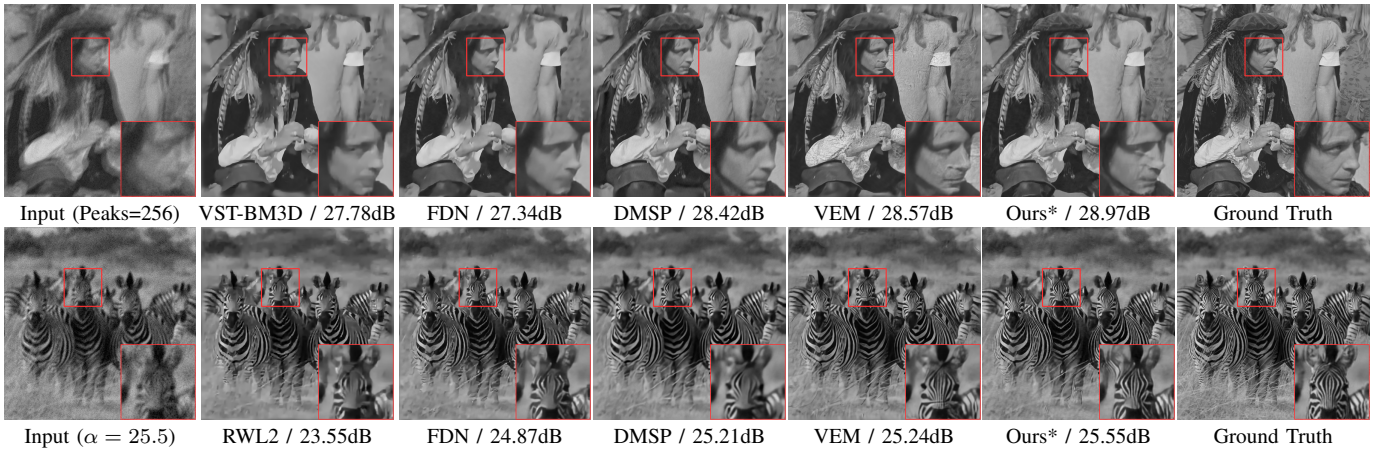


Fig. 8: Visual comparison of deblurring results on images 'Man' & 'Zebra' in the presence of Poisson noise.

in the experiment about AWGN with varying strength. This can indeed test the generalization in handling unseen noise.

The PSNR results are listed in Table V for comparison. It can be seen that our approach consistently outperforms most compared methods in all cases of both noise simulation schemes and it outperforms VEM in 13 out of 18 noise strengths. This clearly indicates the capability of our approach for handling non-Gaussian noise. See also Fig. 8 for the deblurred images generated by different approaches on two degraded images including 'Man' and 'Zebra' in the experiment. Regarding 'Man', the man's face in our deblurred image is clearer than that of other compared approaches. Regarding 'Zebra', the stripes on the zebras in our deblurred image contain more details in comparison to other approaches.

E. Test on Deblurring With Kernel Uncertainty

For evaluating the robustness of the proposed approach to kernel error, we deblur blurry images using the estimated blur kernels from three existing kernel estimation approaches, including Perrone and Favaro [52], Pan *et al.* [53], and Yang and Ji [36]. We select EPLL [24], IRCNN [29], FCNN [1] VEM [31] for comparison. The noise in test is configured as AWGN with strength $\sigma = 1\%$. The results are listed in Table VI. It can be seen that our approach performs better than all other compared methods. See Fig. 9 for some visual results on a blurry image 'Lena'. It can be found that the hair of Lena in our deblurred image is more similar to ground truth, compared to other approaches. Notice that the hair restored by VEM is over-sharpened; see Remark 2 for a discussion.

TABLE VI: Average PSNR(dB)/SSIM($\times 10^{-4}$) of deblurring results on Set18 with estimated kernels estimated by three approaches, in the presence of AWGN with $\sigma = 1\%$. Best results are marked in **blue**.

Method	Perrone and Favaro	Pan *et al.*	Yang and Ji
EPLL	26.07/7812	26.24/7875	26.71/8027
IRCNN	27.00/7906	25.90/7819	26.25/7984
FCNN	26.85/7881	26.79/7948	27.39/8057
VEM	27.02/7925	26.46/7949	26.94/8045
Ours	**27.38/7990**	**27.05/8003**	**27.66/8164**

F. Ablation Study and More Analysis

For further analyzing the proposed approaches and verifying the effectiveness of each of its main components, the following ablation studies and experiments are conducted.

1) CV-CNN vs. RV-CNN: To verify the benefit of using CV-CNN in our approach, we construct an RV counterpart (RV-CNN) of our CV-CNN and form a baseline by replacing our CV-CNN with the RV-CNN in our model for comparison. The RV-CNN is constructed as follows. By viewing the Gabor coefficient map (CV) as a two-channel RV feature map, we replace all the CV-Conv and CV-HS layers in our CV-CNN with

Fig. 9: Visual comparison of deblurring results on image 'Lena' in the presence of kernel error. The first column contains ground truth kernel (top) and estimated kernel from Yang and Ji (bottom).

their RV counterparts, and scale up the numbers of all channels so as to ensure the generated RV-CNN has the same model size as our CV-CNN. The baseline is tested using AWGN with varying strengths, under the same training and test protocols in the previous experiments. The quantitative comparison of using CV-CNN vs. using RV-CNN is given in Table VII. It can be seen that using CV-CNN performs consistently better than using RV-CNN on all noise strengths across all datasets. Such results have demonstrated the benefits from using the CV representation and CV-CNN-based processing in our approach. Also, even only using RV-CNN, the results are competitive to the previous compared approaches. This is attributed to the use of Gabor-domain prior and CA module for better inversion.

TABLE VII: Average PSNR(dB) of deblurred images of proposed approach using RV-CNN and CV-CNN respectively on image deblurring in the presence of AWGN with varying strengths. Best results are marked in **blue**.

Dataset	Sun *et al.*'s		Levin *et al.*'s		Set18	
CNN	RV	CV	RV	CV	RV	CV
$\sigma = 1\%$	32.40	**32.75**	34.29	**34.86**	32.05	**32.32**
$\sigma = 2\%$	30.18	**30.61**	31.73	**32.37**	30.15	**30.43**
$\sigma = 3\%$	29.16	**29.45**	30.28	**30.78**	29.10	**29.29**
$\sigma = 4\%$	28.24	**28.49**	28.98	**29.45**	28.06	**28.34**

2) CV-HS vs. CV-ReLU: To verity the effectiveness of h-swish over the often-seen ReLU in CV-CNN, we construct a baseline by replacing the CV-HS activation function with the CV-ReLU activation function [41] defined by applying ReLU on the real part and imaginary part respectively. The baseline's performance is evaluated on image deblurring in the presence of AWGN with varying strengths, using the same training and test protocols in the previous experiments. The quantitative comparison between the baseline and our approach is given in Table VIII. Overall, our approach with the CV-HS performs better than the baseline using the CV-ReLU. Such results have demonstrated the effectiveness of CV-HS over the CV-ReLU.

3) Gabor vs. wavelet vs. learned filters: To verity the benefits of using the Gabor-domain prior in the proposed approach, we construct two baselines by replacing the Gabor filters with the linear spine wavelet filters (RV) [54] and learned filters respectively. In the first baseline, the wavelet has the same number of high-pass filters as the Gabor transform we use. The CV-CNN is run by viewing its RV input from wavelet transform as the CV ones with imaginary parts set to zeros. In the second baseline, the replaced filers are also CV with the same number as our Gabor filters, and they are jointly initialized and learned with the NN model. The performance of the baselines is evaluated on the image deblurring in the presence of AWGN with varying strength, using the same protocols in the previous experiments.

TABLE VIII: Average PSNR(dB) of deblurred images using CV-ReLU and CV-HS activation functions respectively on image deblurring in the presence of AWGN with varying strengths. Best results are marked in **blue**.

Dataset	Sun *et al.*'s		Levin *et al.*'s		Set18	
Activation	ReLU	HS	ReLU	HS	ReLU	HS
$\sigma = 1\%$	32.51	**32.75**	34.43	**34.86**	32.13	**32.32**
$\sigma = 2\%$	30.25	**30.61**	31.90	**32.37**	30.22	**30.43**
$\sigma = 3\%$	29.20	**29.45**	30.49	**30.78**	29.14	**29.29**
$\sigma = 4\%$	28.33	**28.49**	29.07	**29.45**	28.16	**28.34**

The quantitative comparison between the baselines and our approach is given in Table IX. In all the tests, our approach with Gabor filters performs better than the baseline using wavelet filters, particularly when the noise strength is not high. This is mainly due to that the used Gabor filters have optimal orientation selectivity, *i.e.* the filters are oriented along different directions. Compared to RV wavelet filters which only fit horizontal and vertical oriented edges well, the Gabor filters can better fit image edges with different orientations. As a result, Gabor filters provide a more refined prior for the inversion process. Moreover, our approach with Gabor filters also performs better than the baseline using learned filters. Indeed, the learned filters even yield worse results than wavelet filters. The reason is probably that the learned filters are not guaranteed to becoming band-pass filters with good properties for the inversion process and therefore they may limit the model's generalizability.

4) CA module vs. predefined weights: To evaluate the performance gain brought by the CA module, we remove CA and replace its output with a predefined constant β after fine tuning. The quantitative comparison of using CA module vs. using predefined constant is given in Table X, which is on the image deblurring in the presence of AWGN with known strengths. We can see that the auxiliary parameter sequences $\{\beta_j^{(1)}, \cdots, \beta_j^{(T)}\}$ for all j in the inversion process are important for the deblurring, and our CA module works very well to estimate proper parameter sequences. See Fig. 10 for the visualization of the estimated auxiliary parameters of the trained model when being used to deblur one image *w.r.t.* different noise strengths. It can be observed that the

TABLE IX: Average PSNR(dB) of deblurred images using learned filters, wavelet filters and Gabor filters respectively on image deblurring in the presence of AWGN with varying strengths. Best results are marked in **blue**.

Dataset	σ	Wavelet	Learned	Gabor
Sun *et al.*'s	$\sigma = 1\%$	31.98	31.29	**32.75**
	$\sigma = 2\%$	30.38	29.92	**30.61**
	$\sigma = 3\%$	29.24	28.96	**29.45**
	$\sigma = 4\%$	28.33	28.19	**28.49**
Levin *et al.*'s	$\sigma = 1\%$	34.08	31.70	**34.86**
	$\sigma = 2\%$	31.02	30.38	**32.37**
	$\sigma = 3\%$	30.47	29.33	**30.78**
	$\sigma = 4\%$	29.14	28.39	**29.45**
Set18	$\sigma = 1\%$	31.59	30.56	**32.32**
	$\sigma = 2\%$	30.29	29.43	**30.43**
	$\sigma = 3\%$	29.18	28.55	**29.29**
	$\sigma = 4\%$	28.23	27.79	**28.34**

higher the noise strength is, the larger the magnitudes of the auxiliary parameters are. This is reasonable, since for higher noise strength, the deblurring should rely more on the denoised part while less on the noisy input.

TABLE X: Average PSNR(dB) of deblurred images of proposed approach using CA module and fixed weights respectively for setting $\{\beta_j^{(t)}\}_{j,t}$ on image deblurring in the presence of AWGN with known strengths. Best results are marked in **blue**. The fixed weights are tuned for performance, which are $0.01, 0.05, 0.10$ for $\sigma = 1\%, 3\%, 5\%$ respectively.

Dataset	Sun *et al.*'s		Levin *et al.*'s		Set18	
$\{\beta_j^{(t)}\}_{j,t}$	Fixed	CA	Fixed	CA	Fixed	CA
$\sigma = 1\%$	32.36	**33.10**	34.19	**35.44**	32.17	**32.79**
$\sigma = 3\%$	28.79	**29.54**	30.16	**30.85**	28.42	**29.40**
$\sigma = 5\%$	27.50	**28.07**	28.15	**28.80**	27.45	**27.86**

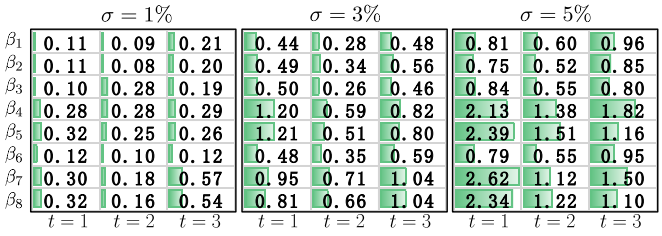


Fig. 10: Estimated auxiliary parameters $\{\beta_j^{(t)}\}_{j,t}$ on a blurry image *w.r.t.* different noise strengths. Longer green bars correspond to larger values.

5) Performance gain vs. number of blocks: By varying the number of blocks in our approach, we study its influence to the deblurring performance. Our test in conducted in image deblurring in the presence of AWGN with known strength. The PSNR results are shown in Fig. 11. It can be seen that the performance of our approach has a significant growth with the increase of number of blocks when $T \leq 3$. After $T = 3$, the performance saturates.

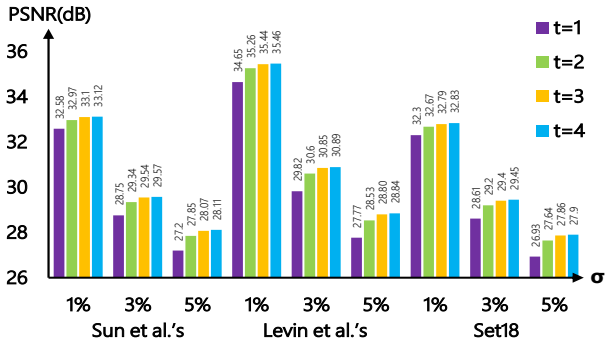


Fig. 11: PSNR(dB) of proposed approach using different numbers of blocks in image deblurring in the presence of AWGN with known strength on three test datasets.

6) Influence of number of Gabor filters: Recall that our results in the previous subsections are based on $L = 8$ Gabor filters. It is interesting to see how the number of Gabor filters affects the deblurring performance. Thus, we generate $L = 3, 15$ two-dimensional high-pass Gabor filters via the scheme of [9] and adopt them in our approach respectively with evaluation in image deblurring in the presence of AWGN with known strength. The PSNR versus number of Gabor filters is illustrated in Fig. 12. It can be seen that using only 3 Gabor filters leads to noticeable performance decrease, while further improvement over 8 Gabor filters is observed when using 15 Gabor filters. Note that the complexity of our approach does not increase much with more filters used, as only the input layer and output layer of the CV-CNN, as well as the CA module is enlarged accordingly.

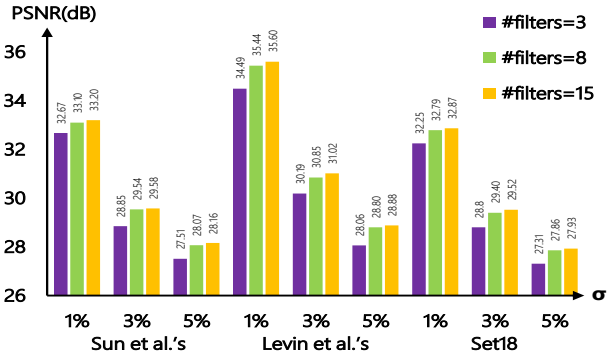


Fig. 12: PSNR(dB) of proposed approach using different numbers of filters in image deblurring in the presence of AWGN with known strength on three test datasets.

7) Study of the cases that challenge our method: In Fig. 13, we show a couple of less-successful cases from the results of our method. In the first case, our result has some details lost on the chair's texture. In the second case, our result still has certain blurring effect. The reason for both the cases is probably that the denoising of CV-CNNs is overdone. See also the supplementary file. We will further investigate how to handle these cases in our future study on non-blind deblurring.

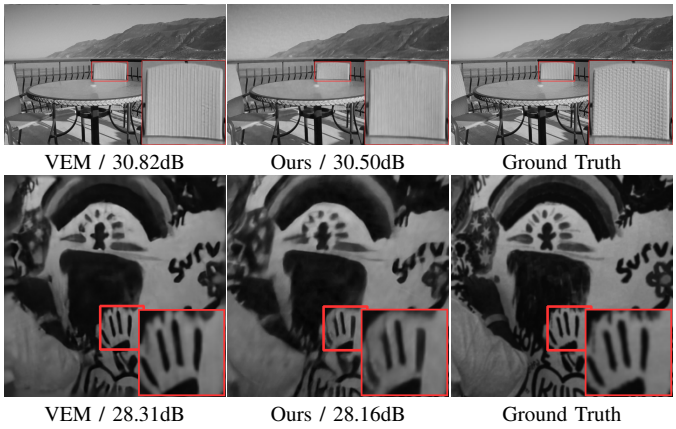


Fig. 13: Illustration of some less-successful results from the proposed method and the visual comparison to other methods.

V. Conclusion

In this paper, we proposed a non-blind image deblurring approach which is built upon the unfolded optimization of a deconvolution model equipped with a Gabor-domain denoising prior. In the unrolling, a CV-CNN is developed to implement the Gabor denoising prior. The proposed approach gains many benefits from the Gabor-domain representation such as strong orientation sensitivity and optimal joint space-frequency resolution. Furthermore, the developed CV-CNN can well exploit the CV representation from the Gabor transform for better handling the unseen noise distributions presented in the inversion process of deblurring. We tested the performance of the proposed approach in non-blind image deblurring with different noise settings. The effectiveness of proposed approach has been demonstrated by its superior performance over the SOTA ones. In future, we would like extend our approach to handle other image recovery tasks.

Appendix

Proof of Statement 1. The optimization model

$$\min_{\boldsymbol{X}} \|\boldsymbol{Y}-\boldsymbol{X}*\boldsymbol{K}\|_{\mathrm{F}}^2+\sum_{j=1}^{J}\beta_j^{(t)}\|\boldsymbol{A}_j^{(t)}-\boldsymbol{D}_j*\boldsymbol{X}\|_{\mathrm{F}}^2 \tag{21}$$

can be interpreted as an MAP estimator, which is calculated by (12) with $\beta_j^{(t)}:=(\sigma_j^{(t)})^{-2}\sigma^2$. Then we have

$$\begin{aligned}
&\mathbb{E}\{|\boldsymbol{R}_j^{(t)}[k]|^2\}=\mathbb{E}\{|(\boldsymbol{D}_j*\boldsymbol{Y})[k]-(\boldsymbol{K}*\boldsymbol{A}_j^{(t)})[k]|^2\}\\
&=\mathbb{E}\{|(\boldsymbol{D}_j*\boldsymbol{K}*\boldsymbol{X})[k]-(\boldsymbol{K}*\boldsymbol{A}_j^{(t)})[k]+(\boldsymbol{D}_j*\boldsymbol{N})[k]|^2\}\\
&=\mathbb{E}\{|(\boldsymbol{K}*\boldsymbol{N}_j^{(t)})[k]+(\boldsymbol{D}_j*\boldsymbol{N})[k]|^2\}\\
&=c_1\sigma^2+c_{2,j}(\sigma_j^{(t)})^2,
\end{aligned} \tag{22}$$

where $c_1, c_{2,j}>0$ are two constants determined by $\boldsymbol{K}, \boldsymbol{D}_j$ respectively. Therefore,

$$\frac{1}{N}\|\boldsymbol{R}_j^{(t)}\|_{\mathrm{F}}^2\approx\mathbb{E}\{|\boldsymbol{R}_j^{(t)}[k]|^2\}=c_1\sigma^2+c_{2,j}(\sigma_j^{(t)})^2 \tag{23}$$

$$=\sigma^2(c_1+c_{2,j}/\beta_j^{(t)}), \tag{24}$$

which completes the proof.

Back-propagation in CV-CNNs. The back-propagation about the CV convolution kernels is similar to that of their RV counterparts, except that the related operations are defined on complex numbers. Specifically, let $\boldsymbol{K}, \boldsymbol{F}$ denote a CV kernel and an input CV feature map respectively. Let $\boldsymbol{B}=\boldsymbol{F}*\boldsymbol{K}$ and $f(\boldsymbol{B})$ is a scalar function on $\boldsymbol{B}$. This sufficiently covers the calculation of the gradients encountered in the training of CV-CNNs. By the chain rule in complex analysis, we have $\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{K}}=\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{B}}\frac{\partial \boldsymbol{B}}{\partial \boldsymbol{K}}=\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{B}}*\boldsymbol{F}$. Since $\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{B}}$ and $\boldsymbol{F}$ are both CV. Thus, $\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{K}}$ is also CV with the form: $\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{K}}=\Re(\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{K}})+i\cdot\Im(\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{K}})$. Then we have $\Re(\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{K}})=\Re(\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{B}})*\Re(\boldsymbol{A})-\Im(\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{B}})*\Im(\boldsymbol{A})$ and $\Im(\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{K}})=\Re(\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{B}})*\Im(\boldsymbol{A})+\Im(\frac{\partial f(\boldsymbol{B})}{\partial \boldsymbol{B}})*\Re(\boldsymbol{A})$.

References

[1] J. Zhang, J. Pan, W. S. Lai, R. Lau, and M. H. Yang, "Learning fully convolutional networks for iterative non-blind deconvolution," in *Proc. CVPR*, 2017, pp. 6969–6977. 1, 4, 7, 10

[2] J. Kruse, C. Rother, and U. Schmidt, "Learning to push the limits of efficient fft-based image deconvolution," in *Proc. ICCV*, 2017, pp. 4586–4594. 1, 4, 7, 8, 9

[3] T. Meinhardt, M. Moeller, C. Hazirbas, and D. Cremers, "Learning proximal operators: using denoising networks for regularizing inverse imaging problems," in *Proc. ICCV*, 2017, pp. 1799–1808. 1, 2, 4

[4] M. Jin, S. Roth, and P. Favaro, "Noise-blind image deblurring," in *Proc. CVPR*, 2017, pp. 3834–3842. 1, 4, 7, 8

[5] S. A. Bigdeli, M. Jin, P. Favaro, and M. Zwicker, "Deep mean-shift priors for image restoration," in *Proc. NIPS*, 2017, pp. 763–772. 1, 4, 7, 8, 9

[6] S. Vasu, V. R. Maligireddy, and A. N. Rajagopalan, "Non-blind deblurring: Handling kernel uncertainty with cnns," in *Proc. CVPR*, 2018, pp. 3272–3281. 1, 2

[7] W. Dong, P. Wang, W. Yin, and G. Shi, "Denoising prior driven deep neural ketwork for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 10, pp. 2305–2318, 2019. 1, 2, 4, 7

[8] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3142–3155, 2017. 2

[9] H. Ji, Z. Shen, and Y. Zhao, "Digital gabor filters do generate mra-based wavelet tight frames," *Appl Comput. Harmonic Anal.*, vol. 47, no. 1, pp. 87–108, 2019. 2, 3, 4, 9, 12

[10] J. G. DAUGMAN, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Optical Society America A*, vol. 2, no. 7, pp. 1160–1169, 1985. 2

[11] A. V. Oppenheim and J. S. Lim, "The importance of phase in signals," *Proc. The IEEE*, vol. 69, no. 5, pp. 529–541, 1981. 2

[12] A. Hirose and S. Yoshida, "Generalization characteristics of complex-valued feedforward neural networks in relation to signal coherence," *IEEE Trans. Neural Networks Learning Syst.*, vol. 23, no. 4, pp. 541–551, 2012. 2

[13] I. Danihelka, G. Wayne, B. Uria, N. Kalchbrenner, and A. Graves, "Associative long short-term memory," *arXiv preprint arXiv:1602.03032*, 2016. 2

[14] T. Plotz and S. Roth, "Benchmarking denoising algorithms with real photographs," in *Proc. CVPR*, 2017, pp. 1586–1595. 2

[15] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, 2011. 2, 7

[16] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, V. Le, Quoc, and H. Adam, "Searching for mobilenetv3," in *Proc. ICCV*, 2019. 3, 6

[17] T. F. Chan and C.-K. Wong, "Total variation blind deconvolution," *IEEE Trans. Image Process.*, vol. 7, 1998. 3

[18] J.-F. Cai, H. Ji, C. Liu, and Z. Shen, "Blind motion deblurring from a single image using sparse approximation," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009, pp. 104–111. 3, 4

[19] D. Krishnan and R. Fergus, "Fast image deconvolution using hyper-laplacian priors," in *Proc. NIPS*, 2009, pp. 1033–1041. 3

[20] A. Danielyan, V. Katkovnik, and K. Egiazarian, "Bm3d frames and variational image deblurring," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1715–1728, 2012. 3, 7, 8
[21] W. Dong, L. Zhang, G. Shi, and X. Li, "Nonlocally centralized sparse representation for image restoration," *IEEE Trans. Image Process.*, vol. 22, no. 4, pp. 1620–1630, 2013. 3
[22] Y. Quan, H. Ji, and Z. Shen, "Data-driven multi-scale non-local wavelet frame construction and image recovery," *Journal of Scientific Computing*, vol. 63, pp. 307–329, 2015. 3
[23] S. Roth and M. J. Black, "Fields of experts: a framework for learning image priors," in *Proc. CVPR*, 2005, pp. 860–867. 3
[24] D. Zoran and Y. Weiss, "From learning models of natural image patches to whole image restoration," in *Proc. ICCV*, 2011, pp. 479–486. 3, 7, 10
[25] C. J. Schuler, H. C. Burger, S. Harmeling, and B. Scholkopf, "A machine learning approach for non-blind image deconvolution," in *Proc. CVPR*, 2013, pp. 1067–1074. 3
[26] L. Xu, J. S. J. Ren, C. Liu, and J. Jia, "Deep convolutional neural network for image deconvolution," in *Proc. NIPS*, 2014, pp. 1790–1798. 3
[27] W. Ren, J. Zhang, L. Ma, J. Pan, X. Cao, W. Zuo, W. Liu, and M.-H. Yang, "Deep non-blind deconvolution via generalized low-rank approximation," in *Proc. NIPS*, 2018, pp. 295–305. 3
[28] U. Schmidt and S. Roth, "Shrinkage fields for effective image restoration," in *Proc. CVPR*, 2014, pp. 2774–2781. 4, 7
[29] K. Zhang, W. Zuo, S. Gu, and L. Zhang, "Learning deep cnn denoiser prior for image restoration," in *Proc. CVPR*, vol. 2, 2017, pp. 2808–2817. 4, 7, 10
[30] D. Gong, Z. Zhang, Q. Shi, A. v. d. Hengel, C. Shen, and Y. Zhang, "Learning deep gradient descent optimization for image deconvolution," *IEEE transactions on neural networks and learning systems*, 2020. 4, 8
[31] Y. Nan, Y. Quan, and H. Ji, "Variational-em-based deep learning for noise-blind image deblurring," in *Proc. CVPR*, 2020, pp. 3623–3632. 4, 7, 8, 9, 10
[32] Y. Nan and H. Ji, "Deep learning for handling kernel/model uncertainty in image deconvolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2388–2397. 4, 9
[33] D. Gilton, G. Ongie, and R. Willett, "Neumann networks for linear inverse problems in imaging," *IEEE Trans. Comput. Imaging*, vol. 6, pp. 328–343, 2020. 4
[34] L. Chen, F. Fang, T. Wang, and G. Zhang, "Blind image deblurring with local maximum gradient prior," in *Proc. CVPR*, 2019, pp. 1742–1750. 4
[35] L. Pan, R. Hartley, M. Liu, and Y. Dai, "Phase-only image based kernel estimation for single image blind deblurring," in *Proc. CVPR*, 2019, pp. 6027–6036. 4
[36] L. Yang and H. Ji, "A variational em framework with adaptive edge selection for blind motion deblurring," in *Proc. CVPR*, 2019, pp. 10 159–10 168. 4, 10
[37] H. Gao, X. Tao, X. Shen, and J. Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *Proc. CVPR*, 2019, pp. 3843–3851. 4
[38] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proc. CVPR*, 2019, pp. 5971–5979. 4
[39] E. Oyallon and S. Mallat, "Deep roto-translation scattering for object classification," in *Proc. CVPR*, 2015, pp. 2865–2873. 4
[40] N. Guberman, "On complex valued convolutional neural networks," *arXiv preprint arXiv:1602.09046*, 2016. 4
[41] C. Trabelsi, O. Bilaniuk, D. Serdyuk, S. Subramanian, J. F. Santos, S. Mehri, N. Rostamzadeh, Y. Bengio, and C. J. Pal, "Deep complex networks," in *Proc. ICLR*, 2018. 4, 11
[42] M. Tygert, J. Bruna, S. Chintala, Y. LeCun, S. Piantino, and A. Szlam, "A mathematical motivation for complex-valued convolutional networks," *Neural Comput.*, vol. 28, no. 5, pp. 815–825, 2016. 4
[43] D. E. Worrall, S. J. Garbin, D. Turmukhambetov, and G. J. Brostow, "Harmonic networks: deep translation and rotation equivariance," in *Proc. CVPR*, 2017, pp. 7168–7177. 4
[44] I. Aizenberg, D. Paliy, J. Zurada, and J. Astola, "Blur identification by multilayer neural network based on multivalued neurons," *IEEE Trans. Neural Networks*, vol. 19, no. 5, pp. 883–898, 2008. 4
[45] U. Schmidt, J. Jancsary, S. Nowozin, S. Roth, and C. Rother, "Cascades of regression tree fields for image restoration," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 4, pp. 677–689, 2016. 7
[46] L. Sun, S. Cho, J. Wang, and J. Hays, "Edge-based blur kernel estimation using patch priors," in *Proc. ICCP*, 2013, pp. 1–8. 7
[47] A. Levin, Y. Weiss, F. Durand, and W. T. Freeman, "Understanding and evaluating blind deconvolution algorithms," in *Proc. CVPR*, 2009, pp. 1964–1971. 7
[48] ——, "Efficient marginal likelihood optimization in blind deconvolution," in *Proc. CVPR*, 2011, pp. 2657–2664. 7
[49] C. A. Metzler, P. Schniter, A. Veeraraghavan, and R. G. Baraniuk, "prdeep: robust phase retrieval with a flexible deep network," in *Proc. ICML*, 2018. 9
[50] J. Li, Z. Shen, R. Yin, and X. Zhang, "A reweighted l2 method for image restoration with poisson and mixed poisson-gaussian noise," *Inverse Probl. Imaging (Springfield)*, vol. 9, no. 3, pp. 875–894, 2015. 9
[51] L. Azzari and A. Foi, "Variance stabilization for noisy+ estimate combination in iterative poisson denoising," *IEEE Signal Processing Letters*, vol. 23, pp. 1086–1090, 2016. 9
[52] D. Perrone and P. Favaro, "Total variation blind deconvolution: the devil is in the details," in *Proc. CVPR*, 2014, pp. 2909–2916. 10
[53] J. Pan, D. Sun, H. Pfister, and M. H. Yang, "Blind image deblurring using dark channel prior," in *Proc. CVPR*, 2016, pp. 1628–1636. 10
[54] B. Dong and Z. Shen, *MRA based wavelet frames and applications*, ser. IAS Lecture Notes Series. Park City Mathematics Institute, 2010, vol. 19. 11

Yuhui Quan received the Ph.D. degree in Computer Science from South China University of Technology in 2013. He worked as the postdoctoral research fellow in Mathematics at National University of Singapore from 2013 to 2016. He is currently the associate professor in Computer Science at South China University of Technology. His research interests include image processing, sparse representation and deep learning.

PeiKang Lin is currently an M.sc candidate Computer Science at South China University of Technology. He is working on image recovery.

Yong Xu received the B.S., M.S., and Ph.D. degrees in mathematics from Nanjing University, Nanjing, China, in 1993, 1996, and 1999, respectively. He is currently a professor in Computer Science at South China University of Technology. His research interests include image processing and analysis.

Yuesong Nan is currently a Ph.D. candidate in Mathematics at National University of Singapore. His research interests include image processing, inverse problems, and sparse coding.

Hui Ji received the B.Sc. degree in Mathematics from Nanjing University in China, the M.Sc. degree in Mathematics from National University of Singapore and the Ph.D. degree in Computer Science from the University of Maryland, College Park. In 2006, the joined National University of Singapore as an assistant professor in Mathematics. Currently, he is an associate professor in mathematics at National University of Singapore. His research interests include computational harmonic analysis, optimization, computational vision, image processing and machine learning.