# CSC2529 Project Report: Weakly-supervising the Deep Priors for Blind Deconvolution

Zhixiang Chi
Student ID: 998884239
University of Toronto, Canada
zhixiang.chi@mail.utoronto.ca

**Abstract**—In this project, we are going to investigate the deep image prior method for blind image deblurring. The current method uses two randomly initialized networks to model the latent clean image and blur kernel. Both networks are optimized by reconstructing the blurry input image via the blurry image formation model for a fixed number of iterations. There are several drawbacks observed for the existing work, 1) lack of direct supervision on the latent clean image; 2) non-adaptive and sub-optimal early stopping policy to prevent the model from overfitting. Several potential improvements are proposed using a pre-trained deblurring network as a weak supervisor to provide direct supervision and an early stopping strategy. Code: Project code.

**Index Terms**—Blind Image Deconvolution, Deep Image Prior, Self-supervision

## 1 Introduction

Unavoidable factors such as camera shake, object motion, inaccurate focus, etc., always result in blurry images. Such blurry artifacts not only degenerate the image fidelity for human viewers but also degrade the downstream computer vision tasks (e.g., image classification [1], object segmentation [2], etc.). Removing such blurry artifacts and restoring the latent clean image is extremely challenging as the convolution operation is hardly invertible. Despite its highly ill-posed properties, it has been a popular research topic and extensive efforts have been devoted over the past decades [3], [4]. With the unprecedented increase in computational budget, deep models have achieved superior performance in many tasks [1]. Deep models also contribute to the image deblurring task and set the state-of-the-art results in various settings, including blind image deblurring [5].

Popular and effective methods involving deep models for image deblurring are end-to-end training paradigms [6]. The models are optimized iteratively on a large-scale dataset with input-output pairs (blurry and clean image pairs) [7]. The models are expected to learn deblurring correspondence and encode such knowledge in their weights to achieve generalization on unseen testing images. The main limitation of this research stream is that the same trained model is employed for all testing scenarios. As each image exhibits its own statistics, these generic models may perform poorly when the test data distribution does not match the training ones [8]. Furthermore, additional artifacts may be introduced with incomplete blurry removal.

A recent method that performs optimization on each of the blurry images is proposed (SelfDeblur) [9]. SelfDeblur is inspired by the concept of Deep Image Prior (DIP) [10] which is optimized to reproduce the degraded image. The main observation of DIP is the high noise impedance property of neural networks. Based on the observations and it is validated that the neural network is more capable to model natural images than noises. In other words, during the process of reconstructing the degraded image, the neural network will produce a clean natural image first. Following a similar trend, SelfDeblur utilizes two separate neural networks with random noises as input and produces the latent clean image and blur kernel simultaneously. The blurry image can be reconstructed via convolution operation using two outputs. A high-level illustration of SelfDeblur is shown in Fig. 1.

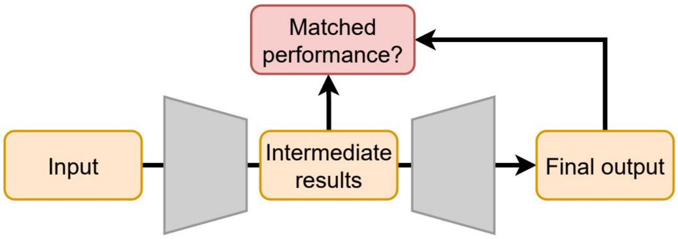

Fig. 1. High level illustration of SelfDeblur [9]. The latent clean image is only the intermediate result, while the network optimization is done at the final outputs. It is not guaranteed that optimizing the final output leads to an improvement in intermediate results.

Several observations on SelfDeblur can be observed. First, the optimization is based on the reconstruction loss by evaluating the output blurry image. However, the expected clean image is only the intermediate result. As shown in [11], there is a misalignment between the quality of intermediate and final results. And ignoring the supervision of the intermediate results will degrade the intermediate results but improve the final output in an end-to-end learning framework. In analogy, in the work of SelfDeblur, the intermediate supervising is missing, thus, the misalignment of performance between two outputs persists. On the other hand, both networks are randomly initialized and expected to fit only the specific data. Without learning abundant features from large-scale training may results in limited prior. Last but not least, a stopping criterion is difficult to be determined as the assessment of the expected output is impractical. Therefore, SeflDeblur uses a fixed number of iterations for all images which might be sub-optimal.

In this work, we investigate several potential improvements that could be helpful on top of SeflDeblur. Inspired by [12], instead of ignoring the middle supervision, a relaxed loss function with error tolerance is able to improve the final outputs. However, in our case, the clean image is not obtainable to guide the intermediate results. Follow the work in Semi-Supervised Few-Shot Classification [13], pseudo labels can be provided by a trained network to utilize unlabeled data. We utilize a deblurring network pre-trained on the large-scale dataset to generate a pseudo-clean image as a surrogate ground truth. An adaptive stopping method is also developed to balance the trade-off between training time and image quality.

To sum up, the contributions of this work are as follows:

- We propose to utilize a pre-trained deblurring network to weakly supervise the intermediate latent clean image.
- Early stopping methods are investigated to reduce the iteration numbers.
- Other techniques are also investigated for potential improvement.

## 2 Related Work

**Image Deblurring.** Deep neural networks (DNNs) have been widely employed for image deblurring. Some early works utilize DNNs as separate modules in the conventional optimization-based framework [14], [15], [16], [17]. For example, DNNs are used to only predict the complex Fourier coefficients of the blur kernel [17]; to estimate the motion information of blury images [16]. With the proposal of large-scale deblurring dataset, end-to-end traning methods are more favoured. Nah *et al.* [7] proposed a multi-scale architecture to progressively restore the latent sharp image. Since then, various networks were proposed under end-to-end manner and set the state-of-the-art.

**Weakly Supervised Learning.** Getting strong supervision is sometime impractical as some task may require intensive human labeling. Weakly supervised learning can be divided into different categories, such as incomplete supervision where a subset of the data is unlabeled [13]; inexact supervision, where the training data are given with only coarse-grained labels in a hierarchical label setting [18]; inaccurate supervision, where the obtained label cannot be fully trusted [19].

## 3 Proposed Method

### 3.1 Preliminaries

In this work, we mainly focus on the task of blind image deconvolution where the blur kernel is unknown. We start from the image degradation model:

$$\mathbf{y} = \mathbf{x} * \mathbf{H} + \mathbf{N}, \tag{1}$$

where $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{N}$ are the clean image, degraded image and noise, respectively. $\mathbf{H}$ and $*$ are the task-dependent degradation function and operator. For example, for blury image formation, $\mathbf{H}$ is the blur kernel and $*$ is the convolution [7]. while for image in-painting, $\mathbf{H}$ can be a mask image and $*$ is the element-wise multiplication [20].

To restore a degraded image, we aim to develop a restoration method denoted as function $f(\cdot)$. When given a degraded image $\mathbf{y}$, it generates a restored image $\hat{\mathbf{x}} = f(\mathbf{y})$ that is closer to the ground truth clean image $\mathbf{x}$. For deep learning-based method, especially the end-to-end ones, we denote the mapping function parameterized by $\theta$ as $f_\theta(\cdot)$. For learning-based methods, $f_\theta(\cdot)$ is learned from large-scale degraded/clean image pairs, so that the restoration prior or knowledge is encoded in the weights $\theta$ [21]. The models are normally iteratively optimized by a loss function which measures the distance between expected output and ground truth.

As an alternative research direction, Deep Image Prior (DIP) [10] finds that an untrained deep model is also capable to capture some of the low-level statistics of natural images. In such setting, the neural network is going to reconstruct the input degraded image and allows the network itself to learn natural clean image in the middle of training. Given each degraded image, the optimization of DIP is formulated as:

$$\theta^* = \arg\min_\theta E(f_\theta(z); \mathbf{y}), \ \mathbf{y}^* = f_{\theta^*}(z). \tag{2}$$

$E(\cdot)$ is the data fidelity term, which normally aims to minimize the distance between two inputs, therefore, making the network outputs closer to the given data. $z$ is the noise that follows Gaussian Distribution.

Through iterative optimization of Eq. 2. The network $f_\theta(z)$ aims to reproduce the degraded image. However, as discovered in [10], such parametrization offers high impedance to noise but low impedance to signals. In other words, the optimization process produces natural clean images first before fitting to the degraded image. Thus, an early stopping for such degradation reconstruction optimization leads to a image restoration solution. It has been shown the effectiveness on restoration tasks, such as image denoising, super-resolution, in-painting.

However, for the task of blind image deblurring, DIP is degenerated as it has the limitation on capturing the prior of blur kernels [9]. Thus, two separate generative networks $\mathcal{G}_x$ and $\mathcal{G}_k$ are proposed to replace $f_\theta$. $\mathcal{G}_x$ aims to capture the image prior, while $\mathcal{G}_k$ is expected to model the blur kernel $k$. The optimization is then becomes:

$$\min_{(\mathcal{G}_x, \mathcal{G}_k)} \left\| \mathcal{G}_k(z_k) \otimes \mathcal{G}_x(z_x) - \mathbf{y} \right\|^2, \tag{3}$$

where $z_k$ and $z_x$ are the input noise for each generator and $\otimes$ is the convolution operator. The goal of Eq. 3 is to generate a clean image and the corresponding blur kernel so that they can be transformed into the input blurry image via blur degradation model. Note, such supervision is self-supervised without any ground-truth clean image. We denote the loss term in Eq. 3 as reconstruction loss $\mathcal{L}_r$.

There are several drawbacks observed:

- The clean image is the intermediate output of the whole system, but the supervision is applied to the

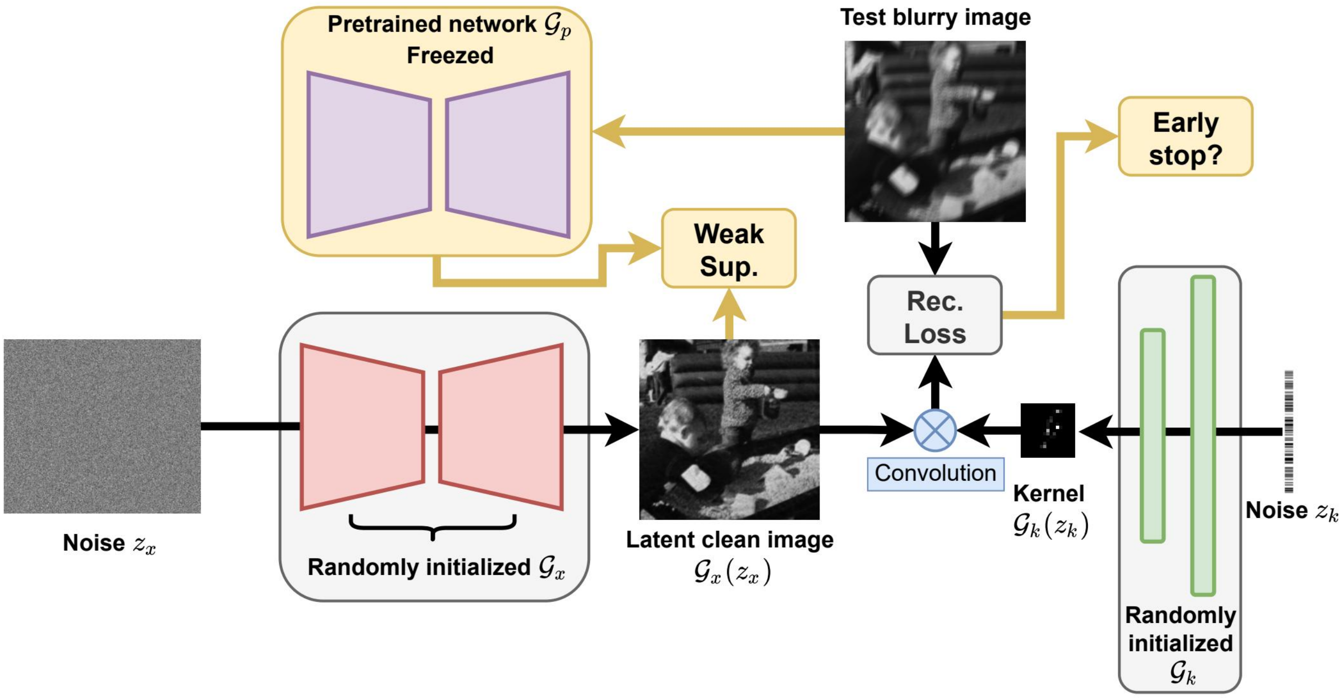

Fig. 2. Overview of the proposed method. The latent clean image and blur kernel are learned by reconstructing the test blurry image via blurry image formation model. Build up on this framework, we utilize a pretrained deblurring network to deblur the input image and treat its output as a weak supervision to supervise the intermediate clean image. Early stop strategy is also investigated.

final reconstructed blurry image. It leads to indirect optimization of the expected output, and the training objective and evaluate protocol do not match.

- Different input blurry images may require different number of iterations. The current method has a fixed iteration number for all images, which is sub-optimal. Although the intermediate output is constrained by the reconstruction loss, its convergence cannot be guaranteed.

### 3.2 Weakly Supervised Learning

In this project, we are going to explore if providing a supervision directly at the expected output is beneficial. Concretely, the direct supervision on the expected intermediate output might enhance the prior extracting process. However, the only available information is the degraded input image $\mathbf{y}$, which reduces the flexibility of such method.

To alleviate such problem, we follow the self-training method [13], which use the trained model to label the unlabeled data. Specifically, we employ a deep model $\mathcal{G}_p$ that is trained on a large-scale dataset for the deblurring task:

$$\bar{\mathbf{x}} = \mathcal{G}_p(\mathbf{y}). \tag{4}$$

We name $\bar{\mathbf{x}}$ as the surrogate ground truth which is the deblurred version of $\mathbf{y}$ using $\mathcal{G}_p$. Although $\bar{\mathbf{x}}$ highly depends on the architecture of $\mathcal{G}_p$ and the training dataset, and may contain defected results, it has learned through abundant data samples to extract features. $\bar{\mathbf{x}}$ can be used as a weak supervision to guide $\mathcal{G}_x(z_x)$. Therefore, we propose to add the following loss:

$$\mathcal{L}_{weak} = E(\mathcal{G}_x(z_x), \bar{\mathbf{x}}). \tag{5}$$

$E$ measures the distance between $\mathcal{G}_x(z_x)$ or $\bar{\mathbf{x}}$ which can be $L1/L2$ norm, or other similarity metrics.

Note, $\bar{\mathbf{x}}$ also represents the knowledge of the trained model $\mathcal{G}_p$. Thus, using Eq. 5 is analogy to the knowledge distillation technique [22]. Specifically, the knowledge of a well-trained model is transferred to a model with lower learning capability. In our case, we aim to transfer the deblurring features from a model learned with large-scale dataset.

### 3.3 Early stopping

The implementation of SelfDeblur has a fixed number of iterations for all images (e.g., 5000). It might be sub-optimal in terms of cost-performance tradeoffs. In addition, different blurry images exhibit various difficulty levels, thus, its needed to investigate a stopping strategy.

In this work, we investigate a simple stopping criteria which looks at the stability of the training process. We define a sequence of training data (e.g. training loss) $\{e_1, e_2, ..., e_n\}$, where $n$ is the $n^{th}$ iteration. The training is stopped if:

$$e_n = \min\{e_n, e_2, ..., e_{n+w}\}, \tag{6}$$

where $w$ is the size of sliding window. Eq. 6 means if the training data $e_n$ is not decreased further for the next $w$ iterations, the training will stop.

### 3.4 Further investigation

Eq. 3 measures how close the reconstructed image compared to the input blurry image. Intuitively, if they are close to each other, they should also be close when transform by a function (e.g. a trained network). So a further investigation can be conducted is to pass both images to a deblurring network and optimize to reduce the distance between outputs. We utilize such method as a regularization as:

$$\mathcal{L}_{reg} = \left\|\mathcal{G}_p(y) - \mathcal{G}_p(\hat{y})\right\|_2 , \tag{7}$$

where $\hat{y}$ is the reconstructed blurry image. Eq. 7 measures how the reconstructed image can be deblurred by a pretrained network. Its effectiveness is validated in the experiment section.

# 4 Experimental Results

**Dataset:** Due to the limitation on both time and computational resource, we follow SelfDeblur to evaluate our method on Levin *el al.* dataset [23]. It contains 4 clean images and 8 blur kernels. Each image is convolved with all the blur kernels to generate the 8 blurry images. Therefore, there are 32 blurry images with resolution of $256 \times 256$.

**Pre-trained deblurring network:** We utilize two network pre-trained on large-scale dataset: DWDN [24] which is trained on 5,000 images and DMPHN [25] which is trained on 2000 images in GoPro dataset [7].

**Implementation details:** We follow the same training and evaluate environment as SelfDeblur. Specifically, the input are initialized as Gaussian noise. Adam optimizer is used, and the initial learning rate is set to 0.01 with decay by a factor of 0.5 at 2,000, 3,000, 4,000 iterations. For evaluation, we use PSNR and SSIM to assess the quality of output clean image. Note, the blur kernel is unaware of the directions, the output image might be misaligned with the ground truth clean image. Therefore, we follow SelfDeblur to use a small search window (5) to do a match.

**Training objectives:** We combine the loss functions we discussed in the previous section with different weights as:

$$\mathcal{L} = \mathcal{L}_r + \alpha\mathcal{L}_{weak} + \beta\mathcal{L}_{reg}. \tag{8}$$

$\alpha$ and $\beta$ balance the contribution from different concepts. We test different weighting values in the next section. As in SelfDeblur, after $1500^{th}$ iteration, $\mathcal{L}_r$ is switched to $1 - SSIM$.

**Early stopping:** We the criteria for $e$ regarding the early stopping policy as MSE between $y$ and $\hat{y}$

# 5 Experiments

In this section, we first conduct some ablation studies to evaluate different components and hyper-parameters. We then compare with the state-of-the-art methods qualitatively and quantitatively.

## 5.1 Ablation studies

**Pre-trained networks and balancing weights:** We first check the performance of two pre-trained methods (DWDN [24] and DMPHN [25]) and different values of $\alpha$. We keep $\beta = 0$ to ignore $\mathcal{L}_{reg}$ for this experiments. Table 1 and Table 2 show the resulting PSNR and SSIM. Several conclusion can be drawn from these tables. 1) The deblurring quality of DWDN [24] and DMPHN [25] is quite low, that might due to the distribution shift between training and testing data. 2) Although they are under-performed, their results can still be helpful to improve the latent clean image of SelfDeblur. 3) Increasing the value of $\alpha$ is the same as putting more weights on $\mathcal{L}_{weak}$. It allows the network to generate results that is closer to DWDN [24] or DMPHN [25], thus results in lower PSNR/SSIM. 4) with proper $\alpha$ ($\alpha = 0.001$), the best performance is achieved. We will set $\alpha = 0.001$ for subsequent experiments.

TABLE 1
PSNR/SSIM values with weakly supervision of DMPHN [25] with various values of $\alpha$.

| | DMPHN | $\alpha = 1$ | $\alpha = 0.1$ | $\alpha = 0.01$ | $\alpha = 0.001$ |
|---|---|---|---|---|---|
| PSNR | 25.20 | 26.11 | 26.92 | 31.07 | 33.89 |
| SSIM | 0.771 | 0.791 | 0.803 | 0.879 | 0.935 |

TABLE 2
PSNR/SSIM values with weakly supervision of DWDN [24] with various values of $\alpha$.

| | DWDN | $\alpha = 1$ | $\alpha = 0.1$ | $\alpha = 0.01$ | $\alpha = 0.001$ |
|---|---|---|---|---|---|
| PSNR | 25.56 | 26.65 | 28.98 | 31.55 | 33.41 |
| SSIM | 0.730 | 0.753 | 0.812 | 0.881 | 0.924 |

**Effect from regularization:** We conduct experiments to show if $\mathcal{L}_{reg}$ can be helpful. For simplicity, we use DWDN. As reported in Table 3, adding $\mathcal{L}_{reg}$ hampers the intermediate latent clean image. It mighe due to the conflict optimization between $\mathcal{L}_{reg}$ and $\mathcal{L}_{weak}$.

**Early stopping:** We set the stopping criteria $e$ as the MSE between $y$ and $\hat{y}$. And test various sliding window sizes as $\{5, 10, 30, 50\}$. We report the PSNR/SSIM values when the stopping criteria is triggered and training is stopped. The number of training iteration is also recorded. Table 4 shows the results. As we can see, at different stages of the training process, the resulting deblurring performance is different. With $w = 30$ it quite matches the results of tranining using full 5,000 iterations. However, one thing to notice is that the learning rate decay happens at $2000^{th}, 3000^{th}, 4000^{th}$ iterations but the early stopping is set after $1500^{th}$ iteration before the first learning rate decay. Thus, setting early stopping after the learning rate decay might be more intuitive. We set this as one of the future works.

TABLE 3
PSNR/SSIM values with weakly supervision of DWDN [24] with various values of $\alpha$.

| | $\alpha = 0.001$ | $\beta = 0.001$ |
|---|---|---|
| PSNR | 33.41 | 33.19 |
| SSIM | 0.924 | 0.911 |

TABLE 4
PSNR/SSIM and number of iterations for different sliding window sizes.

| Stopping criteria: MSE | | | | |
|---|---|---|---|---|
| $w$ | 5 | 10 | 30 | 50 |
| PSNR | 32.63 | 33.24 | 33.29 | 32.78 |
| SSIM | 0.919 | 0.935 | 0.936 | 0.921 |
| # Iterations | 1510 | 1520 | 1573 | 1691 |

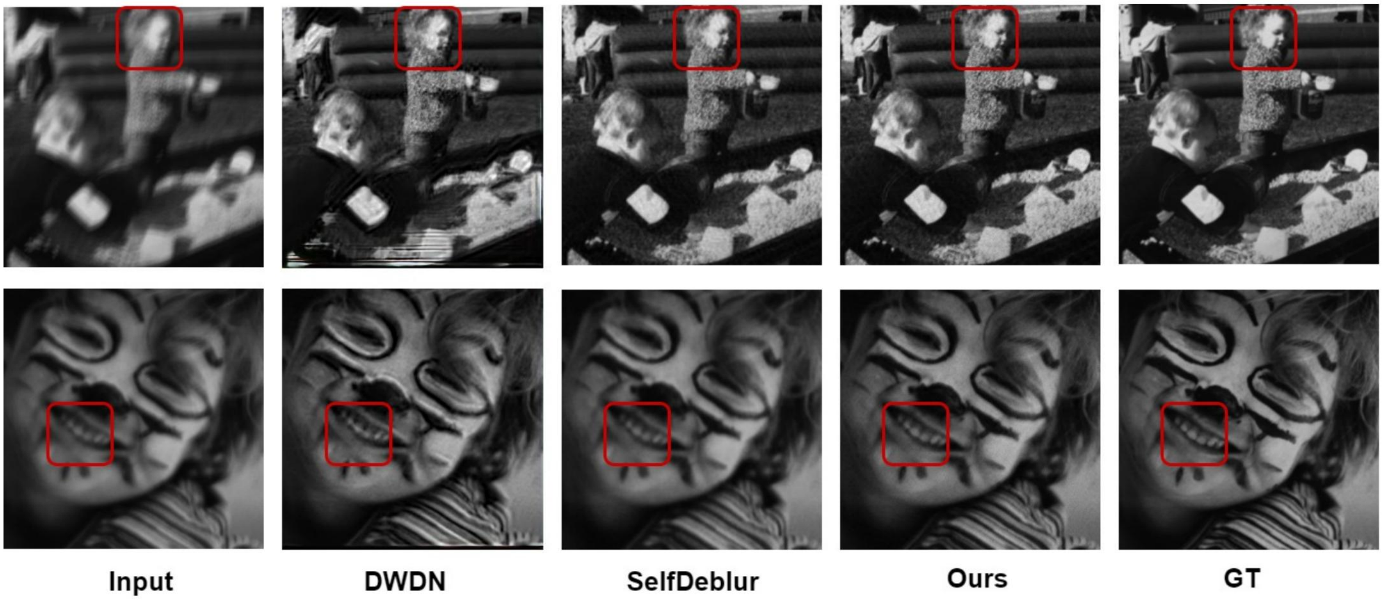

Fig. 3. Qualitative comparison.

TABLE 5
Comparison with the state-of-the-art methods.

| | SelfDeblur | DMPHN | DWDN | +DMPHN | +DWDN |
|---|---|---|---|---|---|
| PSNR | 33.07 | 25.19 | 25.56 | 33.89 | 33.41 |
| SSIM | 0.931 | 0.771 | 0.729 | 0.935 | 0.924 |

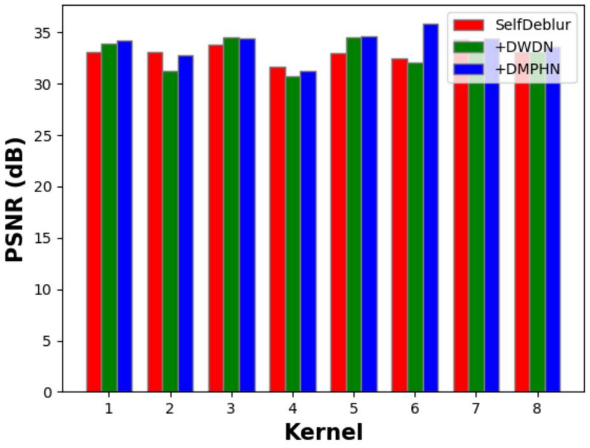

Fig. 4. Per kernel PSNR for different methods.

### 5.2 Comparison with the state-of-the-art

**Qauntitative comparison:** Table 5 shows the comparison between state-of-the-art methods. Compared to the baseline (SelfDeblur), the proposed wealy supervision is able to improve it.

**Per kernel/image analysis:** Fig. 4 and Fig. 5 report the average PSNR for every kernel or every image with different kernels. For most of the cases, adding weak supervision is able to improve the performance.

**Qualitative analysis:** Fig. 3 shows the qualitative analysis.

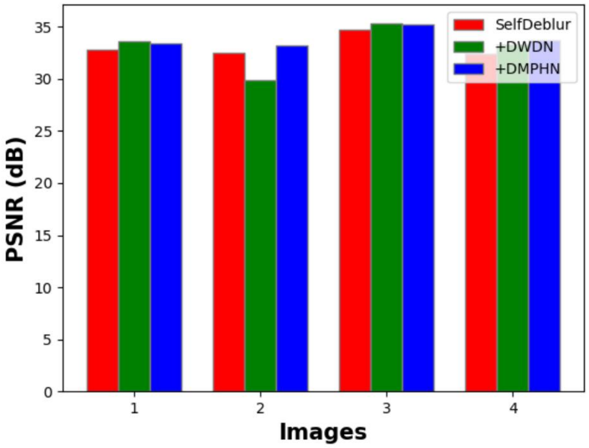

Fig. 5. Per image PSNR for different methods.

DWDN may results in some artifacts, but it also provides some clean areas for better supervision. Therefore, DWDN improves the final results.

## 6 Discussion and Future Works

From the experimental analysis, it seems that the core contribution of this work is effective. However, there are still quite a few things need to be investigated, such as further investigation for more comprehensive analysis on the early stopping methods. On the other hand, the defective performance of adding $\mathcal{L}_{reg}$ needs further justification, such as analyze the gradient before and after adding $\mathcal{L}_{reg}$.

## 7 Conclusion

In this work, we observed the main limitation of the deep prior method for blind image deconvolution due to the indirect supervision during training. We proposed a weak supervised learning to add relaxed supervision in the intermediate results. Experimental results show that our method is able to improve the baseline.

## References

[1] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[3] A. Gupta, N. Joshi, C. Lawrence Zitnick, M. Cohen, and B. Curless, "Single image deblurring using motion density functions," in *European conference on computer vision*. Springer, 2010, pp. 171–184.

[4] S. Harmeling, H. Michael, and B. Schölkopf, "Space-variant single-image blind deconvolution for removing camera shake," *Advances in Neural Information Processing Systems*, vol. 23, 2010.

[5] J. Rim, G. Kim, J. Kim, J. Lee, S. Lee, and S. Cho, "Realistic blur synthesis for learning image deblurring," *arXiv preprint arXiv:2202.08771*, 2022.

[6] C. Mou, Q. Wang, and J. Zhang, "Deep generalized unfolding networks for image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 399–17 410.

[7] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3883–3891.

[8] Z. Chi, Y. Wang, Y. Yu, and J. Tang, "Test-time fast adaptation for dynamic scene deblurring via meta-auxiliary learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 9137–9146.

[9] D. Ren, K. Zhang, Q. Wang, Q. Hu, and W. Zuo, "Neural blind deconvolution using deep priors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[10] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Deep image prior," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018.

[11] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision*, vol. 127, no. 8, pp. 1106–1125, 2019.

[12] Z. Chi, R. Mohammadi Nasiri, Z. Liu, J. Lu, J. Tang, and K. N. Plataniotis, "All at once: Temporally adaptive multi-frame interpolation with advanced motion modeling," in *European Conference on Computer Vision*. Springer, 2020, pp. 107–123.

[13] X. Li, Q. Sun, Y. Liu, Q. Zhou, S. Zheng, T.-S. Chua, and B. Schiele, "Learning to self-train for semi-supervised few-shot classification," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[14] C. J. Schuler, M. Hirsch, S. Harmeling, and B. Schölkopf, "Learning to deblur," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 7, pp. 1439–1451, 2015.

[15] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2015.

[16] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi, "From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[17] A. Chakrabarti, "A neural approach to blind motion deblurring," in *European Confererence on Computer Vison*, 2016.

[18] J. Foulds and E. Frank, "A review of multi-instance learning assumptions," *The knowledge engineering review*, vol. 25, no. 1, pp. 1–25, 2010.

[19] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.

[20] J. Xie, L. Xu, and E. Chen, "Image denoising and inpainting with deep neural networks," *Advances in neural information processing systems*, 2012.

[21] Y. Tai, J. Yang, and X. Liu, "Image super-resolution via deep recursive residual network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017.

[22] G. Hinton, O. Vinyals, J. Dean *et al.*, "Distilling the knowledge in a neural network," *arXiv preprint arXiv:1503.02531*, vol. 2, no. 7, 2015.

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

[24] J. Dong, S. Roth, and B. Schiele, "Deep wiener deconvolution: Wiener meets deep learning for image deblurring," *Advances in Neural Information Processing Systems*, vol. 33, pp. 1048–1059, 2020.

[25] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5978–5986.