MSSNet: Multi-Scale-Stage Network for Single Image Deblurring

Kiyeon Kim, Seungyong Lee, and Sunghyun Cho
POSTECH, Pohang, Korea
{kiyeon, leesy, s.cho}@postech.ac.kr

Abstract. Most traditional single image deblurring methods before deep learning adopt a coarse-to-fine scheme that estimates a sharp image at a coarse scale and progressively refines it at finer scales. While this scheme has also been adopted in several deep learning-based approaches, recently a number of single-scale approaches have been introduced showing superior performance to previous coarse-to-fine approaches in terms of quality and computation time. In this paper, we revisit the coarse-to-fine scheme and analyze the defects of previous coarse-to-fine approaches. Based on the analysis, we propose Multi-Scale-Stage Network (MSSNet), a novel deep learning-based approach to single image deblurring with our remedies to the defects. MSSNet adopts three remedies: stage configuration reflecting blur scales, an inter-scale information propagation scheme, and a pixel-shuffle-based multi-scale scheme. Our experiments show that our remedies can effectively resolve the defects of previous coarse-to-fine approaches and improve the deblurring performance.

Keywords: deblurring, restoration, neural network, CNN

1 Introduction

Single image deblurring aims to restore a sharp image from a blurry one caused by camera shake or object motion. As blur severely degrades the image quality and the performance of other tasks such as object detection, deblurring has been extensively studied for decades [7, 4, 20, 28, 23, 1, 22, 34, 42, 31, 41, 3].

Most classical approaches before deep learning estimate a blur kernel and a latent sharp image through alternating optimization [7, 29, 4, 19, 37, 20, 38, 33, 23, 5]. For computational efficiency and accuracy in estimating a blur kernel and latent image, a coarse-to-fine scheme has been widely adopted by classical approaches [29, 4, 37, 33, 38, 5]. The coarse-to-fine scheme estimates a small blur kernel and latent image at a coarse scale and uses them as an initial solution at the next scale. The small sizes of both images and blur at a coarse scale enable computationally efficient estimation. Also, the small blur size at a coarse scale enables more accurate estimation of a blur kernel and latent image. As a result, the coarse-to-fine scheme can quickly provide an accurate initial solution to the next scale, and improve both quality and efficiency of deblurring.
Thanks to the effectiveness of the coarse-to-fine scheme proven by traditional approaches, it has also been adopted to several deep learning-based single image deblurring approaches [22, 34, 8]. These approaches directly restore a latent image from a blurry image without blur kernel estimation. They adopt multi-scale neural network architectures that stack sub-networks for different scales to initially estimate a small-scale latent image and then a large-scale latent image using the small-scale latent image as a guidance. While they do not estimate blur kernels, they share the same motivation with classical approaches: as the image and blur sizes are small at a coarse scale, a deblurred image can be estimated more efficiently and accurately.

Nonetheless, several deep learning-based single-scale approaches have recently been introduced. Specifically, Zhang et al. [42] pointed out the expensive computation time of the previous multi-scale approaches and the relatively low contribution of lower scale results on the final deblurring quality, and proposed an alternative single-scale approach named DMPHN. Following Zhang et al., Suin et al. [31] and Zamir et al. [41] also proposed hierarchical multi-stage methods based on DMPHN. These approaches show superior performance to previous multi-scale approaches both in quality and computation time, making the traditional coarse-to-fine scheme seem obsolete.

In this paper, we address the following questions: The motivations of the coarse-to-fine scheme still look valid, but why do the coarse-to-fine approaches perform worse than recent single-scale approaches? What degrades their performance and how can we fix them? To this end, we revisit the coarse-to-fine scheme and analyze the defects of previous coarse-to-fine approaches that degrade their performance but have been overlooked so far. Based on the analysis, we propose Multi-Scale-Stage Network (MSSNet), a novel deep learning-based coarse-to-fine approach with our remedies to the defects. MSSNet consists of multiple scales and multiple stages at each scale. MSSNet adopts three remedies: stage configuration reflecting blur scales, an inter-scale information propagation scheme, and a pixel-shuffle-based multi-scale scheme. Each remedy is simple and straightforward, resulting in a simple architecture for MSSNet. Nonetheless, our experiments show that each remedies can effectively resolve the defects of previous coarse-to-fine approaches and improve the deblurring performance.

2 Related Work

Traditional single image deblurring methods [7, 13, 29, 4, 19, 37, 20, 38, 33, 23, 5] before deep learning assume blur models that describe how a blurred image is obtained using blur kernels. Unfortunately, they often fail due to their restrictive blur models and the ill-posedness of the problem. To improve deblurring quality, convolutional neural networks (CNNs) have recently been adopted [32, 9, 1, 28, 39]. For example, Schuler et al. [28] and Sun et al. [32] proposed CNNs that estimate blur kernels and a latent image based on traditional blur models. However, as they still rely on blur models, their performances are limited. To overcome such limitation, deep learning-based methods that directly restore sharp images
without blur kernels have been proposed [22, 34, 42, 31, 41, 3]. These methods can be broadly categorized into single- and multi-scale approaches with respect to their network architectures and training strategies.

**Single-Scale Approaches.** Recently, single-scale multi-stage architectures [42, 31, 41, 2] are gaining popularity. Zhang et al. [42] proposed DMPHN, a multi-stage network that stacks multiple encoder-decoder networks to gradually remove blur from an input image. Based on DMPHN, Suin et al. [31] proposed a dynamic filtering module to remove spatially varying blurs. Zamir et al. [41] proposed MPRNet, which progressively removes blur by giving supervision at each stage. Chen et al. [2] introduced half-instance normalization to the multi-stage architecture. Besides multi-stage architectures, Purohit et al. [25] proposed a deep single-stage architecture based on DenseNet [12]. However, these single-scale approaches do not use initial solutions estimated from coarse scales, so they are less efficient and accurate as will be shown in Sec. 5.

**Multi-Scale Approaches.** Multi-scale approaches are typically based on multi-scale neural network architectures that stack sub-networks in a hierarchical way, and training strategies that train each sub-network to produce deblurred images at different scales. DeepDeblur [22], which is the first end-to-end deep learning-based method, adopts a multi-scale neural network to directly restore a latent image from a blurry input in a coarse-to-fine manner. SRN [34] adopts a UNet-based architecture [27] for each scale. Gao et al. [8] also proposed a UNet-based multi-scale architecture with a different parameter sharing strategy. Hu et al. [10] proposed a pyramid neural architecture search network to automatically design an optimal multi-scale deblurring architecture. However, their performance is limited by the drawbacks of their network architectures as will be discussed in Sec. 3. Cho et al. [3] recently proposed MIMO-UNet, which adopts a single UNet [27] with multi-scale loss terms. This approach is, however, different from a conventional coarse-to-fine approach as it has a large encoder that processes an input image in a fine-to-coarse manner. Furthermore, as Sec. 5 will show, our MSSNet outperforms MIMO-UNet with much fewer parameters and computations.

**Transformer-based Approaches.** Image restoration approaches that adopt transformers [35, 6] have recently been proposed such as Uformer [36] and Restormer [40]. They achieve superior deblurring performance by addressing the shortcomings of CNNs, e.g., limited receptive fields. Nevertheless, in this work, we restrict the scope of our analysis to conventional CNN-based approaches for the ease of analysis, and moreover, empirically show that our CNN-based approach can achieve comparable performance with smaller computation times.

## 3 Shortcomings of Previous Coarse-to-Fine Approaches

This section analyzes defects of previous coarse-to-fine approaches, and discusses our ideas to remedy them. MSSNet with our remedies is presented in Sec. 4.

Fig. 1 illustrates the network architectures of previous coarse-to-fine approaches [22, 34, 8]. While SRN [34] adopts additional recurrent connections between consecutive scales to achieve additional performance gain, which is omitted...
in the figure, the previous coarse-to-fine approaches share essentially the same deblurring process. All the methods first build an image pyramid by downsampling an input blurred image. Then, from the coarsest scale, they estimate a deblurred image from a downsamplied blurred image, upsample the deblurred image, and feed it to the sub-network at the next scale. The sub-network at the next scale then estimates a deblurred image from the blurred image at the current scale using the deblurred image from the previous scale as a guidance. All the sub-networks at different scales share the same network architecture. In the following, we analyze the shortcomings of these approaches one by one and present our ideas to address them.

Network architectures disregarding blur scales. The first shortcoming of the previous approaches is their network architectures that disregard blur scales. Blur spreads a pixel value in a latent image over an area of the blur size. Thus, to restore the pixel value at a certain pixel, it is essential to use receptive fields larger than the blur size to aggregate information spread over the area. Consequently, larger blur sizes require larger receptive fields or deeper neural networks. Likewise, a coarse-to-fine approach needs deeper sub-networks for finer scales. While the previous coarse-to-fine approaches use a deblurred image from the previous scale to deblur the blurred image at the current scale [22, 34, 8], large receptive fields are still required for finer scales. In multi-scale approaches, a deblurred image from a lower scale lacks fine details as it is estimated from a downsamplied image, and such fine details must be restored from the blurred image at a finer scale. Restoring detail at one pixel inevitably needs to aggregate information spread over an area of the blur size regardless of a result from the previous scale. Thus, it is still more effective to have deeper sub-networks for finer scales as will be shown in our experiments.

Ineffective information propagation across scales. The previous coarse-to-fine approaches pass the pixel values of a deblurred result from a coarse scale to the next scale [22, 34, 8]. This causes a significant loss of abundant information encoded in the feature vectors at coarse scales, and eventually degrades the deblurring performance.

Information loss caused by downsampling. To produce multi-scale input blurred images, the previous approaches build an image pyramid by repeatedly downsampling an input image [22, 34, 8]. Unfortunately, downsampling causes significant information loss. Specifically, a downsampling operation reduces the
pixels not only in the input image but also in its deblurred result by 1/4, which severely limits the quality of a guidance to the next scale. To overcome this, in our approach, we present a multi-scale scheme based on the pixel-shuffle [30] operation that reduces the spatial resolution without information loss.

4 Multi-Scale-Stage Network

4.1 Network Architecture

In this section, we present MSSNet, which is designed based on the analysis in Sec. 3. Fig. 2 illustrates the architecture of MSSNet. MSSNet is composed of three scales following previous coarse-to-fine approaches [22, 34, 8]. We denote each scale by $S_1$, $S_2$, and $S_3$ from the coarsest to finest scales, respectively. MSSNet takes a single input blurred image $B$ and estimates a deblurred image $L$ in a coarse-to-fine manner. For effective restoration, MSSNet adopts the residual learning scheme, which has been widely adopted in various restoration tasks [14, 44, 18, 24, 41, 3], i.e., MSSNet predicts a residual image $R$, which is added to the input blurred image $B$ to obtain a deblurred output $L = B + R$. A detailed architecture of MSSNet can be found in the supplementary material.
MSSNet is specifically designed to reflect blur scales, to facilitate effective inter-scale information propagation, and to avoid information loss caused by downsampling. We describe each component in the following.

**Stage Configuration Reflecting Blur Scales.** To reflect blur scales, the sub-networks of MSSNet at finer scales are designed to have deeper architectures. Specifically, each scale of MSSNet has one, two and three stages from $S_1$ to $S_3$, respectively, where each stage consists of a single light-weight UNet module [27]. We denote each UNet module by $U_{ij}$ where $i$ and $j$ are scale and stage indices, respectively. The modules share the same network architecture but have different weights. Each module is trained to produce residual features that can be converted to a residual image and added to a blurred image to produce a deblurred image. More details on the training of MSSNet is explained in Sec. 4.2.

**Inter-Scale Information Propagation.** Whereas the existing multi-scale networks deliver an upsampled deblurred image from a coarse scale to the next scale as an initial solution, MSSNet delivers upsampled residual features to facilitate effective information propagation between scales. Specifically, at the end of a coarse scale, residual features are bilinearly upsampled and processed through a $1 \times 1$ conv layer. Then, the resulting features are concatenated to the features from a blurred image at the next scale and convolved with $3 \times 3$ filters to produce fused features. The fused features are then fed into the UNet modules to produce deblurred residual features at the current scale.

**Pixel-shuffle-Based Multi-Scale Scheme.** To avoid information loss caused by the downsampling operations when producing multi-scale input blurred images, we propose a pixel-shuffle [30] based multi-scale scheme. Specifically, from the input blurred image $B$ of size $W \times H$, we generate multi-scale input images as follows. For the finest scale $S_3$, the input blurred image $B$ is used. The input image downsampled to a different scale is denoted by $B_i$, where $i$ is a scale index, i.e., $B_3 = B$, and $B_2$ is a downsampled version of $B$ of size $W/2 \times H/2$.

For $S_2$, $B_2$ is not used, but $B_3$ is unshuffled to obtain four images of size $W/2 \times H/2$, which are stacked along the channel direction to generate an input tensor $X_2$ for $S_2$. As $B$ is an RGB image with three color channels, the size of $X_2$ is $W/2 \times H/2 \times 12$, so $X_2$ has the same spatial size as $B_2$ but still has the same amount of information as $B_3$. Then, $X_2$ is fed into the feature extractor module ($E_2$ in Fig. 2) and processed through the stages at $S_2$. Note that, despite $X_2$ having the same amount of information as $B_3$, the computation cost increase for $S_2$ is relatively small because we use features extracted from $X_2$ by the feature extractor module. Moreover, thanks to $X_2$ having richer information than $B_2$, the sub-network at $S_2$ can produce a more accurate result.

For the coarsest scale $S_1$, $B_3$ is first downsampled to obtain $B_2$, and then the same unshuffling process as for $S_2$ is applied to obtain an input tensor $X_1$ for $S_1$. Another possible choice is to directly unshuffle $B_3$ and obtain $X_1$ of $W/2 \times H/2 \times 48$, but we empirically found that this performs slightly worse. While the pixel-shuffle-based multi-scale architecture can already enhance deblurring quality when trained with conventional loss terms as will be shown in Sec. 5, we
propose a pixel-shuffle-based training strategy to minimize information loss and enhance deblurring quality in Sec. 4.2.

Cross-Stage and Cross-Scale Feature Fusion. MSSNet also adopts the cross-stage feature fusion scheme proposed in [41]. The cross-stage feature fusion scheme connects network modules in consecutive stages with additional connections (dotted pink lines in Fig. 2) to help information flow more effectively between stages. Fig. 3(a) describes the cross-stage feature fusion scheme. We refer the readers to [41] for more details on the cross-stage feature fusion scheme. In addition, we also introduce cross-scale feature fusion (dotted green lines in Fig. 2) to facilitate more effective information flow between consecutive scales. The cross-scale feature fusion scheme is described in Fig. 3(b).

4.2 Training and Loss Functions

During training, we guide each stage of MSSNet to produce a deblurred image. To this end, an auxiliary layer is attached to every stage to produce a deblurred image, except for the last one in $S_3$ that already has such layers. Specifically, for $S_3$, an auxiliary conv layer is attached at the end of $U_1^3$ and $U_2^3$ as shown in Fig. 4. The attached conv layers take features from the UNet modules and produce residual images $R_1^3$ and $R_2^3$. Each residual image is then added to $B_3$ to produce deblurred results $L_1^3$ and $L_2^3$. We also denote the final deblurred result $L$ by $L_3^3$.

For $S_1$ and $S_2$, we use a slightly different training strategy as the sub-networks at $S_1$ and $S_2$ take unshuffled images as input. Specifically, at the end of each stage at $S_1$ and $S_2$, a conv layer and a pixel-shuffle layer are attached as shown in Fig. 4. The attached layers at the stages at $S_1$ and $S_2$ produce residual images of sizes $W/2 \times H/2$ and $W \times H$, respectively. We denote the deblurred results from the auxiliary layers by $L_i^j$ where $i$ and $j$ are scale and stage indices, respectively.

We train MSSNet using two types of loss functions: a content loss $L_c$ and a frequency reconstruction loss $L_f$. The content loss $L_c$ is defined as:

$$L_c = \frac{1}{N_1} \| L_1 - L_{gt\downarrow} \|_1 + \sum_{j=1}^{2} \frac{1}{N_2} \| L_j^3 - L_{gt} \|_1 + \sum_{j=1}^{3} \frac{1}{N_3} \| L_j^3 - L_{gt} \|_1,$$  

(1)

where $L_{gt}$ is the ground-truth blurred image, and $L_{gt\downarrow}$ is a downsampled version of $L_{gt}$. $N_1$, $N_2$ and $N_3$ are normalization factors, which are set $N_1 = W/2 \times H/2 \times$
and \( N_2 = N_3 = W \times H \times 3 \). The frequency reconstruction loss was proposed in [3] to restore high-frequency details from blurred image by minimizing the difference between blurred image and ground-truth in the frequency domain. The frequency reconstruction loss is defined as:

\[
\mathcal{L}_f = \frac{1}{N_1} \| \mathcal{F}(L_1^i) - \mathcal{F}(L_{gt}^i) \|_1 + \sum_{j=1}^{2} \frac{1}{N_2} \| \mathcal{F}(L_j^i) - \mathcal{F}(L_{gt}) \|_1 + \sum_{j=1}^{3} \frac{1}{N_3} \| \mathcal{F}(L_j^i) - \mathcal{F}(L_{gt}) \|_1,
\]

where \( \mathcal{F} \) is Fourier transform. Our final loss is \( \mathcal{L}_{final} = \mathcal{L}_c + \lambda \mathcal{L}_f \) where \( \lambda = 0.1 \).

5 Experiments

Implementation Details. For evaluation, we trained MSSNet on the GoPro dataset [22] with 256 \( \times \) 256 patches randomly cropped and augmented with random horizontal and vertical flipping. We trained our model for 3,000 epochs (396,000 iterations) with batch size 16. We used the Adam optimizer [15] with cosine annealing [21]. We set the initial learning rate to \( 2 \times 10^{-4} \) and gradually decreased it to \( 1 \times 10^{-6} \). To evaluate the performance on real-world blurred images, we also use the RealBlur dataset [26]. To this end, we trained MSSNet using the GoPro [22], BSD-B [26], and RealBlur training sets following the RealBlur benchmark [26]. We trained the model for 100 epochs (397,400 iterations). The other training details are the same as above. The computation times of all models are measured on a PC with an NVIDIA GeForce RTX 3090 GPU.

5.1 Comparison with Previous Methods

We compare MSSNet with state-of-the-art methods. Table 1 shows a quantitative comparison on the GoPro test set [22]. All the methods in the table were trained with the GoPro training set. Among the compared methods, DeepDeblur [22], SRN [34] and PSS-NSC [8] are coarse-to-fine approaches. MIMO-UNet [3] is trained using multi-scale loss terms, but not a conventional coarse-to-fine approach as it is based on a single UNet architecture. MIMO-UNet+ is a variant of MIMO-UNet with more parameters, and MIMO-UNet++ is MIMO-UNet+ with self-ensemble. All the other methods are single-scale approaches. For a fair comparison, MSSNet is trained for the smallest number of iterations among the methods in the table. Refer to the supplementary material for more details.

As shown in Table 1, recent single-scale approaches tend to perform better than coarse-to-fine approaches except for MIMO-UNet [3] and its variants. On the other hand, MSSNet clearly outperforms all the other methods in PSNR and SSIM thanks to our remedies. Specifically, MSSNet performs better than MIMO-UNet+ by more than 0.5dB with fewer parameters and fewer computations. Compared to MIMO-UNet++, a self-ensemble version of MIMO-UNet+,
Table 1. Quantitative evaluation on the GoPro test dataset [22]. Blue: coarse-to-fine approaches. Purple: single-scale approaches. Green: transformer-based approaches. MIMO-UNet and its variants are based on a single UNet with multi-scale losses [3]. The PSNR and SSIM values of previous methods are from the original papers. The computation times of all the methods are measured in the same environment described earlier. The numbers of parameters, MACs, and computation times of RADN [25], SAPHN [31] and PyNAS [10] are unavailable as their source codes are not publicly released yet.

| Models                  | PSNR (dB) | SSIM  | Param (M) | MACs (G) | Time (s) |
|-------------------------|-----------|-------|-----------|----------|----------|
| DeepDeblur [22]         | 29.08     | 0.914 | 11.72     | 4729     | 1.290    |
| DMPHN [42]              | 30.25     | 0.935 | 7.23      | 1100     | 0.137    |
| SRN [34]                | 30.26     | 0.934 | 8.06      | 20134    | 0.736    |
| PyNAS [10]              | 30.62     | 0.941 | N/A       | N/A      | N/A      |
| PSS-NSC [8]             | 30.92     | 0.942 | 2.81      | 3255     | 0.316    |
| MT-RNN [24]             | 31.15     | 0.945 | 2.6       | 2315     | 0.323    |
| SDNet4 [42]             | 31.20     | 0.945 | 21.7      | 3301     | 0.414    |
| MIMO-UNet [3]           | 31.73     | 0.951 | 6.8       | 944      | 0.133    |
| RADN [25]               | 31.76     | 0.953 | N/A       | N/A      | N/A      |
| SAPHN [31]              | 32.02     | 0.953 | N/A       | N/A      | N/A      |
| MSSNet-small (Ours)     | 32.02     | 0.953 | 6.75      | 634      | 0.104    |
| MIMO-UNet+ [3]          | 32.45     | 0.957 | 16.1      | 2171     | 0.290    |
| MPRNet [41]             | 32.66     | 0.959 | 20.1      | 10927    | 1.023    |
| MIMO-UNet++ [3]         | 32.68     | 0.959 | 16.1      | 8683     | 1.169    |
| HINet [2]               | 32.90     | 0.960 | 88.67     | 2401     | 0.247    |
| Restormer [40]          | 32.92     | 0.959 | 26.13     | 1983     | 1.123    |
| MSSNet (Ours)           | 33.01     | 0.961 | 15.59     | 2159     | 0.255    |
| Uformer-B [36]          | 33.06     | 0.967 | 50.88     | 2236     | 1.105    |
| MSSNet-large (Ours)     | 33.39     | 0.964 | 28.15     | 4235     | 0.457    |

MSSNet still outperforms by 0.33dB with a 4× fewer computations. Also, compared to HINet [2], MSSNet achieves 0.11dB higher PSNR with 5.7× fewer parameters and fewer computations while slightly slower. The table also shows that our CNN-based models (MSSNet and MSSNet-large) outperform recent transformer-based approaches [36, 40] despite their smaller computation times.

We also include two variants of MSSNet: MSSNet-small and MSSNet-large, in this evaluation. Their detailed architectures are provided in the supplement. Compared to MIMO-UNet and SRN, which have larger model sizes, MSSNet-small achieves a higher PSNR and SSIM with smaller computation time. While SAPHN [31] achieves similar PSNR and SSIM values to those of MSSNet-small, ours performs much faster according to the computation time reported in their paper. Specifically, the reported computation time of SAPHN measured on a Titan Xp GPU is 0.77 sec., while that of MSSNet-small on the same GPU is 0.19 sec. MSSNet-large has about twice the parameters of MSSNet, which is still 3× fewer than HINet, and its computation time is more than twice shorter than those of MIMO-UNet++ and MPRNet. Nevertheless, it achieves 33.39 dB in PSNR, significantly exceeding all the other methods by a large margin. Fig. 5 shows a qualitative comparison on the GoPro dataset [22].
Table 2. Quantitative evaluation on RealBlur [26]. The models in the upper part of the table are trained on the GoPro dataset [22] and tested on the RealBlur test sets. The models in the lower part are trained and tested on each of the RealBlur-R and -J datasets. MIMO-UNet++ [3] provides only a model trained on the RealBlur-J dataset.

| Models                  | RealBlur-R |            | RealBlur-J |            |
|-------------------------|------------|------------|------------|------------|
|                         | PSNR       | SSIM       | PSNR       | SSIM       |
| Hu et al. [11]          | 33.67      | 0.916      | 26.41      | 0.803      |
| DeepDeblur [22]         | 32.51      | 0.841      | 27.87      | 0.827      |
| DeblurGAN [16]          | 33.79      | 0.903      | 27.97      | 0.834      |
| Pan et al. [23]         | 34.01      | 0.916      | 27.22      | 0.790      |
| Xu et al. [38]          | 34.46      | 0.937      | 27.14      | 0.830      |
| DeblurGAN-v2 [17]       | 35.26      | 0.944      | 28.70      | 0.866      |
| Zhang et al. [43]       | 35.48      | 0.947      | 27.80      | 0.847      |
| SRN [34]                | 35.66      | 0.947      | 28.56      | 0.867      |
| SDNet4 [42]             | 35.70      | 0.948      | 28.42      | 0.860      |
| MPRNet [41]             | 35.99      | 0.952      | 28.70      | 0.873      |
| MSSNet (Ours)           | 35.93      | 0.953      | 28.79      | 0.879      |
| DeblurGAN-v2 [17]       | 36.44      | 0.935      | 29.69      | 0.870      |
| SRN [34]                | 38.65      | 0.965      | 31.38      | 0.909      |
| MPRNet [41]             | 39.31      | 0.972      | 31.76      | 0.922      |
| MIMO-UNet++ [3]         | N/A        | N/A        | 32.05      | 0.921      |
| MSSNet (Ours)           | 39.76      | 0.972      | 32.10      | 0.928      |

Fig. 5. Qualitative evaluation on the GoPro dataset [22].

figure, our results show clearly restored sharp details while those of the others have remaining blur.

We also study the generalization ability and performance of MSSNet on real-world blurred images. Table 2 shows a quantitative evaluation on the RealBlur dataset [26], which consists of real-world blurred images. The methods in the upper section in the table are trained on the GoPro dataset [22], while those in the lower section are trained on the RealBlur-R and RealBlur-J datasets. Among the methods trained on the GoPro datasets, MSSNet achieves the highest SSIM for the RealBlur-R test set, and the highest PSNR and SSIM for the RealBlur-J test set. Also, among the methods trained on the RealBlur datasets, MSSNet
Table 3. Performance comparison among a single-scale architecture with four stages and our multi-scale architectures. MSSNet-Single is a single-scale architecture, while MSSNet-Multi and MSSNet-Multi-Small are multi-scale architectures. ‘Initial’ and ‘Final’ are the initial and final results of each architecture, respectively.

|               | MSSNet-Single | MSSNet-Multi-Small | MSSNet-Multi |
|---------------|---------------|--------------------|--------------|
| PSNR (Initial / Final) | 29.11 / 31.59 | 29.51 / 31.58 | 30.09 / 31.75 |
| Params (M) / MACs (G)    | 4.39 / 660.69  | 4.38 / 574.82  | 6.61 / 621.60 |

achieves the highest PSNR and SSIM. Fig. 6 shows a qualitative comparison on the ReaBlur-J dataset [26]. In all the examples, the results of the other methods show either remaining blur and incorrectly restored details. On the other hand, our results show better restored details. Additional qualitative examples are provided in the supplement.

5.2 Ablation Study and Analysis

We validate the effectiveness of the coarse-to-fine approach, and then analyze the effect of each technical component in our model. For analysis, we test several variants of MSSNet. All the models in the analysis are trained and tested on the GoPro training and test sets [22], respectively. For ease of analysis, all the variants of MSSNet in the ablation studies use neither the pixel-shuffling scheme nor the cross-stage and cross-scale feature fusion scheme if not otherwise noted.

Coarse-to-Fine vs Single-Scale. As discussed in Sec. 1, the coarse-to-fine approach can quickly estimate a high-quality initial solution using coarse scales. Specifically, compared to performing a single stage of deblurring at the original scale, performing multiple stages at a coarse scale can be computationally more efficient. Moreover, thanks to the small blur size at a coarse scale, it can estimate a more accurate result, which serves as an initial solution for a finer scale, which leads to a final deblurring result of higher quality.

To verify this, in Table 3, we compare three variants of MSSNet. MSSNet-Single is a single-scale model with four stages at the original scale. MSSNet-Multi and MSSNet-Multi-Small are multi-scale models with the same number of scales and stages as MSSNet. MSSNet-Single and MSSNet-Multi has the same number of parameters for each stage. On the other hand, MSSNet-Multi-Small
has fewer parameters for each stage at \( S_1 \) and \( S_2 \) so that its total number of parameters is similar to that of MSSNet-Single. Its architecture details are in the supplementary material. The multi-scale models use our pixel-shuffle-based approach, but none of the models use the cross-stage and cross-scale feature fusion schemes. While the multi-scale models have six stages in total, three of them are at coarser scales. As a result, both multi-scale model require smaller amounts of computation than MSSNet-Single as shown in the table.

In Table 3, ‘Initial’ and ‘Final’ indicates the initial and final results of the single-scale and multi-scale models. An initial solution of the single-scale model indicates a deblurring result of the first stage obtained using an auxiliary conv layer, while an initial solution of the multi-scale models indicates a deblurring result of the last stage at \( S_2 \) obtained using auxiliary conv and pixel-shuffle layers. We compare these as they serve as initial solutions for the last three stages. As shown in the table, despite its smaller computation cost, MSSNet-Multi produces higher-quality initial and final deblurring results. Also, although MSSNet-Multi-Small has a similar number of parameters and a much smaller computation cost, it still achieves a similar PSNR for the final result to that of MSSNet-Single. This proves the advantage of the coarse-to-fine approach against the single-scale approach.

### Stage Configuration Reflecting Blur Scales

Our first remedy that we adopt into our MSSNet is the stage configuration reflecting blur scales. To verify its effect as a common rule, we conduct two ablation studies using DeepDeblur [22] and MSSNet. Table 4 compares three variants of DeepDeblur [22]. D444 and D444L have four residual blocks at each scale, while D246 adopts our stage configuration scheme and has two, four and six residual blocks at \( S_1 \), \( S_2 \) and \( S_3 \), respectively. To match the computation cost of D246, we also prepare D444L, which has more channels at each residual block. The table shows that D246 outperforms both of the others in terms of PSNR and SSIM, especially, despite its fewer parameters and a smaller computation cost than those of D444L. Note that our first remedy is to simply use a better stage configuration reflecting blur scales. Nonetheless, this result shows that this simple remedy can clearly improve the deblurring performance.
In the second experiment, we compare two variants of MSSNet in Table 5. The variants have different numbers of stages at different scales as informed in the table, but share the same network architecture for the UNet modules. The deblurring performance is not only affected by the number of stages, but also by the computation amount and the number of parameters. To isolate the impact of the stage configuration on the deblurring performance from other factors, each of the tested models in this experiment shares the network weights across different stages. In Table 5, M123 has the same stage configuration as MSSNet. M552 has fewer stages at $S_3$ but more stages at coarse scales so it requires the same amount of computation. The table shows that M123 clearly outperforms M552, validating our argument on the stage configuration. Additional experiments with different settings, e.g., models without parameter sharing, are provided in the supplementary material.

**Inter-Scale Feature Propagation.** In the next ablation study, we verify the effect of our inter-scale feature propagation scheme. In this study, we also investigate how to fuse the solution from a coarse scale with the input to the finer scale. To this end, we compare three variants of MSSNet: MSS(Image,Concat), MSS(Feature,Skip) and MSS(Feature,Concat). MSS(Image,Concat) has auxiliary conv layers at the end of $S_1$ and $S_2$ to convert features to residual images. The residual images are added to the input blurred images of the corresponding sizes to produce deblurred results. The deblurred results are then upsampled and concatenated to the blurred images at the next scales. This model corresponds to the previous coarse-to-fine approaches that transfer pixel values from coarse to fine scales. MSS(Feature,Skip) transfers features from coarse to fine scales as done in MSSNet. However, features from coarse scales are not concatenated but added to the features of the blurred images at the next scales. As the subnetworks estimate residual features, adding them to the features of blurred images will produce initial deblurred features at finer scales. MSS(Feature,Concat) uses our inter-scale feature propagation scheme that concatenates features from coarse scales to the features of the blurred images at the next scales.

Table 6 compares the performance of the variants. The results confirm that using features instead of pixel values clearly improves the deblurring quality as features provide richer information. The table also shows that MSS(Feature,Concat) performs slightly better than MSS(Feature, Skip), although it requires slightly more parameters, validating our approach. Again, our remedy is simple and requires negligible amount of additional parameters and computation, but the improvement is clear.

**Table 6. Ablation study on the scale information propagation.**

| Model               | PSNR  | SSIM  | Params (M) | MACs (G) |
|---------------------|-------|-------|------------|----------|
| MSS(Image,Concat)   | 31.42 | 0.947 | 6.59       | 613.1    |
| MSS(Feature,Skip)   | 31.52 | 0.948 | 6.59       | 621.8    |
| MSS(Feature,Concat) | 31.54 | 0.949 | 6.61       | 621.1    |
Table 7. Ablation study on the pixel-shuffle-based multi-scale approach. PUS: pixel-unshuffle. PS: pixel-shuffle.

| PUS | PS | PSNR | SSIM | Params (M) | MACs (G) |
|-----|----|------|------|------------|----------|
| ✓   | ✓  | 31.54| 0.949| 6.61       | 621.1    |
| ✓   | ✓  | 31.67| 0.950| 6.61       | 621.6    |
| ✓   | ✓  | 31.75| 0.951| 6.61       | 621.6    |

**Pixel-Shuffle-Based Multi-Scale Scheme.** We then verify the effect of our pixel-shuffle-based multi-scale scheme. As discussed in Sec. 4, our pixel-shuffle-based multi-scale scheme consists of pixel-unshuffle layers that generate input tensors, and auxiliary pixel-shuffle layers used only in the training phase. To verify the effect of each component, we compare the performance of three variants of MSSNet: 1) without both pixel-unshuffle and shuffle layers, 2) with only the pixel-unshuffle layers, and 3) with both layers in Table 7. The first model takes downsampled images as input as done in previous coarse-to-fine approaches, and its sub-networks at $S_1$ and $S_2$ are trained to produce intermediate results of the corresponding sizes. The second model takes tensors generated by pixel-unshuffling layers as input, but its sub-networks in $S_1$ and $S_2$ are trained in the same manner as the first model. The third model corresponds to our approach.

As Table 7 shows, introducing the pixel-unshuffling and shuffling layers introduces a negligible increase in the number of parameters. On the other hand, the pixel-unshuffling layers clearly improve the deblurring quality as they provide richer information than downsampling. Also, the auxiliary pixel-shuffling layers further improve the deblurring quality as they enable higher-quality supervision.

6 Conclusion

In this work, we analyzed the defects of previous coarse-to-fine single image deblurring approaches. Based on our analysis, we proposed a novel coarse-to-fine approach with our remedies: stage configuration reflecting blur scales, inter-scale feature propagation, and pixel-shuffle-based multi-scale network architecture. The experiment results prove the effectiveness of our remedies.

**Limitations and Future Work.** While MSSNet achieves the state-of-the-art performance, it still fails on many real-world blurred images especially with large blur as other methods. Extending MSSNet for handling large blur can be an interesting future work. Our analysis is focused on conventional CNN-based architectures. Extending our work to transformer-based architectures would be an interesting future work. We also plan to examine the performance of MSSNet on other types of image degradation.

**Acknowledgements.** This work was supported by IITP grants funded by the Korea government (MSIT) (Artificial Intelligence Graduate School Program (POSTECH) No.2019-0-01906 and SW Star Lab No.2015-0-00174) and National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (NRF2018R1A5A1060031, No. 2020R1C1C1014863).
References

1. Chakrabarti, A.: A neural approach to blind motion deblurring. In: ECCV. pp. 221–235 (2016)
2. Chen, L., Lu, X., Zhang, J., Chu, X., Chen, C.: HiNet: Half instance normalization network for image restoration. In: CVPR workshops. pp. 182–192 (2021)
3. Cho, S.J., Ji, S.W., Hong, J.P., Jung, S.W., Ko, S.J.: Rethinking coarse-to-fine approach in single image deblurring. In: ICCV. pp. 4641–4650 (2021)
4. Cho, S., Lee, S.: Fast motion deblurring. ACM Trans. on Graph. 28(5), 1–8 (2009)
5. Cho, S., Lee, S.: Convergence analysis of map based blur kernel estimation. In: ICCV. pp. 4808–4816 (2017)
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16×16 words: Transformers for image recognition at scale. In: ICLR (2021)
7. Fergus, R., Singh, B., Hertzmann, A., Roweis, S.T., Freeman, W.T.: Removing camera shake from a single photograph. ACM Trans. on Graph. 25(3), 787–794 (2006)
8. Gao, H., Tao, X., Shen, X., Jia, J.: Dynamic scene deblurring with parameter selective sharing and nested skip connections. In: CVPR. pp. 3848–3856 (2019)
9. Hradiš, M., Kotera, J., Zemek, P., Sroubek, F.: Convolutional neural networks for direct text deblurring. In: BMVC. pp. 6.1–6.13 (2015)
10. Hu, X., Ren, W., Yu, K., Zhang, K., Cao, X., Liu, W., Menze, B.: Pyramid architecture search for real-time image deblurring. In: ICCV. pp. 4298–4307 (2021)
11. Hu, Z., Cho, S., Wang, J., Yang, M.H.: Deblurring low-light images with light streaks. In: CVPR. pp. 3382–3389 (2014)
12. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: CVPR. pp. 2261–2269 (2017)
13. Jia, J.: Single image motion deblurring using transparency. In: CVPR. pp. 1–8 (2007)
14. Kim, J., Lee, J.K., Lee, K.M.: Accurate image super-resolution using very deep convolutional networks. In: CVPR. pp. 1646–1654 (2016)
15. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: ICLR (2015)
16. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: DeblurGAN: Blind motion deblurring using conditional adversarial networks. In: CVPR. pp. 8183–8192 (2018)
17. Kupyn, O., Martyniuk, T., Wu, J., Wang, Z.: DeblurGAN-v2: Deblurring (orders-of-magnitude) faster and better. In: ICCV. pp. 8878–8887 (2019)
18. Lai, W.S., Huang, J.B., Ahuja, N., Yang, M.H.: Fast and accurate image super-resolution with deep laplacian pyramid networks. IEEE Trans. Pattern Analysis and Machine Intelligence 41(11), 2599–2613 (2018)
19. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Understanding and evaluating blind deconvolution algorithms. In: CVPR. pp. 1964–1971 (2009)
20. Levin, A., Weiss, Y., Durand, F., Freeman, W.T.: Efficient marginal likelihood optimization in blind deconvolution. In: CVPR. pp. 2657–2664 (2011)
21. Loshchilov, I., Hutter, F.: SGDR: Stochastic gradient descent with warm restarts. In: ICLR (2017)
22. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: CVPR (2017)
23. Pan, J., Sun, D., Pfister, H., Yang, M.H.: Blind image deblurring using dark channel prior. In: CVPR. pp. 1628–1636 (2016)
24. Park, D., Kang, D.U., Kim, J., Chun, S.Y.: Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In: ECCV. pp. 327–343 (2020)
25. Purohit, K., Rajagopalan, A.N.: Region-adaptive dense network for efficient motion deblurring. In: AAAI. vol. 34, pp. 11882–11889 (2020)
26. Rim, J., Lee, H., Won, J., Cho, S.: Real-world blur dataset for learning and benchmarking deblurring algorithms. In: ECCV. pp. 184–201 (2020)
27. Ronneberger, O., Fischer, P., Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. pp. 234–241 (2015)
28. Schuler, C.J., Hirsch, M., Harmeling, S., Schölkopf, B.: Learning to deblur. IEEE Trans. Pattern Analysis and Machine Intelligence 38(7), 1439–1451 (2015)
29. Shan, Q., Jia, J., Agarwala, A.: High-quality motion deblurring from a single image. ACM Trans. on Graph. 27(3), 1–10 (2008)
30. Shi, W., Caballero, J., Huszár, F., Totz, J., Aitken, A.P., Bishop, R., Rueckert, D., Wang, Z.: Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In: CVPR. pp. 1874–1883 (2016)
31. Suin, M., Purohit, K., Rajagopalan, A.: Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In: CVPR. pp. 3606–3615 (2020)
32. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: CVPR. pp. 769–777 (2015)
33. Sun, L., Cho, S., Wang, J., Hays, J.: Edge-based blur kernel estimation using patch priors. In: ICCP. pp. 1–8 (2013)
34. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: CVPR (2018)
35. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.U., Polosukhin, I.: Attention is all you need. In: NIPS. vol. 30 (2017)
36. Wang, Z., Cun, X., Bao, J., Zhou, W., Liu, J., Li, H.: Uformer: A general u-shaped transformer for image restoration. In: CVPR. pp. 17683–17693 (2022)
37. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: ECCV. pp. 157–170 (2010)
38. Xu, L., Zheng, S., Jia, J.: Unnatural l0 sparse representation for natural image deblurring. In: CVPR. pp. 1107–1114 (2013)
39. Xu, X., Pan, J., Zhang, Y.J., Yang, M.H.: Motion blur kernel estimation via deep learning. IEEE Trans. Image Processing 27(1), 194–205 (2018)
40. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H.: Restormer: Efficient transformer for high-resolution image restoration. In: CVPR. pp. 5728–5739 (2022)
41. Zamir, S.W., Arora, A., Khan, S., Hayat, M., Khan, F.S., Yang, M.H., Shao, L.: Multi-stage progressive image restoration. In: CVPR. pp. 14821–14831 (2021)
42. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: CVPR. pp. 5978–5986 (2019)
43. Zhang, J., Pan, J., Ren, J., Song, Y., Bao, L., Lau, R.W., Yang, M.H.: Dynamic scene deblurring using spatially variant recurrent neural networks. In: CVPR. pp. 2521–2529 (2018)
44. Zhang, K., Zuo, W., Chen, Y., Meng, D., Zhang, L.: Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. IEEE Trans. Image Processing 26(7), 3142–3155 (2017)