Efficient Multi-scale Network with Learnable Discrete Wavelet Transform for Blind Motion Deblurring

Xin Gao*[1,2] Tianheng Qiu*[2,3,4]
Xinyu Zhang†[2] Hanlin Bai[1] Kang Liu[1] Xuan Huang†[4] Hu Wei[4] Guoying Zhang†[1] Huaping Liu[2]
*Equal Contribution † Coresponding Author
[1]China University of Mining & Technology-Beijing [2]Tsinghua University
[3]University of Science and Technology of China
[4]Hefei Institutes of Physical Science,Chinese Academy of Sciences

Abstract

Coarse-to-fine schemes are widely used in traditional single-image motion deblur; however, in the context of deep learning, existing multi-scale algorithms not only require the use of complex modules for feature fusion of low-scale RGB images and deep semantics, but also manually generate low-resolution pairs of images that do not have sufficient confidence. In this work, we propose a multi-scale network based on single-input and multiple-outputs(SIMO) for motion deblurring. This simplifies the complexity of algorithms based on a coarse-to-fine scheme. To alleviate restoration defects impacting detail information brought about by using a multi-scale architecture, we combine the characteristics of real-world blurring trajectories with a learnable wavelet transform module to focus on the directional continuity and frequency features of the step-by-step transitions between blurred images to sharp images. In conclusion, we propose a multi-scale network with a learnable discrete wavelet transform (MLWNet), which exhibits state-of-the-art performance on multiple real-world deblurred datasets, in terms of both subjective and objective quality as well as computational efficiency. Our code is available on https://github.com/thqiu0419/MLWNet.

1. Introduction

Most current top-performing single-image blind deblurring algorithms are based on DNNs, which can be structurally categorized into single-scale [2, 10, 12, 28, 29, 42] and multi-scale approaches [3, 9, 13, 27, 37]. Compared to single-scale methods, multi-scale methods are built on the idea of moving from coarse-to-fine, and thus they decompose the challenging single-image blind deblurring problem into easier-to-solve sub-problems to restore the blurred image step-by-step.

Earlier DNNs exploiting the coarse-to-fine concept usually employ estimators of different scales to produce restored images gradually [20, 21, 27]. However, not only is the extracted semantic information not sufficiently representative, but the use of the same complexity at different scales may lead to redundant computations; More advanced multi-scale algorithms [3] use a global encoder-decoder, which significantly reduces the algorithm's runtime, multiple upsampling and downsampling leads to insufficient restoration of detailed information, and therefore such algorithms require the introduction of additional fusion modules to encode input images for fusion with deeper semantics. In addition, existing multi-scale algorithms use multiple-input and multiple-output (MIMO) architectures, which require the introduction of manually constructed downsampled image pairs, and usually employ simple interpolation algorithms to generate low-resolution images for the sake of efficiency, which is obviously not reliable.

We note that the gradual insertion of images used by existing coarse-to-fine algorithms is redundant. Referring to numerous algorithms [6, 8, 15, 17, 26, 31] that are widely used in downstream tasks, the network starts with a single input and also extracts features at different scales. These

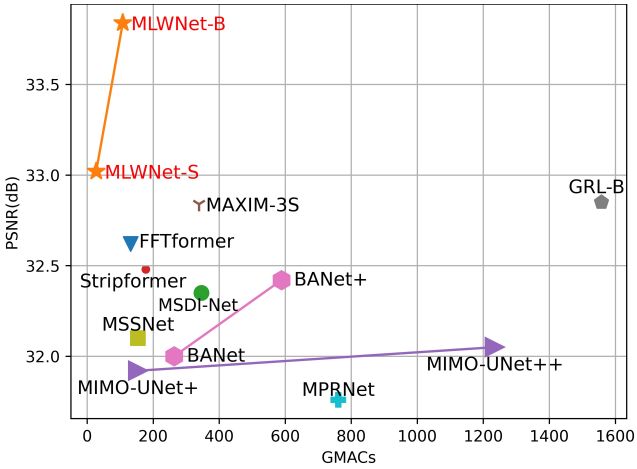


Figure 1. Performance comparison on the RealBlur-J [23] test dataset in terms of PSNR and GMACs. Our proposed MLWNet achieves superiority in comparison with other state-of-the-arts.

features can be aggregated by simple feature fusion to ultimately obtain competitive results. That is, a DNN itself possesses the ability to learn effective features at different scales. This inspired us to design a multi-scale architecture that retains an original image as input and sequentially restores images at each scale in output stage. This is not only more theoretically sound, but also eliminates the time required to generate pairs of images of different resolutions, as well as the huge complexity increase brought about by fusion modules between RGB domain and deep features.

The coarse-to-fine algorithm has another inherent defect. During progressive restoration, the solution of the upper-level problem takes the solution of the lower-level as initialization [9]. Although this reduces the difficulty of solving the upper-level problem, due to the smaller resolution of lower-level spatial, the features transmitted upwards are semantically precise but spatially ambiguous, thus the restoration ability of multi-scale network in spatial details is limited. Hence, there is an urgent need to improve the quality of restoration of high-frequency detail part. A simple approach is to introduce a frequency domain transform as an alternative to a spatial domain transform. This would provide the algorithm with a choice and direct its attention to different frequencies. In this paper, we consider using the discrete wavelet transform for the following reasons:

1. Many recent state-of-the-art deblurring algorithms [10, 19, 42] have introduced the discrete Fourier transform (DFT) as a frequency prior, which provides information that helps the algorithm to identify and select high- and low-frequency components that need to be preserved during restoration. Compared to DFT, the discrete wavelet transform (DWT) is better suited to deal with images containing more abrupt signals [5].
2. Realistic blur and synthetic blur have significant distributional differences [24]. In the real world, due to the short exposure time of a camera, realistic blur has a specific directionality [29], i.e., the blur trajectory is regionally continuous; conversely, synthetic blur has an unnatural and discontinuous trajectory. To fully utilize the potential deblurring guidance brought by this trajectory continuity, we use 2D-DWT to reveal blur directionality, distinguish changes in the blur signal along different directions, and provide the algorithm with a reliable basis for deblurring through adaptive learning.

To make 2D-DWT fit the data distribution and feature layer space more closely, we implemented 2D-DWT with an adaptive data distribution by using group convolution, transferring feature space from the spatial domain into the wavelet domain, generating sub-signals with different frequency features and different directional features, and then constructing wavelet losses for them in order to constrain them by self-supervision. Experimental results demonstrate that the proposed method has advanced performance in terms of accuracy and efficiency (Fig. 1).

In summary, our contributions are as follows:

- We propose a single-input and multiple-output(SIMO) multi-scale baseline for progressive image deblur, which reduces the complexity of existing multi-scale deblurring algorithms and improves the overall efficiency of image restoration networks.
- We construct a learnable discrete wavelet transform node(LWN) to provide reliable frequency and direction selection for the algorithm, which promotes the algorithm's restoration of the high-frequency components of edges and details.
- For the network to work better, reasonable multi-scale loss is proposed to guide pixel-by-pixel and scale-by-scale restoration. We also created reasonable self-supervised losses for the learnable wavelet transform to limit the learning direction of the wavelet kernel.
- We demonstrate the effectiveness of our algorithm under several motion blur datasets, especially real datasets, and obtain highly competitive results.

2. Related Work

Single-scale Deblurring Algorithm. Image deblurring have developed rapidly in recent years [2, 10, 11, 14, 39, 40]. To preserve richer image details, Kupyn et al. [11] viewed deblurring as a special case of image-to-image conversion, and for the first time utilized a GAN to recover images with richer details. To provide a high-quality and simple baseline, Chen et al. [2] proposed a nonlinear activation-free network that obtained excellent results. Zamir et al. [38] proposed a transformer with an encoder–decoder architecture that can be applied to higher-resolution images and utilizes cross-channel attention. Kong et al. [10] utilized a domain-based self-attention solver to reduce the transformer complexity and suppress artefacts. However, such single-scale algorithms estimate complex restoration problems directly and may require the design of restorers of higher complexity, which is suboptimal in terms of efficiency [9].

Multi-scale Deblurring Algorithm. Multi-scale motion deblur methods aim to achieve progressive recovery using a multi-stage or multi-sub-network approach. The multi-scale algorithm previously used in the field of image deblurring is based on MIMO [3, 13, 20, 21, 27]. Recently, Cho et al. [3] proposed MIMO-UNet, which employs a multi-output single decoder as well as a multi-input single encoder to simulate a multi-scale architecture consisting of stacked networks, thereby greatly simplifying complexity. Zamir et al. [37] designed a multi-stage progressive image recovery network to simplify the information flow between multiple sub-networks. As shown in Sec. 3.1, our proposed multi-scale approach further simplifies the structure of multi-scale networks and better balances the accuracy and complexity.

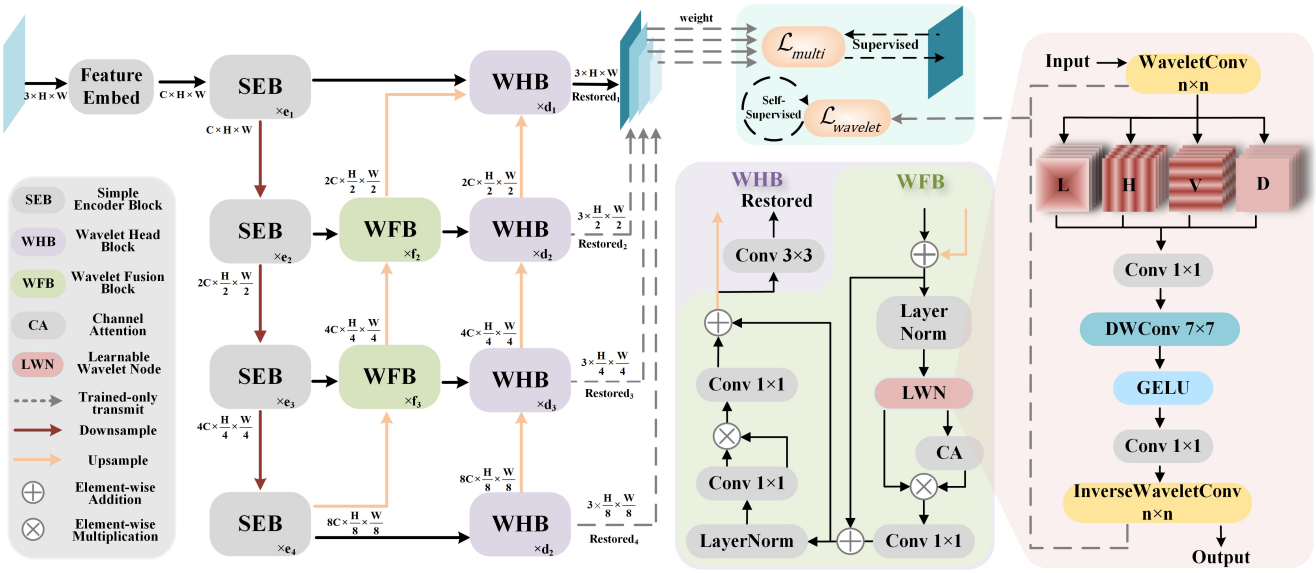


Figure 2. The overall architecture of the proposed MLWNet, the SEB is a simple module designed with reference [2], the WFB and WHB apply the LWN that implements the learnable 2D-DWT. In training phase, supervised learning is performed using $\mathcal{L}_{multi}$ and self-supervised restraint of the wavelet kernel is performed using $\mathcal{L}_{wavelet}$. In testing phase, only the highest scale restored images is output.

Application of Frequency in Deblurring. Frequency is an important property of images, in recent years frequency domain has been introduced into numerous DNNs to be solved for various tasks [10, 22, 25, 32, 35, 41]. Mao et al. [19] proposed DeepRFT, which introduced a residual module based on the Fourier transform into an advanced algorithm. Kong et al. [10] proposed FFTformer, which introduced the Fourier transform into a feed-forward network to effectively utilize different frequency information. Zou et al. [42] proposed SDWNet, which utilized frequency domain information as a complement to spatial domain information. Different from these methods, we introduce a learnable 2D-DWT capable of adapting to data distribution and feature space, more suitable not only for dealing with digital images rich in mutation signals, but also for dealing with real blurring.

3. Proposed Method

Our proposed MLWNet(Fig. 2) aims to explore an efficient multi-scale architectural approach to achieve high-quality blind deblurring of a single image. First, we designed a novel and highly scalable SIMO multi-scale baseline to address the performance bottleneck faced by partially multi-scale networks. It takes a single image as an input and gradually generates a series of sharp images from the bottom up. Then, we propose a learnable wavelet transform node (LWN) for image deblurring, and it enhances the proposed algorithm's ability to restore detail information.

3.1. Multi-scale Baseline

Our multi-scale network maintains an encoder-decoder architecture, but retains the original image with the highest resolution as input, and restores sharp images of various scales sequentially in the output stage. In this way, we eliminate the complexity of the fusion module when dealing RGB images of different scales and with deep semantics.

For ease of understanding, we will introduce here the Encoder phase, multi-scale semantic Fusion phase (hereafter referred to as Fusion phase), and Decoder phase sequentially. As shown in Fig. 2, the Encoder stage consists of several gray Simple Encoder Blocks(SEB), where information flows top-down for feature extraction; the Fusion stage consists of several green Wavelet Fusion Blocks(WFB), where information flows bottom-up for semantic fusion at different scales; and the Decoder stage consists of several purple Wavelet Head Blocks(WHB), and the bottom-up flow of information enables gradual restoration. Compared with WHB, SEB replaces LWN with a 1×1 conv for channel scaling, and a 3×3 depth-conv for feature extraction.

The Encoder stage is responsible for full feature extraction, downsampling feature maps once after each block, and then passing the feature maps to the Fusion stage or the Decoder stage as needed, whose outputs E^i_{out} to block of the $i-th$ layer can be denoted as:

$$E^i_{out} = \begin{cases} \phi_i\left(embed(x)\right), & i = i_{max} \\ \phi_i\left(E^{i-1}_{out}\right), & otherwise \end{cases} \tag{1}$$

where ϕ_i represent the block of corresponding layer of Encoder. The Fusion stage adapts and fuses information from different scales of semantics produced in Encoder to generate intermediate representations with deep semantics and shallow details. The Fusion stage's outputs F^i_{out} to the block of $i-th$ layer is expressed as in Eq. 2, where δ_i

represents the block of the corresponding layer of Fusion.

$$F_{out}^{i}=\begin{cases}\delta_i\left(E_{out}^{i}+E_{out}^{i-1}\right), & i=i_{min}\\ \delta_i\left(F_{out}^{i-1}+E_{out}^{i}\right), & otherwise\end{cases} \tag{2}$$

The Decoder stage utilizes the information passed by both the Encoder and Fusion stages to progressively upsample the feature maps and generating pre-output feature maps at each scale separately. Output D_{out}^{i} to the block of the $i-th$ layer is denoted as in Eq. 3. Here ξ_i represents the block of the corresponding layer of the Decoder.

$$D_{out}^{i}=\begin{cases}\xi_i\left(E_{out}^{i}\right), & i=i_{min}\\ \xi_i\left(D_{out}^{i-1}+F_{out}^{i}\right), & i>i_{min} \text{ and } i<i_{max}\\ \xi_i\left(D_{out}^{i-1}+F_{out}^{i-1}+E_{out}^{i}\right), & i=i_{max}\end{cases}$$

Each block in the Decoder is followed by a 3×3 convolution to generate a restored image at each scale. To improve inference efficiency, all but the highest-scale outputs are generated in the training phase to compute the multi-scale loss. We note that this design also have a similar role with auxiliary head [36] in facilitating the model's learning of the recovered uniform patterns.

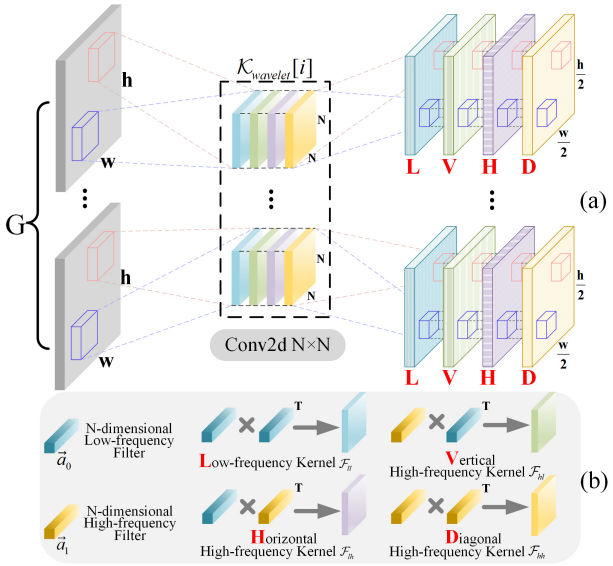


Figure 3. (a)The process of learnable 2D-wavelet convolution. (b)The construction process of the $N\times N$ 2D-wavelet kernel.

3.2. Learnable Discrete Wavelet Transform

To alleviate the defects of multi-scale architectures in the recovery of detail information, we design LWN and constructed the WFB and WHB based on it. Unlike the Fourier transform, the wavelet transform has adaptive time-frequency resolution, and performs excellently on digital images rich in mutation signal. Its use in conjunction with DNNs has gained increasing widespread attention [7, 16, 18, 34].

For ease of understanding, the 1D-DWT case is given here first. For the input discrete signal t, given the wavelet function $\psi_{j,k}(t)=2^{\frac{j}{2}}\psi(2^{j}t-k)$ for scaling factor j, time factor k, and scale function $\phi_{j,k}(t)=2^{\frac{j}{2}}\phi(2^{j}t-k)$, the decomposition of the input signal t at j_0 is given by:

$$f(t)=\sum_{j>j_0}\sum_{k}d_{j,k}\psi_{j,k}(t)+\sum_{k}c_{j_0,k}\phi_{j_0,k}(t) \tag{3}$$

In this process, $d_{j,k}=\langle f(t),\psi_{j,k}(t)\rangle$ represents the detail coefficients (i.e., high-frequency components), $c_{j_0,k}=\langle f(t),\phi_{j_0,k}(t)\rangle$ represents the approximation coefficients (i.e., low-frequency components) so that the input signal can be disassembled step by step into a rich wavelet domain signal. In order to use wavelet transform in DNN, we introduce the analysis vectors $\vec{a}_0[k]=\left\langle\frac{1}{\sqrt{2}}\phi(\frac{t}{2}),\phi(t-k)\right\rangle$, $\vec{a}_1[k]=\left\langle\frac{1}{\sqrt{2}}\psi(\frac{t}{2}),\phi(t-k)\right\rangle$ to represent the high and low frequency filters respectively, then:

$$\begin{aligned}c_{j+1,p}&=\sum_{k}\vec{a}_0[k-2p]c_{j,k}\\ d_{j+1,p}&=\sum_{k}\vec{a}_1[k-2p]c_{j,k}\end{aligned} \tag{4}$$

According to Eq. 4, the decomposition of the original signal over the wavelet basis can be viewed as a recursive convolution of the original signal with a specific filter using a step size of 2. Similarly, the inverse wavelet transform can be realized by transposed convolution of two synthetic vectors $\vec{s}_0$ and $\vec{s}_1$. We extend the above situation to 2D by efficiently extracting axial high-frequency information using 2D discrete wavelet transform (2D-DWT), and designing learnable methods to make it adaptive to the data distribution and feature layers. Fig. 3(b) presents the construction process of the forward wavelet convolution kernel in 2D, which can be expressed as:

$$\begin{aligned}&\mathcal{F}_{ll}=\vec{a_0}\times\vec{a_0}^{T},\mathcal{F}_{lh}=\vec{a_0}\times\vec{a_1}^{T}\\ &\mathcal{F}_{hl}=\vec{a_1}\times\vec{a_0}^{T},\mathcal{F}_{hh}=\vec{a_1}\times\vec{a_1}^{T}\\ &\mathcal{K}_{w}=\text{cat}\left(\mathcal{F}_{ll},\mathcal{F}_{lh},\mathcal{F}_{hl},\mathcal{F}_{hh}\right)\end{aligned} \tag{5}$$

We set $\vec{a_0}$ and $\vec{a_1}$ as learnable filters; $\mathcal{F}_{ll}$, $\mathcal{F}_{lh}$, $\mathcal{F}_{hl}$, and $\mathcal{F}_{hh}$ are low-frequency, horizontal high-frequency, vertical high-frequency, and diagonal high-frequency convolution operators obtained from vector outer products; The wavelet kernel $\mathcal{K}_w$ is spliced from above convolution operators.

Then we implement learnable wavelet transform as a group convolution like Fig. 3(a), given a set of input feature maps $\mathcal{X}_{in}\in(C,H,W)$, its projection in wavelet domain $\mathcal{X}_{out}\in\left(4C,\frac{H}{2},\frac{W}{2}\right)$ (low-frequency, horizontal high-frequency, vertical high-frequency, and diagonal high-frequency component) is generated through the wavelet

convolution. Similarly, the construction method of the inverse kernel and the forward propagation can be easily deduced, since they are inverse processes of each other.

Then, a natural question can then be posed: how to ensure the correctness of the wavelet kernel learning and ensure that it does not degrade into a general group convolution and cause performance degradation?

A common practice is to introduce the principle of perfect reconstruction [16, 34] to constrain adaptive wavelets, for a complex $z \in \mathbb{C}$, given a filter $\vec{x}$ that applies to the z-transform, its z-transform can be expressed as $X(z) = \sum_{n\in\mathbb{Z}} \vec{x}(n)z^{-n}$. Then $\vec{a}_0$,$\vec{a}_1$,$\vec{s}_0$,$\vec{s}_1$ can be obtained as their corresponding z-transforms A_0,A_1,S_0,S_1, which must then be satisfied if perfect reconstruction is desired:

$$A_0(-z)S_0(z) + A_1(-z)S_1(z) = 0 \tag{6}$$

$$A_0(z)S_0(z) + A_1(z)S_1(z) = 2 \tag{7}$$

Eq. 6 is known as alaising cancellation condition, which is used to the cancellation of the aliasing effects arising from the downsampling. We will give a detailed derivation of Eq. 6 and Eq. 7 in the Appendix.

After wavelet convolution, to achieve frequency restoration in the wavelet domain, the input $\mathcal{X}$ wavelet domain component was separated into a separate dim, we use depth-wise convolution with an expansion factor of r for wavelet domain feature extraction and transformation after the wavelet positive transform, and 1×1 convolution for channel expansion and scaling, and finally a learnable wavelet inverse transform is used to reduce the wavelet domain feature maps to the spatial domain for output, which constitutes the LWN. We follow the rules of [2] and design structurally similar Wavelet Head Block (WHB) and Wavelet Fusion Block (WFB), both with LWN as an important base module, but used for semantic aggregation and multiscale output at different scales, respectively; thus, WHB adds the recovery convolution branch after WFB.

3.3. Loss

Multi-scale Loss. Follow PSNR loss [2], we propose multi-scale loss function as the main loss of the algorithm for calculating the pixel difference between restored image and GT at each scale:

$$\mathcal{L}_{multi}(x, y) = \sum_{k=1}^{K} w_k \times \mathcal{L}_{psnr}\left(x_k, y_k\right) \tag{8}$$

where x, y denote the output and GT, respectively, k denotes the downsampling level, w_k represents the weights under the corresponding scale. Our design of w_k is based on the simple finding that we cannot produce absolutely accurate downsampled clear images, and that the accuracy gap is progressively enlarged with decreasing scales. To ensure that lower scale output layers do not negatively affect the final output, we empirically set the loss weight w_k to $\frac{1}{k}$ for the corresponding k-scale.

Wavelet Loss. Both Eq. 6 and Eq. 7 can be easily converted losses optimized toward a minimum of 0, here we use the mean square error. Then, based on the convolvable nature of the z transform, it is possible to construct convolutionally equivalent substitutions for polynomial multiplications for the perfect reconstruction condition:

$$\begin{aligned} L_{wavelet}(\theta_i) &= \left(\sum_{k}^{N-1} \left(\langle a_0, s_0\rangle_k + \langle a_1, s_1\rangle_k\right) - \hat{V}_{\lfloor \frac{N}{2} \rfloor}\right)^2 \\ &= \left(\sum_{k}^{N-1} \left(\left\langle (-1)^k a_0, s_0 \right\rangle_k + \left\langle (-1)^k a_1, s_1 \right\rangle_k\right)\right)^2 \end{aligned} \tag{9}$$

where k denotes the position of the filter when it performs the convolution, $\hat{V}$ is a vector with a center position value of two, and θ_i denotes the constructed filter

Overall Loss.Based on the multi-scale loss and wavelet loss, the loss function used in this study is:

$$\mathcal{L}_{total}(x, y) = \mathcal{L}_{wavelet}(\theta_i) + \mathcal{L}_{multi}(x, y) \tag{10}$$

4. Experimental Results

4.1. Datasets and Implementation Details

We evaluate our method on the realistic datasets RealBlur [23] and RSBlur [24], as well as the synthetic dataset GoPro [20], they are all composed of blurry-sharp image pairs. During training we use AdamW optimizer ($\alpha = 0.9$ and $\beta = 0.9$) for a total of 600K iterations. The initial value of the learning rate is 10^{-3} and is updated with cosine annealing schedule. The patch size is set to 256 × 256 pixels and we followed the training strategy of [28, 38]. For data augmentation, we only use random flips and rotations. More experimental results can be found in the Appendix.

4.2. Comparison with State-of-the-Art Methods

We have compared our method with several advanced deblurring methods and use the PSNR and SSIM to evaluate the quality of restored images.

Evaluations on the RealBlur dataset. Since our proposed method is driven by real-world deblurring, we first conduct comparisons on RealBlur [23]. The quantitative analysis results are shown in Tab. 1, the PSNR value of our method is 0.91dB and 0.49dB higher than state-of-the-art GRL on the RealBlur-J and RealBlur-R datasets respectively. Compared to other multi-scale architectures, our method has fewer model computational and running time while the performance is better. Fig. 4 shows visual comparison on RealBlur-J dataset, our method obtains restored text with

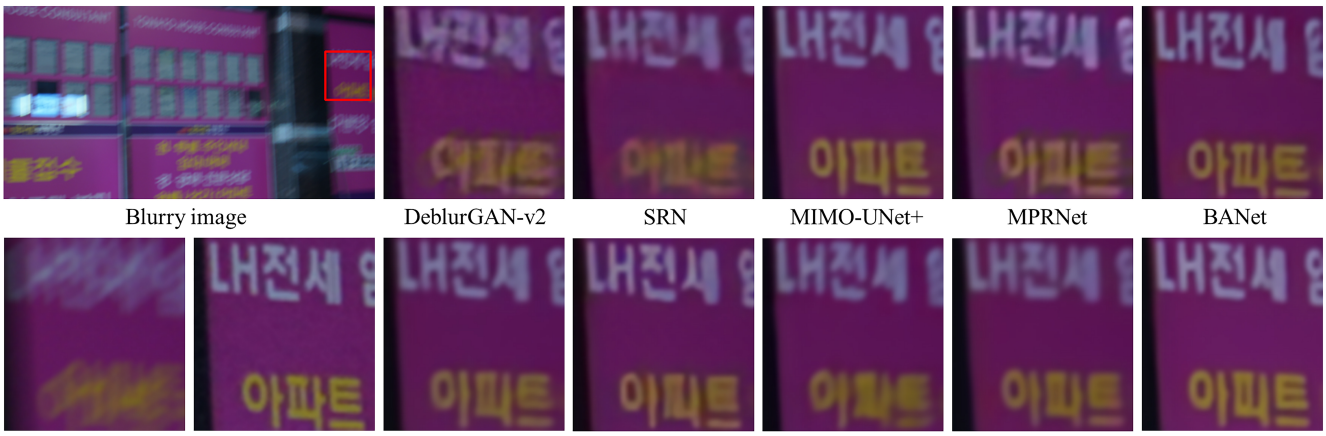


Figure 4. Visual comparisons on the RealBlur-J dataset [23]. The proposed method generates an image with clearer characters.

Method	RealBlur-J		RealBlur-R		Avg.
	PSNR	SSIM	PSNR	SSIM	runtime
DeblurGAN-v2 [12]	29.69	0.870	36.44	0.935	0.04s
SRN [27]	31.38	0.909	38.65	0.965	0.07s
MPRNet [37]	31.76	0.922	39.31	0.972	0.09s
SDWNet [42]	30.73	0.896	38.21	0.963	0.04s
MIMO-UNet+ [3]	31.92	0.919	-	-	0.02s
MIMO-UNet++ [3]	32.05	0.921	-	-	-
DeepRFT+ [19]	32.19	0.931	39.84	0.972	0.09s
BANet [29]	32.00	0.923	39.55	0.971	0.06s
BANet+ [29]	32.42	0.929	39.90	0.972	0.12s
Stripformer [28]	32.48	0.929	39.84	0.974	0.04s
MSSNet [9]	32.10	0.928	39.76	0.972	0.06s
MSDI-Net [13]	32.35	0.923	-	-	0.06s
MAXIM-3S [30]	32.84	0.935	39.45	0.962	-
FFTformer [10]	32.62	0.933	40.11	0.973	0.13s
GRL-B [14]	32.82	0.932	40.20	0.974	1.28s
MLWNet-S	33.02	0.933	-	-	0.04s
MLWNet-B	**33.84**	**0.941**	**40.69**	**0.976**	0.05s

Table 1. Quantitative evaluations on the RealBlur dataset [23]. The experimental results were trained under the corresponding datasets respectively, and average runtime is tested on 256×256 patchs.

richer edge details, while achieving the best contrast, sharpness, brightness, and structural details of the image.

Evaluations on the RSBlur dataset. We further conduct experiments on the latest real-world deblurring dataset RSBlur [24] and follow the protocol of this dataset for fair comparison. Tab. 2 summarizes the quantitative evaluation results of our method with advanced algorithms. The proposed MLWNet achieved the highest PSNR and SSIM, which were 34.94 and 0.880 respectively. In addition, the model we trained on RSBlur dataset also achieved the best results on RealBlur-J dataset. We show some visual comparisons in Fig. 5. We note that our method outperforms other methods in low-light blurry scenes, proving that the proposed method makes textures and structures clearer.

Method	RSBlur		RealBlur-J	
	PSNR	SSIM	PSNR	SSIM
SRN [27]	32.53	0.840	29.86	0.886
MIMO-Unet [3]	32.73	0.846	29.53	0.876
MIMO-Unet+ [3]	33.37	0.856	29.99	0.889
MPRNet [37]	33.61	0.861	30.46	0.899
Restormer [38]	33.69	0.863	30.48	0.891
Uformer-B [33]	33.98	0.866	30.37	0.899
SFNet [4]	34.35	0.872	30.26	0.897
MLWNet-B	**34.94**	**0.880**	**30.53**	**0.905**

Table 2. Quantitative evaluations trained on the RSBlur dataset [24], the RealBlur-J dataset was used for testing only.

Method	[23]→ [24]		MACs(G)
	PSNR	SSIM	
DeblurGAN-v2 [12]	30.15	0.766	42.0
MPRNet [37]	29.56	0.785	760.8
MIMO-UNet+ [3]	29.69	0.792	154.4
BANet [29]	30.19	0.806	263.9
BANet+ [29]	30.24	0.809	588.7
MSSNet [9]	29.86	0.806	154.0
FFTformer [10]	29.70	0.787	131.8
MLWNet-B	**30.91**	**0.818**	108.2

Table 3. Quantitative evaluation for generalizability shows the results of models trained on the RealBlur-J dataset and tested on the RSBlur dataset, MACs are measured on 256 × 256 patches.

In addition, we evaluate the RSBlur dataset using models trained only on RealBlur-J to fairly compare the generalization of methods to real scenarios. As shown in Tab. 3, our method produced results with highest PSNR value of 30.91. The results of Tab. 2 and Tab. 3 show that our model has a better generalization ability.

Evaluations on the GoPro dataset. We conduct a exten-

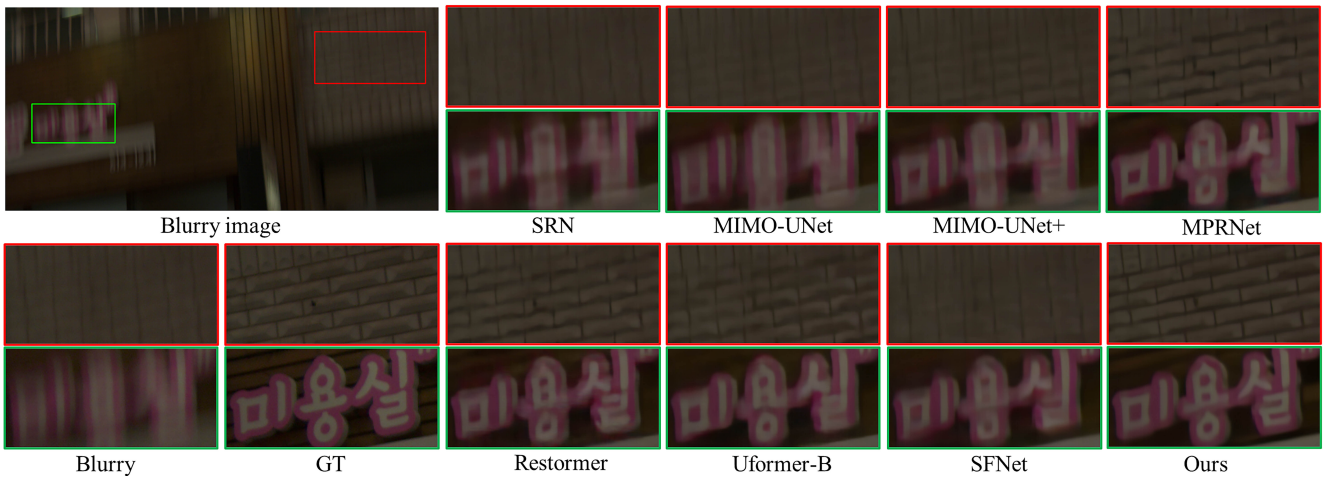


Figure 5. Visual comparisons on the RSBlur dataset [24]. The deblurring performance of the proposed method in low-light is impressive, The recovery of characters and texture structures far exceeds other advanced methods.

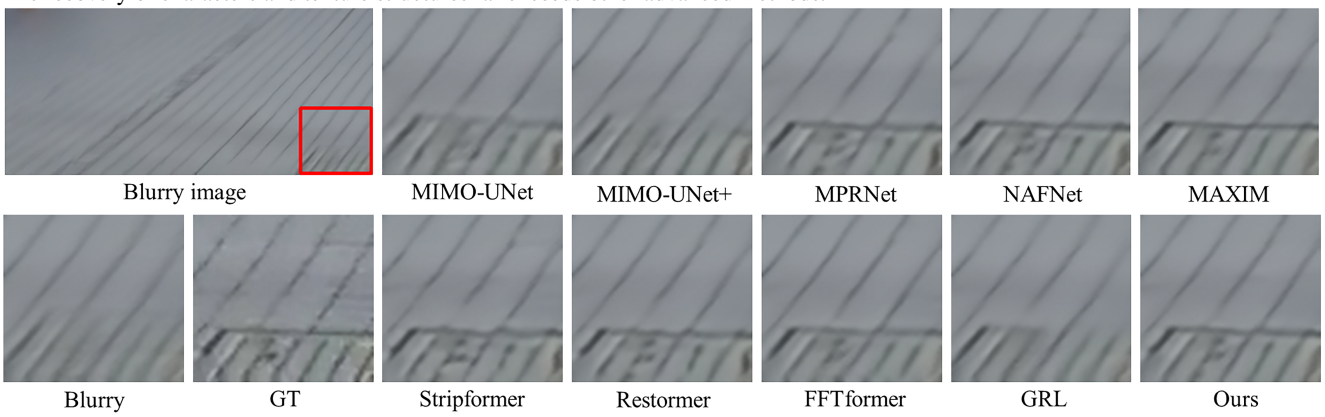


Figure 6. Visual comparisons on the GoPro dataset [20]. Our method better preserves texture information without sharpening.

sive comparison with advanced algorithms on GoPro [20]. Tab. 4 shows the quantitative evaluation results. The gap between our method and the state-of-the-art FFTformer is only 0.001 in SSIM value, while the running time is reduced by 60%. We show a visual comparison on the GoPro dataset in Fig. 6. We can see that our method recovers blurred textures well without sharpening. For the reasons why the proposed MLWNet does not achieve optimal performance in synthetic blurry, we conducted a detailed analysis in Sec. 5.

5. Analysis and Discussion

In this section, we provide a more in-depth analysis and show the contribution of each component of the proposed method. We perform 20w iterations on the RealBlur-J dataset for abtation with a batch size of 8 to train our method using a version of the model with a width of 32, and the results are shown in Tab. 6.

Single-scale vs. Multi-scale. Given that each node of the baseline network is composed of SEB, we have deformed the multi-scale and single-scale according to the architecture. Tab. 5 shows that the multi-scale architecture improves the PSNR and SSIM values to varying degrees, proving that

Method	GOPRO		Avg.runtime
	PSNR	SSIM	
DeblurGAN-v2 [12]	29.55	0.934	0.04s
SRN [27]	30.26	0.934	0.07s
DMPHN [39]	31.20	0.945	0.21s
SDWNet [42]	31.26	0.966	0.04s
MPRNet [37]	32.66	0.959	0.09s
MIMO-UNet+ [3]	32.45	0.957	0.02s
DeepRFT+ [19]	33.23	0.963	0.09s
MAXIM-3S [30]	32.86	0.961	-
Stripformer [28]	33.08	0.962	0.04s
MSDI-net [13]	33.28	0.964	0.06s
Restormer [38]	33.57	0.966	0.08s
NAFNet [2]	33.69	0.967	0.04s
FFTformer [10]	**34.21**	**0.969**	0.13s
GRL-B [14]	<u>33.93</u>	<u>0.968</u>	1.28s
MLWNet-B	33.83	<u>0.968</u>	0.05s

Table 4. Quantitative evaluations trained and tested on the GoPro dataset [20]. Our proposed MLWNet obtains competitive results with a combination of time efficiency and accuracy.

the SIMO strategy we employ helps to eliminate motion blur to some extent. As shown in Sec. 3.1, we use SIMO for coarse-to-fine restoration, and only use multiple outputs to calculate multi-scale losses during training.

WFB and WHB. We default to using WFB and WHB via $\mathcal{L}_{wavelet}$ statute in this section, we also proved the effectiveness of LWN, because LWN is a subpart of the first two. After applying them, the PSNR index improved by 0.25dB compared to the multi-scale baseline, which shows that LWN and its extension modules can easily help multi-scale algorithms improve detail recovery capabilities. Fig. 7 shows the four types of feature maps generated after forward learnable wavelet convolution. We can see that the high and low frequency information of the input feature map are mixed together, and after wavelet group convolution, the network exerts different attention on the feature information in different directions or frequencies.

Effectiveness of $\mathcal{L}_{wavelet}$. By adding unreduced WFB and WHB to the baseline without using wavelet loss, the performance improvement of the baseline is minimal and does not break the bottleneck of multiscale detail recovery, which suggests that the wavelet convolution degenerates into a general group convolution that only serves to deepen the network, proving that merely deepening the network is ineffective for the restoration of detail information.

Method	SISO	MIMO	SIMO
PSNR/SSIM	32.29/0.924	32.19/0.928	**32.37/0.929**
MACs(G)	**19.24**	21.83	19.29

Table 5. Comparison in various input and output modes.

SIMO	WFB	WHB	$\mathcal{L}_{wavelet}$	PSNR	SSIM	MACs(G)
✓				32.37	0.929	19.29
✓	✓	✓		32.40	0.928	28.21
✓		✓	✓	32.49	0.928	25.28
✓	✓		✓	32.57	0.929	22.22
✓	✓	✓	✓	32.62	0.931	28.21

Table 6. Ablation study on components of the proposed MLWNet. We set the baseline network to use SEB in its entirety, and models that do not use SIMO will represent single scales using SISO.

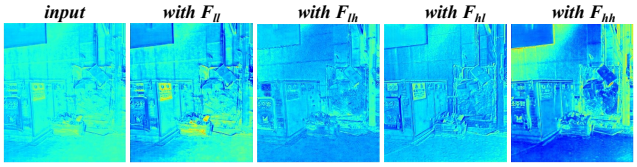
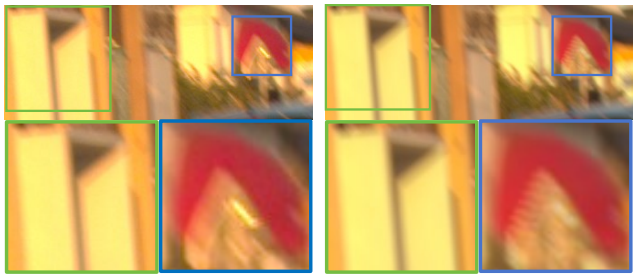


Figure 7. Feature maps representing high-and low-frequency components generated after learnable wavelet convolution. Zoom in on the screen for the best view.

Limitations and analysis. It is observed from Tab. 1-3 that our method fails to achieve the same expected high performance with synthetic blur as it does with realistic blur. We try to analyze the reasons for this phenomenon via dataset comparison and experimental results. First, the average frame synthesis method used in the synthetic blur dataset leads to unnatural discontinuous blur trajectories, thereby introducing strange high-frequency information interference (see Fig. 8). Second, synthetic data tends to produce an unnatural mixture of high and low frequencies. It is easy to mix with the average color area at the edge of the texture, causing the confusion of high-low frequency information.



(a) Realistic Blur (b) Synthetic Blur

Figure 8. The difference between realistic blur (a) and synthetic blur (b). In the green box, the synthetic blur appears with color averaging resulting in high and low frequency confusion, and in the blue box has unnatural discontinuous trajectories.

Noise Level	L1	L2	L3	L4	L5
GoPro	34.81	34.66	33.76	33.19	32.63
RealBlur-J	33.92	33.81	33.97	33.93	33.54

Table 7. Performance comparison at different noise difference levels, where L3 contains the noise difference mean.

Finally, there is a large difference in noise levels between synthetic blur and real blur. We follow [1] to calculate the noise level of each testset, and divided the noise difference between clear images and blurred images of GoPro and RealBlur-J into 5 levels for separate testing. As shown in Tab. 7, we note that as the noise difference increases, our method maintains good performance on RealBlur-J, while the performance drops fast on GoPro. This shows that our method is robust to the noise of real blur, and that there are differences in the noise between synthetic blur and real blur.

6. Conclusion

In this paper, we propose a SIMO-based multi-scale architecture to achieve efficient motion deblurring. We have developed a learnable discrete wavelet transform module, which not only improves the algorithm's ability to recover details, but is also more applicable to the real world. In addition, we construct a reasonable multi-scale loss to guide the recovery of blurred images pixel by pixel and scale by scale, and constrain the learning direction of the wavelet kernel with self-supervised loss to achieve better image deblurring. Through extensive experiments on multiple real and synthetic datasets, we demonstrate that it outperforms existing state-of-the-art methods in terms of quality and efficiency of image restoration, especially with optimal deblurring performance and generalization in real scenes.

References

[1] Guangyong Chen, Fengyuan Zhu, and Pheng Ann Heng. An efficient statistical method for image noise level estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 477–485, 2015. 8

[2] Liangyu Chen, Xiaojie Chu, Xiangyu Zhang, and Jian Sun. Simple baselines for image restoration. In *European Conference on Computer Vision*, pages 17–33. Springer, 2022. 1, 2, 3, 5, 7

[3] Sung-Jin Cho, Seo-Won Ji, Jun-Pyo Hong, Seung-Won Jung, and Sung-Jea Ko. Rethinking coarse-to-fine approach in single image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4641–4650, 2021. 1, 2, 6, 7

[4] Yuning Cui, Yi Tao, Zhenshan Bing, Wenqi Ren, Xinwei Gao, Xiaochun Cao, Kai Huang, and Alois Knoll. Selective frequency network for image restoration. In *International Conference on Learning Representations, ICLR*, 2023. 6

[5] Ingrid Daubechies. The wavelet transform, time-frequency localization and signal analysis. *IEEE transactions on information theory*, 36(5):961–1005, 1990. 2

[6] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 7036–7045, 2019. 1

[7] Wooseok Ha, Chandan Singh, Francois Lanusse, Srigokul Upadhyayula, and Bin Yu. Adaptive wavelet distillation from neural networks through interpretations. *Advances in Neural Information Processing Systems*, 34:20669–20682, 2021. 4

[8] Bumsoo Kim, Jonghwan Mun, Kyoung-Woon On, Minchul Shin, Junhyun Lee, and Eun-Sol Kim. Mstr: Multi-scale transformer for end-to-end human-object interaction detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 19578–19587, 2022. 1

[9] Kiyeon Kim, Seungyong Lee, and Sunghyun Cho. Mssnet: Multi-scale-stage network for single image deblurring. In *European Conference on Computer Vision*, pages 524–539. Springer, 2022. 1, 2, 6

[10] Lingshun Kong, Jiangxin Dong, Jianjun Ge, Mingqiang Li, and Jinshan Pan. Efficient frequency domain-based transformers for high-quality image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5886–5895, 2023. 1, 2, 3, 6, 7

[11] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8183–8192, 2018. 2

[12] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 8878–8887, 2019. 1, 6, 7

[13] Dasong Li, Yi Zhang, Ka Chun Cheung, Xiaogang Wang, Hongwei Qin, and Hongsheng Li. Learning degradation representations for image deblurring. In *European Conference on Computer Vision*, pages 736–753. Springer, 2022. 1, 2, 6, 7

[14] Yawei Li, Yuchen Fan, Xiaoyu Xiang, Denis Demandolx, Rakesh Ranjan, Radu Timofte, and Luc Van Gool. Efficient and explicit modelling of image hierarchies for image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 18278–18289, 2023. 2, 6, 7

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 1

[16] Pengju Liu, Hongzhi Zhang, Wei Lian, and Wangmeng Zuo. Multi-level wavelet convolutional neural networks. *IEEE Access*, 7:74973–74985, 2019. 4, 5

[17] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8759–8768, 2018. 1

[18] Wei Liu, Qiong Yan, and Yuzhi Zhao. Densely self-guided wavelet network for image denoising. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 432–433, 2020. 4

[19] X Mao, Y Liu, W Shen, Q Li, and Y Wang. Deep residual fourier transformation for single image deblurring. arxiv 2021. *arXiv preprint arXiv:2111.11745*, 2021. 2, 3, 6, 7

[20] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3883–3891, 2017. 1, 2, 5, 7

[21] Dongwon Park, Dong Un Kang, Jisoo Kim, and Se Young Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *European Conference on Computer Vision*, pages 327–343. Springer, 2020. 1, 2

[22] Zequn Qin, Pengyi Zhang, Fei Wu, and Xi Li. Fcanet: Frequency channel attention networks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 783–792, 2021. 3

[23] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*, pages 184–201. Springer, 2020. 1, 5, 6

[24] Jaesung Rim, Geonung Kim, Jungeon Kim, Junyong Lee, Seungyong Lee, and Sunghyun Cho. Realistic blur synthesis for learning image deblurring. In *European conference on computer vision*, pages 487–503. Springer, 2022. 2, 5, 6, 7

[25] Roman Suvorov, Elizaveta Logacheva, Anton Mashikhin, Anastasia Remizova, Arsenii Ashukha, Aleksei Silvestrov, Naejin Kong, Harshith Goka, Kiwoong Park, and Victor Lempitsky. Resolution-robust large mask inpainting with fourier convolutions. In *Proceedings of the IEEE/CVF winter*

conference on applications of computer vision, pages 2149–2159, 2022. 3

[26] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 10781–10790, 2020. 1

[27] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8174–8182, 2018. 1, 2, 6, 7

[28] Fu-Jen Tsai, Yan-Tsung Peng, Yen-Yu Lin, Chung-Chi Tsai, and Chia-Wen Lin. Stripformer: Strip transformer for fast image deblurring. In *European Conference on Computer Vision*, pages 146–162. Springer, 2022. 1, 5, 6, 7

[29] Fu-Jen Tsai, Yan-Tsung Peng, Chung-Chi Tsai, Yen-Yu Lin, and Chia-Wen Lin. Banet: a blur-aware attention network for dynamic scene deblurring. *IEEE Transactions on Image Processing*, 31:6789–6799, 2022. 1, 2, 6

[30] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5769–5780, 2022. 6, 7

[31] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7464–7475, 2023. 1

[32] Luyuan Wang and Yankui Sun. Image classification using convolutional neural network with wavelet domain inputs. *IET Image Processing*, 16(8):2037–2048, 2022. 3

[33] Zhendong Wang, Xiaodong Cun, Jianmin Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: A general u-shaped transformer for image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 17683–17693, 2022. 6

[34] Moritz Wolter and Jochen Garcke. Adaptive wavelet pooling for convolutional neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 1936–1944. PMLR, 2021. 4, 5

[35] Yanchao Yang and Stefano Soatto. Fda: Fourier domain adaptation for semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4085–4095, 2020. 3

[36] Changqian Yu, Changxin Gao, Jingbo Wang, Gang Yu, Chunhua Shen, and Nong Sang. Bisenet v2: Bilateral network with guided aggregation for real-time semantic segmentation. *International Journal of Computer Vision*, 129: 3051–3068, 2021. 4

[37] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14821–14831, 2021. 1, 2, 6, 7

[38] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5728–5739, 2022. 2, 5, 6, 7

[39] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5978–5986, 2019. 2, 7

[40] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Bjorn Stenger, Wei Liu, and Hongdong Li. Deblurring by realistic blurring. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2737–2746, 2020. 2

[41] Yijie Zhong, Bo Li, Lv Tang, Senyun Kuang, Shuang Wu, and Shouhong Ding. Detecting camouflaged object in frequency domain. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4504–4513, 2022. 3

[42] Wenbin Zou, Mingchao Jiang, Yunchen Zhang, Liang Chen, Zhiyong Lu, and Yi Wu. Sdwnet: A straight dilated network with wavelet transformation for image deblurring. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1895–1904, 2021. 1, 2, 3, 6, 7