# Multi-Scale Separable Network for Ultra-High-Definition Video Deblurring

Senyou Deng[1], Wenqi Ren[1, 2, ✉], Yanyang Yan[1], Tao Wang[3], Fenglong Song[3], and Xiaochun Cao[1]
[1]Institute of Information Engineering, Chinese Academy of Sciences, Beijing, P.R. China;
[2]State Key Lab. for Novel Software Technology, Nanjing University; [3]Huawei Noah's Ark Lab.
{dengsenyou,yanyanyang,caoxiaochun}@iie.ac.cn; rwq.renwenqi@gmail.com;
{wangtao10,songfenglong}@huawei.com.

## Abstract

*Although recent research has witnessed a significant progress on the video deblurring task, these methods struggle to reconcile inference efficiency and visual quality simultaneously, especially on ultra-high-definition (UHD) videos (e.g., 4K resolution). To address the problem, we propose a novel deep model for fast and accurate UHD Video Deblurring (UHDVD). The proposed UHDVD is achieved by a separable-patch architecture, which collaborates with a multi-scale integration scheme to achieve a large receptive field without adding the number of generic convolutional layers and kernels. Additionally, we design a residual channel-spatial attention (RCSA) module to improve accuracy and reduce the depth of the network appropriately. The proposed UHDVD is the first real-time deblurring model for 4K videos at 35 fps. To train the proposed model, we build a new dataset comprised of 4K blurry videos and corresponding sharp frames using three different smartphones. Comprehensive experimental results show that our network performs favorably against the state-of-the-art methods on both the 4K dataset and public benchmarks in terms of accuracy, speed, and model size.*

## 1. Introduction

Ultra-High-Definition (UHD, i.e., 12 megapixels or 4K) becomes a trend during the last several years. Many device manufacturers have released new devices (e.g., smartphones and DSLR cameras) with 4K support. Unfortunately, irregular camera shakes and high-speed movements often generate undesirable blurs in captured UHD videos. The blurred video leads to visually low quality and hampers high-level vision tasks [27].

Numerous image and video deblurring methods have been proposed to recover the sharp frames from a captured blurry video. Conventional methods usually make assumptions on motion blurs and latent frames. Among these methods, motion blurs are usually modeled as uniform kernels [35, 57, 54] or non-uniform kernels (e.g., region-wise [11, 1, 9, 45, 4] and pixel-wise [10, 33]). While the sharp frames are usually constrained by hand-crafted image priors [19, 39, 24, 8] to regularize the solution space. However, these assumptions do not usually hold for real cases, which leads to an inaccurate estimation of the blur kernel and the quality of the deblurred image is not desirable.

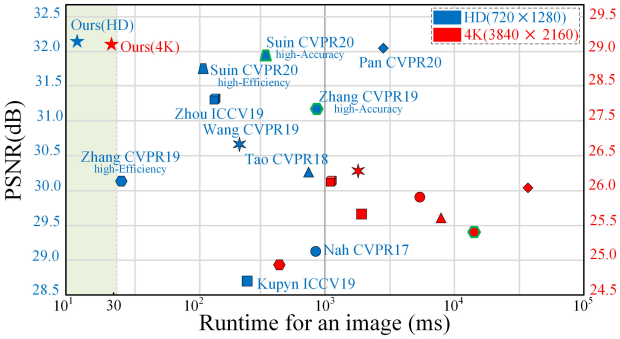



Figure 1. PSNR(dB) vs. runtime(ms) of several deblurring methods and our method on different datasets. The green region indicates real-time inference at 30 fps. The blue PSNR and icons are methods on the HD dataset and the red ones with the same shape are on the 4K dataset. Clearly, our method is better not only in efficiency but also in accuracy. Furthermore, we try to handle the 4K resolution and achieve considerable results.

To address these issues, deep learning deblurring algorithms have been proposed recently. These methods use convolutional neural networks (CNNs) to explicitly learn features from blurry input and regress the blur kernel [34, 7, 38] or directly recover the clean image [26, 52, 53, 56, 50]. These algorithms can remove blur effects caused by camera shakes and object motions, and achieve state-of-the-art results on image deblurring task. However, existing CNN-based methods have two major problems. The first one is that the computation and memory usage are too large for practical applications, especially when the resolution of input images is high. For example, the recent video deblurring method of CDVD-TSP [30] needs about four seconds and one minute to deblur a single frame from HD (720p) and

UHD (4K) videos, respectively. The second one is that most existing CNN-based video deblurring methods lack flexibility in dealing with different types of information due to less discrimination ability learning between blur and sharp pairs. Therefore, generating detailed textures from blurred videos is still a non-trivial problem.

To overcome the above limitations, we present a new UHDVD network which has advantages of high efficiency, low memory overhead, and high quality deblurring performance. Our method is partially motivated by the patch-hierarchical image deblurring methods [50, 37] where multi-patch hierarchy is fed into the network. This scheme achieves great improvements on 720p image deblurring at very high efficiency. However, the multi-patch hierarchy [50, 37] has the same spatial resolution at different levels and require slow algorithms to layout the patches and stitch them together, which hinders the reconstruction ability of the deep network and reduces the feature extraction speed. We note that a low-resolution image is easier to recover than high-resolution since there are less class information and fewer modes (i.e., edges and textures) [13]. Therefore, we propose a novel separable-patch architecture combined with a multi-scale integration scheme, which allows to capture the global structure on the coarse scale and process multiple patches of each scale in parallel within an iteration.

In addition, most existing deblurring algorithms employ a cascaded network to help the latent frame restoration [30, 50]. However, to the best of our knowledge, simply stacking the same network to construct deeper networks can hardly obtain better improvements [37]. To achieve more expressive and intelligent video deblurring capability, we further propose a cascaded residual channel and spatial attention (RCSA) module to improve deblurring performance without sacrificing speed. The proposed RCSA is able to adaptively learn more useful channel-wise features and emphasize the most informative region on the feature map for video deblurring.

The main contributions of this paper are summarized as:

- We propose a novel UHDVD network by using a separable-patch architecture combined with a multi-scale integration scheme. To the best of our knowledge, our proposed model is the first video deblurring model that can deblur 4K videos in real-time by parallelizing multiple patches.
- We design a cascaded RCSA module to improve feature representation power and discriminative ability, ensuring high deblurring performance.
- We establish a 4K video deblurring dataset including both synthesized and real captured videos. We evaluate the proposed model on the proposed benchmark and public datasets [25, 26, 36] and show that the proposed method performs favorably against state-of-the-arts.

## 2. Related Work

To address the ill-posed nature of the deblurring problem, traditional methods make different assumptions and use appropriate priors. These include total variation [32], sparse image priors [22, 5], gradient distributions [16, 2], patch priors [24, 39], $l_0$-norm regularizers [46, 20], etc. One of the limitations of these prior-based approaches is that they does not always hold for dynamic scenes containing depth variations and moving objects.

Recently, due to the immense success of deep learning in the computer vision field, many CNN-based approaches have also been proposed for image deblurring [51, 12, 28]. In these methods, the idea is to learn a mapping between the blurry input and the corresponding sharp image using a CNN architecture. In addition, Generative Adversarial Nets (GANs) have also been exploited for image deblurring due to the texture generation ability [18, 17]. However, these models usually involve large-size of network parameters and consume long processing times, which cannot satisfy the ever-increasing demand for real-time deblurring, especially for UHD videos.

**Multi-Scale and Multi-Patch Networks.** Coarse-to-fine (i.e., multi-scale) methods have been popular in the conventional deblurring literature [46, 16], recent CNN-based methods also use the multi-scale mechanism to mimic conventional coarse-to-fine approaches. Nah et al. [26] propose the first multi-scale CNN-based deblurring network, which starts from a coarse scale of the blurred input and then progressively deblur the input at higher scales until the full resolution latent image is recovered. Tao et al. [41] introduce a scale-recurrent network by training shared parameters across scales. The approach can preserve image structures and motion information from the previous coarser scales based on the recurrent network. Gao et al. [6] improve the multi-scale CNN [41] by selective sharing parameters and modules in each scale. However, these multi-scale networks are usually large and suffer from expensive inference time.

To address this challenge, a hierarchical multi-patch model [50] is proposed to exploit the motion information at different scales by feature aggregation over multiple patches. Suin et al. [37] combine the multi-patch hierarchical and a global attention mechanism to avoid cascading network along depth. Zamir et al. [48] use a similar scheme in a multi-stage architecture to obtain better results but increase the computation time. These three multi-patch networks support deblurring of 720p images in real time, but still struggle to reconcile efficiency and deblurring performance on full-high-definition (FHD, $1920 \times 1080$ resolution) and UHD videos (e.g., 4K resolution). The performance of some representative methods on 720P and UHD datasets are shown in Figure 1.

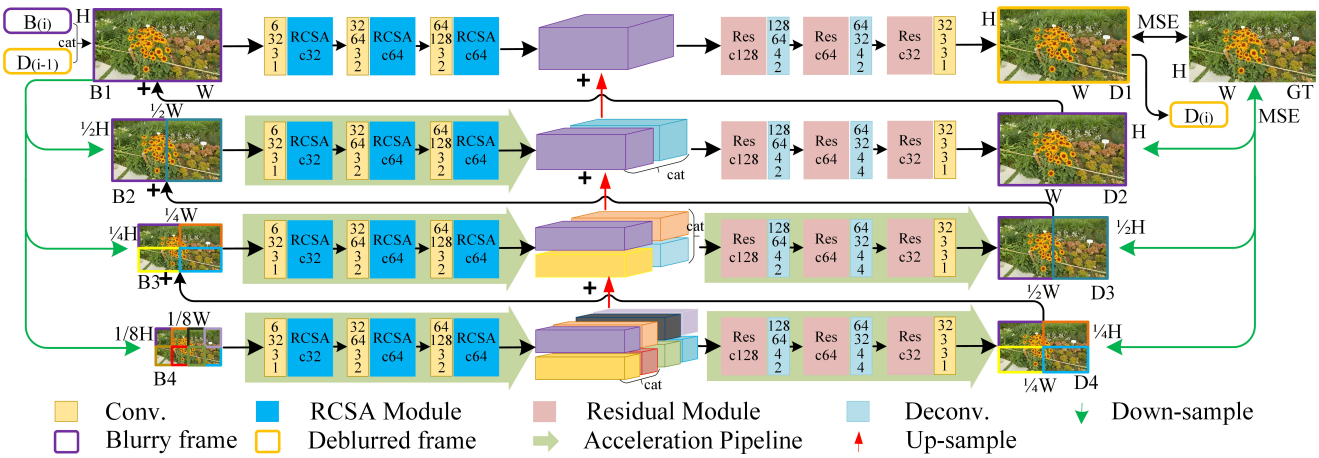



Figure 2. Our proposed UHDVD model and its some layer configurations. Symbol "+" is a summation akin to the target data.

**FHD and UHD Image Enhancement.** A few methods have been proposed to recover clear images from FHD or UHD degraded inputs by learning bilateral regularizer [14] or 3D Lookup Tables [49]. However, all these methods reconstruct the final output by using some sophisticated interpolation techniques from a down-sampled version. In contrast to these approaches, our network directly deblur images at the full-resolution inputs on the finest scale and is the first real-time deblurring model for 4K videos at 35 fps.

## 3. Proposed Method

The general idea of our proposed network is to integrate the multi-scale and multi-patch schemes properly, and we further propose a separable-patch strategy to dramatically accelerate reference implementations. The architecture of our UHDVD is shown in Figure 2.

Inspired by the work of [36], which demonstrates that simply stacking neighboring frames without any alignment performs better than the single frame based method. In our network, given a blurred video $\mathbf{B}$, the previous deblurred frame ($\mathbf{D_{(i-1)}}$) is concatenated with the current blurry frame ($\mathbf{B_{(i)}}$) at channel dimension as our network input to improve deblurring result. Thus, the input channel of the first convolution layer in each scale network is 6 instead of 3. The concatenated input is then half-downsampled ordinally at different 4 scales (**B1, B2, B3, B4**), and a corresponding sharp images (**D1, D2, D3, D4**) is recovered at each scale. The sharp one at Scale 1 (**D1**) is the final output. Based on this scheme, we can set a larger "crop size" in training process to expand the receptive field, which means more feature information [3, 47, 43] can be captured and can improve the final deblurred results. In addition, the number of split patches in each scale is multiplied by scales. The input for each scale is generated by dividing the original image input ($\mathbf{B_{i,\ i=1,2,3,4}}$) into multiple non-overlapping patches. The maximum number of patch ($B_{i,j}$) for each scale is set as $J = [1,\ 2,\ 4,\ 8]$. These process can be modeled as:

$$D^s_{i,j}, F^s_{i,j} = Net^s(B^s_{i,j},\ D^s_{i-1,j},\ D^{s+1}_{i,j},\ F^{s+1}_{i,j};\ \theta^{p_s}), \quad (1)$$

where $s$ is the scale index, with $s = 1$ representing the finest scale, $j$ and $i$ are the patch index and the video frame index, respectively; $D^s_{i,j}$ and $B^s_{i,j}$ are our network output and input at the $s$-th scale and $j$-th patch of $i$-th frame, respectively; $D^{s+1}_{i,j}$ denotes the deblurred $j$-th patch of $i$-th frame at upper scale; $Net^s$ represents the proposed 4K video deblurring network with training parameters denoted as $\theta^{p_s}$. Since the network is also recurrent, the middle state features $F^s_{i,j}$ flow across scales from $s+1$ to $s$.

As shown in Figure 2, our real-time 4K video deblurring network is composed of 4 similar encoder-decoder architectures at each scale. Each encoder branch contains 3 convolutions with the kernel size of $3 \times 3$ and stride 1, and each convolution layer is followed by a RCSA module. Meanwhile, in each decoder branch, the residual modules is in the front of every deconvolutional layer. The kernel size of the first deconvolution in the decoder is $4 \times 4$ with stride 2, the second deconvolution's kernel size is $4 \times 4$ with stride 4 to expand the output size twice so that its size is equal to the input of the upper layer except the scale 1. The third layer of the decoder is a normal convolution with its output channel is 3. The red arrows represent the middle feature maps $F^s_{i,j}$ in (1), which is double up-sampled from $F^{s+1}_{i,j}$.

### 3.1. Asymmetrical Encoder-decoder Architecture

The symmetrical encoder-decoder structure has been proven to be effective in many methods [26, 50, 41, 23], which first progressively transforms input data into feature

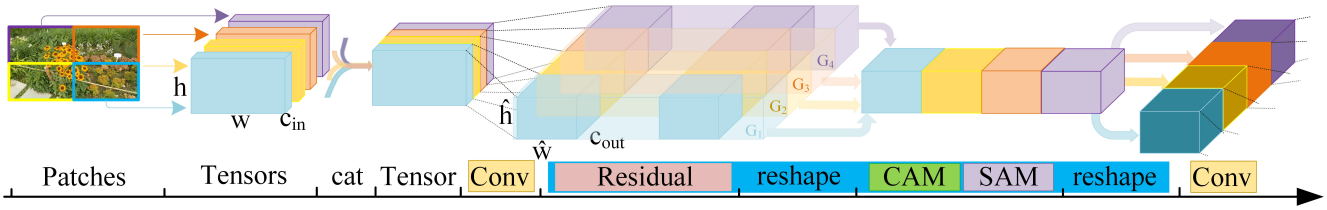



Figure 3. The separable-patch acceleration architecture. Taking the encoder branch of scale 3 before the second RCSA module as an example, where $G_i$ is the group number in convolutional layers.

maps with smaller spatial sizes and more channels (in encoder), and then transforms them back to the shape of the input (in decoder). Skip-connection between corresponding layers are widely used to combine different levels of feature information. Usually, more convolution layers are added in every level to further increase network depth to improve the accuracy. However, directly employing the symmetrical encoder-decoder structure is not the best choice for our work with the following reasons. First, we aim to process 4K resolution videos in real-time, so it is still a great challenge by using traditional encoder/decoder structure since the sizes of middle feature maps from 4K inputs are still very large compared with common 720P images. Second, using more convolution layers at every level of encoder-decoder modules will make the network slow to converge (with flat convolution at each scale) although this method can reduce the size of the processed image.

Based on these considerations, we propose an approximate asymmetrical encoder-decoder structure inspired by super-resolution framework [40]. In our new architecture, the transformation between encoder and decoder is different from the traditional architecture. The asymmetry is mainly reflected in the different modules that we used in encoder and decoder branches. In the decoder branch we just use three normal light residual module revealed in Figure 4(a) after each standard deconvolution to reduce the parameter numbers so that the computation speed can be greatly improved. Each of the three residual modules contains 1 convolution layer with kernel size of $3 \times 3$ and stride 1, then followed by a ReLU activation function and another same convolutional layer. Relatively, we used RCSA module instead of the residual module (Figure 4(a)) in the encoder branch. The convolution in the encoder and the deconvolution in the decoder is also asymmetric in channel dimension.

### 3.2. Separable-Patch Acceleration Architecture

To further improve the inference speed of the UHDVD model to reach the goal that can deblur a 4K resolution video within 30 ms at a single GPU, we design the separable-patch acceleration architecture to handle multiple patches or feature maps at the same time. As shown in Figure 3, the process of this architecture is linear like a pipeline. At the beginning, the multiple patches (e.g., $n = 4$) are concatenated together as a new tensor in channel dimension, and its size is $[batch\_size,\ n * c_{in},\ h,\ w]$. The tensor is processed by the subsequent convolutional layer with setting the parameter $groups = n$.

Obviously, the computing burden of the new tensor is $((n * c_{in}) * (n * c_{out}) * kernel\_size^2)/groups$, while it is equal to $n$ original tensors. But the benefits are that we can change these $n$ serial computations to a parallel computation, this will greatly reduce the computation time. After the computation in residual module, we reshape the tensor to the size $[batch\_size * n,\ c_{out},\ h,\ w]$ so that it can be computed synchronously in channel attention module and spatial attention module, respectively. The output will be taken as the input of the next RCSA module and this acceleration is going to continue until we get the middle feature maps or restored images of the scale.

### 3.3. RCSA Module

We further propose a new RCSA module that contains a channel attention module and a spatial attention module [55, 6] in the deblurring network. The architecture of the RCSA is shown in Figure 4(b). Following the recent success of transformer architecture in natural language processing domain [42] and image processing tasks [37, 21, 31], the main building blocks of RCSA is the channel attention and spatial attention which calculates the response at channel and spatial dimensions.

The Channel Attention Module (**CAM**) is made of two adaptive pooling computation: average pooling and maximum pooling. After each pooling calculation is a standard convolutional layer with its input channel number is the same as the output channel of previous convolutional layer which can be seen on blue blocks (RCSA Modules, c32/c64/c128) in Figure 2, the output channel is 1/8 of input channel, kernel size is $1 \times 1$ and `bias` is false. Then, there is a ReLU activation function and another same convolution which input and output channels are the exactly opposite of the front convolution.

$$C(x) = sigmoid(M_C(P_{avg}(x)) + M_C(P_{max}(x))), \quad (2)$$

where $P_{avg}$ and $P_{max}$ are average pooling and maximum

pooling, respectively; $M_C$ is the processing module described above. Two processed pooling results added together as the input of $sigmoid$ function.

The Spatial Attention Module (**SAM**) only has one convolutional layer with input channel is 2, output channel is 1, kernel size is $3 \times 3$, padding size is 1 and bias is false. The input data is firstly processed by average and maximum calculation respectively at $dim = 1$ and then concatenated together at the same $dim$.

$$S(x) = sigmoid(M_S(Avg(x, 1), Max(x, 1))), \quad (3)$$

where $M_S$ is the special convolution described above and the output will de calculated by $sigmoid$ function before passing to the next layer. The subsequent calculation is as follows:

$$\begin{aligned} O_C &= C(x) \times x, \\ O_{RCSA} &= S(O_C) \times O_C + x, \end{aligned} \quad (4)$$

where $O_C$ is the output of CAM module while $O_{RCSA}$ is the output of RCSA module, the operation "$\times$" denotes point-wise multiplication.

The structure of RCSA module is simple and the depth of it is light. It does not significantly affect computing speed but improves the deblurring results to a certain extent. The following experiments have proved this conclusion.

### 3.4. Loss Function

As a video deblurring network, different from approach [56] that use the previous middle feature map with extra computation, we directly use the whole previous deblurred frame to guarantee the temporal continuity by concatenation. So we do not have to spend extra time to calculate the optical flow and its loss [7, 30].

Meanwhile, the coarse-to-fine approach desires that every mid-level outputs are the deblurred image of the corresponding scales. Thus, the training loss of our proposed UHDVD network is the MSE loss between the image content of the network output and the ground truth frame at each scale,

$$\mathcal{L}_{i\_MSE} = \sum_{s=1}^{S} \frac{\mathcal{K}_s}{C_i^s H_i^s W_i^s} ||D_i^s - G_i^s||_2^2, \quad (5)$$

where $D_i^s$ and $G_i^s$ are the deblurred image and the ground truth at the $s$-th scale of $i$-th frame, respectively; $C_i^s$, $H_i^s$, $W_i^s$ are dimensions of multi-scale image; $\mathcal{K}_s$ is the weights for each scale. We empirically set $\mathcal{K}_{1,\,2,\,3,\,4} = [0.7,\ 0.15,\ 0.1,\ 0.05]$. In addition, $S$ is the number of scales in our network, we set $S$ to 4 in the paper. Besides, we add the Total Variation (TV) loss to avoid stripe artifect in the recover image. So the total loss is formulated as:

$$\mathcal{L}_{total} = \mathcal{L}_{i\_MSE} + \beta \mathcal{L}_{i\_TV}, \quad (6)$$

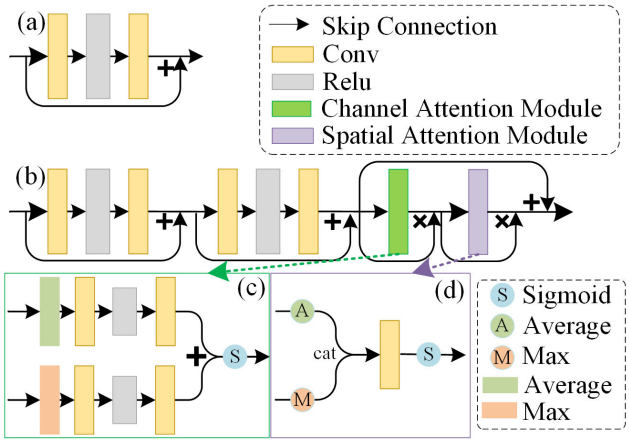



Figure 4. The structure of Residual module (a) and RCSA module (b) in UHDVD. (c) and (d) are CAM and SAM in RCSA module. Symbol "$\times$" is point-wise multiplication and "+" is addition.

The $\beta$ is set for $1e^{-7}$ to control the impact of TV loss. Note that our 4K real-time video deblurring network does not rely on other complicated loss functions such as adversarial loss [26, 17] and optical flow loss [30], only using the MSE and TV loss can achieve competitive results as demonstrated in next section.

## 4. Experiments

In this section, we evaluate the proposed algorithm on both synthetic datasets and real-world 4K videos with comparisons to the state-of-the-art image/video deblurring methods in terms of accuracy and visual effect. For fair comparisons, we also evaluate our method on public 720p datasets with these methods. The new 4KRD dataset will be made available to the public for further discussion and research. Results on more frames and real blurry videos can be found in the supplementary material.

### 4.1. Implementation Details

All our experiments are implemented in PyTorch and evaluated on a single NVIDIA Tesla V100 GPU with 32GB RAM. The batch size is set to 1 during training because every frame needs its previous deblurred frame as an extra feature. The Adam optimizer [15] is used to train our models with patch size of 512×512. The initial learning rate is set to 0.0001 and the decay rate to 0.1. We normalize frame to the range of [0,1] and subtract 0.5.

### 4.2. Dataset

Due to there is no public high-quality 4K dataset for deblurring study, we choose the scheme of [27] to generate a 4K Resolution Deblurring (4KRD) dataset. The proposed dataset covers a diversity of characters, people, artificial or

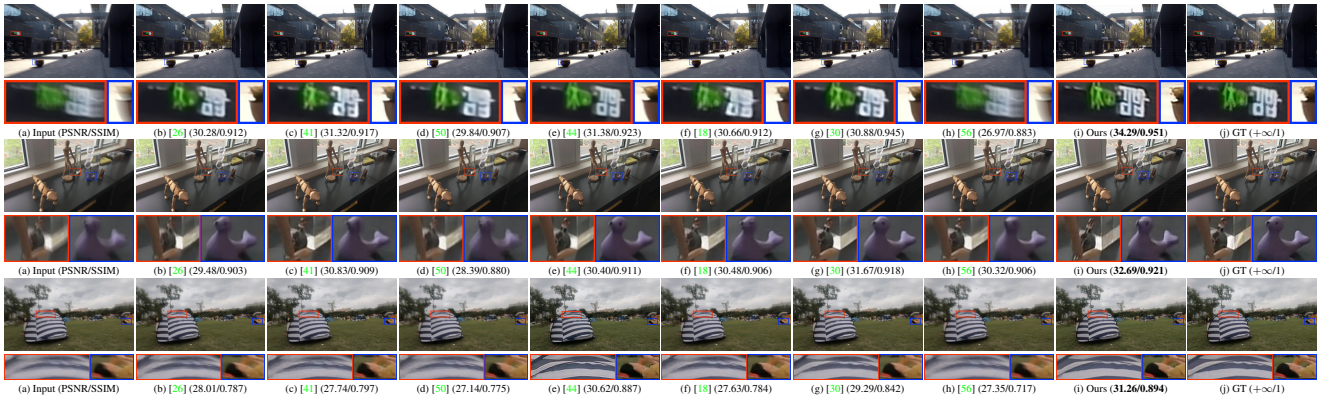



Figure 5. Quantitative evaluations on the HD deblurring datasets, from top to bottom are GoPro [26], DVD [36] and REDS [25], respectively. Our proposed method generates much clearer images with higher PSNR and SSIM values compared to MSResNet [26], SRN [41], DMPHN-Stack(4) [50], EDVR [44], DeblurGAN-v2 [18], CDVD-TSP [30], and STFAN [56]. (**Zoom in for best view)**

Table 1. Quantitative comparisons against existing deblurring methods (MSResNet [26], SRN [41], DeblurGAN-v2 [18], DMPHN-Stack(4) / DMPHN-(1-2-4-8) [50], EDVR [44], CDVD-TSP [30], STFAN [56]) on four deblurring benchmarks, The runtime (writing generated images to disk is not considered) is expressed in millisecond of an image. We use **bold** and underline to indicate the best and the second-best performances, respectively. The $^\star$ denotes that EDVR [44] uses the validation data for training on the REDS dataset.

| Datasets | GoPro [26] (test) | | | | | | | | DVD [36] (10 clips) | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Methods | [26] | [41] | [18] | [50] | [44] | [30] | [56] | Ours | [26] | [41] | [18] | [50] | [44] | [30] | [56] | Ours |
| PSNR | 28.45 | 30.10 | 29.55 | 31.20/30.25 | 26.87 | **31.67** | 28.63 | 31.33 | 28.98 | 29.10 | 28.54 | 30.47/29.91 | 30.27 | 32.13 | 31.24 | **32.19** |
| SSIM | 0.917 | 0.932 | 0.934 | **0.945**/0.935 | 0.843 | 0.928 | 0.863 | 0.921 | 0.885 | 0.899 | 0.925 | 0.881/0.866 | 0.917 | 0.927 | 0.934 | **0.937** |
| Time | 747.8 | 731.7 | 293.6 | 1029.3/30.9 | 384.6 | 4216.6 | 150.4 | **12.7** | 775.8 | 783.6 | 312.2 | 987.9/30.4 | 289.2 | 4098.2 | 177.2 | **13.2** |
| Datasets | REDS [25] (validation) | | | | | | | | 4KRD (13 clips) | | | | | | | |
| Methods | [26] | [41] | [18] | [50] | [44]$^\star$ | [30] | [56] | Ours | [26] | [41] | [18] | [50] | [44] | [30] | [56] | Ours |
| PSNR | 26.49 | 25.40 | 25.61 | 25.18/25.06 | **30.63** | 26.29 | 25.49 | 27.53 | 25.81 | 25.58 | 25.64 | 24.99/24.91 | 26.36 | 26.43 | 26.14 | **27.88** |
| SSIM | 0.742 | 0.734 | 0.731 | 0.724/0.724 | **0.850** | 0.774 | 0.719 | 0.815 | 0.778 | 0.759 | 0.763 | 0.757/0.748 | 0.803 | 0.793 | 0.800 | **0.813** |
| Time | 802.6 | 823.3 | 350.8 | 1069.9/29.3 | 325.7 | 3765.6 | 155.7 | **13.9** | 7543.4 | 8723.3 | 3283.4 | 10378.1/399.4 | 2428.1 | 26922.9 | 953.2 | **27.9** |

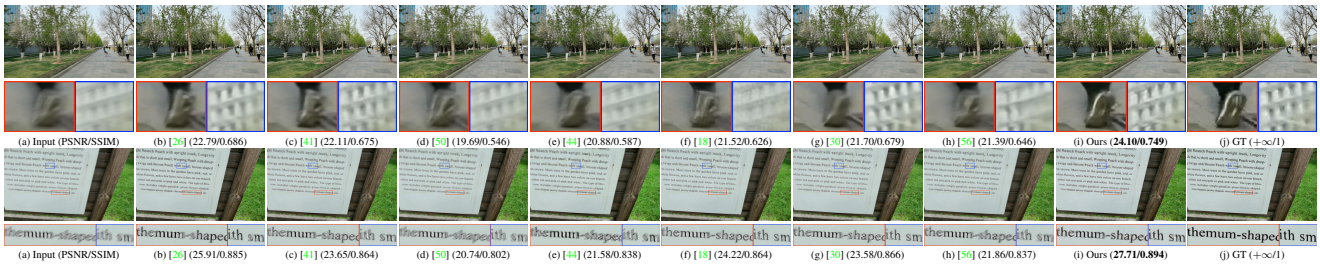
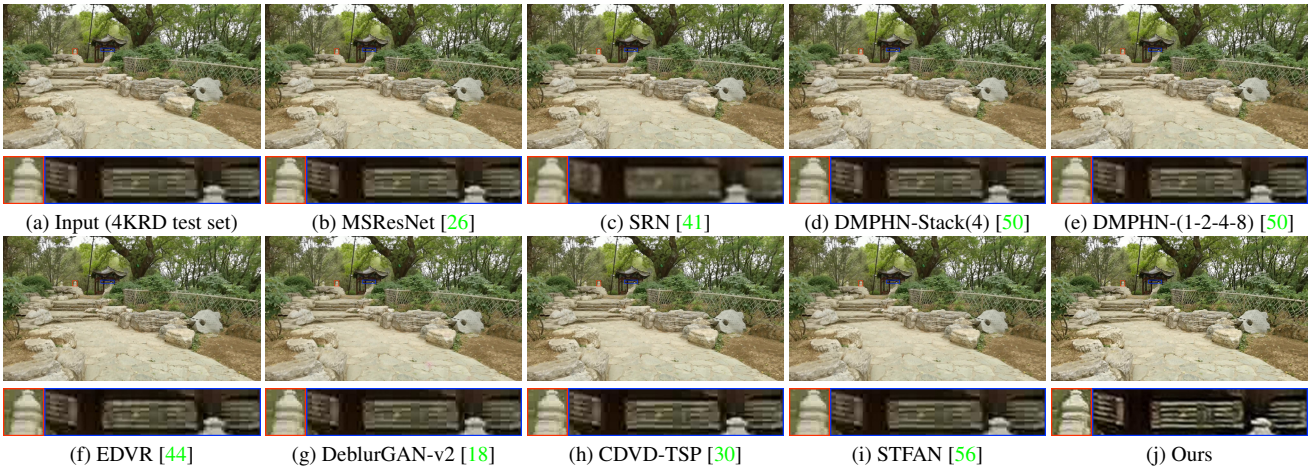



Figure 6. Quantitative evaluations on our 4K resolution deblurring datasets. Our UHDVD generates much clearer images with higher PSNR and SSIM compared to MSResNet [26], SRN [41], DMPHN-Stack(4) [50], EDVR [44], DeblurGAN-v2 [18], CDVD-TSP [30], and STFAN [56]. (**Zoom in for best view**)

natural objects, indoor scenes, outdoor landscapes and city street views, etc. The generating process is made up of two main parts: frame interpolation and dataset synthesis. The video capturing equipments are mainstream flagship mobile phones at the time, e.g., iPhone 11 Pro Max, Samsung S20 Ultra, and HUAWEI Mate 30 Pro. We also use a DJI Osmo Mobile 3 to stabilize mobile phones so that the captured videos as clear as possible.

High frame rates are necessary for the subsequent multi-frame fusion to ensure the continuity of frames in the syn-



(a) Input (4KRD test set) (b) MSResNet [26] (c) SRN [41] (d) DMPHN-Stack(4) [50] (e) DMPHN-(1-2-4-8) [50]

(f) EDVR [44] (g) DeblurGAN-v2 [18] (h) CDVD-TSP [30] (i) STFAN [56] (j) Ours

Figure 7. Qualitative evaluations on our 4KRD real test datasets. Our proposed UHDVD model generates much clearer results in both detail and full image. (**Zoom in for best view**)

(a) Input (GoPro) (b) w/o (26.27/0.783) (c) w/ (27.64/0.824) (d) GT (+∞/1) (a) Input (REDS) (b) w/o (27.44/0.787) (c) w/ (29.48/0.860) (d) GT (+∞/1)

(a) Input (DVD) (b) w/o (25.65/0.814) (c) w/ (27.22/0.836) (d) GT (+∞/1) (a) Input (4KRD) (b) w/o (26.17/0.871) (c) w/ (27.46/0.874) (d) GT (+∞/1)

Figure 8. Quantitative evaluations on different datasets with (w/) the whole RCSA module or not (w/o). (**Zoom in for best view**)

thetic dataset. However, we cannot directly capture 4K videos at high-frame-rates with smartphones since hardware configuration limitations. Therefore, we use the frame interpolation method [29] to interpolate the recorded 4K videos from 30/60 fps to 480 fps as like the scheme of [27]. Then we generate blurry frames by averaging a series of successive sharp frames. In addition to our 4K resolution dataset, we also use three public deblurring datasets of GoPro [26], DVD [36], and REDS [25] to test our UHDVD model. Specially, since the test ground truth is not available for the REDS [25] dataset, we select the validation set as our test data.

### 4.3. Performance Comparisons

In this section, we compared our UHDVD method with the state-of-the-art video deblurring methods of [56, 36, 44] and image deblurring approaches of [18, 26, 41, 50]. We evaluate these methods by three criteria: PSNR, SSIM, and average runtime of an image on each dataset. All these methods are tested in the same server environment.

**Quantitative Evaluation.** Table 1 shows that our proposed method performs favorably against the state-of-the-art algorithms on the four datasets. The run time of all the methods reported in this table is based on the same test environment. On the DVD benchmark [36] and our 4KRD dataset, our algorithm obtain the best results in terms of PSNR and SSIM, while on the GoPro [26] and REDS [25] datasets, we are also the suboptimal method. Although EDVR [44] achieves the best results on the REDS dataset, we note that this method uses all the validation videos of REDS to train their model. In addition, since CDVD-TSP [30] explicitly uses temporal information of multiple frames, this method exceeds us on the GoPro dataset. However, our algorithm is 300× faster than this approach. Figure 5 shows three visual examples from the GoPro [26], DVD [36] and REDS [25] datasets, respectively. Figure 6 gives two examples of the 4KRD dataset. Our method achieved better results on the visual effects.

**Qualitative Evaluation.** To further validate the generalization ability of our network, we also qualitatively compare the proposed network with other algorithms on real blurry frames on our 4K real videos. As illustrated in Figure 7, the proposed method can restore clearer frames with more details than other methods. These comparison results show

Table 2. The PSNR and Time of an image for UHDVD with (w/) Separable-Patch Acceleration Architecture (SPAA) or not (w/o).

| | 720×1280 | | 2160×3840 | |
|---|---|---|---|---|
| | PSNR | Time | PSNR | Time |
| w/o SPAA | 29.76 | 32.5 ms | 27.62 | 65.6 ms |
| w/ SPAA | 29.74 | 12.5 ms | 27.61 | 27.9 ms |

Table 3. The PSNR/Time of our proposed UHDVD model on different datasets with (w/) the whole RCSA module or not (w/o).

| Datasets | GoPro | DVD | REDS | 4KRD |
|---|---|---|---|---|
| w/o RCSA | 30.64/12.4 | 31.57/12.3 | 26.64/12.5 | 27.38/22.6 |
| w/ RCSA | 31.33/12.7 | 32.19/13.2 | 27.53/13.9 | 27.88/27.9 |

that our UHDVD method can robustly handle unknown real blurs in most scenes. For example, the kiosk lattice in our restored image contain sharper structures and details than the results generated by other approaches.

### 4.4. Effectiveness of Separable-Patch Architecture

To validate the effectiveness of Separable-Patch Acceleration Architecture (SPAA), we conduct experiments with 1000 blurry frames on 720p and 4K, respectively. We count the average time of each task with the millisecond for per image. The results are shown in Table 2. It is clearly shown that we have more than doubled the computing speed by using the proposed acceleration architecture, while the PSNRs are almost the same. These results demonstrate that our proposed separable-patch acceleration architecture is pivotal to increase the speed and enable 4K image deblurring in real-time. Although the proposed SPAA is simple in theory, and the multi-patch scheme used in the SPAA architecture is similar to the work [37, 50], the parallel process of several patches at the same time is SPAA's target and it has been proved that this can reduce operation time significantly.

### 4.5. Effectiveness of RCSA Module

To validate the effectiveness of RCSA module in our network, we trained a new model without RCSA module on 4KRD dataset. The baseline model only uses two layers of the residual block without any CAM and SAM. Except for this difference, everything else is exactly the same as the initial model. The quantitative results are shown in Table 3. It indicates that our UHDVD model achieves about 0.5 dB gain than the model without using RCSA in terms of PSNR. Meanwhile, the calculation speed of the two models is almost the same by using the separable-patch acceleration pipeline. Some visual results are shown in Figure 8. From these examples, we can see that the proposed UHDVD obtains better results in image details. The visual performance also demonstrate the effectiveness of the RCSA module.

### 4.6. Running Time

Our model (UHDVD) can process a 2160×3840 frame within 30 ms, which means our model supports real-time 4K video deblurring task at 35fps. DMPHN [50] has also tried to reach real-time deblurring on images of 720p resolution, but they achieve this by reducing the accuracy. From the quantitative results of DMPHN in Table 1, it can be observed that their high-speed version (without stack) yields lower PSNR than the one with stack in all test datasets. Additionally, their high-speed version still does not work to reach real-time on 4K resolution. Our model is 10× faster than the method of DMPHN-(1-2-4-8) on 4K resolution videos. Furthermore, our model also improves the operational efficiency on 720p and reaches the speed of 12.7 ms per frame. It should be pointed out that we follow the prototype in [49, 50], the time we considered is just the GPU process time, there are runtime overheads related to I/O operations, which is directly proportional to the size of deblurring images. So the real-time processing means GPU's real-time in the strict sense.

The following factors contribute to our real-time: i) multi-scale scheme reduces the input image size of the first three scales; ii) multi-patch and separable-patch acceleration architecture increases the speed of calculation; iii) relatively few network layers and parameter amounts.

## 5. Conclusion

In this paper, we proposed a 4K video real-time deblurring network by using an asymmetrical encoder-decoder architecture. We integrated multi-scale and multi-patch schemes in a unified framework to improve efficiency and accuracy at the same time. Different from other methods, we use the asymmetrical encoder-decoder structure to build our network with fewer convolution layers to save calculation cost. In addition, we proposed the RCSA module to further improve the efficiency and adopted the separable-patch acceleration architecture to reach the real-time processing speed at 35 fps on 4K resolution videos. To study the 4K deblurring, we created the first public 4K resolution video dataset. Quantitative and qualitative results show that our proposed method performs favorably against the relevant state-of-the-art deblurring methods on both synthetic and real-world datasets.

**Acknowledgments.** This work is supported by the National Key R&D Program of China under Grant 2019YFB1406500, Beijing Municipal Education Commission Cooperation Beijing Natural Science Foundation (No. KZ 201910005007), National Natural Science Foundation of China (No. 61802403, U1803264, 62025604), Beijing Nova Program (No. Z201100006820074), Youth Innovation Promotion Association CAS.

## References


[1] Leah Bar, Benjamin Berkels, Martin Rumpf, and Guillermo Sapiro. A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In *ICCV*, 2007. 1

[2] Liang Chen, Faming Fang, Tingting Wang, and Guixu Zhang. Blind image deblurring with local maximum gradient prior. In *CVPR*, 2019. 2

[3] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 3

[4] Sunghyun Cho, Yasuyuki Matsushita, and Seungyong Lee. Removing non-uniform motion blur from images. In *ICCV*, 2007. 1

[5] Weisheng Dong, Lei Zhang, Guangming Shi, and Xiaolin Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE TIP*, 20(7):1838–1857, 2011. 2

[6] Hongyun Gao, Xin Tao, Xiaoyong Shen, and Jiaya Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *CVPR*, 2019. 2, 4

[7] Dong Gong, Jie Yang, Lingqiao Liu, Yanning Zhang, Ian Reid, Chunhua Shen, Anton Van Den Hengel, and Qinfeng Shi. From motion blur to motion flow: a deep learning solution for removing heterogeneous motion blur. In *CVPR*, pages 2319–2328, 2017. 1, 5

[8] Zhe Hu and Ming-Hsuan Yang. Learning good regions to deblur images. *IJCV*, 115(3), 2015. 1

[9] Tae Hyun Kim, Byeongjoo Ahn, and Kyoung Mu Lee. Dynamic scene deblurring. In *ICCV*, pages 3160–3167, 2013. 1

[10] Tae Hyun Kim and Kyoung Mu Lee. Segmentation-free dynamic scene deblurring. In *CVPR*, pages 2766–2773, 2014. 1

[11] Hui Ji and Kang Wang. A two-stage approach to blind spatially-varying motion deblurring. In *CVPR*, 2012. 1

[12] Zhe Jiang, Yu Zhang, Dongqing Zou, Jimmy Ren, Jiancheng Lv, and Yebin Liu. Learning event-based motion deblurring. In *CVPR*, 2020. 2

[13] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 2

[14] Soo Ye Kim, Jihyong Oh, and Munchurl Kim. Deep sr-itm: Joint learning of super-resolution and inverse tone-mapping for 4k uhd hdr applications. In *ICCV*, 2019. 3

[15] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[16] Dilip Krishnan, Terence Tay, and Rob Fergus. Blind deconvolution using a normalized sparsity measure. In *CVPR*, 2011. 2

[17] Orest Kupyn, Volodymyr Budzan, Mykola Mykhailych, Dmytro Mishkin, and Jiří Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *CVPR*, pages 8183–8192, 2018. 2, 5

[18] Orest Kupyn, Tetiana Martyniuk, Junru Wu, and Zhangyang Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*, 2019. 2, 6, 7

[19] Wei-Sheng Lai, Jian-Jiun Ding, Yen-Yu Lin, and Yung-Yu Chuang. Blur kernel estimation using normalized color-line prior. In *CVPR*, 2015. 1

[20] Lerenhan Li, Jinshan Pan, Wei-Sheng Lai, Changxin Gao, Nong Sang, and Ming-Hsuan Yang. Learning a discriminative prior for blind image deblurring. In *CVPR*, 2018. 2

[21] Ding Liu, Bihan Wen, Yuchen Fan, Chen Change Loy, and Thomas S Huang. Non-local recurrent network for image restoration. In *NeurIPS*, pages 1673–1682, 2018. 4

[22] Ziwei Liu, Raymond A Yeh, Xiaoou Tang, Yiming Liu, and Aseem Agarwala. Video frame synthesis using deep voxel flow. In *ICCV*, pages 4463–4471, 2017. 2

[23] Xiaojiao Mao, Chunhua Shen, and Yu-Bin Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *NeurIPS*, pages 2802–2810, 2016. 3

[24] Tomer Michaeli and Michal Irani. Blind deblurring using internal patch recurrence. In *ECCV*, 2014. 1, 2

[25] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *CVPRW*, 2019. 2, 6, 7

[26] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891, 2017. 1, 2, 3, 5, 6, 7

[27] Seungjun Nah, Radu Timofte, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, and Kyoung Mu Lee. Ntire 2019 challenge on video deblurring: Methods and results. In *CVPRW*, 2019. 1, 5, 7

[28] Yuesong Nan, Yuhui Quan, and Hui Ji. Variational-em-based deep learning for noise-blind image deblurring. In *CVPR*, 2020. 2

[29] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *CVPR*, pages 670–679, 2017. 7

[30] Jinshan Pan, Haoran Bai, and Jinhui Tang. Cascaded deep video deblurring using temporal sharpness prior. In *CVPR*, pages 3043–3051, 2020. 1, 2, 5, 6, 7

[31] Niki Parmar, Ashish Vaswani, Jakob Uszkoreit, Łukasz Kaiser, Noam Shazeer, Alexander Ku, and Dustin Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018. 4

[32] Daniele Perrone and Paolo Favaro. Total variation blind deconvolution: The devil is in the details. In *CVPR*, 2014. 2

[33] Wenqi Ren, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *ICCV*, pages 1077–1085, 2017. 1

[34] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE TPAMI*, 38(7):1439–1451, 2015. 1

[35] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *ACM TOG*, 27(3):1–10, 2008. 1

[36] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *CVPR*, pages 1279–1288, 2017. 2, 3, 6, 7

[37] Maitreya Suin, Kuldeep Purohit, and AN Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *CVPR*, pages 3606–3615, 2020. 2, 4, 8

[38] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *CVPR*, pages 769–777, 2015. 1

[39] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays. Edge-based blur kernel estimation using patch priors. In *ICCP*, 2013. 1, 2

[40] Xin Tao, Hongyun Gao, Renjie Liao, Jue Wang, and Jiaya Jia. Detail-revealing deep video super-resolution. In *ICCV*, pages 4472–4480, 2017. 4

[41] Xin Tao, Hongyun Gao, Xiaoyong Shen, Jue Wang, and Jiaya Jia. Scale-recurrent network for deep image deblurring. In *CVPR*, pages 8174–8182, 2018. 2, 3, 6, 7

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017. 4

[43] Panqu Wang, Pengfei Chen, Ye Yuan, Ding Liu, Zehua Huang, Xiaodi Hou, and Garrison Cottrell. Understanding convolution for semantic segmentation. In *WACV*, pages 1451–1460, 2018. 3

[44] Xintao Wang, Kelvin CK Chan, Ke Yu, Chao Dong, and Chen Change Loy. Edvr: Video restoration with enhanced deformable convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 6, 7

[45] Jonas Wulff and Michael Julian Black. Modeling blurred video with layers. In *ECCV*, pages 236–252, 2014. 1

[46] Li Xu, Shicheng Zheng, and Jiaya Jia. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*, 2013. 2

[47] Fisher Yu and Vladlen Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511.07122*, 2015. 3

[48] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. *arXiv preprint arXiv:2102.02808*, 2021. 2

[49] Hui Zeng, Jianrui Cai, Lida Li, Zisheng Cao, and Lei Zhang. Learning image-adaptive 3d lookup tables for high performance photo enhancement in real-time. *IEEE TPAMI*, 2020. 3, 8

[50] Hongguang Zhang, Yuchao Dai, Hongdong Li, and Piotr Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*, pages 5978–5986, 2019. 1, 2, 3, 6, 7, 8

[51] Jiawei Zhang, Jinshan Pan, Jimmy Ren, Yibing Song, Linchao Bao, Rynson WH Lau, and Ming-Hsuan Yang. Dynamic scene deblurring using spatially variant recurrent neural networks. In *CVPR*, 2018. 2

[52] Kaihao Zhang, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, and Hongdong Li. Adversarial spatio-temporal learning for video deblurring. *IEEE Transactions on Image Processing*, 28(1):291–301, 2018. 1

[53] 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. 1

[54] Kai Zhang, Wangmeng Zuo, and Lei Zhang. Deep plug-and-play super-resolution for arbitrary blur kernels. In *CVPR*, 2019. 1

[55] Zunjin Zhao, Bangshu Xiong, Shan Gai, and Lei Wang. Improved deep multi-patch hierarchical network with nested module for dynamic scene deblurring. *IEEE Access*, 8:62116–62126, 2020. 4

[56] Shangchen Zhou, Jiawei Zhang, Jinshan Pan, Haozhe Xie, Wangmeng Zuo, and Jimmy Ren. Spatio-temporal filter adaptive network for video deblurring. In *ICCV*, pages 2482–2491, 2019. 1, 5, 6, 7

[57] Yipin Zhou and Nikos Komodakis. A map-estimation framework for blind deblurring using high-level edge priors. In *ECCV*, 2014. 1