A Deep Motion Deblurring Network based on Per-Pixel Adaptive Kernels with Residual Down-Up and Up-Down Modules

Hyeonjun Sim Munchurl Kim
School of EE, Korea Advanced Institute of Science and Technology (KAIST)
Daejeon, Korea
{flhy5836,mkimee}@kaist.ac.kr

Abstract

Due to the object motion during the camera exposure time, latent pixel information appears scattered in a blurred image. A large dataset of dynamic motion blur and blur-free frame pairs enables deep neural networks to learn deblurring operations directly in end-to-end manners. In this paper, we propose a novel motion deblurring kernel learning network that predicts the per-pixel deblur kernel and a residual image. The learned deblur kernel filters and linearly combines neighboring pixels to restore the clean pixels in its corresponding location. The per-pixel adaptive convolution with the learned deblur kernel can effectively handle non-uniform blur. At the same time, the generated residual image is added to the adaptive convolution result to compensate for the limited receptive field of the learned deblur kernel. That is, the adaptive convolution and the residual image play different but complementary roles each other to reconstruct the latent clean images in a collaborative manner. We also propose residual down-up (RDU) and residual up-down (RUD) blocks that help improve the motion deblurring performance. The RDU and RUD blocks are designed to adjust the spatial size and the number of channels of the intermediate feature within the blocks. We demonstrate the effectiveness of our motion deblurring kernel learning network by showing intensive experimental results compared to those of the state-of-the-art methods.

1. Introduction

An image enhancement aims to restore the latent image of the observed image with degradation. Unlike the high-level vision task which requires annotation efforts from human resource, relatively abundant data in a manageable image format are available in low-level image enhancement task.

Nevertheless, for some self-supervised tasks such as super-resolution, frame interpolation, it is easy to obtain training data, but relatively difficult to have clean-hazy image pairs and clean-blurry image pairs in dehazing and deblurring problems, respectively. Therefore, the availability of several large datasets such as O-HAZE, I-HAZE [3, 2, 1], GOPRO deblurring dataset [14], and REDS [13] dataset have encouraged the researchers to study data-driven methodologies for dehazing and motion deblurring problems.

Figure 1. Examples of deblurring results on GOPRO dataset [14]: (a) Input blurry image; (b) Result of Tao *et al*. [27]; (c) Result of our proposed network; (d) Clean image.

This paper studies an end-to-end deblurring method given multiple blur and blur-free frame pairs. In this study, we challenge deblurring problems for the images with non-uniform blurs synthesized by the accumulation of multiple consecutive sharp frames. The accumulation process approximates that a fast-moving object appears over several pixels during the camera exposure time. So the amount of motion blur varies depending on the pixel locations due

to the speeds, depths and motion types of moving objects and camera shakes. To handle non-uniform blur, we propose a deblurring kernel learning network that learns input-adaptive per-pixel deblur kernels at each pixel location for the input blurry frames. Our kernel learning network is trained to restore each clean pixel by combining the corresponding surrounding pixels in the blurry images. The adaptive convolution of the blurry frame and the generated *de*blur kernels act as blind deconvolution [29, 31] without explicit estimation of blur kernels. This process collects scattered latent pixel information and filters the disturbing information.

Sometimes the movement of an object is so large that the pixels are scattered too far apart and most of the information is lost in the blurred image. In this case, it may be better to create the pixels directly rather than combining the surrounding pixels. This can be achieved by training deep neural networks to predict the RGB pixels directly. Hence, we incorporate a direct pixel estimation branch and combine it with an adaptive convolution branch in our network. The two branches share most of the parameters through the mainstream which is then separated into two branches with several convolution layers. This enables both the adaptive convolution and the direct pixel estimation in a collaborative manner to produce the reliable final output. Fig. 1 shows the motion deblurred results in comparison between our proposed network and Tao *et al*. [27].

Our deep motion deblurring network is based on the U-Net [21] structure where the convolution layers in the encoder and decoder parts are replaced with novel residual down-up (RDU) blocks and residual up-down (RUD) blocks, respectively. The RDU blocks and RUD blocks inherit the properties of the encoder and decoder structures, respectively, as simplified building blocks of two layers. Each RDU block halves the spatial size of its input feature and doubles the number of channels through the first convolution layer, and yields its output back to the original spatial size and channel number through the second layer for residual learning. On the other hand, each RUD block doubles the spatial size of its input and reduces the number of channels through the first transposed convolution layer, and returns the intermediate feature back to the original shape. Peng *et al*. [20] used the RDU blocks without increased channels for a classification network to speed up while maintaining the classification performance. Besides, Yu *et al*. [30] reported that even if the total number of parameter is the same, increasing and decreasing the intermediate feature channels in a block helps improve performance in super-resolution. Our RDU block is basically a combination of the two methods [20, 30]. Additionally, we extended the idea of the RDU block to the RUD block for improved restoration performance.

We demonstrate the effects of our proposed network with the adaptive convolution and the residual RDU and RUD blocks in Experiment Section 4.

2. Related Works

2.1. Deblurring Approaches

The image blur is generally defined as follows :

$$B = L \otimes K_b + N, \tag{1}$$

where B represents blurred patch, L latent patch, K_b blur kernel, N additive observation noise.

Some previous methods predict the underlying blur kernels and then deconvolve the kernels with the blurred image to generate latent image satisfying Eq. 1 and additional regularization or image prior information [23, 5]. Some researchers have tried to estimate blur kernels using deep neural networks in uniform [22] or non-uniform setting [26]. Xu *et al*. [29] developed a convolutional neural network (CNN) based method that serves as the deconvolution. They analyzed that convolutional filters should have large-sized receptive fields to reliably approximate the deconvolution.

Thanks to the availability of high-frame-rate cameras, it becomes easier to have many clean images and to generate synthesized motion blurred images [14, 25]. As a result, end-to-end deblurring methods to predict a blur-free image without kernel prediction have been developed by learning the mapping functions from blur to blur-free images. Nah *et al*. [14] proposed a coarse-to-fine deblurring network consisting of residual blocks [11] using multi-scale blur inputs. Tao *et al*. [27] improved Nah's work [14] with encoder-decoder residual blocks. Their network shares the parameters across the scales to reduce the model size and stabilize the training. Zhang *et al*. [31] analyzed that recurrent neural networks (RNN) can be trained for the deconvolution [29] in a feature domain. They incorporated a sub-network that predicts the spatially variant per-pixel kernels of the RNNs. The blur input is deblurred gradually as it passes through the RNNs. In our case, the generated per-pixel kernel is applied directly to the input blurry image and complemented by adding the residual image. The deblurring network proposed by Chen *et al*. [4] first predicts three blur-free frames from three blurred frames. Then bidirectional optical flows are computed by a pre-trained network to approximate a blur kernel with the optical flows. The approximated blur kernel reblurs the predicted sharp frame to reconstruct blurry input again. The cycle consistency loss between the blur input and the reblurred image is additionally used to improve the deblurring performance.

2.2. Adaptive Convolution

Jia *et al*. [9] introduced a novel network that output a single filter or location-specific filters. The generated single fil-

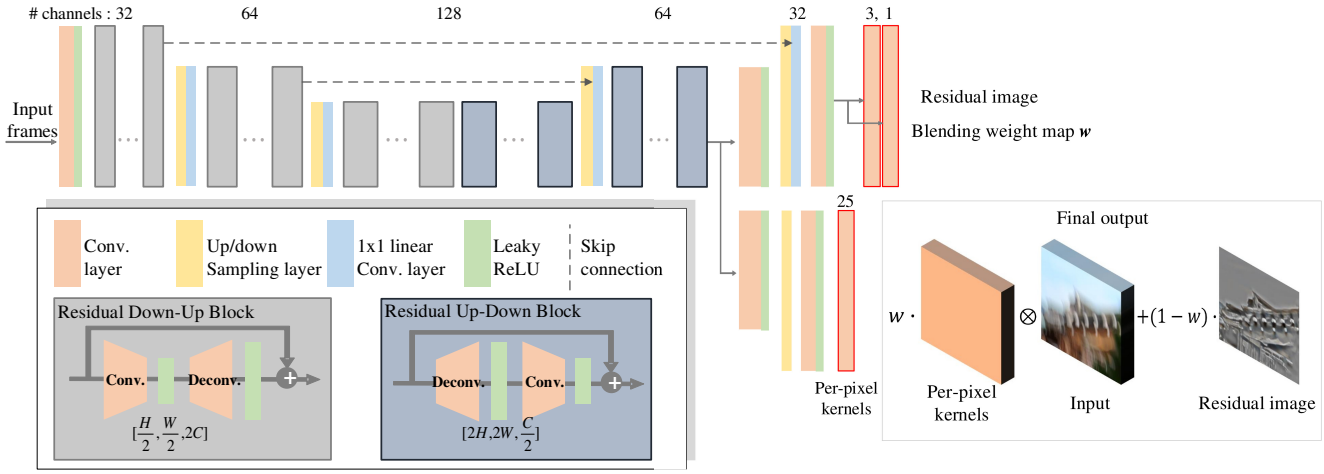

Figure 2. The architecture of the proposed motion deblurring network. The encoder and decoder parts consist of our proposed RDU and RUD blocks, respectively. The encoder of the network takes as input the three consecutive frames concatenated along channel axis where the middle frame is the target frame, and the decoder outputs 25-channel per-pixel kernels, a residual image and a blending weight map. The target frame is locally filtered by the generated kernels. The final output is the weighted sum of the locally convolved output and the residual image defined by Eq. 2.

ter is applied to an image and the generated location-specific filters (or dynamic filter) are applied to pixel locations. The location-specific filtering is also studied in frame interpolation by Niklaus *et al*. [16, 17]. In [16], a latent intermediate frame is predicted from adjacent frames. Their network produces spatially-adaptive filters and the filters are applied to both of the previous and next frames. Nikalus *et al*. [17] replace the square-shape dynamic filters with two separable 1-dimensional filters. By doing so, they can enlarge the receptive fields of the kernels since the kernel sizes could be reduced from N^2 to $2N$ when the receptive field size is N. In video super-resolution task, Jo *et al*. [10] achieved superior results using the adaptive convolution and the residual images. For each location, 16 kernels are predicted for an up-scaling factor of 4. They designed their network to share the parameters until two branches are separated. The concept of their work is somewhat similar to ours in the sense that adaptive kernel learning, residual learning and parameter sharing, but their work solves video super-resolution based on a densely-connected CNN structure while our network is based on a U-NET with RDU and RUD blocks to solve the motion deblurring problems.

3. Proposed Methods

Our deblurring network takes a target frame and its previous and next frames as a concatenated input. Then the network generates three outputs with per-pixel 5×5-sized (25-length in the channel depth) adaptive kernels, a residual image of three channels and a blending weight map. The generated kernel at each location *locally* filters the target frame with its corresponding 5×5 windows. This local filtering is also referred to as adaptive convolution [16, 17] or dynamic filtering [9, 10]. However, this operation is often denoted as a dot product (element-wise multiplication followed by summation) since the convolution kernel is not shared entirely. The second output is a 3-channel image called a *residual image* [10], since it is added to the locally filtered output. The third output, blending weight map, has a single channel and determines how to blend two preceding outputs. Therefore, the final output can be represented as follows:

$$\hat{L} = w \cdot B \otimes K_d + (1 - w) \cdot R, \tag{2}$$

where B represents the input blurred image, K_d the per-pixel kernel, R the residual image, w the blending weight map, $\otimes$ the adaptive convolution.

3.1. APP-K Branch for small/medium Motion Blur and BWM-R Branch for Large Motion Blur

As shown in Fig. 2, our motion deblurring network has two branches in its decoder part: one is an adaptive per-pixel kernel based motion deblurring (APP-K) branch and the other is a blending weight map and residual image estimation (BWM-R) branch. Since the APP-K branch learns adaptive per-pixel kernels of 5×5 size, it is more capable of deblurring small and medium motion blur images and is less effective for large motion blur. On the other hand, the BWM-R branch learns both the residual images and blending weight map with large receptive fields, which is beneficial for large motion deblurring.

For the APP-K branch, the values of each 5×5-sized adaptive per-pixel deblur kernel are linearly combined with

a corresponding input blurry pixel and its square neighbor pixels via the dot product operation to generate a clean pixel. Due to the one-to-one correspondence between the kernels and the pixel locations, the motion of individual pixel can be handled appropriately. The deblur kernel filters only the information that is important in restoring the original clean pixel and excludes other noisy pixels. The output pixel lies in a range of the surrounding pixel values within a 5×5-sized region. Therefore, this adaptive convolution based on the estimated per-pixel deblur kernels is less effective for large motion blur since the pixel information of a clean image is heavily scattered or spread. In this case, the BWM-R branch compensates the coarse output of the APP-K branch by adding the residual images by the blending weight maps. This is because the residual images can be robustly created directly from the feature maps, which is beneficial for large motion blur. It should be noted that the added BWM-R branch focuses on the restoration of sharp edges that are severely blurred due to large motion rather than smooth image structures.

3.2. Residual Down-Up and Up-Down Blocks

Our motion deblurring network is designed based on building blocks, denoted as residual down-up (RDU) blocks and residual up-down (RUD) blocks. We change the spatial size and the number of channels of the intermediate feature in the RDU and RUD blocks. As shown in Fig. 2, our motion deblurring network is based on the U-Net structure with two parts: encoder and decoder. The encoder extracts the feature of the input, and the decoder plays the reconstruction process. For the low-level features close to the input images, their neighboring pixels tend to be highly correlated with each other with much redundant information. Therefore, in general classification [24, 7] or auto-encoder based regression models [18], the spatial sizes of feature maps are reduced while the numbers of feature map channels are increased in order to extract more various feature information [28]. The U-Net architecture [21, 8, 12, 6] adds skip connections between the encoder and the decoder to prevent loss of information caused by reducing spatial size. By taking the advantage of the U-Net architecture, we design a RDU block having two back-to-back layers of convolution and transposed convolution as a base block of the encoder. In the RDU block, an input tensor is fed into the convolution layer with stride 2 followed by the transposed convolution layer with stride 2. The convolution layer halves the spatial size while doubling the number of channels of the input tensor. The transposed convolution layer, reversely, doubles the spatial size and reduce the number of channels back to the original size. At the end of the block, input tensor is added for the purpose of residual learning to preserve detail information.

The proposed RUD block is also designed as the base block of the decoder by having the two back-to-back layers of transposed convolution and convolution. The decoder should restore a clean image by increasing a spatial size gradually from bottleneck feature of the network. In the RUD block, the transposed convolution layer with stride 2 enlarges the spatial size of input but reduces the number of channels to alleviate the computational burden. The convolution layer with stride 2 returns the tensor back to the original size and the identity addition is performed. Since the convolution is operated in the zoomed-in feature maps, fine reconstruction can be possible.

Table 1. Configuration of our proposed network in Fig. 2. For each 'RDU' or 'RUD' block, the number in parenthesis indicates the repetition number of the same blocks in a cascade. '*' indicates concatenation of input tensors along the channel axis. The size of each estimated per-pixel kernel is 5×5, so 'conv_8' layer has 25 channels.

Input			Output		
Layer	H,W	C	Layer	H,W	C
Input	1	9	conv_1	1	32
conv_1	1	32	RDU_1 ($\times9$)	1	32
RDU_1	1	32	avg_pool_1	1/2	32
avg_pool_1	1/2	32	$1\times1\text{conv}_1$	1/2	64
$1\times1\text{conv}_1$	1/2	64	RDU_2 ($\times9$)	1/2	64
RDU_2	1/2	64	avg_pool_2	1/4	64
avg_pool_2	1/4	64	$1\times1\text{conv}_2$	1/4	128
$1\times1\text{conv}_2$	1/4	128	RDU_3 ($\times4$)	1/4	128
RDU_3	1/4	128	RUD_1 ($\times4$)	1/4	128
RUD_1	1/4	128	NN_1	1/2	128
NN_1, RDU_2	1/2	192*	$1\times1\text{conv}_3$	1/2	64
$1\times1\text{conv}_3$	1/2	64	RUD_2 ($\times9$)	1/2	64
RUD_2	1/2	64	conv_2	1/2	64
conv_2	1/2	64	NN_2	1	64
NN_2, RDU_1	1	96*	$1\times1\text{conv}_4$	1	32
$1\times1\text{conv}_4$	1	32	conv_3	1	32
conv_3	1	32	conv_4	1	3
conv_3	1	32	conv_5	1	1
RUD_2	1/2	64	conv_6	1/2	64
conv_6	1/2	64	NN_3	1	64
NN_3	1	64	conv_7	1	32
conv_7	1	32	conv_8	1	25

3.3. Proposed Network Architecture

Fig. 2 shows the overall architecture of our motion deblurring network. Table 1 shows the detailed configuration of our motion deblurring network. The concept of our network is somewhat similar to the structure of Jo *et al*. [10] in the sense that both the per-pixel kernels and the residual images are generated to construct the final output. Our network is based on the U-Net structure to reduce the input image size by a factor of 4 similar to Tao *et al*. [27]. The final output is the weighted sum of adaptive convolution output and the residual image (Eq. 2), and L_1 loss function is

Table 2. Results of ablation study on our proposed motion deblurring network.

Variations	Model 1	Model 2	Model 3	Model 4	Model 5	Model 6	Model 7
RDU and RUD block				✓	✓	✓	✓
BWM-R		✓	✓		✓	✓	✓
APP-K	✓		✓	✓		✓	✓
Multi-frame	✓	✓	✓	✓	✓	✓	
PSNR(dB)	31.03	31.24	31.33	31.36	31.43	**31.86**	31.55
Runtime(s)	0.86	0.76	0.95	0.63	**0.43**	0.59	0.62

applied to the final output only. Hence the parameters are shared along the mainstream until two branches are separated as shown in Fig. 2. Since the kernel has a different characteristic compared to images, the skip connection is only connected to the BWM-R branch. Leaky ReLU is used as an activation function except for the output layers in our network. Nearest neighbor and average pooling are selected as up- and down-sampling methods, respectively, between different scale levels. All convolutional layers have 5×5 filters and all transposed convolution layers have 4×4 to avoid the checkerboard artifact [19].

4. Experiments

4.1. Dataset

We trained our network on REDS [13] training dataset, which is provided by NTIRE2019 Video Deblurring Challenge [15]. It consists of blur and blur-free frame pairs of size 720×1280. Blurry images are synthesized from multiple sharp frames captured by high-frame-rate cameras. There are 240 video sequences, 100 frames per each video sequence, totally 24,000 blur and blur-free frame pairs in the training set. The REDS validation set contains 30 video sequences, total 3,000 image pairs. But we evaluate the motion deblurring methods on every 10 deblurred frames of each sequence (frame number 09,19,...) as the development phase of the NTIRE2019 Video Deblurring Challenge. The quantitative and qualitative results on the partial of REDS validation dataset are reported in Section 4.3 and 4.4.

4.2. Training Details

In order to provide useful information of adjacent frames in learning the motion deblurring, the previous and next frames are concatenated to the target frame along the channel axis. The previous and next frames are shifted up to 2 pixels to the opposite directions respectively for the purpose of robust training. This augmentation strategy gives a small amount of random motion to stationary scenes. Training patches are cropped to a 160×160 size and then are randomly flipped and rotated. Batch size is set to 4, and the initial learning rate is set to 1e-4 with a linear decrease to zero after 10% of training. We trained the network for about 1.5 million iterations. A L1 loss and the Adam optimizer are selected as training options. Implementation was done with tensorflow library. Training and testing are operated on the machine with intel core i7-7700k@4.20GHz CPU, NVIDIA TITAN Xp GPU. The training takes about 3 days.

4.3. NTIRE 2019 Video Deblurring Challenge

Our full model trained with entire training set has yielded average 33.38dB PSNR for 300 validation frames without using the simple geometric self-ensemble [11] which can further improve the PSNR performance by augmenting the input frames to 8 combinations (2 flips, 4 rotations). The augmented 8 combinations are fed into the network, aligned back to the original shape, respectively. Then we take the mean of the 8 combinations. The simple ensemble has increased the average PSNR value to 33.86dB (+0.48dB). This indicates that there is room for the performance improvement with longer training iterations. Fig. 5 shows some examples of deblurring results for the REDS validation images. Our proposed network model was ranked 3rd in the NTIRE2019 Video Deblurring Challenge [15] for the REDS test data.

4.4. Ablation Study

To compare the effectiveness of our proposed method, we performed ablation study. Due to limited time and a computation resource for the experiments of ablation study, we used 128×128-size training patches and utilized only the first 20% of each video sequence of the REDS training dataset. The numbers of training iterations were also limited to the 20% of the total iteration number to train our full model in Section 4.3. Table 2 summarizes the ablation study results. If the 'RDU and RUD block' is unchecked in Table 2, the RDU and RUD blocks are replaced with general residual blocks which consist of two convolution layers of fixed spatial size and the number of channels. When the 'BWM-R' is used alone without the APP-K, the network directly predict clean images without APP-K and the blending maps. If 'Multi-frame' is checked, the input is constituted with a concatenation of three consecutive frames along the channel axis where the middle frame is the target frame. If not, only the target frame is fed into the networks. All the seven variations in this ablation study are adjusted to have a

(a) (b) (c) (d) (e) (f)

Figure 3. Subjective comparison of motion deblurring examples for the ablation study on two models (Model 4 and Model 6 in Table 2) : (a) Input blur image; (b) Output of Model 4 that only contains adaptive convolution (APP-K branch) without the residual images (BWM-R branch); (c) Adaptive convolution output of Model 6 that has both APP-K and BWM-R branches; (d) Residual image output of Model 6; (e) Final output of Model 6 according to Eq. 2; and (f) Ground truth clean image.

(a) (b) (c) (d)

Figure 4. Comparison of Model 2 and Model 6 in Table 2 : (a) Input blur image; (b) Output of Model 2; (c) Output of Model 6; and (d) Ground truth clean image.

similar number of parameters for fair comparison. Our proposed network with all components (Model 6 in Table 2) has achieved the best PSNR performance among the seven models for the validation set. In addition, the multiple-frame input improves PSNR performance over the single frame input. We define the baseline model that uses the general residual blocks as its base building block and directly estimates the latent images without the APP-K, which is denoted as Model 2 in Table 2. Fig. 4 compares the results of the baseline (Model 2) and our proposed model (Model 6). It can be noted in Fig. 4 that Model 6 can restore the severely blurred edges and strips well, yielding sharp edges and thin shoulder straps.

Adaptive Convolution As shown in Table 2, Model 1 and Model 4 yield relatively lower PSNR performance compared to Model 2 and Model 5, respectively, due to a limited kernel size. Fig. 3-(b) shows the results of Model 4. As shown in the middle picture of Fig. 3-(b), Model 4 fails to restore the left-side tree whose pixel information is heavily scattered over a large area due to fast motion. This implies that the motion deblurring task should be assisted by hallucinating the pixels from scratch.

Fig. 3-(c) shows the results of the APP-K branch output of Model 6. The APP-K branch of Model 6 only removed a smaller amount of motion blur shown in Fig. 3-(c), compared to the results of Model 4 shown in Fig. 3-(b). This is because, the APP-K branch of Model 6 performs a coarse prediction and help the BWM-R branch of Model 6 focus on restoring the details (edges) of the clean images. From a perspective of subjective quality, the final output of Model 6 yields superior results over Model 4, as shown in Fig. 3-(e). Also, from a perspective of the overall PSNR performance, Model 6 outperforms Model 4 and Model 5 that directly estimates the clean images, as shown in Table 2.

To further analyze the effectiveness of the per-pixel adaptive kernels, we have applied adaptive separable convolution [17] for Model 6 by replacing 5×5 kernels with 1-dimensional horizontal and vertical kernels of length 13, which is denoted as Model 6-s. Form the experiments, Model 6-s results in average 31.77dB PSNR, which is slightly lower than that of Model 6. Although the usage of

(a) (b) (c)

Figure 5. Examples of the deblurring results by our proposed full model that was trained with the entire training dataset, called REDS dataset [13], used for the NTIRE2019 Video Deblurring Challenge: (a) Input blur image; (b) Motion-deblurred output of our full model; and (c) Ground truth clean image.

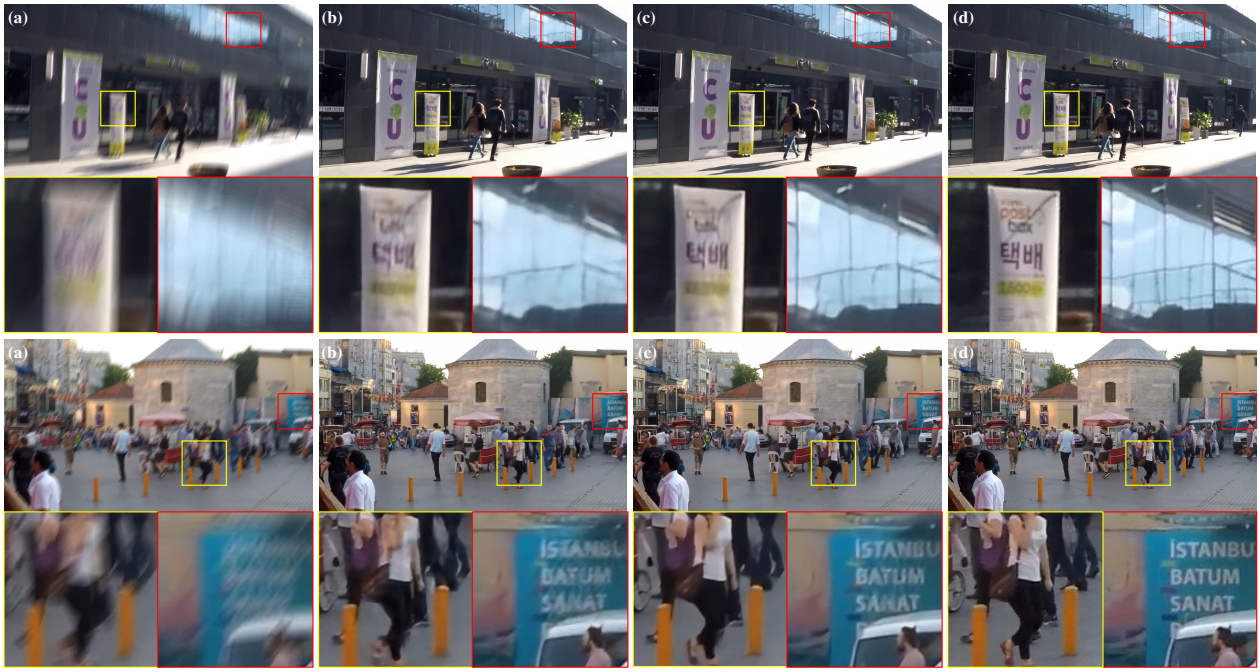

Figure 6. Comparison of motion deblurred results of our proposed full model and Tao *et al*. [27] for a benchmark dataset: (a) Input blur image; (b) Results of Tao *et al*. [27]; (c) Results of our network, and (d) Ground truth clean image.

Table 3. Quantitative performance comparison for different combinations of general residual, RDU and RUD blocks in the encoder and decoder of our proposed motion deblurring network.

Encoder	R	RDU	RUD	RUD	RDU
Decoder	R	RDU	RUD	RDU	RUD
PSNR	31.27	31.61	31.64	31.61	**31.86**

the 1-dimensional horizontal and vertical kernels of length 13 has increased the receptive field (RF) to 13×13, the effect of the enlarged RF is not significant since the BWM-R branch already enjoys a large RF in generating the residual images of Model 6. Moreover, the 5×5 kernel has a smaller range but can deal with neighboring pixels more densely than the 1-d kernels.

Residual Down-Up and Up-Down Blocks The PSNR result of Table 2 shows that proposed RDU and RUD blocks are effective compared to the residual blocks in terms of efficiency and performance, *i.e*., Model 4–6 yield higher PNSR values and smaller run-time than Model 1–3, respectively. It is worthwhile to mention that the efficient structures of the RDU and RUD blocks can compensate for the increased computational burden by the APP-K branch. Consequently, Model 6 (0.59 second per 720×1280) runs faster than our baseline, Model 2 (0.76 second per 720×1280).

For more detailed analysis, the usage of RDU and RUD blocks has been inspected in a perspective of their different combinations in the encoder and decoder of Model 6. Table 3 shows the PSNR performance for variations of Model 6. It should be noted that the numbers of parameters are not kept the same for different combinations of RDU and RUD blocks due to their structural differences. In Table 3, the k-th variation model of Model 6 is denoted as VM k. VM1 has the general residual blocks (denoted as 'R') as it building blocks in both encoder and decoder. VM5 that has the RDU blocks ('RDU') in the encoder and the RUD blocks ('RUD') in the decoder is equivalent to Model 6. The VM2, VM3, VM4 and VM5 yield higher PSNR values than VM1. From the comparison between VM4 and VM5 in Table 3, the RDU blocks are suitable for the encoder to extract more features and to reduce spatial redundancy while the RUD blocks are appropriate for the decoder to restore details.

4.5. Evaluation on Benchmark Dataset

To evaluate our network on a benchmark dataset, we trained our model with GOPRO dataset provided by Nah *et al*. [14]. The training dataset contains 720×1280-sized 2,103 blur and blur-free frame pairs. We used the blurred images that are not gamma corrected. Our proposed network is compared with the state-of-the-art methods, Nah *et al*. [14], Zhang *et al*. [31] and Tao *et al*. [27]. For a fair comparison, only a single frame without the adjacent frames is input to our proposed network in both the training and testing phases. Also, we did not conduct the geometric self-ensemble described in Section 4.3. The size of training patched is set to 128×128 and the training for our proposed network has taken about 3 days. The other training options are the same as those in Section 4.2. Table 4 lists the average PSNR and SSIM values of our proposed network and the three state-of-the-art methods for the 1,111 GOPRO test images. As shown in Table 4, our proposed network exhibited the best PSNR and SSIM performances, compared to the three state-of-the-art methods. Fig. 6 shows some deblurring results of GOPRO test images for subjective comparison. As shown in Fig. 6, our network produces much cleaner images, especially for heavily motion-blurred images, than the method of Tao *et al*. [27].

Table 4. PSNR/SSIM and run-time comparisons of our network and the three state-of-the-art methods for the GOPRO dataset [14].

Method	PSNR(dB)	SSIM	runtime(s)
Nah *et al*. [14]	28.62	0.9094	N/A
Zhang *et al*. [31]	29.19	0.9306	N/A
Tao *et al*. [27]	30.26	0.9342	**0.32**
Ours-shallow	**31.34**	**0.9474**	0.62

5. Conclusion

We proposed a deep motion deblurring network with novel base blocks, residual down-up (RDU) and residual up-down (RUD) blocks. Furthermore, our network is featured with an adaptive per-pixel kernel (APP-K) module to restore image details for small/medium motion blur images and with a residual image estimation with blending weight map generation (BWM-R) module to precisely restore the latent sharp edges from heavily motion-blurred images. Our deep motion deblurring network can effectively combine the adaptive convolution results and the residual images. The replacement of the residual building blocks with the RDU and RUD blocks in the encoder and decoder of an U-Net architecture allows for PSNR performance. From experiments, it is shown that our deep motion deblurring network outperformed the state-of-the-art methods in PSNR and SSIM perspectives for the benchmark dataset. Our proposed motion deblurring network has ranked the 3rd in the NTIRE2019 Video Deblurring Challenge.

Acknowledgement

This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. 2017-0-00419, Intelligent High Realistic Visual Processing for Smart Broadcasting Media).

References

[1] Cosmin Ancuti, Codruta O Ancuti, and Radu Timofte. Ntire 2018 challenge on image dehazing: Methods and results. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 891–901, 2018. 1

[2] Cosmin Ancuti, Codruta O Ancuti, Radu Timofte, and Christophe De Vleeschouwer. I-haze: a dehazing benchmark with real hazy and haze-free indoor images. In *International Conference on Advanced Concepts for Intelligent Vision Systems*, pages 620–631. Springer, 2018. 1

[3] Codruta O Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 754–762, 2018. 1

[4] Huaijin Chen, Jinwei Gu, Orazio Gallo, Ming-Yu Liu, Ashok Veeraraghavan, and Jan Kautz. Reblur2deblur: Deblurring videos via self-supervised learning. In *2018 IEEE International Conference on Computational Photography (ICCP)*, pages 1–9. IEEE, 2018. 2

[5] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. *ACM Transactions on graphics (TOG)*, 28(5):145, 2009. 2

[6] Alexey Dosovitskiy, Philipp Fischer, Eddy Ilg, Philip Hausser, Caner Hazirbas, Vladimir Golkov, Patrick Van Der Smagt, Daniel Cremers, and Thomas Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015. 4

[7] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4

[8] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 4

[9] Xu Jia, Bert De Brabandere, Tinne Tuytelaars, and Luc V Gool. Dynamic filter networks. In *Advances in Neural Information Processing Systems*, pages 667–675, 2016. 2, 3

[10] Younghyun Jo, Seoung Wug Oh, Jaeyeon Kang, and Seon Joo Kim. Deep video super-resolution network using dynamic upsampling filters without explicit motion compensation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3224–3232, 2018. 3, 4

[11] Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 136–144, 2017. 2, 5

[12] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015. 4

[13] Seungjun Nah, Sungyong Baik, Seokil Hong, Gyeongsik Moon, Sanghyun Son, Radu Timofte, and Kyoung Mu Lee. Ntire 2019 challenges on video deblurring and super-resolution: Dataset and study. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1, 5, 7

[14] 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, 8

[15] Seungjun Nah, Radu Timofte, et al. Ntire 2019 challenge on video deblurring: Methods and results. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 5

[16] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive convolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 670–679, 2017. 3

[17] Simon Niklaus, Long Mai, and Feng Liu. Video frame interpolation via adaptive separable convolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 261–270, 2017. 3, 6

[18] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015. 4

[19] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016. 5

[20] Junran Peng, Lingxi Xie, Zhaoxiang Zhang, Tieniu Tan, and Jingdong Wang. Accelerating deep neural networks with spatial bottleneck modules. *arXiv preprint arXiv:1809.02601*, 2018. 2

[21] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015. 2, 4

[22] Christian J Schuler, Michael Hirsch, Stefan Harmeling, and Bernhard Schölkopf. Learning to deblur. *IEEE transactions on pattern analysis and machine intelligence*, 38(7):1439–1451, 2016. 2

[23] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *Acm transactions on graphics (tog)*, 27(3):73, 2008. 2

[24] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 4

[25] Shuochen Su, Mauricio Delbracio, Jue Wang, Guillermo Sapiro, Wolfgang Heidrich, and Oliver Wang. Deep video deblurring for hand-held cameras. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1279–1288, 2017. 2

[26] Jian Sun, Wenfei Cao, Zongben Xu, and Jean Ponce. Learning a convolutional neural network for non-uniform motion blur removal. In *Proceedings of the IEEE Conference on*

Computer Vision and Pattern Recognition, pages 769–777, 2015. 2

[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, 4, 7, 8

[28] Min Wang, Baoyuan Liu, and Hassan Foroosh. Design of efficient convolutional layers using single intra-channel convolution, topological subdivisioning and spatial" bottleneck" structure. *arXiv preprint arXiv:1608.04337*, 2016. 4

[29] Li Xu, Jimmy SJ Ren, Ce Liu, and Jiaya Jia. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, pages 1790–1798, 2014. 2

[30] Jiahui Yu, Yuchen Fan, Jianchao Yang, Ning Xu, Zhaowen Wang, Xinchao Wang, and Thomas Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018. 2

[31] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2521–2529, 2018. 2, 8