BANet: Blur-aware Attention Networks for Dynamic Scene Deblurring

Fu-Jen Tsai* [1], **Yan-Tsung Peng*** [2], **Yen-Yu Lin** [3], **Chung-Chi Tsai** [4], **and Chia-Wen Lin** [1]

[1]National Tsing Hua University [2]National Chengchi University
[3]National Chiao Tung University [4]Qualcomm Technologies, Inc.
*equal contribution

Abstract

Image motion blur usually results from moving objects or camera shakes. Such blur is generally directional and non-uniform. Previous research efforts attempt to solve non-uniform blur by using self-recurrent multi-scale or multi-patch architectures accompanying with self-attention. However, using self-recurrent frameworks typically leads to a longer inference time, while inter-pixel or inter-channel self-attention may cause excessive memory usage. This paper proposes blur-aware attention networks (BANet) which accomplish accurate and efficient deblurring via a single forward pass. Our BANet utilizes region-based self-attention with multi-kernel strip pooling to disentangle blur patterns of different magnitudes and orientations and with cascaded parallel dilated convolution to aggregate multi-scale content features. Extensive experimental results on the GoPro and HIDE benchmarks demonstrate that the proposed BANet performs favorably against the state-of-the-arts in blurred image restoration and can provide deblurred results in real-time.

1. Introduction

Dynamic scene deblurring or blind motion deblurring aims to restore a blurred image with little knowledge about the blur kernel. Scene blurring caused by camera shakes, moving objects, low shutter speeds, or low frame rates not only degrades the quality of taken images/videos but also results in information loss. Therefore, removing such blurring artifacts to recover image details becomes essential to many subsequent vision applications where clean and sharp images are appreciated. Although significant progress has been made in both conventional and deep-learning-based approaches, we observe a compromise between accuracy and speed. Owing to this observation, we target at developing an efficient and effective algorithm in this paper for blurred image restoration with its current performance in accuracy and speed shown in Figure 1.

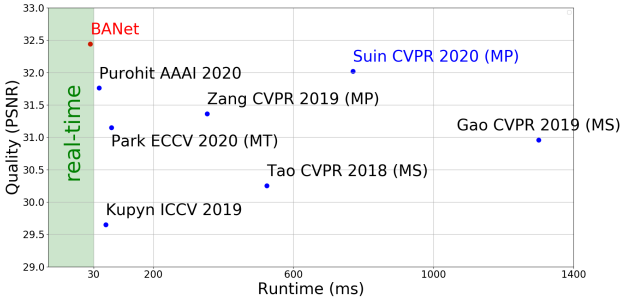


Figure 1. Performance comparison on the GoPro test dataset in terms of deblurring quality and runtime. The proposed BANet performs favorably against the state-of-the-art methods in both accuracy and efficiency.

Deep-learning-based approaches usually reach superior results, given their better feature representation capability toward dynamic scenes. Among the state-of-the-art architectures for deblurring, self-recurrent models have been widely adopted to leverage blurred image repeatability in either *multiple scales* (MS) [5, 18, 27], *multiple patch levels* (MP) [26, 33], or *multiple temporal behaviors* (MT) [21], as shown in Figure 2(a)–(c). Specifically, the MS models distill multi-scale blur information in a self-recurrent manner and restore blurred images based on the extracted coarse-to-fine features [5, 18, 27]. However, scaling a blurred image to a lower resolution often results in losing edge information [21]. In contrast, the MP models split an input blurred image into multiple patches to estimate and then remove motion blurs of different scales [26, 33]. However, splitting the blurred input and features into equal-sized non-overlapping patches may cause discontinuities in contextual information, which is sub-optimal for handling non-uniform blur in dynamic scenes. In [21], an MT structure is proposed to eliminate non-uniform blur and generate better results progressively. Yet, these existing self-recurrent models, including MS, MP, and MT, cannot achieve real-time high-quality deblurring.

In addition to model architectures, recent research stud-

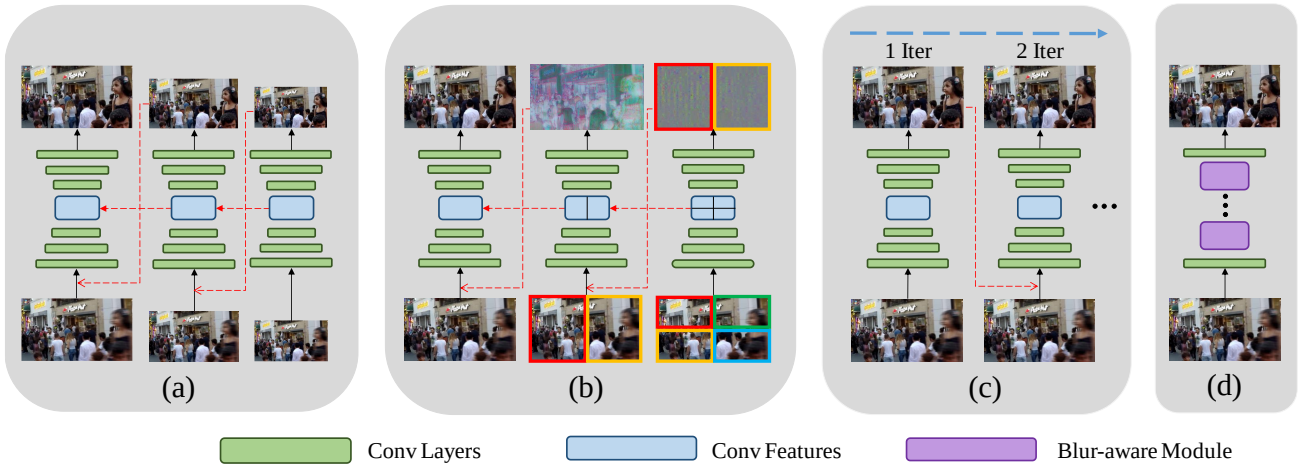


Figure 2. Network architecture comparisons among (a) MS, (b) MP, (c) MT, and (d) our BANet. Recurrent models are typically less efficient. BANet completes deblurring via a single forward pass.

ies [23, 26] further exploit self-attention to address blur non-uniformity. Suin *et al*. [26] utilize MP-based processing with self-attention to extract features for areas with global and local motions. However, using a self-recurrent mechanism to generate multi-scale features often leads to a longer inference time. To shorten the latency, Purohit and Rajagopalan [23] selectively aggregate features through learnable pixel-wise attention [34] enabled by deformable convolutions for modeling local blurs in a single forward pass. Despite its effectiveness, self-attention exploring pixel-wise or channel-wise correlations via trainable filters often causes high memory usage, thus only applicable to small-scale features [23]. Furthermore, motion blurs coming from moving objects manifest smeared effects and produce directional and local averaging artifacts which cannot be handled well by inter-pixel/channel correlations.

This paper proposes a *blur-aware attention networks* (BANet) to overcome the aforementioned issues. BANet is an efficient yet effective single-forward-pass model, as illustrated in Figure 2(d), which achieves the state-of-the-art deblurring performance while working in real-time, as shown in Figure 1. Specifically, our model stacks multiple layers of the *blur-aware attention module* (BAM) for removing motion blurs. Based on an observation of directional and regional averaging artifacts caused by dynamic blurs, the proposed BAM derives region-wise attention by using computationally inexpensive local averaging to globally and locally capture blurred patterns of different magnitudes and orientations. It also leverages cascaded parallel dilated convolutions to extract features without suffering from blur information loss as an MS model does. As a result, BANet possesses a superior deblurring capability and can well support subsequent real-time applications. In short, our contributions are two-fold: First, we design a novel BAM module which is capable of disentangling blur contents of different degrees in dynamic scenes. Second, our efficient single-forward-pass deep networks perform favorably against the state-of-the-art methods while running 27x faster than the best visual-quality competitor [26].

2. Related Work

Conventional Methods. Dynamic Scene image deblurring is a highly ill-posed problem since blurs stem from various factors in the real world. Conventional image deblurring studies often make different assumptions, such as uniform [3, 4, 16, 24] or non-uniform [6, 7, 8, 13, 30] blur, and image priors [1, 12, 19, 20, 31], to model blur characteristics. Namely, these methods impose different constraints to the estimated blur kernels, latent images, or both with handcrafted regularization terms for blur removal. Nevertheless, these methods often lead to solving a non-convex optimization problem and involve heuristic parameter tuning that is entangled with the camera pipeline; thus, they cannot generalize well to complex real-world examples.

Deblurring via Learning. Learning-based approaches with self-recurrent modules gain great success in single-image deblurring. Particularly, the *corarse-to-fine* schemes can gradually restore a sharp image on different resolutions (MS) [5, 18, 27], fields of view (MP) [26, 33], or temporal characteristics (MT) [21]. Despite the success, self-recurrent models usually lead to longer inference runtime. Recently, non-recurrent methods [14, 15, 23, 32] were proposed for efficient deblurring. For instance, Kupyn *et al*. [14, 15] suggest using conditional generative adversarial networks to restore blurred images. However, these methods do not well address non-uniform blur in dynamic scenes, often causing blur artifacts in the deblurring results. To address this issue, Yuan *et al*. [32] propose a spatially variant deconvolution network with optical flow estimation to guide deformable convolutions and capture moving ob-

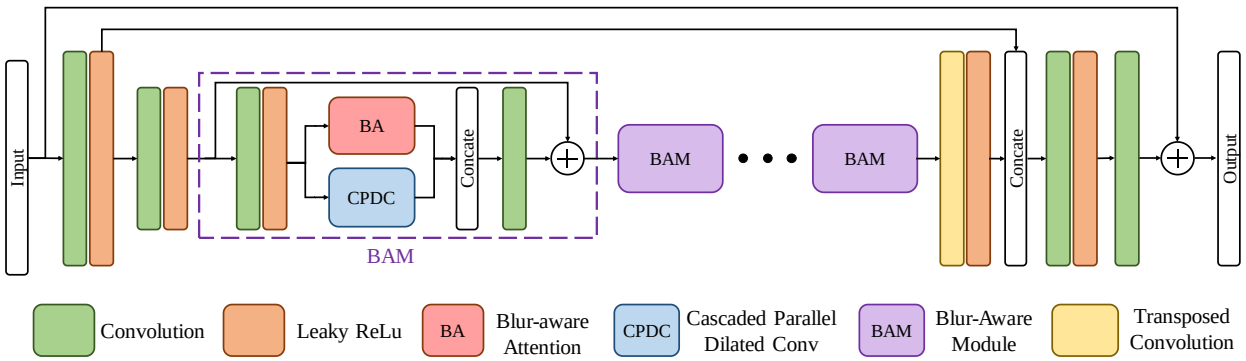


Figure 3. Architecture of the proposed blur-aware attention networks (BANet). The blur-aware modules (BAM) serve as the building blocks of BANet. The first BAM is detailed in the purple dotted box while the rest are represented by solid purple boxes.

jects during model training. However, optical flow may not always correlate with blur, which may cause less effective deblurring.

Self-attention. Self-attention (SA) [28] has been widely adopted to advance the fields of image processing [22, 34] and computer vision [10, 29]. Recent advances [23, 26] revealed that attention is beneficial for learning inter-pixel correlations to emphasize different local features for removing non-uniform blur. Specifically, Purohit *et al*. [23] proposed to deblur using SA to explore pixel-wise correlation for non-local feature adaptation. However, since SA requires much memory in $\mathcal{O}(H^2W^2)$ space, where H and W are the height and width of the input to SA. Thus, the method only applies SA to the smallest-scale features (from a 1280×720 blurred input to 160×90 SA's input), limiting the efficacy of SA. Also, motion blurs cause directional and local averaging artifacts, which may not be well addressed by merely pixel-wise SA. Suin *et al*. [26] proposed an MP architecture with less memory-intensive SA by using global average pooling with space complexity $\mathcal{O}(d_a d_c HW)$, where d_a is the channel dimension of the components *query* and *key* in SA, d_c is the dimension of the component *value*, and $d_a d_c < HW$. Despite the method's less space complexity, compressing pixel information into the channel domain may lose spatial information, thus degrading deblurring performance. In contrast, we propose an efficient and low-memory-cost regional averaging SA to capture non-uniform blur information more accurately. It is with space complexity $\mathcal{O}(CHW)$, where C is the number of output channels. It can deblur high-resolution input images and achieve superior performance in real-time.

3. Proposed Approach

We present the blur-aware attention networks (BANet) to address the potential issues in two commonly used techniques for deblurring: self-recurrence and self-attention. Self-recurrent algorithms result in longer inference time due to repeatedly accessing input blurred images. Self-attention based on inter-pixel or inter-channel correlations is memory intensive and cannot explicitly capture regional blurring information. Instead, the proposed BANet is a one-pass residual network consisting of a series of the stacked blur-aware modules (BAM), which serve as the building blocks, to effectively disentangle different patterns of blurriness.

As illustrated in Figure 3, BANet starts with two convolutional layers, which contain a stride of 2 to downsample the input image to half resolution. BANet employs one transposed convolutional layer to upsample features to the original size. In between, we stack a set of BAMs to correlate regions with similar blur and extract multi-scale content features. A BAM consists of two components: blur-aware attention (BA) and cascaded parallel dilated convolution (CPDC). BA distills global and local blurring information, and CPDC captures multi-scale blurred patterns. Combining BA and CPDC, the BAM is a residual-like architecture that derives both global and local multi-scale blurring features in a learnable manner. We detail the two key components, BA and CPDC, in the following.

3.1. Blur-aware Attention (BA)

To accurately restore the motion area displaying directional and averaging artifacts caused by moving objects and camera shakes, we propose a region-based self-attention module, called blur-aware attention (BA), to capture such effects in the global (image) and local (patch) scales. As shown in Figure 4, BA contains two cascaded parts: multi-kernel strip pooling (MKSP) and attention refinement (AR). MKSP catches multi-scale blurred patterns of different magnitudes and orientations, followed by AR to refine these patterns locally.

Multi-Kernel Strip Pooling (MKSP). Hou *et al*. [9] present an SP method that uses horizontal and vertical one-pixel long kernels to extract long-range band-shape context information for scene parsing. SP averages the input features within a row or a column individually and then fuses the two thin-strip features to discover global cross-

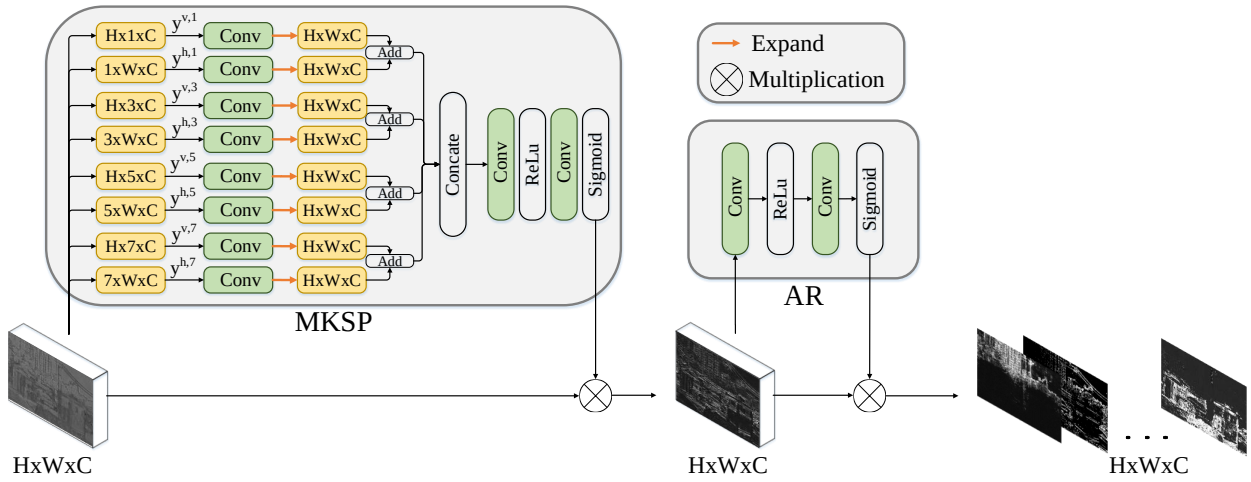


Figure 4. Architecture of blur-aware attention (BA). It cascades two parts, including multi-kernel strip pooling (MKSP) and attention refinement (AR). It is developed to disentangle blurred contents in an efficient way. See the text for details.

region dependencies. Let the input feature map be a three-dimensional (3D) tensor $\mathbf{x} = [x_{i,j,c}] \in R^{H\times W\times C}$, where C denotes the number of channels. Applying SP to $\mathbf{x}$ generates a vertical and a horizontal tensor followed by a 1D convolutional layer with a kernel size of 3. This produces $\mathbf{y}^v = [y^v_{i,c}] \in R^{H\times C}$ and $\mathbf{y}^h = [y^h_{j,c}] \in R^{W\times C}$, where $y^v_{i,c} = \frac{1}{W}\sum_{j=0}^{W-1} x_{i,j,c}$ and $y^h_{j,c} = \frac{1}{H}\sum_{i=0}^{H-1} x_{i,j,c}$. The SP operation fuses the two tensors into a 3D tensor $\mathbf{y} = [y_{i,j,c}] \in R^{H\times W\times C}$, where $y_{i,j,c} = y^v_{i,c} + y^h_{j,c}$, and then turns the fused tensor into an attention mask $\mathbf{M}_{sp}$ as

$$\mathbf{M}_{sp} = \sigma_{sig}(f_1(\mathbf{y})), \tag{1}$$

where f_1 is a 1×1 convolutional layer and σ_{sig} is the sigmoid function.

Motivated by SP, we propose MKSP that adopts strip pooling with different kernel sizes and orientations to discover regional and directional averaging artifacts caused by dynamic blurs. MKSP combines and compares multiple sizes/scales of averaging results followed by concatenation and convolution to catch blurred patterns of different magnitudes and orientations. The idea behind our design is that operations on multi-scale results, *e.g.* the difference between consecutive kernel sizes, reveal the scales of blurred patterns. We apply convolutional layers to automatically discover these blur-aware operations on the feature level to learn irregular attended features rather than a fixed cropping method on the image level used in MP methods [26, 33]. MKSP averages the input tensors within rows and columns to generate $H\times n\times C$ and $n\times W\times C$ long features, where $n \in \{1, 3, 5, 7\}$. Thus, MKSP generates four pairs of tensors, each of which has a vertical and a horizontal tensor followed by a 1D (for $n = 1$) or 2D (for the rest) convolutional layer with the kernel size of 3 or 3×3, respectively. This produces $\mathbf{y}^{v,n} \in R^{H\times n\times C}$ and $\mathbf{y}^{h,n} \in R^{n\times W\times C}$, where the vertical tensor is

$$y^{v,n}_{i,j,c} = \frac{1}{K_h}\sum_{k=0}^{K_h-1} x_{i,(j\cdot S_h+k),c}, \tag{2}$$

where the horizontal stride $S_h = \lfloor \frac{W}{n} \rfloor$ and the horizontal-strip kernel size $K_h = W - (n-1)S_h$. Symmetrically, the horizontal tensor is defined by

$$y^{h,n}_{i,j,c} = \frac{1}{K_v}\sum_{k=0}^{K_v-1} x_{(i\cdot S_v+k),j,c}, \tag{3}$$

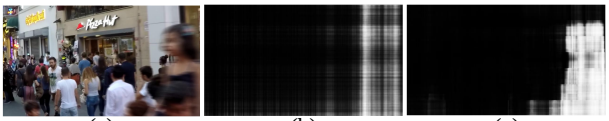
where the vertical stride $S_v = \lfloor \frac{H}{n} \rfloor$ and the vertical-strip kernel size $K_v = H - (n-1)S_v$.

Next, MKSP, after a 1D (for $n = 1$) or 2D (for the rest) convolutional layer, fuses each pair of tensors $(\mathbf{y}^{v,n}, \mathbf{y}^{h,n})$ into a 3D tensor $\mathbf{y}^n \in R^{H\times W\times C}$ by

$$y^n_{i,j,c} = y^{v,n}_{i,\lfloor \frac{n\times j}{W} \rfloor,c} + y^{h,n}_{\lfloor \frac{n\times i}{H} \rfloor,j,c}. \tag{4}$$

Similar to SP, we concatenate all the fused tensors to yield an attention mask as $\mathbf{M}_{mksp} = f_{AR}(\mathbf{y}^1 \oplus \mathbf{y}^3 \oplus \mathbf{y}^5 \oplus \mathbf{y}^7)$, where $\oplus$ stands for concatenation operation, and $f_{AR}(\cdot)$ represents a non-linear mapping function consisting of two 3×3 convolutional layers. The first layer uses the ReLU activation function, and the second a sigmoid function. As shown in Figure 5, the proposed MKSP can generate attention masks that better fit objects or local scenes than those by using SP with only $H\times 1$ and $1\times W$ kernels used, which yields rough band-shape masks.

Attention Refinement (AR). After obtaining the globally attended features by the element-wise multiplication of attention masks $\mathbf{M}_{mksp}$ and input tensor $\mathbf{x}$, we further refine



(a) (b) (c)

Figure 5. Visualization of the attention masks in our deblurring model. (a) An input blurred image. (b) & (c) The attention masks obtained using (b) strip pooling (SP) and (c) our MKSP.

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

Figure 6. (a) Input blurred image. (b)-(d) Comparisons among the attended feature maps by using different components of the proposed BA including (b) AR, (c) MKSP, and (d) MKSP + AR.

these features locally via a simple attention mechanism using $f_{AR}(\cdot)$. The final output of our BA block through the MKSP and AR stages is computed as

$$f_{AR}(\tilde{\mathbf{x}}) \otimes \tilde{\mathbf{x}}, \tag{5}$$

where $\otimes$ represents element-wise multiplication, and $\tilde{\mathbf{x}} = \mathbf{M}_{mksp} \otimes \mathbf{x}$ denotes the global features extracted using MKSP. Figures 6(c) and (d) demonstrate that cascading MKSP with AR can refine the attended feature maps.

The proposed BA facilitates the attention mechanism applied to deblurring since it requires less memory, *i.e.* $\mathcal{O}(HWC)$ where C represents channel dimensions, than those adopted in [23, 26]. It disentangles blurred contents with different magnitudes and orientations. Figure 7 showcases three examples of blur content disentanglement by using BA, where we witness that background scenes are differentiated from the foreground scenes because objects closer to the camera move faster, thus more blurred. Figure 9 shows more examples of attention maps yielded by BA, which implicitly acts as a gate for propagating relevant blur contents.

3.2. Cascaded Parallel Dilated Convolution (CPDC)

Atrous convolution, also called *dilated convolution*, has been widely applied to computer-vision tasks [2, 17] for enlarging receptive fields and extracting features from objects with different scales without increasing the kernel size. Inspired by this, we design a *cascaded parallel dilated convolution* (CPDC) block with multiple dilation rates to capture

(a) (b)

Figure 7. Three disentanglement examples of blurred patterns of different degrees using our BA. (a) Input blurred images and (b) attended feature maps on different regions.

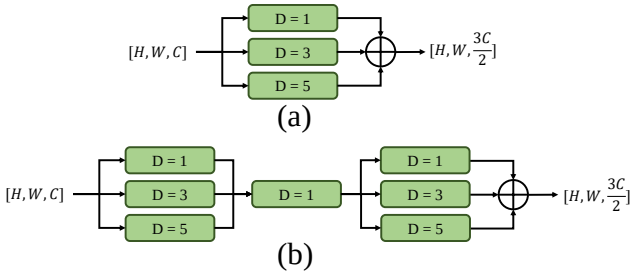


Figure 8. Architectures of (a) parallel dilated convolution (PDC) and (b) cascaded parallel dilated convolution (CPDC).

multi-scale blurred objects. Instead of stacking dilated convolutional layers with different rates in parallel, which we call *parallel dilated convolution* (PDC), our CPDC block cascades two sets of PDC with a single convolutional layer working as a channel attention bridge. It can distill patterns more beneficial to deblurring before passing through the second PDC. As an example, Figure 8(a) shows a PDC block that has three 3×3 dilated convolutional layers, each of which has a dilation rate D, where $D = 1, 3$, and 5, and each of which outputs features with half the number of input channels. After concatenation, the number of the output channel of the PDC block increases by 1.5 times. As shown in Figure 8(b), our CPDC block consists of two PDC blocks bridged by a 3×3 convolutional layer, which would be more effective in aggregating multi-scale content information for deblurring.

4. Experiments

This section evaluates the proposed method. In the following, we first describe the experimental setup, then compare our method with the state-of-the-arts, and finally conduct ablation studies to analyze individual components.

4.1. Experimental Setup

We evaluate the blur-aware attention networks (BANet) on two image deblurring benchmark datasets: 1) GoPro [18]

Figure 9. Visualization of the blur-aware attended features where moving objects in the blurred images are highlighted while background is mostly excluded. These blur-aware masks are crucial for handling blurry images with diverse blur patterns.

Table 1. Evaluation results on the benchmark GoPro testing set. The best score in its column is highlighted in bold and the second best is underlined. Symbol $*$ represents that the code was not released; thus we cite the results from the original papers or evaluate on the released deblurred images

Method	PSNR↑	SSIM↑	Time$_{(ms)}$ ↓
MSCNN [18]	30.40	0.936	943
SRN [27]	30.25	0.934	650
DSD [5]	30.96	0.942	1300
DeblurGanv2 [15]	29.55	0.934	42
DMPHN [33]	31.36	0.947	354
LEBMD* [11]	31.79	0.949	–
EDSD* [32]	29.81	0.934	**10**
DBGAN+* [35]	31.10	0.942	–
MTRNN [21]	31.13	0.944	53
RADN* [23]	31.85	0.953	38
SAPHN* [26]	32.02	0.953	770
BANet (stack-10)	**32.44**	**0.957**	<u>28</u>

Table 2. Evaluation results on the benchmark HIDE dataset. The best score in its column is highlighted in bold and the second best is underlined. Symbol $*$ represents that the code was not released; thus we cite the results from the original papers or evaluate on the released deblurred images

Method	PSNR↑	SSIM↑	Time$_{(ms)}$
DeblurGanv2 [15]	27.40	0.882	<u>42</u>
SRN [27]	28.36	0.904	424
HAdeblur* [25]	28.87	<u>0.930</u>	–
DSD [5]	29.10	0.913	1200
DMPHN [33]	29.10	0.918	341
MTRNN [21]	29.15	0.918	53
SAPHN* [26]	29.98	<u>0.930</u>	–
BANet (stack-10)	**30.27**	**0.931**	**26**

consists of $3,214$ pairs of blurred and sharp images of resolution 720×1280, where $2,103$ pairs are used for training and the rest are for testing, and 2) HIDE [25] contains $2,025$ pairs of HD images, all for testing. We train our model for $3,000$ epochs using Adam optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and the loss function is the Mean Squared Error (MSE) between the deblurred image and its ground truth. We set the initial learning rate to 10^{-4}. After the first 50 epochs, it starts to linearly decay until 10^{-7} after another 150 epochs. Following [21, 32], we randomly crop the input into 256×256 patches, along with random flipping or rotation for data augmentation. Lastly, we implement our model with PyTorch library on a computer equipped with Intel Xeon Silver 4210 CPU and NVIDIA 2080ti GPU.

4.2. Experimental Results

Quantitative Analysis: We compare our method with 11 latest approaches [5, 11, 15, 18, 21, 23, 26, 27, 32, 33, 35] that also handle dynamic deblurring on GoPro [18] testing set. For HIDE [25], we choose seven recent deblurring methods [5, 15, 21, 25, 26, 27, 33] depending on compared methods' availability. Table 1 lists the metric scores *i.e.* PSNR, SSIM and run-time obtained on the GoPro testing set with stacking ten (stack-10) BAMs version. We can observe that the self-recurrent models [5, 18, 21, 26, 27] consume comparatively longer times than the non-recurrent ones [15, 23], including ours. We record the average runtime of all the models using a single GPU. As reported in Table 1, our BANet runs significantly faster than the others while achieving the best metric scores. In particular, BANet outperforms the best competitor [26] by 0.42 dB in PSNR while running about $27\times$ faster. Similarly, in Table 2, BANet significantly outperforms all the compared methods.

Qualitative Analysis: Figures 10 and 11 respectively show the qualitative comparisons on GoPro testing set with previous state-of-the-arts [5, 15, 21, 23, 33] and on HIDE with [5, 15, 21, 27, 33]. Thanks to the proposed blur-aware module, our model can effectively restore images with sharper edges and richer details in dynamic scenes involving camera shakes and object motions. For instance, the existing state-of-the-arts [5, 15, 21, 23, 33] do not well recover the text regions in the first three rows and the ground

Figure 10. Qualitative comparisons on GoPro [18] test dataset. The deblurred results listed from left to right are from DeblurGanV2 [15], DSD [5], MTRNN [21], DMPHN [33], RADN [23] and ours (The deblurred images of SAPHN [26] were not compared since not released).

Figure 11. Qualitative comparisons on HIDE [25] test dataset. The deblurred results listed from left to right are from DeblurGanV2 [15], SRN [27], DSD [5], MTRNN [21], DMPHN [33] and ours (The deblurred images of SAPHN [26] were not compared since not released).

pattern in the last row of Figure 10. In Figure 11, the compared methods [5, 15, 21, 27, 33] do not work well on deblurring the faces in the first two rows and the texts in the other two rows, while BANet recovers these regions better.

4.3. Ablation Study

BAM with Different Components: In Table 3, we explore the performance change with different component unions in our Blur-Aware Module (BAM) based on the GoPro testing set. Adding a simple attention refinement (AR) mechanism to both PDC (Net1 vs. Net2) or PDC+MKSP (Net3 vs. Net4) for deblurring increases PSNR by around 0.2 dB. Substituting PDC in Net4 with CPDC (Net5), our proposed version of BAM leads to a significant performance gain with a reasonable additional runtime cost. Thanks to its mechanism for locating blur regions based on both global attention and local convolutions, our BAM attains the best performance while achieving real-time practical significance.

Numbers of Stacked BAMs: Using more layers to enlarge the receptive field may improve performance for computer vision or image processing tasks. However, for deblurring, stacking more layers does not guarantee better performance [26], and might consume extra inference time. However, using our residual learning-based BAM design, we can stack multiple layers to expand the effective receptive field for better deblurring. In Table 4, we show performance comparisons with various numbers of BAMs stacked in our model based on the GoPro testing set. We list four versions: stack-4, stack-8, stack-10, and stack-12, corresponding to

Table 3. Ablation study based on GoPro testing set for using different component combinations in the BAM of our BANet (stack-8)

Method	PDC	AR	MKSP	CPDC	PSNR	Time (ms)
Net1	√				31.29	7
Net2	√	√			31.47	9
Net3	√		√		31.83	13
Net4	√	√	√		32.02	16
Net5 (BAM)		√	√	√	32.23	23

Table 4. Performance comparisons of BANet stacking different numbers of BAMs based on the GoPro test dataset

Method	stack-4	stack-8	stack-10	stack-12
PSNR (dB)	31.44	32.23	32.44	32.46
Time (ms)	12	23	28	35

Table 5. PSNR performance (in dB) of SP and MKSP on HIDE (PSNR_H) and GoPro (PSNR_G) based on our stack-8 model with PDC. Subscripts denote kernel size combinations

	SP	MKSP_{135}	MKSP_{1357}	MKSP_{13579}
PSNR_H	29.68	**29.81**	**29.81**	29.76
PSNR_G	31.76	31.79	31.83	**31.84**

Table 6. Ablation study of CPDC (w/o BA) compared to PDC (w/o BA) on HIDE (PSNR_H) and GoPro (PSNR_G)

	PDC_{135}	$\text{PDC}^2{}_{135}$	CPDC_{135}
PSNR_H	28.97	29.27	**29.80**
PSNR_G	31.29	31.57	**32.04**
Size (MB)	**74.9**	128.4	128.9

Table 7. Ablation study of CPDC (w/o BA) with different dilated rates on HIDE (PSNR_H) and GoPro (PSNR_G)

	CPDC_{13}	CPDC_{135}	CPDC_{1357}
PSNR_H	28.97	**29.80**	29.57
PSNR_G	31.81	**32.04**	31.92

4, 8, 10, and 12 BAMs used in BANet. Although the quantitative performance improves with the number of BAMs, the improvement became marginal after 12. Therefore, we choose 10 resblocks for its excellent balance between efficiency and visual quality.

Effectiveness of MKSP and CPDC: In Table 5, we investigate the effects of kernel combination of MKSP on GoPro and HIDE datasets. It shows that MKSP outperforms SP, and MKSP with four kernel sizes of 1, 3, 5, and 7 performs the best. In Table 6, we verify that CPDC, which uses a single convolution as a channel attention bridge, can work better than PDC. For a fair comparison, we also compare CPDC against a PDC variant that stacks two PDCs in a series, called PDC^2, with a similar parameter size, and CPDC still performs better. As for the number of kernels in CPDC, Table 7 shows that CPDC with dilated rates 1, 3, and 5 performs the best.

4.4. Blur-aware Attention vs. Self-Attention

Purohit *et al.* [23] utilize a similar self-attention (SA) mechanism proposed in [34] for deblurring. It helps connect regions with similar blurs to facilitate global access to relevant features across the entire input feature maps. However, its high memory usage makes applying it to images of high resolution infeasible. Thus, SA is usually employed in network layers on a smaller scale like in [23], where important blur information would be lost due to down-sampling. In contrast, our proposed region-based attention is more suitable to correlate regions with similar blur characteristics. Moreover, it can be applied to images with a larger resolution thanks to its low memory consumption.

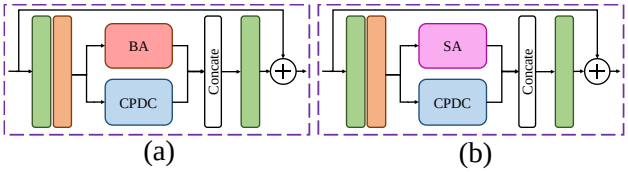


Figure 12. Architecture comparisons between (a) our original BAM and (b) BA replaced by SA [34] in BAM.

Table 8. Performance comparison between BA and SA [34] using BANet (stack-4) as a backbone tested on GoPro. "$*$" represents deblurring eight sub-images instead of an entire image

SA*	BA*	BA	PSNR	Time (ms)
√			30.59	1240
	√		31.30	670
		√	31.44	12

To further demonstrate our BA's efficacy, we compare the SA [34] with BA using our BANet (stack-4) as a backbone network, as shown in Figure 12(b). Due to the high memory demand for SA ($\mathcal{O}((HW)^2)$) to process 720×1280 images, we adopt our stack-4 model for training. When testing the networks, we separate the input image into eight sub-images for SA to process each with a single 2080ti GPU. We provide the deblurring results using BA with or without splitting the input image into eight sub-images for a fair comparison. Table 8 shows the results under the scenario of dividing the input into eight sub-images, demonstrating BA still outperforms SA for deblurring based on our BANet. Also, dividing the input image for deblurring using a single GPU increases the run-time, and adopting BA runs significantly faster. Last but not least, since our BA has lower memory usage, we can process the original input image without division to attain better visual quality.

5. Conclusion

In this paper, we propose a novel blur-aware attention network (BANet) for single image deblurring. BANet consists of the stacked blur-aware modules (BAMs) to disentangle blur contents of different magnitudes and orientations and the cascaded parallel dilated convolution (CPDC) module to aggregate multi-scale content features, for more accurate and efficient dynamic scene deblurring. We have investigated and examined our design through demonstrations of attention masks and attended feature maps as well as extensive ablation studies and performance comparisons. Our extensive experiments demonstrate that the proposed BANet achieves real-time deblurring and performs favorably against state-of-the-art deblurring methods on the GoPro and HIDE benchmarks.

References

[1] L. Chen, F. Fang, T. Wang, and G. Zhang. Blind image deblurring with local maximum gradient prior. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019. 2

[2] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017. 5

[3] S. Cho and S. Lee. Fast motion deblurring. In *ACM Trans. on Graphics*, 2009. 2

[4] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman. Removing camera shake from a single photograph. In *ACM Trans. on Graphics*, 2006. 2

[5] H. Gao, X. Tao, X. Shen, and J. Jia. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019. 1, 2, 6, 7

[6] A. Gupta, N. Joshi, L. Zitnick, M. Cohen, and B. Curless. Single image deblurring using motion density functions. In *Proc. Euro. Conf. Comput. Vis.*, 2010. 2

[7] S. Harmeling, H. Michael, and B. Schölkopf. Space-variant single-image blind deconvolution for removing camera shake. In *Proc. Neural Inf. Process. Syst.*, 2010. 2

[8] M. Hirsch, C. J. Schuler, S. Harmeling, and B. Schölkopf. Fast removal of non-uniform camera shake. In *Proc. Int. Conf. Comput. Vis.*, 2011. 2

[9] Q. Hou, L. Zhang, M.-M. Cheng, and J. Feng. Strip Pooling: Rethinking spatial pooling for scene parsing. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020. 3

[10] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018. 3

[11] Z. Jiang, Y. Zhang, D. Zou, J. Ren, J. Lv, and Y. Liu. Learning event-based motion deblurring. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020. 6

[12] N. Joshi, C. L. Zitnick, R. Szeliski, and D. J. Kriegman. Image deblurring and denoising using color priors. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2009. 2

[13] T. H. Kim and K. M. Lee. Segmentation-free dynamic scene deblurring. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2014. 2

[14] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018. 2

[15] O. Kupyn, T. Martyniuk, J. Wu, and Z. Wang. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *Proc. Int. Conf. Comput. Vis.*, 2019. 2, 6, 7

[16] X. L and J. J. Two-phase kernel estimation for robust motion deblurring. In *Proc. Euro. Conf. Comput. Vis.*, 2010. 2

[17] S. Liu, D. Huang, and a. Wang. Receptive field block net for accurate and fast object detection. In *Proc. Euro. Conf. Comput. Vis.*, 2018. 5

[18] S. Nah, T. H. Kim, and K. M. Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017. 1, 2, 5, 6, 7

[19] J. Pan, Z. Hu, Z. Su, and M. Yang. Deblurring text images via l0-regularized intensity and gradient prior. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2014. 2

[20] J. Pan, D. Sun, H. Pfister, and M.-H. Yang. Deblurring images via dark channel prior. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018. 2

[21] D. Park, D. U. Kang, J. Kim, and S. Y. Chun. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *Proc. Euro. Conf. Comput. Vis.*, 2020. 1, 2, 6, 7

[22] N. Parmar, A. Vaswani, J. Uszkoreit, Ł. Kaiser, N. Shazeer, A. Ku, and D. Tran. Image transformer. *arXiv preprint arXiv:1802.05751*, 2018. 3

[23] K. Purohit and A. N. Rajagopalan. Region-adaptive dense network for efficient motion deblurring. In *Proc. AAAI Conf. Artificial Intell.*, 2020. 2, 3, 5, 6, 7, 8

[24] Q. Shan, J. Jia, and A. Agarwala. High-quality motion deblurring from a single image. In *ACM Trans. on Graphics*, 2008. 2

[25] Z. Shen, W. Wang, J. Shen, H. Ling, T. Xu, and L. Shao. Human-aware motion deblurring. In *Proc. Int. Conf. Comput. Vis.*, 2019. 6, 7

[26] M. Suin*, K. Purohit*, and A. N. Rajagopalan. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020. 1, 2, 3, 4, 5, 6, 7

[27] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia. Scale-recurrent network for deep image deblurring. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018. 1, 2, 6, 7

[28] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Proc. Neural Inf. Process. Syst.*, 2017. 3

[29] X. Wang, R. Girshick, A. Gupta, and K. He. Non-local neural networks. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2018. 3

[30] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce. Non-uniform deblurring for shaken images. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010. 2

[31] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao. Image deblurring via extreme channels prior. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017. 2

[32] Y. Yuan, W. Su, and D. Ma. Efficient dynamic scene deblurring using spatially variant deconvolution network with optical flow guided training. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020. 2, 6

[33] H. Zhang, Y. Dai, H. Li, and P. Koniusz. Deep stacked hierarchical multi-patch network for image deblurring. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2019. 1, 2, 4, 6, 7

[34] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena. Self-attention generative adversarial networks. In *Proc. Machine Learning Research.*, 2019. 2, 3, 8

[35] K. Zhang, W. Luo, Y. Zhong, L. Ma, B. Stenger, W. Liu, and H. Li. Deblurring by realistic blurring. In *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2020. 6