

# Event-guided Multi-patch Network with Self-supervision for Non-uniform Motion Deblurring

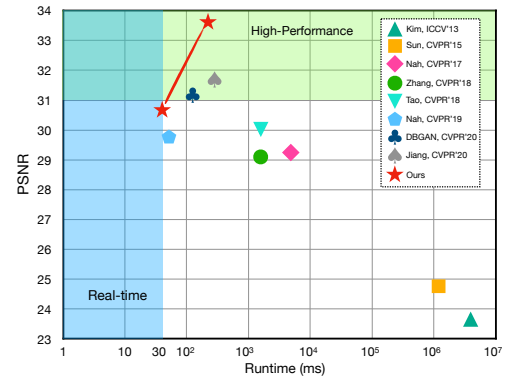
Hongguang Zhang · Limeng Zhang · Yuchao Dai · Hongdong Li · Piotr Koniusz

Received: 04.09.2021 / Accepted: 20.10.2022

**Abstract** Contemporary deep learning multi-scale deblurring models suffer from many issues: 1) They perform poorly on non-uniformly blurred images/videos; 2) Simply increasing the model depth with finer-scale levels cannot improve deblurring; 3) Individual RGB frames contain a limited motion information for deblurring; 4) Previous models have a limited robustness to spatial transformations and noise. Below, we extend our preliminary paper [59] by several mechanisms to address the above issues: I) We present a novel self-supervised event-guided deep hierarchical Multi-patch Network (MPN) to deal with blurry images and videos via fine-to-coarse hierarchical localized representations; II) We propose a novel stacked pipeline, StackMPN, to improve the deblurring performance under the increased network depth; III) We propose an event-guided architecture to exploit motion cues contained in videos to tackle complex blur in videos; IV) We propose a novel self-supervised step to expose the model to random transformations (rotations, scale changes), and make it robust to Gaussian noises. Our MPN achieves the state of the art on the GoPro and VideoDeblur datasets with a  $40\times$  faster runtime compared to current multi-scale methods. With 30ms to process an image at  $1280 \times 720$  resolution, it is the first real-time deep motion deblurring model for 720p images at 30fps. For StackMPN, we obtain significant improvements over 1.2dB on the GoPro dataset by increasing the network depth. Utilizing the event information and self-supervision further boost results to 33.83dB.

H. Zhang (the corresponding author) is an Assistant Professor at the Systems Engineering Institute, AMS, Beijing, 100141, China.  
E-mail: zhang.hongguang@outlook.com

L. Zhang is a PhD student at Shanghai Jiao Tong University. · Y. Dai is a Professor at the Northwestern Polytechnical University. · H. Li is a Professor at the Australian National University. · P. Koniusz is a Senior Research Scientist in Data61/CSIRO and a Senior Honorary Lecturer at the Australian National University.  
E-mail: piotr.koniusz@data61.csiro.au

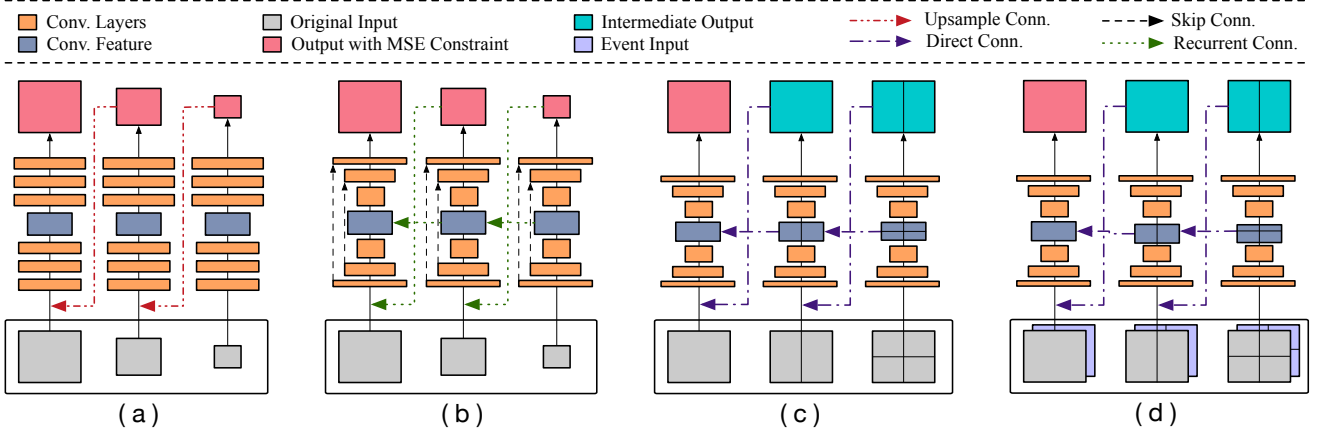


**Fig. 1:** The PSNR vs. runtime of state-of-the-art deep image deblurring methods and our method on the GoPro dataset [28]. The blue region indicates real-time inference, whereas the green region represents high performance motion deblurring (over 30 dB). Our method achieves the best performance at 30 fps for  $1280 \times 720$  images. The event-guided version and the stacked variants of our model further improve the performance at a cost of somewhat increased runtime.

## 1 Introduction

The objective of non-uniform blind image deblurring is to remove the undesired blur caused by the camera motion and the scene dynamics [28, 49, 32]. Prior to the success of deep learning, conventional deblurring methods used to employ a variety of constraints or regularizations to approximate the motion blur filters, involving an expensive non-convex non-linear optimization, and overly restrictive assumption of spatially-uniform blur kernel, resulting in a poor deblurring of complex blur patterns.

Deblurring methods based on deep Convolutional Neural Networks (CNNs) [21, 43] learn the regression between a blurry input image and the corresponding sharp image in an end-to-end manner [28, 49]. To exploit deblurring cues at varying processing levels, the “coarse-to-fine” scheme has



**Fig. 2:** Comparison between different network architectures. From left to right: (a) multi-scale [28], (b) scale-recurrent [49], (c) our hierarchical multi-patch architecture. We do not employ any skip or recurrent connections which simplifies our model. (d) our event-guided multi-patch network architecture, in which the event representations are concatenated with original blurry frames as two-stream inputs. Best viewed in color.

been extended to deep CNN scenarios by a multi-scale network architecture [28] and a scale-recurrent architecture [49]. Under the “coarse-to-fine” scheme, a sharp image is gradually restored at different resolutions in a pyramid. Nah *et al.* [28] demonstrated the ability of CNN models in removing motion blur from multi-scale blurry images, where a multi-scale loss function is devised to mimic conventional coarse-to-fine approaches. Following a similar pipeline, Tao *et al.* [49] shared network weights across scales to improve training and model stability, thus achieving highly effective deblurring compared with [28]. However, there still exist major challenges in current deep deblurring methods:

- Under the coarse-to-fine multi-scale scheme, most networks use a large number of training parameters due to large filter sizes. Thus, the multi-scale and scale-recurrent methods suffer from an expensive runtime (see Fig. 1) and struggle to improve the deblurring quality.
- Increasing the network depth for a low-resolution input in multi-scale approaches does not seem to improve the deblurring performance [28].
- The model is not capable of capturing motion information from RGB frames under complex blur, thus they cannot effectively address video deblurring.
- The learnt model has limited robustness to spatial transformations and random noises, which limits its usefulness in real-world applications.

In this paper, we address the challenges with the multi-scale and scale-recurrent architectures. We propose a novel architecture which exploits deblurring cues at different scales via a *hierarchical multi-patch* model. Specifically, we propose a simple yet effective multi-level CNN model called deep Multi-Patch Network (MPN) which uses multi-patch hierarchy as input. In this way, the residual cues from deblurring local regions are passed via residual-like links to

the next level of network which deals with coarser regions. Feature aggregation over multiple patches has been used in image classification [23, 14, 26, 20]. For example, [23] proposes Spatial Pyramid Matching (SPM) which divides images into coarse-to-fine grids in which histograms of features are computed. In [20], a second-order fine-grained image classification model uses feature embeddings of overlapping patches and positional embeddings for aggregation. Sun *et al.* [48] learn a patch-wise motion blur kernel through a CNN for restoration via an expensive energy optimization.

The advantages of our network are threefold: 1) As the inputs at different levels have the same spatial resolution, we apply residual-like learning which requires smaller filter sizes and leads to a fast inference; 2) We use an SPM-like model exposed to more training data at the finest level due to relatively more patches available for that level; 3) Our architecture encourages model to learn to deblurring from easier tasks (small patches) to harder tasks (large patches), a gradual learning process that encourages the consistency of deblurring over different locations and spatial sizes.

To overcome the limitation of *stacking depth* in multi-scale and multi-patch models, simply increasing the model depth by introducing additional coarser or finer grids cannot improve the overall deblurring performance of known models. Thus, we present the novel stacked version of our MPN, whose performance can be effectively and continuously improved by stacking multiple submodels.

As an extension of our preliminary paper [59], we propose the event-guided MPN to deal with complex motion blurs in rapidly evolving scenes. To this end, we employ the Dynamic and Active Pixel Sensor (DAVIS) to simultaneously produce the grey-scale Active Pixel Sensor (APS) and event frames, in which object motions are captured at a very high temporal resolution ( $1\mu s$ ), thus increasing the ability of our model to deblur complex real-world blurs.

Moreover, we notice that deep deblurring models have a limited robustness to different types of transformations and perturbations, *e.g.*, random rotations, scale transforms, and Gaussian noises. For example, once we apply a weak Gaussian noise to blurry images on input, the PSNR score sharply drops to around 20dB. Therefore, we propose a novel self-supervised robust training strategy to explicitly align deblurred outputs of an input image and its augmented version (deblurred output of augmented image is de-augmented prior to alignment), thus enhancing the robustness of our model. Our contributions are summarized below:

- I. We propose an end-to-end CNN hierarchical model that performs deblurring in the fine-to-coarse grids by exploiting multi-patch localized-to-coarse operations. Each finer level acts in the residual manner by contributing its residual image to the coarser level, thus letting each level of network focus on different scales of blur. We perform baseline comparisons in the common testbed (where possible) for fair comparisons.
- II. We identify the limitation to stacking depth of current deep deblurring models and introduce a novel stacking approach which effectively overcomes this limitation.
- III. We introduce the use of events in deep multi-patch architecture to capture richer motion information, thus help the model deblur videos with complex blurs and scenes.
- IV. We propose to apply an auxiliary self-supervised consistency loss leveraging pretext augmentation tasks to enhance the robustness of model w.r.t. different geometric transformations and photometric distortions, thus reducing overfitting to specific training poses, which helps deblur real-world images.

Our experiments demonstrate clear benefits of our event-guided SPM-like model in non-uniform motion deblurring. To the best of our knowledge, our model is the first multi-patch take on blind motion deblurring, *e.g.*, MPN is the first model that supports deblurring of 720p images real-time (at 30fps). The self-supervised step is demonstrated useful in deep deblurring scenario also for the first time.

## 2 Related Work

Below we discuss the related works on image deblurring. Conventional image deblurring methods [4, 16, 57, 24, 35, 17, 15, 40] fail to remove non-uniform motion blur due to the use of spatially-invariant deblurring kernel. Moreover, their complex computational inference leads to long processing times, which cannot satisfy the ever-growing needs for real-time deblurring.

**Deep Deblurring.** Recently, CNNs have been applied for non-uniform image deblurring to deal with the complex motion blur in a time-efficient manner [58, 48, 28, 39, 30, 46].

Xu *et al.* [58] proposed a deconvolutional CNN which removes blur in non-blind setting by recovering a sharp image given the estimated blur kernel. Their network uses separable kernels which can be decomposed into a small set of filters. Sun *et al.* [48] estimated and removed a non-uniform motion blur from an image by learning the regression between  $30 \times 30$  image patches and their corresponding kernels. The conventional energy-based optimization scheme was employed to estimate the latent sharp image.

Su *et al.* [46] presented a deep learning framework to process blurry video sequences and accumulate information across frames. This method does not require spatially-aligned pairs of samples. Nah *et al.* [28] exploited a multi-scale CNN to restore sharp images in an end-to-end fashion from images whose blur is caused by various factors. A multi-scale loss was employed to mimic the coarse-to-fine pipeline in conventional deblurring approaches.

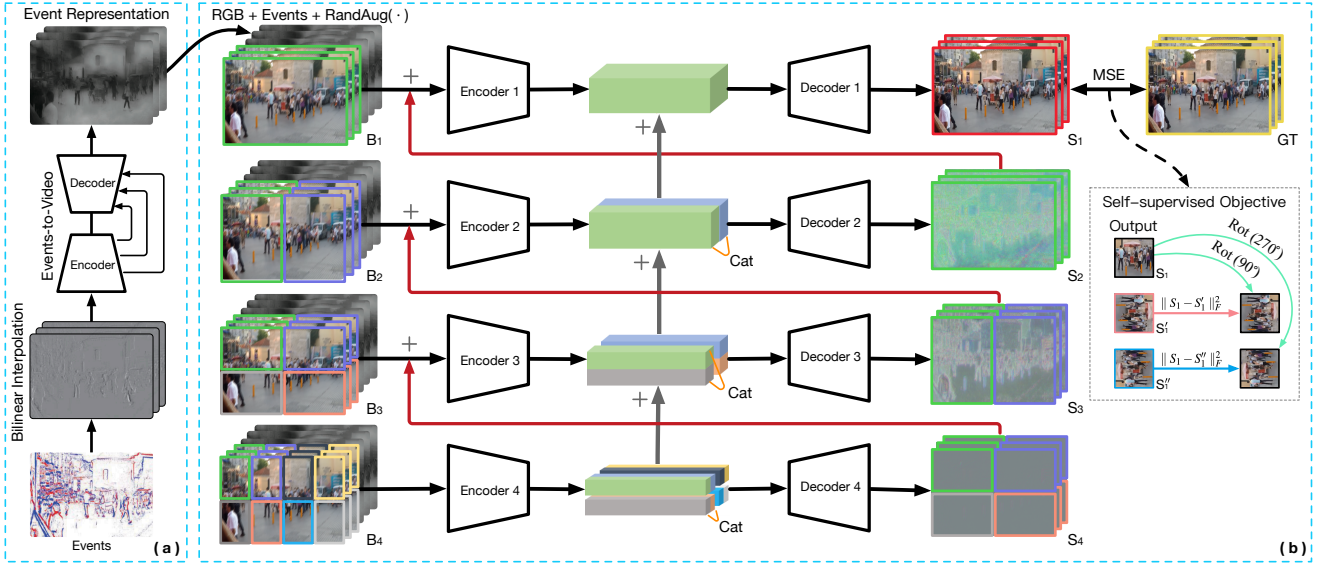
Recurrent Neural Network (RNN) is often used in deblurring due to its sequential information processing. Take as an example a network [62] consisting of three deep CNNs and one RNN. The RNN is used as a deconvolutional decoder on feature maps extracted by the first CNN module. Another CNN module learns weights for each layer of RNN. The last CNN module reconstructs the sharp image. Scale-Recurrent Network (SRN-DeblurNet) [49] uses ConvLSTM cells to aggregate feature maps from coarse-to-fine scales. Finally, Nah *et al.* [29] proposed a recurrent network, which iteratively updates the hidden state with existing parameters.

Generative Adversarial Nets (GANs) have also been employed in deblurring due to their advantage in preserving texture details and generating photorealistic images. Kupyn *et al.* [22] presented a conditional GAN which produces high-quality deblurred images via the Wasserstein loss.

Gao *et al.* [11] proposed a novel selective parameter sharing scheme to improve the dynamic deblurring task. Though their approach achieves impressive results, the complicated nested connections lead to very long processing runtime, which cannot satisfy real-time applications.

Notably, some recent works have been based on our multi-patch network [59]. Suin *et al.* [47] proposes a modified MPN, which can handle the blur variations across different spatial locations, and adaptively process test images to improve the performance. Dipta *et al.* [6] propose a fast multi-patch architecture to address image dehazing task.

**Event-based modeling.** Events are playing an important role in recent motion deblurring tasks. Event cameras such as DAVIS [3] and DVS [25] record log intensity changes at the microsecond scale with negligible motion blurs, allowing them to compensate the lost information from motion blur. The output of event camera is a stream of events formed into quadruplets  $(x, y, t, p)$  that encode the position of brightness changes, time and polarity.



**Fig. 3:** Our proposed Event-guided Multi-Patch Network (E-MPN) consists of two parts: (a) a generator of event representation, based on a 10-layer residual U-Net, (b) multi-patch deblurring network which consists of multi-level coarse-to-fine branches. As the patches do not overlap with each other, they may cause boundary artifacts which are removed by the consecutive upper levels of our model. Symbol  $+$  is a summation akin to residual networks.

Recent works study how to directly transform events into sharp images, *e.g.*, Bardow *et al.* [1] simultaneously estimated the optical flow and intensity images with a fixed-length sliding spatial-temporal window by solving an energy minimizing problem. Barua *et al.* [2] proposed to learn a sparse patch-based dictionary to match event patches with gradient patches, then use the so-called Poisson integration to reconstruct the intensity images. Munda *et al.* [27] restored intensity images through the manifold regularization. Rebecq *et al.* [37] proposed a novel recurrent network to reconstruct videos from a stream of events. As events are asynchronous, they are raised if there is a local intensity change within the scene, so single events can model static scenes/textures, and sequences of events can model very rapid motions.

DAVIS [3] can simultaneously output events and Active Pixel Sensor (APS) intensity images that contain the static texture. It directly integrates events on the APS frame and refreshes the event accumulation. Scheerlinck *et al.* [38] proposed an asynchronous event-driven complementary filter to integrate the APS frame with events for continuous-time intensity estimation. Pan *et al.* [33] formulated a deblurring task as an optimization problem that solves a single variable non-convex problem with a double integral model. Jiang *et al.* [18] presented a convolution recurrent network to integrate visual and temporal knowledge at the global and local scales. With a novel directional event filtering module, sharp edge boundary guidance is extracted which increases the quality of reconstructed details. The eSL-Net model [52] constructed an event-based sparse learning network to im-

prove the deblurring performance. E-CIR [44] proposed to leverage events to construct the parametric bases, and introduced a refinement module to propagate visual features among frames. Wang *et al.* [55] proposed to recreate intensity images using an asynchronous Kalman filter based on a unified event and frame uncertainty model. The images reconstructed using these methods, however, include artifacts due to the accumulation of event noises.

**Self-supervised learning.** A network can be trained with so-called pretext tasks, *e.g.*, predicting augmentation labels, or predicting easily obtainable self-information as auxiliary objective to improve the performance by making network ‘aware’ of auxiliary tasks. Self-supervised learning has been used in object recognition [8, 7, 13, 42, 51], video representation learning [9, 41, 10, 50, 54, 53], and also few-shot image and video recognition [12, 45, 61, 60].

Two types of self-supervision are popular: i) contrastive loss; and ii) prediction of label of pretext task. For instance, Gidaris *et al.* [13] predict labels of random image rotations, Doersch *et al.* [7] predict the relative pixel positions, Dosovitskiy *et al.* [8] learn to discriminate a set of surrogate classes, and approaches [12, 45] improve the few-shot performance by predicting labels of image rotations and jigsaw patterns.

In contrast to previous self-supervised pipelines, we leverage self-supervision to promote the consistency of deblurring under augmentations to improve the robustness of model to geometric transformations and photometric distortions. The self-supervision strategy in this paper aligns features obtained from the same augmentation applied at the early and late stage, respectively. This is somewhat related to the



so-called knowledge distillation, which encourages one stream of information to distil its knowledge to the other, therefore improving the deblurring performance of our model. However, notice that such a self-supervision is not the distillation pipeline, *i.e.*, it does not use two network streams such as the teacher and student networks.

### 3 Approach

In this paper, we propose to exploit the multi-patch hierarchy for efficient and effective blind motion deblurring. The overall architecture of our proposed MPN network is shown in Fig. 3 for which we use the (1-2-4-8) model (explained in Sec. 3.2) as an example. Our network is inspired by coarse-to-fine Spatial Pyramid Matching [23], which has been used for the problem of scene recognition [20] to aggregate multiple image patches for better performance. In contrast to the expensive inference in multi-scale and scale-recurrent network models [28, 49] shown in Fig. 2, our approach (also in Fig. 2) uses a residual-like architecture, thus requiring small-size filters which result in fast processing. Despite our model uses a very simple architecture (skip and recurrent connections have been removed), it is very effective. In contrast to Nah *et al.* [28] which uses deconvolution/upsampling links, we use operations such as feature map concatenations, which are possible due to the multi-patch setup we propose. Moreover, our self-supervised unit differs from typical self-supervised representations: we impose the deblurring consistency between augmented and non-augmented images by reversing the augmentation from the deblurred output.

#### 3.1 Encoder-decoder Architecture

Each level of our MPN network consists of one encoder and one decoder whose architecture is illustrated in Fig. 4. Our encoder consists of 15 convolutional layers, 6 residual links and 6 ReLU units. The layers of decoder and encoder are identical except that two convolutional layers are replaced by deconvolutional layers to generate images.

Our encoder and decoder use  $\sim 3.6$  MB parameters due to the small convolutional kernel size and the residual nature of our model, which contribute to the fast deblurring runtime. By contrast, the multi-scale deblurring network [28] has 303.6 MB parameters leading to the slower inference.

#### 3.2 Network Architecture

Fig. 3 shows the architecture of our MPN, in which we use the (1-2-4-8) model for illustration purposes. Notation (1-2-4-8) indicates the numbers of non-overlapping image patches from the coarsest to the finest level, *i.e.*, a vertical

split at the second level,  $2 \times 2 = 4$  splits at the third level, and  $2 \times 4 = 8$  splits at the fourth level. For third and fourth levels that output multi-patch residuals, we impose the region-aware consistency loss between adjacent boundaries.

We denote the initial blurry image input as  $\mathbf{B}_1$ , while  $\mathbf{B}_{i,j}$  is the  $j$ -th patch at the  $i$ -th level. Moreover,  $\mathcal{F}_i$  and  $\mathcal{G}_i$  are the encoder and decoder at level  $i$ ,  $\mathbf{C}_{i,j}$  is the output of  $\mathcal{G}_i$  for  $\mathbf{B}_{i,j}$ , and  $\mathbf{S}_{i,j}$  represents the output patches from  $\mathcal{G}_i$ .

Each level of our network consists of an encoder-decoder pair. The input for each level is generated by dividing the original blurry image input  $\mathbf{B}_1$  into multiple non-overlapping patches. The output of encoder from a lower level (corresponds to finer grid) is added to the output of encoder one level up. The output of decoder from the lower level (corresponds to finer grid) is added to the upper level input grids passed to the input of encoder (one level above) so that the top level contains all information inferred in the finer levels. Note that the numbers of input and output patches at each level are different as the main idea of our work is to make the lower level focus on local information (finer grid) to produce a residual information for the coarser grid (obtained by concatenating convolutional features).

Consider the (1-2-4-8) variant as an example. The deblurring process of MPN starts at the bottom level 4.  $\mathbf{B}_1$  is sliced into 8 non-overlapping patches  $\mathbf{B}_{4,j}, j = 1, \dots, 8$ , which are fed into the encoder  $\mathcal{F}_4$  to produce the following convolutional feature representation:

$$\mathbf{C}_{4,j} = \mathcal{F}_4(\mathbf{B}_{4,j}), \quad j \in \{1, \dots, 8\}. \quad (1)$$

Then, we concatenate adjacent features (in the spatial sense) to obtain a new feature representation  $\mathbf{C}_{4,j}^*$  of the same size as the convolutional feature representation at level 3:

$$\mathbf{C}_{4,j}^* = \mathbf{C}_{4,2j-1} \oplus \mathbf{C}_{4,2j}, \quad j \in \{1, \dots, 4\}, \quad (2)$$

where  $\oplus$  denotes the concatenation operator. The concatenated feature representation  $\mathbf{C}_{4,j}^*$  is passed through the encoder  $\mathcal{G}_4$  to produce  $\mathbf{S}_{4,j} = \mathcal{G}_4(\mathbf{C}_{4,j}^*)$ .

Next, we move one level up to level 3. The input of  $\mathcal{F}_3$  is formed by summing up  $\mathbf{S}_{4,j}$  with patches  $\mathbf{B}_{3,j}$ . Once the output of  $\mathcal{F}_3$  is produced, we add to it  $\mathbf{C}_{4,j}^*$ :

$$\mathbf{C}_{3,j} = \mathcal{F}_3(\mathbf{B}_{3,j} + \mathbf{S}_{4,j}) + \mathbf{C}_{4,j}^*, \quad j \in \{1, \dots, 4\}. \quad (3)$$

At level 3, we concatenate the feature representation of level 3 to obtain  $\mathbf{C}_{3,j}^*$  and pass it through  $\mathcal{G}_3$  to obtain  $\mathbf{S}_{3,j}$ :

$$\mathbf{C}_{3,j}^* = \mathbf{C}_{3,2j-1} \oplus \mathbf{C}_{3,2j}, \quad j \in \{1, 2\}, \quad (4)$$

$$\mathbf{S}_{3,j} = \mathcal{G}_3(\mathbf{C}_{3,j}^*), \quad j \in \{1, 2\}. \quad (5)$$

Note that features at all levels are concatenated along the spatial dimension: imagine neighboring patches being concatenated to form a larger patch.

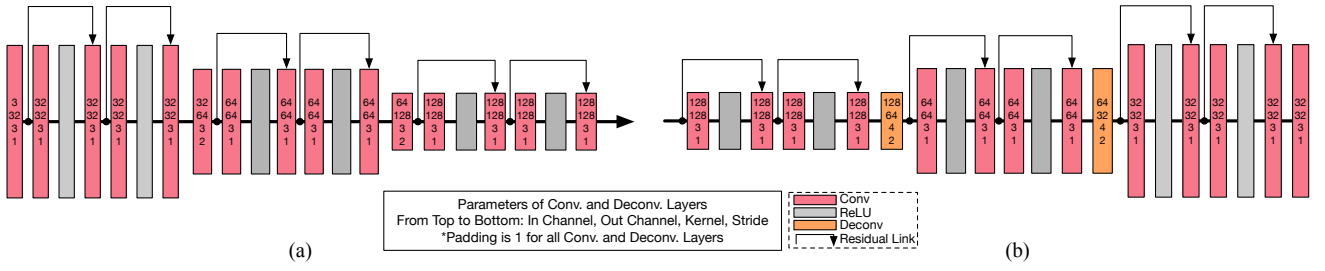


Fig. 4: The architectures and layer configurations of our (a) decoder and (b) encoder.

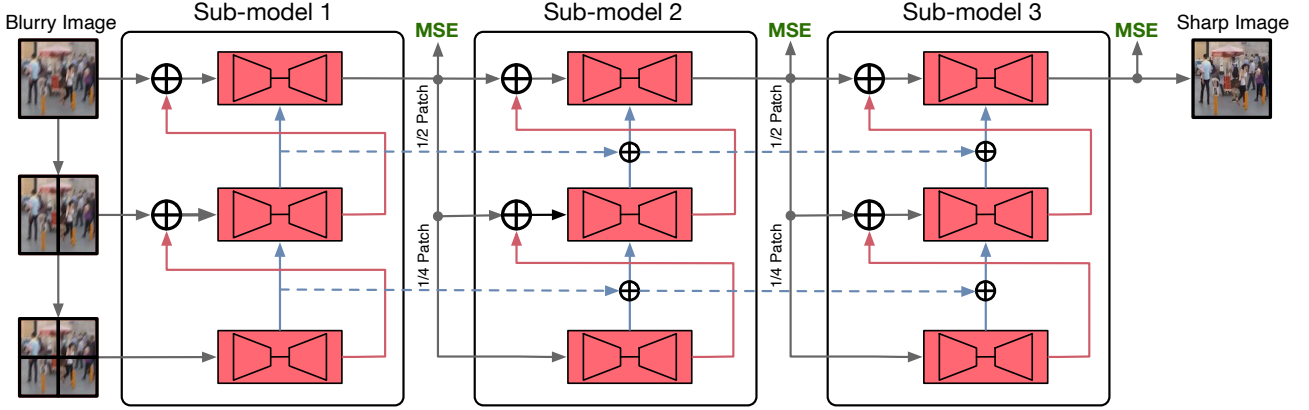


Fig. 5: The architecture of Stacked Multi-patch Network (StackMPN), in which multiple MPNs are serially stacked to distribute the deblurring task among several sub-models, and reduce the training difficulty level by level. Such a design provides the performance gain as the network depth increases.

At level 2, our network takes two image patches  $\mathbf{B}_{2,1}$  and  $\mathbf{B}_{2,2}$  as input. We update  $\mathbf{B}_{2,j}$  so that  $\mathbf{B}_{2,j} := \mathbf{B}_{2,j} + \mathbf{S}_{3,j}$  and pass it through  $\mathcal{F}_2$ :

$$\mathbf{C}_{2,j} = \mathcal{F}_2(\mathbf{B}_{2,j} + \mathbf{S}_{3,j}) + \mathbf{C}_{3,j}^*, \quad j \in \{1, 2\}, \quad (6)$$

$$\mathbf{C}_2^* = \mathbf{C}_{2,1} \oplus \mathbf{C}_{2,2}. \quad (7)$$

The residual map at level 2 is given by:

$$\mathbf{S}_2 = \mathcal{G}_2(\mathbf{C}_2^*). \quad (8)$$

At level 1, the final deblurred output  $\mathbf{S}_1$  is given by:

$$\mathbf{C}_1 = \mathcal{F}_1(\mathbf{B}_1 + \mathbf{S}_2) + \mathbf{C}_2^*, \quad (9)$$

$$\mathbf{S}_1 = \mathcal{G}_1(\mathbf{C}_1). \quad (10)$$

Different from approaches [28, 49] that evaluate the Mean Square Error (MSE) loss at each level, we evaluate the MSE loss only at the output of level 1 (which resembles the residual network). The loss function of MPN is given as:

$$\mathcal{L}_{deblur} = \frac{1}{2} \sum_j \|\mathbf{S}_{1j} - \mathbf{G}_j\|_F^2, \quad (11)$$

where  $\mathbf{G}_j$  denotes the ground-truth sharp image  $j$ . Due to the hierarchical multi-patch architecture, our network follows the principle of residual learning: the intermediate outputs at different levels  $\mathbf{S}_{ij}$  capture image statistics at different scales. Thus, we evaluate the loss function only at the

first level. We have investigated the use of multi-level MSE loss which forces the outputs at each level to be close to the ground truth image. However, as expected, there is no visible performance gain achieved by using the multi-scale MSE loss.

### 3.3 Stacked Multi-Patch Network

As reported by Nah *et al.* [28] and Tao *et al.* [49], adding finer network levels cannot improve the deblurring performance of the multi-scale and scale-recurrent architectures. For our multi-patch network, we have also observed that dividing the blurred image into ever smaller grids does not further improve the deblurring performance. This is mainly due to coarser levels attaining low empirical loss on the training data fast thus excluding the finest levels from contributing their residuals.

In this section, we propose a novel stacking paradigm for deblurring. Instead of making the network deeper vertically (adding finer levels into the network model, which increases the difficulty for a single worker), we propose to increase the depth horizontally (stacking multiple network models), which employs multiple MPN workers horizontally to perform deblurring.

Figure 5 demonstrates how we cascade the MPN to improve the deblurring performance. The stacked model, called StackMPN, stacks multiple “bottom-top” MPNs. Note that the output of sub-model  $i - 1$  and the input of sub-model  $i$  are connected, which means that for the optimization of sub-model  $i$ , output from the sub-model  $i - 1$  is required. All intermediate features of sub-model  $i - 1$  are distilled to sub-model  $i$ . The MSE loss is evaluated at the output of every sub-model  $i$  by minimizing the StackMPN objective:

$$\mathcal{L}_{deblur} = \frac{1}{2} \sum_j \sum_{i=1}^N \|\mathbf{S}_{ij} - \mathbf{G}_j\|_F^2, \quad (12)$$

where  $N$  is the number of sub-models used,  $\mathbf{S}_{ij}$  is the output of sub-model  $i$  (note that definitions of  $\mathbf{S}_{ij}$  differ between Eq. 11 and 12),  $\mathbf{G}$  is the ground-truth sharp image, and  $j$  loops over a subset of images.

Our experiments will illustrate that such a stacked network can significantly benefit from the increased network depth and improve the deblurring performance accordingly. Although our stacked pipeline uses MPN units, we believe they are generic, that is, other deep deblurring methods can be stacked in the similar manner to improve their performance. However, the total processing time may be unacceptable if a costly deblurring model is employed for the basic unit. Thanks to fast and efficient MPN units, we can control the runtime and size of stacking networks within a reasonable range to cater for various applications.

### 3.4 Event-guided MPN

The above proposed MPN model is applied on image deblurring task, thus its ability to deal with realistic and complicated blur patterns is insufficient. To improve deblurring further, one may extend our model to videos in order to restore the blur kernel based on the temporal information.

One may simply extend MPN by concatenating multiple frames along the channel mode to form an input (followed by the adjustment of the number of input and output channels). However, such a model does not fully exploit the temporal information: the channel-wise convolutional operator in the first layer of encoder does not guarantee the model to develop a sufficient implicit model of motion. Thus, we propose to introduce the event representation into our MPN to form a hybrid event-guided deblurring process as in Fig. 3.

Specifically, let us denote  $T$  consecutive blurry frames as  $\{\mathbf{B}^{(t)}\}_{t=1}^T$ , a set of events as  $\{\mathbf{E}^{(t)}\}_{t=1}^T$ ,  $\Delta\mathbf{E}^{(t)}$  as the event information from time  $t$  to  $t + \Delta t$ , and  $\Delta t$ , the infinitesimal time step. Let  $\{\mathbf{S}^{(t)}\}_{t=1}^T$  be restored sharp frames. Inspired by the design of event cameras [33], we have:

$$\mathbf{B}^{(f)} = \frac{1}{T} \sum_{t=f-T/2}^{f+T/2} \mathbf{S}^{(t)}, \quad (13)$$

$$\mathbf{S}^{(t)} = \mathbf{S}^{(f)} e^{c\mathbf{E}^{(t)}}, \quad (14)$$

$$\mathbf{E}^{(t)} = \sum_{h=f}^t \Delta\mathbf{E}^{(h)}, \quad (15)$$

where  $c$  is the threshold determining if an event should be recorded,  $\mathbf{E}^{(t)}$  is the sum of events during time  $f$  to  $t$ .

Above equations show that the blurry frame is generated from latent sharp frames during the exposure time  $[f - T/2, f + T/2]$ , and the sharp frame at time  $t$  can be generated by simply interpolating the sharp frame at time  $t - T$  and the events during time step  $T$ . Combining Eq. 13, 14 and 15, we have:

$$\mathbf{B}^{(f)} = \frac{1}{T} \sum_{t=f-T/2}^{f+T/2} \mathbf{S}^{(t)} = \frac{\mathbf{S}^{(f)}}{T} \sum_{t=f-T/2}^{f+T/2} e^{c\sum_{h=f}^t \Delta\mathbf{E}^{(h)}}, \quad (16)$$

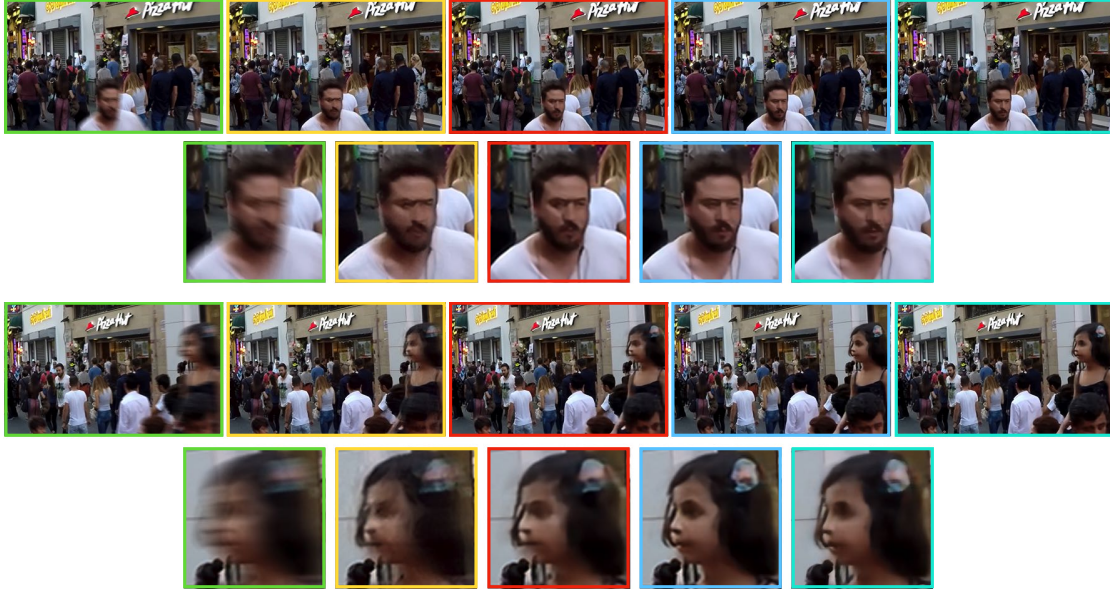
$$\log \mathbf{S}^{(f)} = \log \mathbf{B}^{(f)} - \log \left( \frac{1}{T} \sum_{t=f-T/2}^{f+T/2} e^{c\mathbf{E}^{(t)}} \right). \quad (17)$$

From the above equations, we conclude that the sharp frame is associated with the original blurry frame and the event information. Once the event information from  $f - T/2$  to  $f + T/2$  is collected, the learning formulation of MPN can be re-written as the following two-stream variant:

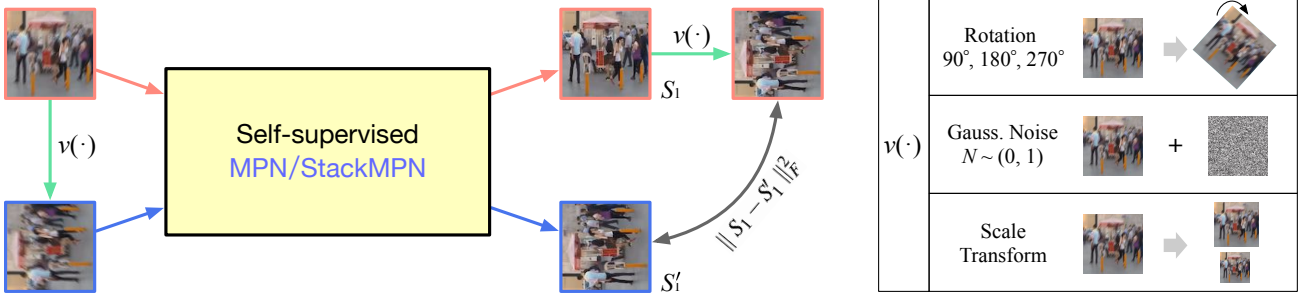
$$\mathbf{S}^{(f)} = \text{MPN}(\mathbf{B}^{(f)}, \mathbf{E}^{(f-T/2:f+T/2)}). \quad (18)$$

We evaluate our model on realistic event datasets [33] in which samples are recorded by DAVIS sensor. For non-event datasets, we employ the method proposed in [36] to simulate the event information from RGB frames.

The event stream needs to be converted to an image-like event representation before being fed into the pipeline for training and evaluation. Integrating the event on a 2D plane is a natural choice, whereas encoding it with spatial-temporal voxel grad can be more effective. Thus, we use the Events-to-Video model [37], which is built upon a 10-layer residual U-Net, to produce 10 adjacent event representations with 0.1s time grid for the central frame. Subsequently, 10 event representations are concatenated with original RGB blurry frames to restore the sharp RGB output. As the events are collected at a very high frame-rate, we naturally capture accurate object motions compared to conventional cameras. Thus, our two-stream deblurring model is expected to significantly improve the deblurring performance in end-to-end manner for both synthetic and realistic blurry images.



**Fig. 6:** Deblurring results. The images from left to right show original blurry images, the results of [28], [49], our MPN and MPN + self-supervision, respectively. As can be seen, our method produces the sharpest and most realistic facial details.



**Fig. 7:** Self-supervised training step. The consistency loss is applied between the augmented deblurred images of vanilla images and the deblurred outputs of augmented counterparts, thus promoting the robustness to various transformations and noises during training and inference. Functions  $v(\cdot)$  denote a chosen augmentation. Note that the self-supervised loss does not enforce the ground truth on the output which limits overfitting.

### 3.5 Boosting MPN with Self-supervision

Below we introduce the self-supervision to boost the performance and robustness of our MPN approach. Figure 7 illustrates the self-supervised aspect of our pipeline, for which we investigate the impact of rotations, scale transform and Gaussian noise augmentations.

**Rotations.** One natural property of blur kernels is the invariance to rotations, which enhances the robustness and generalization ability in realistic scenarios. However, previous works [28, 49, 62] do not guarantee such a property even though they use the random rotation augmentations. Thus, we propose to improve the robustness of MPN to rotations by promoting the consistency between original and rotated restored outputs in a self-supervised manner.

Let  $\mathbf{B}_j$  be a blurred image,  $\text{Deblur}(\cdot)$  be a deblurring network, *e.g.*, MPN or StackMPN,  $\text{Rot}(\cdot)$  be the rotation func-

tion (with random choice of rotation by  $90^\circ$ ,  $180^\circ$ , or  $270^\circ$ ). The self-supervised loss  $L_{ss}$  is defined as:

$$\mathbf{S}_{Nj} = \text{Deblur}(\mathbf{B}_j), \quad (19)$$

$$\mathbf{S}'_{Nj} = \text{Deblur}(v(\mathbf{B}_j)), \quad (20)$$

$$\mathcal{L}_{ss} = \sum_j ||v(\mathbf{S}_{Nj}) - \mathbf{S}'_{Nj}||_F^2, \quad (21)$$

where  $j$  loops over a subset of images,  $N$  is the number of stacked levels of network,  $\text{Deblur}(\cdot)$  is the output from level  $N$  of stacked network, whereas  $v(\cdot)$  performs a chosen augmentation, *e.g.*,  $\text{Rot}(\cdot)$ .

In this manner, the deblurring network is exposed to a variety of orientations and can capture the rotation-invariant blur kernels. With the rotation-based self-supervised loss term, our final objective of MPN is defined as follows:

$$\mathcal{L} = \mathcal{L}_{\text{deblur}} + \alpha \mathcal{L}_{ss}, \quad (22)$$



**Table 1:** Quantitative analysis of our model on the GoPro dataset [28]. *Size* and *Runtime* are expressed in MB and milliseconds. The reported time is the CNN runtime (writing generated images to disk is not considered). Note that we employ (1-2-4) multi-patch architecture for StackMPN and E-StackMPN. We did not investigate deeper stacking networks due to the GPU memory limits and long training times.

Models	PSNR	SSIM	Size	Runtime
♣ Sun <i>et al.</i> [48]	24.64	0.843	54.1	12000
♣ Nah <i>et al.</i> [28]	29.23	0.916	303.6	4300
♣ Zhang <i>et al.</i> [62]	29.19	0.931	37.1	1400
♣ Tao <i>et al.</i> [49]	30.10	0.932	33.6	1600
♣ Gao <i>et al.</i> [11]	30.92	0.942	2.8	N/A
♣ Park <i>et al.</i> [34]	31.15	0.945	2.6	70
♠ Nah <i>et al.</i> [29]	29.97	0.895	N/A	34.7
♠ DBGAN [63]	31.10	0.943	11.8	N/A
♠ Pan <i>et al.</i> [33]	29.06	0.943	N/A	N/A
♠ eSL-Net [52]	30.23	0.870	N/A	N/A
♠ Jiang <i>et al.</i> [18]	31.79	0.949	N/A	N/A
♠ Pan <i>et al.</i> [31]	31.89	0.921	286	6500
♠ Xiang <i>et al.</i> [56]	32.63	0.935	61.8	6000
(@ indicates the number of RGB frames used)				
♣ MPN	30.21	0.935	21.7	17
♣ StackMPN	31.16	0.945	65.1	233
♠ MPN @3	30.68	0.940		
♠ MPN @5	30.89	0.941	21.7	17
♠ MPN @7	30.58	0.939		
♠ StackMPN @5	31.63	<b>0.951</b>	65.1	233
♠ E-MPN	33.14	0.937	64.7	121
♠ E-MPN @5	33.24	0.936	64.7	121
♠ E-StackMPN	33.56	0.939	118.2	338
♠ E-StackMPN @5	<b>33.83</b>	0.941	118.2	338

♣: single image deblur; ♠: video deblur; ♠: event-based deblur.

where  $\alpha \geq 0$  is the hyper-parameter to tune.

**Scale Transformations.** Another well-established property of blurry kernels is their consistency for different scale inputs. Recent works [28, 62] exploit the multi-scale inputs for image deblurring. As discussed in Section 2, the improvement from such a design is not significant compared to our multi-patch network. Ideally, one might replace each level of MPN with a multi-scale architecture, thus making it a multi-scale multi-patch network, denoted as ‘MPN+MSN’ in Table 3, to simultaneously capture non-uniform kernels from different scales and locations. However, such a design is extremely costly w.r.t. network parameters, training overheads and inference time, thus not practical.

Instead, we follow the self-supervision step designed in a similar spirit to the rotation-based self-supervision step, with the goal of promoting the consistency between scales, and capturing multi-scale information in an efficient self-supervised manner. The difference compared to the rotation-based self-supervision is that instead of batch of rotated images, we randomly downsample or upsample the original blurry images, and simultaneously feed the original input and images at different scales to MPN to obtain their outputs. We follow Eq. 21 and 22 but simply use a scale augmenting function  $\text{Scale}(\cdot)$  in the place of  $\text{Rot}(\cdot)$ .

**Table 2:** Ablations on the performance of hierarchical architecture.

Models	PSNR	SSIM	Size	Runtime
MPN(1)	28.70	0.9131	7.2	5
MPN(1-2)	29.77	0.9286	14.5	9
MPN(1-1-1)	28.11	0.9041	21.7	12
MPN(1-2-4)	30.21	0.9345	21.7	17
MPN(1-4-16)	29.15	0.9217	21.7	92
MPN(1-2-4-8)	<b>30.25</b>	<b>0.9351</b>	29.0	30
MPN(1-2-4-8-16)	29.87	0.9305	36.2	101

**Gaussian Noise.** The robustness to an additive pixel noise drawn from the Normal distribution is a desired property to equip a deblurring model with. Previous works [28, 49, 62], including our MPN, have no built-in robustness to Gaussian noises. To demonstrate this point, Table 6 shows that the inference performance sharply decreases once small-valued noises are injected into the blurring test images. Previous works randomly apply noises to blurred images and use them during training. Thus, the model is expected to deal with deblurring and denoising simultaneously.

## 4 Experiments

Below, we present experimental evaluations of several variants of MPN. Firstly, we introduce datasets we use.

### 4.1 Datasets

We train/evaluate our methods on several versions of the GoPro dataset [28] and the VideoDeblurring dataset [46], and perform qualitative analysis on the realistic blurry images [33] to visually compare the deblurring ability of each model. Lastly, we capture some realistic heavily blurred images consisting of both camera motion and object motion to further justify real-life the effectiveness of each method.

**GoPro** dataset [28] consists of 3214 pairs of blurred and clean images extracted from 33 sequences at  $720 \times 1280$  resolution. The blurred images are generated by averaging varying number (7–13) of successive latent frames to produce varied blurs. For a fair comparison, we follow the protocol in [28], which uses 2103 image pairs for training and the remaining 1111 pairs for testing.

**VideoDeblurring** dataset [46] contains videos captured by various devices, such as iPhone, GoPro and Nexus. The quantitative part has 71 videos. Every video consists of 100 frames at  $720 \times 1280$  resolution. Following the setup in [46], we use 61 videos for training and the remaining 10 videos for testing. In addition, we evaluate the model trained on the GoPro dataset [28] on the VideoDeblurring dataset to demonstrate the generalization ability of our method.

**Table 3:** The baseline performance of multi-scale and multi-patch methods on the GoPro dataset [28]. MSN is a baseline that uses our encoder and decoder following the design of [28] (refer to baselines from Section 4 for details). Note that MSN(1) and MPN(1) are in fact the same model.

Models	PSNR	SSIM	Runtime
Nah <i>et al.</i> [28]	29.23	0.9162	4300
MSN(1) MPN(1)	28.70	0.9131	4
MSN(2)	28.82	0.9156	21
MPN(1-2)	29.77	0.9286	9
MSN(3)	28.97	0.9178	27
MPN(1-2-4)	30.21	0.9345	17
MPN + MSN	30.34	0.9351	523

## 4.2 Evaluation Setup and Results

We feed the original high-resolution  $720 \times 1280$  pixel images into MPN. The PSNR, SSIM, model size and runtime are reported in Table 1 for an in-depth comparison with competing state-of-the-art motion deblurring models. For the stacking networks, we employ the (1-2-4) multi-patch architecture in every model unit, considering the runtime and difficulty of training.

For the stacked model, the output of every sub-model is optimized level-by-level, which means the first output has the poorest quality and the last output achieves the best performance. Fig. 8 presents the outputs of Stack(3)-MPN (3 sub-models stacked together) to demonstrate that each sub-model gradually improves the quality of deblurring.

## 4.3 Implementation Details

All our experiments are implemented in PyTorch and evaluated on a single NVIDIA Tesla P100. To train MPN,



**Fig. 8:** Outputs of different sub-models of Stack(3)-MPN. From left to right are the outputs of  $M_1$  to  $M_3$ . The clarity of results improves level-by-level.

we randomly crop images to  $256 \times 256$  pixel size. Subsequently, we extract patches from the cropped images and forward them to the inputs of each level. The batch size is set to 6 during training. The Adam solver [19] is used to train our models for 3000 epochs. The initial learning rate is set to 0.0001 and the decay rate to 0.1. We normalize image to range  $[0, 1]$  and subtract 0.5.

**Performance.** Table 1 shows that our proposed MPN outperforms other competing methods according to PSNR and SSIM measures, which demonstrates the superiority of non-uniform blur removal via the localized information our model uses. The deepest MPN we trained and evaluated is (1-2-4-8-16) due to the GPU memory limitation. The best performance is obtained with the (1-2-4-8) model, for which PSNR and SSIM are higher compared to all current state-of-the-art models. Note that our model is simpler than other competing approaches, *e.g.*, we do not use recurrent units. We note that patches that are overly small (below  $1/16$  size) are not helpful in removing the motion blur.

Moreover, the stacked variant, StackMPN, outperforms shallower MPN by around 1.0dB PSNR. SSIM scores indicate the same trend. The performance of StackMPN can be improved by serially stacking more MPN units, which is consistent with our expectations.

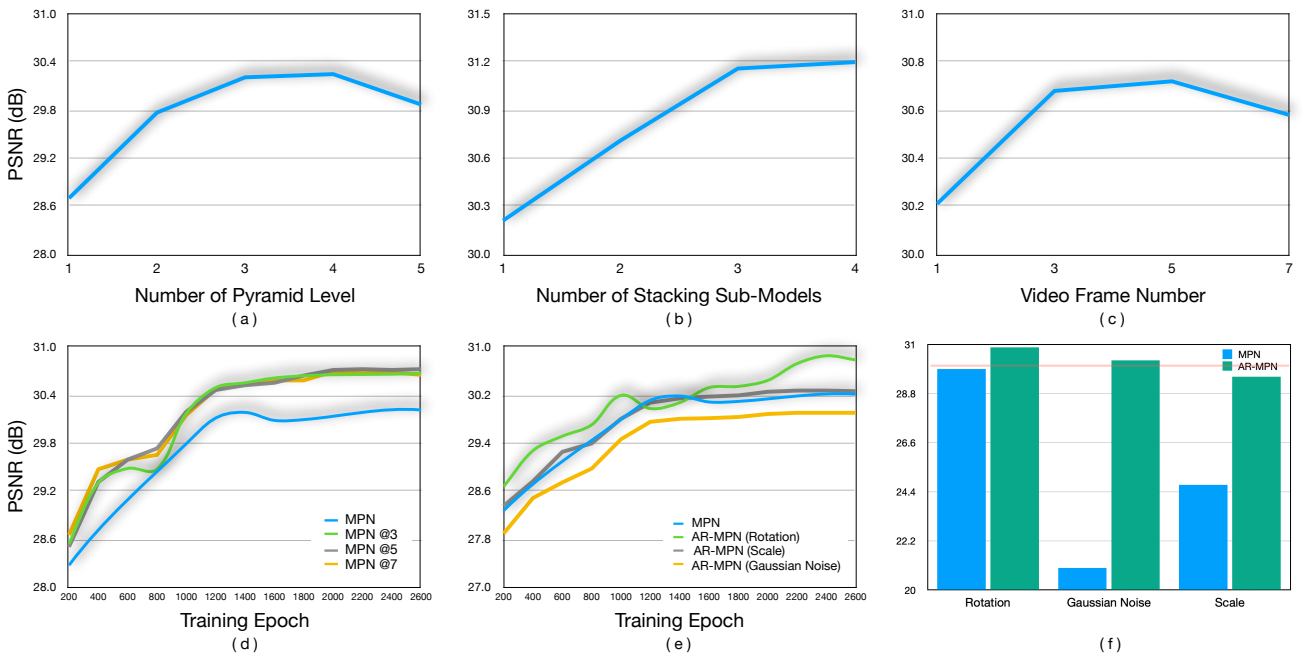
For video deblurring, using multi-frame inputs does not affect the runtime significantly but it improves the PSNR by 0.41dB. However, as we do not investigate advanced strategies for processing multiple frames, which is out of our focus in this paper, the performance cannot be continuously improved by simply using a larger number of frames. The optimal performance is achieved by using 5 blurry frames.

For event-guided deblurring, we observe that the performance of both image deblurring and video deblurring is significantly boosted. To demonstrate this point, E-MPN achieves 33.14dB on the GoPro dataset, which outperforms the MPN by up to  $\sim 2.9$ dB. Similar trend is also observed on video deblurring, which is consistent with our theoretical analysis that associating the event information with blurry images as a composite input to the pipeline should capture accurate motion information, helping the model achieve a better deblurring performance. When we placed the output of the popular TV-L1 optical flow/pretrained FlowNet in place of ‘event representation’ in our pipeline, results of E-StackMPN dropped from 32.57dB to 32.01dB/32.07dB (which is still better than results of [31, 56]) but worse than results of E-StackMPN with ‘event representation’. While event information may be captured with an extra hardware such as an event camera, the event information used on GoPro dataset in our experiments is entirely simulated from RGB frames by Esim [36], which comprises a rendering engine rather than an event camera for ground truth labelling. In case an event camera is available, the quantitative per-

**Table 4:** Quantitative analysis (PSNR) on the VideoDeblurring dataset [46] for models trained on the GoPro dataset. PSDeblur means using Photoshop CC 2015. We select the “single frame” version of approach [46] for fair comparisons.

Methods	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	Average
PSDeblur [46]	24.42	28.77	25.15	27.77	22.02	25.74	26.11	19.75	26.48	24.62	25.08
WFA [5]	25.89	32.33	28.97	28.36	23.99	31.09	28.58	24.78	31.30	28.20	28.35
Su <i>et al.</i> [46]	25.75	31.15	29.30	28.38	23.63	30.70	29.23	25.62	31.92	28.06	28.37
Nah <i>et al.</i> [29]	-	-	-	-	-	-	-	-	-	-	30.80
STFAN [64]	-	-	-	-	-	-	-	-	-	-	31.24
Xiang <i>et al.</i> [56]	-	-	-	-	-	-	-	-	-	-	31.68
Pan <i>et al.</i> [31]	-	-	-	-	-	-	-	-	-	-	31.67
MPN	29.89	33.35	31.82	31.32	26.35	32.49	30.51	27.11	34.77	30.02	30.76
StackMPN	30.48	34.31	32.24	32.09	26.77	33.08	30.84	27.51	35.24	30.57	31.39
E-MPN	31.32	34.73	33.21	32.68	27.85	33.81	31.83	28.46	36.09	31.45	32.14
E-StackMPN	<b>31.68</b>	<b>35.81</b>	<b>33.35</b>	<b>33.17</b>	<b>28.32</b>	<b>34.17</b>	<b>32.15</b>	<b>29.01</b>	<b>36.27</b>	<b>31.75</b>	<b>32.57</b>

♣: single image deblur; ♦: video deblur; ♣: event-based deblur.

**Fig. 9:** Ablation studies. PSNR w.r.t. (a) the number of pyramid levels, (b) stacking units, (c) the number of frames concatenated. Moreover, we show PSNR w.r.t. to (d) the number of video frames used (indicated by @) by MPN, and (e) the type of augmentations as a function of epoch number. Finally, (f) compares the final performance of MPN alone vs. MPN with different self-supervision strategies.**Table 5:** Evaluations of the weight sharing scheme on GoPro [28].

Models	PSNR	SSIM	Size (MB)
MPN(1-2)	29.77	0.9286	14.5
MPN(1-2)-WS	29.22	0.9210	7.2
MPN(1-2-4)	30.21	0.9343	21.7
MPN(1-2-4)-WS	29.56	0.9257	7.2
MPN(1-2-4-8)	30.25	0.9351	29.0
MPN(1-2-4-8)-WS	30.04	0.9318	7.2

formance of our E-MPN should improve further due to the highest quality of event information in such a case.

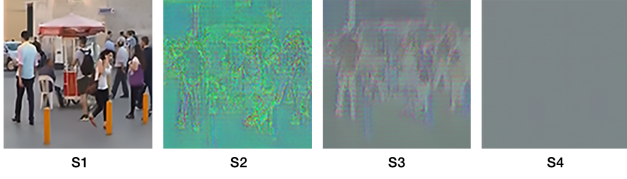
Another downside of using the optical flow is that it encodes the displacement information rather than the change information, *etc.* Event models can cope with fast motions

**Table 6:** Ablation study of the robustness of MPN vs. MPN models with self-supervision under different transforms/noises on GoPro [28]. The robustness to augmentations is measured by randomly applying transformations or noises on test images to measure the PSNR.

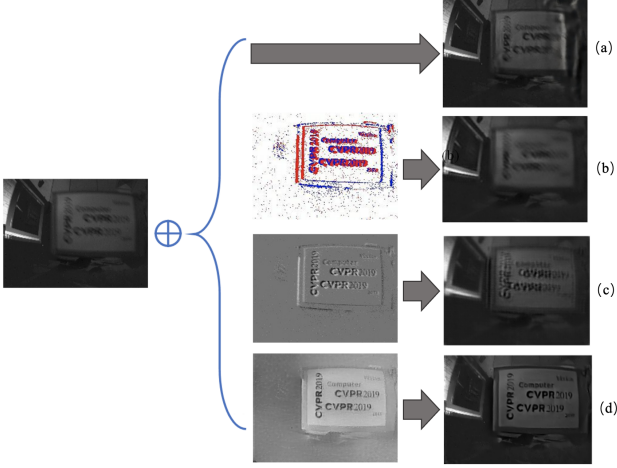
Type of Aug.	None	Rot.	Gauss.	Scale
MPN	30.24	29.89	20.98	24.71
MPN + Rand. Aug.	30.01	29.90	28.05	28.31
MPN + SS (Rot.)	30.85	30.86	-	-
MPN + SS (Gauss.)	30.46	-	30.31	-
MPN + SS (Scale)	30.35	-	-	29.49
MPN + SS (Mix)	30.92	30.91	30.35	29.91

by design, whereas optical flow algorithms are known to fail under large displacement.

The deblurred images from the GoPro dataset are shown in Figures 6, 12 and 9. Specifically, Figure 6 shows the de-



**Fig. 10:** Outputs  $S_i$  for different levels of MPN(1-2-4-8). Images from right to left visualize bottom level  $S_4$  to top level  $S_1$ .



**Fig. 11:** Various strategies of injecting events into our MPN: (a) directly deblurring of original blurred inputs, (b) deblurring of original blurred input combined with the accumulation of events along the temporal mode (both from the input), (c) deblurring on ‘original blurred image + event voxel’, and (d) deblurring model using the concatenation of original blurred image and Event-to-Video representation.

blurring performance of several models on an image containing heavy a motion blur. We zoom in the main object for clarity. Figure 12 shows selected images of different scenes to demonstrate the advantages of our model which produces the sharpest details across all cases. In addition, we present the deblurring performance on realistic blurry images in Figure 13 to show the benefit of our E-MPN, which clearly outperforms previous deep models in such a scenario. Figure 14 presents the performance comparison on realistic heavily blurred images consisting of camera and object motions. Our pipeline achieves better deblurring compared to the baseline models.

**Runtime.** In addition to the superior PSNR and SSIM of our model, to the best of our knowledge, MPN is also the first deep deblurring model that can work real-time. For example, MPN (1-2-4-8) takes 30ms to process a  $720 \times 1280$  image, which means it supports real-time 720p image deblurring at 30fps. However, there are runtime overheads related to I/O operations, so real-time deblurring applications require fast transfers from a video grabber to GPU, larger GPU memory and/or an SSD drive, etc.

The following factors contribute to our fast runtime: i) shallower encoder-decoder with small-size convolutional filters; ii) removal of unnecessary links, e.g., skip or recurrent connections; iii) reduced number of upsampling/deconvolution between convolutional features of different levels.

**Baseline Comparisons.** Despite our model has a much better performance than the multi-scale model [28], it is a somewhat unfair comparison as network architectures of our proposed model and [28] differ significantly. Compared with [28], which uses over 303.6MB parameters, we apply much shallower CNN encoders and decoders with the model size  $10 \times$  smaller. Thus, we create a deep Multi-Scale Network (MSN) that uses our encoder-decoder following the setup in [28] for the baseline comparison (sanity check) between multi-patch and multi-scale methods. As shown in Table 3, the PSNR of MSN is worse than [28], which is expected due to our simplified CNN architecture. Compared with our MPN, the best result obtained with MSN is worse than the MPN(1-2) model. Due to the common testbed, the reported performance of MSN and MPN is the fair comparison of the multi-patch hierarchical and multi-scale models [28].

#### 4.4 Ablation Studies

We visualize the outputs of our MPN unit in Figure 10 to analyze intermediate contributions. As previously alluded to, MPN uses the residual design. Thus, finer levels contain finer but visually less important information compared to the coarser levels. In Fig. 10, we illustrate outputs  $S_i$  of each level of MPN (1-2-4-8). The information contained in  $S_4$  is the finest and most sparse. The outputs become less sparse, sharper and richer in color as we move up level-by-level in MPN.

**Weight Sharing over Each Level.** Below, we investigate weight sharing between the encoder-decoder pairs of all levels of our network to reduce the number of parameters. Table 5 shows that weight sharing results in a slight loss of performance but reduces the number of parameters significantly.

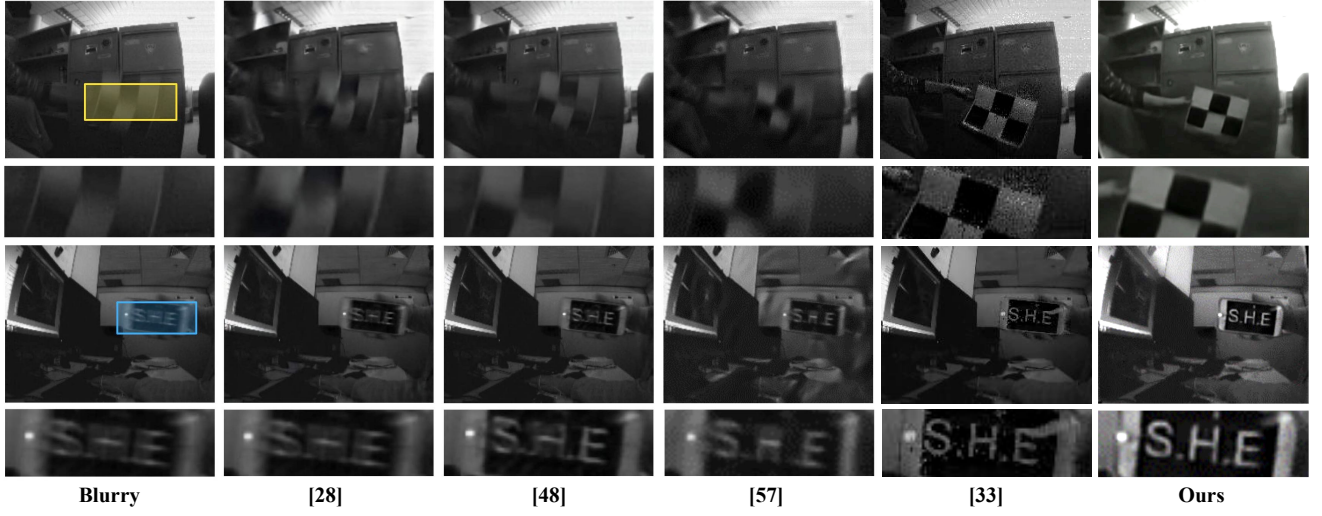
**The Need for Event Pre-processing.** Our network uses the so-called ‘events-to-video’ simulating network to improve the aggregation of event information and RGB frames. To justify its necessity, we compare our E-MPN with two recent deep event-guided deblurring models [52, 44], and we find that adding a front-end network to pre-process event voxels is more effective at extracting the motion information from events than other models. Alternatively, a back-end module is required to help compensate for the performance loss but such a module is not easily explainable in the context of using events.

**Various Strategies of Injecting Events.** To justify the necessity of using Event-to-Videos network in our MPN pipeline,

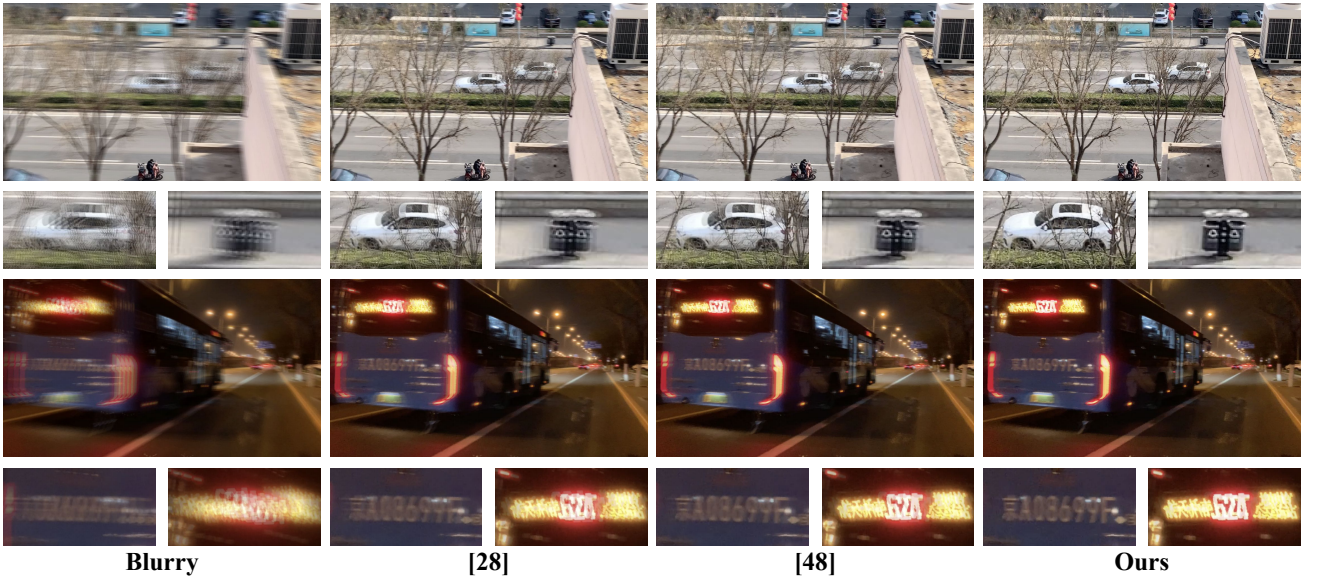




**Fig. 12:** Deblurring performance on the blurry images from the GoPro and the VideoDeblurring datasets. We crop regions indicated by blue and red bounding boxes, and obtain smaller windows as follows. The first column contains the original blurry crops, the second column is the result of [28], the third column is the result of [49]. Our results are presented in the last column. We argue that our model achieves the best visual performance across several different scenes.



**Fig. 13:** The qualitative comparisons of deblurring performance on blurry images [33]. The first column contains original blurry images, the second column is obtained with approach [28], the third column is obtained with approach [49], the fourth column is obtained with approach [33]. Our results are presented in the last two columns. The first three deep deblurring models (from left to right) perform poorly when dealing with complex blurs, whereas applying the event representation E-MPN produces crisp images.



**Fig. 14:** The qualitative comparisons of deblurring performance on realistic blurry images. The first column contains original blurry images. The second column is obtained with approach [28], the third column is obtained with approach [49]. Our results are presented in the last column.

we perform ablation studies w.r.t. different strategies of injecting events. Figure 11 shows that the last model (concatenation of the original blurred image and Event-to-Video representation) significantly outperforms other baselines in terms of the qualitative comparison, which we attribute to the fact that the Event-to-Videos network decodes events to represent the underlying event/motion dynamics within the image domain rather than the event domain. It is unreasonable to expect a deblurring pipeline could learn decode events by itself, and for that very reason we employ the specialized Event-to-Videos network.

**Self-supervised Pipelines.** Combining our proposed self-supervision step with MPN improves the robustness w.r.t. different geometric transformations and photometric noises, as shown in Table 6. For example, applying a low-level additive Gaussian noise on original blurred images during testing decreases the deblurring performance of the original model down to 20.98dB, which demonstrates that the original model cannot deblur noisy images. Once the Gaussian noise self-supervision step is applied, deblurring performance reaches 30.31dB. Applying rotations as self-supervisory task in MPN brings around 0.6dB PSNR improvement on GoPro dataset.



## 5 Conclusions

In this paper, we address the challenging problem of non-uniform motion deblurring by exploiting the multi-patch model as opposed to the widely used multi-scale and scale-recurrent architectures. To this end, we have devised an end-to-end deep multi-patch hierarchical deblurring network. Compared against existing deep deblurring frameworks, our model achieves the state-of-the-art performance (according to PSNR and SSIM) and is able to run at 30fps for 720p images. To overcome the discrepancy between adjacent patch boundaries, we explicitly minimize the  $\ell_2$  metric between these boundaries to promote the global consistency of patches. Our stacked variants StackMPN further improve results over both shallower MPN and competing approaches while being  $\sim 4\times$  faster than the latter models. Our stacking architecture appears to have overcome the limitation to stacking depth which other competing approaches exhibit. Moreover, the novel self-supervised mechanism proposed by us improve the model ability to cope with geometric transformations and photometric noises. Finally, exploiting the camera event representation together with blurred images results in the largest improvements on frames containing complex blur patterns. We hope our work provides several valuable insights for subsequent works on deblurring.

**Acknowledgements.** This research is supported by the Natural Science Foundation of China (Grant No. 62106282).

Code: <https://github.com/HongguangZhang/DMPHN-cvpr19-master>.

**Data availability statement:** All datasets used and studied in this paper are publicly available.

## References

1. Bardow, P., Davison, A.J., Leutenegger, S.: Simultaneous optical flow and intensity estimation from an event camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 884–892 (2016) [4](#)
2. Barua, S., Miyatani, Y., Veeraraghavan, A.: Direct face detection and video reconstruction from event cameras. In: 2016 IEEE winter conference on applications of computer vision (WACV), pp. 1–9. IEEE (2016) [4](#)
3. Brandli, C., Berner, R., Yang, M., Liu, S.C., Delbruck, T.: A 240×180 130 db 3  $\mu$ s latency global shutter spatiotemporal vision sensor. *IEEE Journal of Solid-State Circuits* **49**(10), 2333–2341 (2014) [3, 4](#)
4. Cho, S., Lee, S.: Fast motion deblurring. *ACM Transactions on graphics* **28**(5), 145:1–145:8 (2009) [3](#)
5. Delbracio, M., Sapiro, G.: Hand-held video deblurring via efficient fourier aggregation. *IEEE Transactions on Computational Imaging* **1**(4), 270–283 (2015). DOI 10.1109/TCI.2015.2501245 [11](#)
6. Dipta Das, S., Dutta, S.: Fast deep multi-patch hierarchical network for nonhomogeneous image dehazing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops, pp. 482–483 (2020) [3](#)
7. Doersch, C., Gupta, A., Efros, A.A.: Unsupervised visual representation learning by context prediction. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1422–1430 (2015) [4](#)
8. Dosovitskiy, A., Springenberg, J.T., Riedmiller, M., Brox, T.: Discriminative unsupervised feature learning with convolutional neural networks. In: Advances in neural information processing systems, pp. 766–774 (2014) [4](#)
9. Fernando, B., Bilen, H., Gavves, E., Gould, S.: Self-supervised video representation learning with odd-one-out networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 3636–3645 (2017) [4](#)
10. Gan, C., Gong, B., Liu, K., Su, H., Guibas, L.J.: Geometry guided convolutional neural networks for self-supervised video representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 5589–5597 (2018) [4](#)
11. Gao, H., Tao, X., Shen, X., Jia, J.: Dynamic scene deblurring with parameter selective sharing and nested skip connections. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019) [3, 9](#)
12. Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., Cord, M.: Boosting few-shot visual learning with self-supervision. *arXiv preprint arXiv:1906.05186* (2019) [4](#)
13. Gidaris, S., Singh, P., Komodakis, N.: Unsupervised representation learning by predicting image rotations. *arXiv preprint arXiv:1803.07728* (2018) [4](#)
14. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: Proc. Eur. Conf. Comp. Vis., pp. 346–361. Springer (2014) [2](#)
15. Hyun Kim, T., Mu Lee, K.: Generalized video deblurring for dynamic scenes. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pp. 5426–5434 (2015) [3](#)
16. Jia, J.: Single image motion deblurring using transparency. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pp. 1–8. IEEE (2007) [3](#)
17. Jia, J.: Mathematical models and practical solvers for uniform motion deblurring. (2014) [3](#)
18. Jiang, Z., Zhang, Y., Zou, D., Ren, J., Lv, J., Liu, Y.: Learning event-based motion deblurring. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2020) [4, 9](#)
19. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014) [10](#)
20. Koniusz, P., Zhang, H., Porikli, F.: A deeper look at power normalizations. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pp. 5774–5783 (2018) [2, 5](#)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proc. Adv. Neural Inf. Process. Syst., pp. 1097–1105 (2012) [1](#)
22. Kupyn, O., Budzan, V., Mykhailych, M., Mishkin, D., Matas, J.: Deblurgan: Blind motion deblurring using conditional adversarial networks. *arXiv preprint arXiv:1711.07064* (2017) [3](#)
23. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pp. 2169–2178. IEEE (2006) [2, 5](#)
24. Levin, A.: Blind motion deblurring using image statistics. In: Proc. Adv. Neural Inf. Process. Syst., pp. 841–848 (2007) [3](#)
25. Lichtsteiner, P., Posch, C., Delbruck, T.: A 128×128 120 db 15  $\mu$ s latency asynchronous temporal contrast vision sensor. *IEEE Journal of Solid-State Circuits* **43**(2), 566–576 (2008) [3](#)
26. Lu, X., Lin, Z., Shen, X., Mech, R., Wang, J.Z.: Deep multi-patch aggregation network for image style, aesthetics, and quality estimation. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pp. 990–998 (2015) [2](#)

27. Munda, G., Reinbacher, C., Pock, T.: Real-time intensity-image reconstruction for event cameras using manifold regularisation. *International Journal of Computer Vision* **126**(12), 1381–1393 (2018) [4](#)
28. Nah, S., Kim, T.H., Lee, K.M.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 257 – 265 (2017) [1](#), [2](#), [3](#), [5](#), [6](#), [8](#), [9](#), [10](#), [11](#), [12](#), [13](#), [14](#)
29. Nah, S., Son, S., Lee, K.M.: Recurrent neural networks with intra-frame iterations for video deblurring. In: *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2019) [3](#), [9](#), [11](#)
30. Nimisha, T.M., Singh, A.K., Rajagopalan, A.N.: Blur-invariant deep learning for blind-deblurring. In: *Proc. IEEE Int. Conf. Comp. Vis.*, pp. 4762–4770 (2017) [3](#)
31. Pan, J., Bai, H., Tang, J.: Cascaded deep video deblurring using temporal sharpness prior. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3043–3051 (2020) [9](#), [10](#), [11](#)
32. Pan, L., Dai, Y., Liu, M., Porikli, F.: Simultaneous stereo video deblurring and scene flow estimation. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 6987–6996. *IEEE* (2017) [1](#)
33. Pan, L., Scheerlinck, C., Yu, X., Hartley, R., Liu, M., Dai, Y.: Bringing a blurry frame alive at high frame-rate with an event camera. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6820–6829 (2019) [4](#), [7](#), [9](#), [14](#)
34. Park, D., Kang, D.U., Kim, J., Chun, S.Y.: Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In: *European Conference on Computer Vision*, pp. 327–343. Springer (2020) [9](#)
35. Rajagopalan, A.N., Chellappa, R.: *Motion Deblurring: Algorithms and Systems*. Cambridge University Press (2014) [3](#)
36. Rebecq, H., Gehrig, D., Scaramuzza, D.: Esim: an open event camera simulator. In: *Conference on Robot Learning*, pp. 969–982. PMLR (2018) [7](#), [10](#)
37. Rebecq, H., Ranftl, R., Koltun, V., Scaramuzza, D.: Events-to-video: Bringing modern computer vision to event cameras. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3857–3866 (2019) [4](#), [7](#)
38. Scheerlinck, C., Barnes, N., Mahony, R.: Continuous-time intensity estimation using event cameras. In: *Asian Conference on Computer Vision*, pp. 308–324. Springer (2018) [4](#)
39. Schuler, C., Hirsch, M., Harmeling, S., Scholkopf, B.: Learning to deblur. *IEEE Trans. Pattern Anal. Mach. Intell.* (7), 1439–1451 (2016) [3](#)
40. Sellent, A., Rother, C., Roth, S.: Stereo video deblurring. In: *Proc. Eur. Conf. Comp. Vis.*, pp. 558–575. Springer (2016) [3](#)
41. Sermanet, P., Lynch, C., Chebotar, Y., Hsu, J., Jang, E., Schaal, S., Levine, S.: Time-contrastive networks: Self-supervised learning from pixels (2017) [4](#)
42. Simon, C., Koniusz, P., Harandi, M.: On learning the geodesic path for incremental learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (2021) [4](#)
43. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014) [1](#)
44. Song, C., Huang, Q., Bajaj, C.: E-cir: Event-enhanced continuous intensity recovery. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7803–7812 (2022) [4](#), [12](#)
45. Su, J.C., Maji, S., Hariharan, B.: Boosting supervision with self-supervision for few-shot learning. *arXiv preprint arXiv:1906.07079* (2019) [4](#)
46. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.* (2017) [3](#), [9](#), [11](#)
47. Suin, M., Purohit, K., Rajagopalan, A.: Spatially-attentive patch-hierarchical network for adaptive motion deblurring. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3606–3615 (2020) [3](#)
48. Sun, J., Cao, W., Xu, Z., Ponce, J.: Learning a convolutional neural network for non-uniform motion blur removal. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 769–777 (2015) [2](#), [3](#), [9](#)
49. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 8174–8182 (2018) [1](#), [2](#), [3](#), [5](#), [6](#), [8](#), [9](#), [13](#), [14](#)
50. Tas, Y., Koniusz, P.: CNN-based action recognition and supervised domain adaptation on 3D body skeletons via kernel feature maps. In: *The British Machine Vision Conference (BMVC)* (2018) [4](#)
51. Tas, Y., Koniusz, P.: Simple dialogue system with AUDITED. In: *The British Machine Vision Conference (BMVC)* (2021) [4](#)
52. Wang, B., He, J., Yu, L., Xia, G.S., Yang, W.: Event enhanced high-quality image recovery. In: *European Conference on Computer Vision*, pp. 155–171. Springer (2020) [4](#), [9](#), [12](#)
53. Wang, L., Koniusz, P.: Self-supervising action recognition by statistical moment and subspace descriptors. In: *The ACM International Conference on Multimedia (ACM MM)* (2021). DOI 10.1145/3474085.3475572 [4](#)
54. Wang, L., Koniusz, P., Huynh, D.Q.: Hallucinating IDT descriptors and I3D optical flow features for action recognition with cnns. In: *The IEEE International Conference on Computer Vision (ICCV)* (2019) [4](#)
55. Wang, Z., Ng, Y., Scheerlinck, C., Mahony, R.: An asynchronous kalman filter for hybrid event cameras. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 448–457 (2021) [4](#)
56. Xiang, X., Wei, H., Pan, J.: Deep video deblurring using sharpness features from exemplars. *IEEE Transactions on Image Processing* **29**, 8976–8987 (2020) [9](#), [10](#), [11](#)
57. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: *Proc. Eur. Conf. Comp. Vis.*, pp. 157–170. Springer (2010) [3](#)
58. Xu, L., Ren, J.S., Liu, C., Jia, J.: Deep convolutional neural network for image deconvolution. In: *Proc. Adv. Neural Inf. Process. Syst.*, pp. 1790–1798 (2014) [3](#)
59. Zhang, H., Dai, Y., Li, H., Koniusz, P.: Deep stacked hierarchical multi-patch network for image deblurring. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019) [1](#), [2](#), [3](#)
60. Zhang, H., Koniusz, P., Jian, S., Li, H., Torr, P.H.S.: Rethinking class relations: Absolute-relative supervised and unsupervised few-shot learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 9432–9441 (2021) [4](#)
61. Zhang, H., Zhang, L., Qi, X., Li, H., Torr, P., Koniusz, P.: Few-shot action recognition with permutation-invariant attention. In: *European Conference on Computer Vision (ECCV)* (2020) [4](#)
62. Zhang, J., Pan, J., Ren, J., Song, Y., Bao, L., Lau, R.W., Yang, M.H.: Dynamic scene deblurring using spatially variant recurrent neural networks. In: *Proc. IEEE Conf. Comp. Vis. Patt. Recogn.*, pp. 2521–2529 (2018) [3](#), [8](#), [9](#)
63. Zhang, K., Luo, W., Zhong, Y., Ma, L., Stenger, B., Liu, W., Li, H.: Deblurring by realistic blurring. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2020) [9](#)
64. Zhou, S., Zhang, J., Pan, J., Xie, H., Zuo, W., Ren, J.: Spatio-temporal filter adaptive network for video deblurring. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2482–2491 (2019) [11](#)