This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2020.3035722, IEEE Transactions on Circuits and Systems for Video Technology



# Recursive Neural Network for Video Deblurring

Xiaoqin Zhang, Runhua Jiang, Tao Wang, Jinxin Wang

***Abstract*—Video deblurring is still a challenging low-level vision task since spatio-temporal characteristics across both the spatial and temporal domains are difficult to model. In this paper, to model the temporal information, we develop a non-local block which estimates inter-frame similarity and inter-frame difference. Specially, for modeling the spatial characteristics and restoring sharp frame details, we propose a recursive block that iteratively refines feature maps generated at the last iteration. In addition, a novel temporal loss function is introduced to ensure the temporal consistency of generated frames. Experimental results on public datasets demonstrate that our method achieves state-of-the-art performance both quantitatively and qualitatively.**

***Index Terms*—Video deblurring, Recursive neural network, Temporal consistency**

## I. Introduction

Videos captured in dynamic environments usually contain blur artifacts caused by camera shake, object movement or out-of-focus. The task of video deblurring aims at restoring the sharp frames from the blurry video. It has become an important research topic because it often serves as a basis for many important tasks such as SLAM [1] and object tracking [2].

Compared with single image deblurring which only requires modeling spatial information of a single image, video deblurring also needs to model the continuous information in the temporal domain. It is still a challenging problem that hinders the improvement of video deblurring methods. Therefore, existing video deblurring methods [3], [4], [5], [6] pay a lot of attention to model the temporal information of blurry input frames, but few of them concern about the temporal consistency among generated frames. Consequently, existing methods often generate flickering results [7] that are shown in Fig. 1. Besides the critical need of accurate temporal information, reducing storage and inference time also plays an important role in deploying video deblurring methods. Overall, there exists three major challenges in deep video deblurring: 1) how to effectively model continuous information in the temporal domain, 2) how to restore sharp frame details with parameters as few as possible, and 3) how to ensure the temporal consistency of generated frames.

**Non-local Block:** A key challenge for video deblurring is how to effectively model the temporal information exists in blurry input frames. Prior-based methods [8], [9], [10], [11] formulate this challenge as a non-convex energy minimization framework of which variables include the global motion and the latent frames, while deep learning methods formulate it as a high-level information acquisition problem. For example, Su *et al.* [3] align a sequence of frames to the middle frame via homography or optical flow. Those aligned frames are then fed into a CNN to handle misplacement caused by large motions between frames. Zhang *et al.* [4] first take three consecutive frames as inputs of 3D convolution layers, then employ stacked residual blocks to capture the dynamic variations. Nah *et al.* [5] deliver information from past frames to the current frame in the form of hidden state, and use an RNN-based model to restore the sharp frame. Although these aforementioned methods have achieved great success in modeling temporal information, they seldom consider inter-frame similarity which is an important component of temporal information.

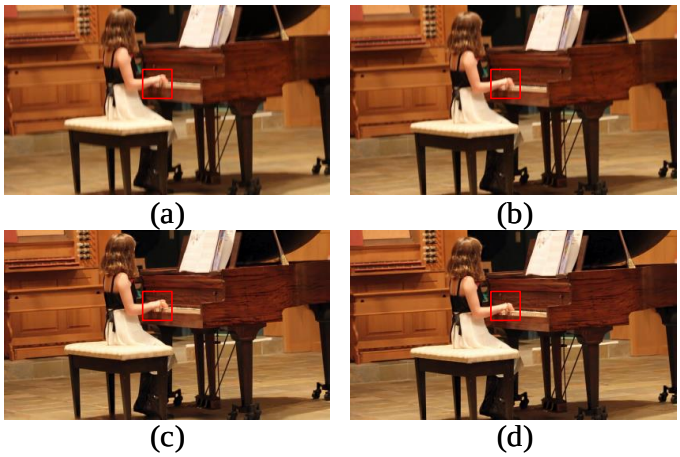



Fig. 1. Flickering appearances of blurry images and deblurred images. (a) and (b) are the real blurry images taken from the VideoDeblurring dataset [3], while (c) and (d) are the deblurred images.

Non-local recurrent neural network [12], which has shown great ability to compute the self-similarity in nature images, can be used to capture temporal information. The rationale behind this conclusion is that the temporal information can be represented by the changing contents (i.e., differences between consecutive frames) and the unchanging contents (i.e., similarities of consecutive frames) in video. Therefore, temporal information can be obtained by estimating the changed and unchanged contents. Specially, the inter-frame difference can be computed by convolution layers as existing methods do [3], [4], and the inter-frame similarity can be obtained by the non-local operation. Based on above analyses, we propose a non-local block which employs the non-local operation and convolution layers to model temporal information composed by inter-frame similarity and inter-frame difference. Compared with existing non-local modules, the proposed non-local block additionally takes the inter-frame difference into consideration

X. Zhang, R. Jiang, T. Wang and J. Wang are with the College of Computer Science and Artificial Intelligence, Wenzhou University, 325035, China (e-mail: {zhangxiaoqinnan, ddghjikle1, taowangzj}@gmail.com, jxwang@stu.wzu.edu.cn). X. Zhang is the corresponding author (E-mail: zhangxiaoqinnan@gmail.com)

and does not require stacking because of its special organization of input frames. More details are presented in Section III-A.

**Recursive Block:** Although existing video deblurring methods have achieved great progress in term of model accuracy, their success strongly relies on complex and deep networks which always cause expensive hardware requirements. To the best of our knowledge, the newest video deblurring method that aims at improving model efficiency (i.e., achieving the same performance with fewer parameters) is IFI-RNN [5], in which the proposed network iteratively updates feature maps via using RNN cells before predicting an output deblurred frame. Although the IFI-RNN improves the model efficiency by using a single RNN cells, the model efficiency of video deblurring methods can be further improved by designing a recursive block.

As discussed by Guo *et al.* [13], the reuse scheme of network layers can make full use of limited parameters and enhance the ability of model presentation. Moreover, Tao *et al.* [14] demonstrate that the reuse scheme can be applied to the single image deblurring problem. Inspired by [13], [14], we find that the model efficiency of video deblurring methods can be further improved by reusing the same block. However, with only these reused parameters, the recursive blocks may suffer from limited receptive filed when modeling spatial information. Therefore, the recursive block is still not employed into video deblurring task. In this work, we address this issue by using a cascade CNN to dynamically modify parameters of the recursive blocks. Overall, the proposed recursive blocks are different from recurrent blocks [13], [14] in two aspects: a) recursive blocks share parameters, while recurrent blocks share hidden states; b) the proposed recursive blocks not only contain a recursive cell (*i.e.*, convolutional LSTM) but also a cascade CNN.

**Temporal Loss Function:** As we mentioned earlier, many video deblurring methods care about modeling the temporal information in the blurry input frames, while few of them consider the temporal consistency of the deblurred outputs. In fact, temporal information existing in outputs which is also referred as temporal consistency has been investigated in other video processing tasks [15], [16]. Methods proposed in [15] first use the optical flow to align output frames, and then compute distinctions between these aligned frames. Bonneel *et al.* [16] propose a gradient-domain technique for the video enhancement task. The gradient-domain technique first infers the temporal regularity from the original unprocessed video, then uses it as a temporal consistency guide to stabilize the processed sequence.

However, as these aforementioned methods are not designed for the video deblurring task, they may suffer from the remaining blur artifacts in the deblurred frames. Inspired by existing methods [4], [17], we propose a temporal loss function to guarantee temporal consistency of generated videos. Unlike perceptual loss [17], which requires a pre-trained VGG model, the temporal loss function computes differences between feature maps generated by the temporal module (i.e., blocks that capture the temporal information about sharp frames), thus does not require extra algorithms. Moreover, the proposed loss function is different from existing methods in the following aspects: a) the proposed loss function aims at supervising temporal information among output frames, while existing methods only focus on temporal information among input frames; b) the proposed loss function is calculated in the feature space, while most existing loss functions are computed in the image space.

The main contributions of this work are summarized as follows:

- We propose a non-local block, which employs non-local operations and convolution layers to capture temporal information composed by the inter-frame similarity and the inter-frame difference.
- We study various types of block recursion for the video deblurring task. In addition, we propose an efficient recursive block to restore sharp frame details.
- We propose a novel loss function to ensure the temporal consistency of generated frames. Specially, the proposed temporal loss function does not need extra algorithms.
- Through comprehensive evaluations, we demonstrate that the proposed network achieves state-fo-the-art performance in terms of accuracy and model efficiency.

The rest of this paper is structured as follows. Section II briefly reviews related works on image deblurring, video deblurring and recursive neural networks. The proposed network is presented in Section III. Experimental results are presented in section IV, and section V is devoted to a conclusion.

## II. Related Work

Our work is closely related to three topics: image deblurring, video deblurring and recursive neural networks. In this section, we briefly review existing methods related to these topics.

### A. Image Deblurring

Image deblurring aims at generating a sharp image from a blurry one. Generally, image deblurring approaches [18] are based on the uniform blur model:

$$I_B = I_S * k + n, \tag{1}$$

where $I_B$ means the blurry image, $I_S$ refers to the sharp image. $k$ is the blur kernel, and $n$ is the additional noise.

Generally, the image deblurring methods can be divided into two categories: non-blind image deblurring [18], [19] and blind image deblurring [20]. Non-blind image deblurring methods assume that the blur kernel $k$ is known in advance, and therefore usually adopt the classical Lucy-Richardson algorithm, which is an iterative algorithm based on Bayesian analysis. On the other hand, blind image deblurring methods are more complicated as they need simultaneously estimate the sharp image $I_S$ and the blur kernel $k$. Therefore, blind image deblurring methods are mainly based on heuristics, image statistics and hypothetical blur kernels. For example, parametric prior models are used to iteratively estimate the motion kernel and the sharp image in [18]. Liu *et al.* [21] later take a pyramid histogram of oriented gradients to estimate blur kernels for modeling the spatially variant blur. In contrast, Zhang *et al.* [22] take a bundle of blur kernels to model the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2020.3035722, IEEE Transactions on Circuits and Systems for Video Technology



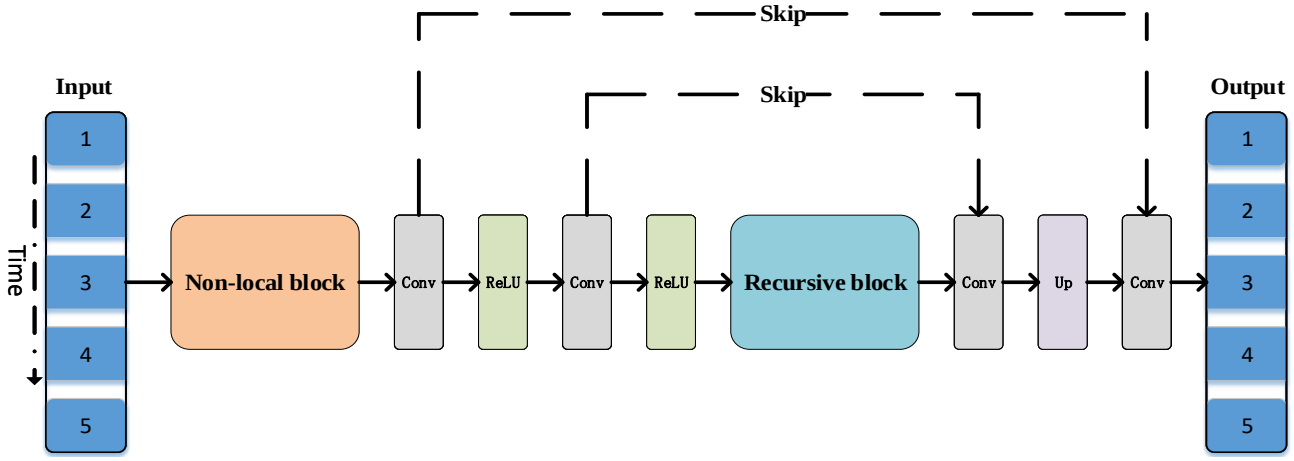



Fig. 2. Overall architecture of the proposed recursive network. The inputs are five consecutive frames, and the outputs are the corresponding five frames.

spatially variant blurs and restore the clear images. Pan *et al.* [9] propose a learning-based approach to estimate the blur kernels for blind image deblurring. Whyte *et al.* [23] employ a parametrized geometric model of the blurring process. Noorozi *et al.* [24] and Nah *et al.* [25] reuse a convolution neural network at multiple scales for deblurring single image. Zhang *et al.* [26] introduce a spatially variant neural network to conduct the image deblurring. Tao *et al.* [14] and Gao *et al.* [27] both propose a multi-scale network with parameter-related schemes to address the image deblurring problem.

### B. Video Deblurring

Compared with image deblurring methods, video deblurring methods consider not only the spatial information in a blurry frame, but also the temporal information among several frames. Li *et al.* [28] first take the temporal information into consideration. Specifically, the video deblurring task is solved by minimizing the energy function based on the multi-image deconvolution in [28]. Paramanand *et al.* [29] conduct the video deblurring task by estimating pixel-wise blur kernels. In [30], a spatio-temporal constraint is proposed to model differences between consecutive frames. Hyun *et al.* [31] consider the effect of depth variations of blur artifacts. Klose *et al.* [32] use pixels at different frames to generate a clear frame. A recurrent neural network is adopted to restore the central frame by learning the spatio-temporal information existing in multiple consecutive frames in [33]. Tan *et al.* [34] introduce a kernel-free method for restoring a sharp frame by estimating the similar contents among continuous frames. DBN [3], which takes multiple frames as the input of the 2D convolution layer, can generate the middle sharp frame. Zhang *et al.* [4] further propose the 3D convolution layer to model the spatio-temporal information. In contrast to [4], [35], a convolution neural network is supervised by a temporal sharpness prior for effectively modeling the temporal relationship [36]. The authors also propose a cascaded training framework to learn compact and effective features. In [37], the deformable convolution operation is taken to learn more representative features, thus the proposed EDVR (Video Restoration framework with Enhanced Deformable Convolutions) achieves better restoration performance than its counterparts. Authors of [38] propose a global spatio-temporal attention module to effectively fuse hierarchical features from past and future frames. On the contrary, Park *et al.* [39] decompose blur artifacts with small local blurs and use the incremental temporal learning to handle these blurs. By considering that blur artifacts are caused by events such as people moving, an effective model is recently proposed in [40] to conduct the video deblurring and interpolation tasks.

### C. Recursive Neural Networks

Due to the importance of modeling the continuous relationship existing in consecutive frames, recurrent neural networks and recursive neural networks have attracted increasingly more attention. The most crucial part of recursive neural networks against recurrent neural networks is that recurrent networks share hidden states along the sequence, while recursive networks share parameters at every layer, which could be considered as a generalization of recurrent networks [13], [41], [42]. Eigen *et al.* [43] propose recursive layers for the image restoration task, but obtain worse performance due to overfitting. Socher *et al.* [44] adopt the recursive blocks when the input dimension is twice that of output. Studies which incorporate recursive connections into convolution neural networks have also shown superiority in several computer vision tasks such as super-resolution [45] and object recognition [46]. Guo *et al.* [13] propose a recursive network which can adaptively decide the loop times of recursive blocks. Zhang *et al.* [47] take the recursive mechanism to efficiently code a single frame. Though the recursive network has achieved remarkable performance in several computer vision tasks [13], [45], [46], it is seldom used to tackle the video deblurring task.

## III. Proposed Approach

As illustrated in Fig. 2, our network is primarily composed of a non-local block, a recursive block and four convolution layers. In this section, we first introduce the non-local block. Then, we illustrate the recursive block and the temporal loss function. At last, we present the whole architecture of our network.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2020.3035722, IEEE Transactions on Circuits and Systems for Video Technology



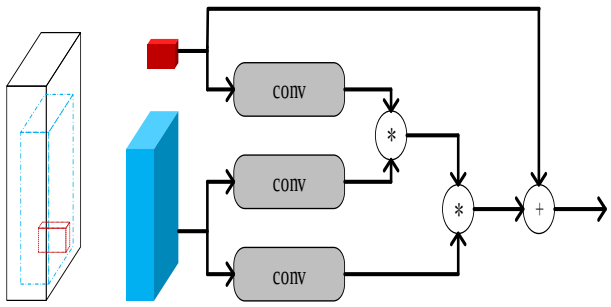



Fig. 3. An illustration of the non-local module proposed in [12]. Tensor with black lines denotes the deep feature map of an entire image. The red tensor indicates the feature maps of a location on the image, while the blue tensor indicates the neighboring features of the red tensor.

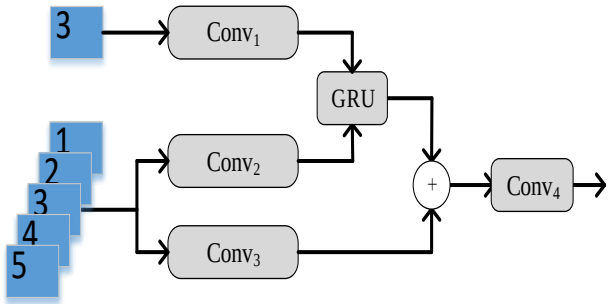



Fig. 4. The proposed non-local block. $Conv_1$ takes the central frame as the input, while $Conv_2$ and $Conv_3$ take all frames as inputs.

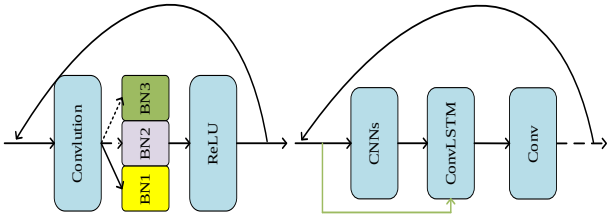



Fig. 5. The left block is the reused block in [13], while the right block is the proposed recursive block. The imaginary line can only go through if the looping is finished. The green line means that the input is also taken as input of another layer.

### A. Non-local Block

Generally speaking, temporal information can be represented by the inter-frame difference and inter-frame similarity as frames in the same video change gradually. Differences among frames can be estimated by convolution layers [3], [4], and similarities can be obtained by a non-local operation that estimates resemblances among inputs [48]. Liu *et al.* [12] first validate that the non-local operation can be formulated by the module illustrated in Fig. 3. Then, they take 12 non-local modules to compose NLRNs (Non-Local Recurrent Networks) and achieve state-of-the-art performance in the field of image restoration. In [48], a video is classified by the Non-local network, which is obtained by inserting non-local operations into 3D convolution networks. For extracting temporal relationship, all input frames are first processed by the 3D convolution layers to generate feature maps. After that, these features are fed into 1, 5 or 10 non-local operations to capture relationship among input frames. According to their experiments, the non-local network with 10 non-local operations always achieves the best performance, but may suffer from lots of parameters [48]. In contrast, this work aims at conducting the video deblurring task, where input frames are more blurry and temporal relationship among these inputs is more complicated [7]. In addition, for most video restoration methods, repeatedly aligning and processing input frames (*i.e.*, functions of 3D convolution layers and non-local operations) are not beneficial to capture temporal information [5]. Therefore, it can be concluded that non-local operations proposed in [48] might not work well on extracting temporal information among blurry frames.

For the above reasons, we propose the non-local block to model temporal information, which can be represented by differences and similarities among several frames. As illustrated in Fig. 4, the non-local block is composed of four convolution layers and a convolutional GRU block [49]. To better define the non-local block, we denote components in this block as $Conv_1, Conv_2, Conv_3, Conv_4$ and $GRU$ which refers to the convolutional GRU block. The input of $Conv_1$ is the central frame of consecutive input frames, while the input of $Conv_2$ and $Conv_3$ are all input frames. Therefore, the function of $GRU$ can be defined as:

$$G = GRU(Conv_1(f_{\frac{1+n}{2}}), Conv_2(f_i)), i = 1, ..., n, \quad (2)$$

where $n$ represents the number of inputs and $G$ represents feature maps generated by $GRU$. Moreover, the operation of $Conv_4$ is defined as

$$C = Conv_4(G + Conv_3(f_i)), i = 1, ..., n. \quad (3)$$

It can be found that, different from existing non-local modules [48], [12], the proposed block organizes the central frame and blurry frames to capture temporal relationship. In addition, as experimental results demonstrate, with only a non-local block, the proposed model can model temporal relationships among the central frame and whole sequence, thus achieves better performance than existing methods [3], [48] that only take convolution layers to capture motions (*e.g.*, inter-frame difference) among consecutive frames.

### B. Recursive Block

Since feature reuse can bring the high-level information back to refine low-level filters, the recursive scheme could iteratively refine the generated feature maps by the same block. Guo *et al.* [13] reuse a block composed of two convolution layers and three batch normalization layers to conduct the image classification task. As illustrated in Fig. 5, the outputs of the first convolution layer go through different batch normalization layers according to the iteration number. However, there are three different batch normalization layers which are not reused at each iteration. Though the three batch normalization layers can reduce the covariance shift [13], we still find that the reuse of high-level information is inadequate since information is processed by different batch normalization layers.

TABLE I
DETAILED CONFIGURATIONS OF THE CNNS. ALL CONVOLUTION LAYERS ARE FOLLOWED BY A RELU LAYER, WHILE THE LAST CONVOLUTION LAYER IS FOLLOWED BY A TANH LAYER. KERNEL SIZES AND PADDING OF ALL CONVOLUTION LAYERS ARE SET TO 3 AND 1, RESPECTIVELY.

| Layers | Operation | Output Channel | Skip |
|---|---|---|---|
| 1 | Conv | 64 | - |
| 2 | Conv | 64 | - |
| 3 | Conv+Maxpool | 128 | - |
| 4 | Conv | 128 | - |
| 5 | Conv+Maxpool | 256 | - |
| 6 | Conv | 256 | - |
| 7 | Conv | 256 | - |
| 8 | Conv | 512 | - |
| 9 | Conv | 512 | - |
| 10 | Conv | 256 | - |
| 11 | Conv+Upsample | 128 | 6 |
| 12 | Conv | 128 | - |
| 13 | Conv+Upsample | 64 | 4 |
| 14 | Conv+Tanh | 64 | - |

TABLE II
CONFIGURATIONS OF VARIANT NON-LOCAL BLOCKS. NUMBERS IN EACH COLUMN INDICATE HOW MANY FRAMES ARE TAKEN AS INPUTS.

| Networks | $Conv_1$ | $Conv_2$ | $Conv_3$ | PSNR | SSIM |
|---|---|---|---|---|---|
| w/o Non | - | - | - | 29.54 | 0.88 |
| SR Non | - | - | - | 27.18 | 0.82 |
| Original Non | - | - | - | 29.65 | 0.88 |
| Non-1 | 3 | 3 | 3 | 29.70 | 0.89 |
| Non-2 | 3 | 5 | 5 | 29.71 | 0.89 |
| RVD | 1 | 5 | 5 | 30.66 | 0.91 |

In this paper, a recursive block which can be presented as the right part of Fig. 5 is proposed to restore details of sharp frames. Inspired by [26], we form the proposed recursive block by a deep CNN (i.e., CNNs), a convolutional LSTM block [50] and a pixel-wise convolution layer. Components of the CNNs are presented in Table I, in which Upsample refers to the bilinear upsampling layer. Specifically, the CNNs takes the reused feature maps as inputs to calculate the useful parts. Then, the convolutional LSTM block and the convolution layer generate feature maps based on the reused feature maps and outputs of the CNNs. Let us denote outputs of the recursive block as $R$ and the looping times as $T$. Then, the function of the recursive block can be represented as follows:

$$R_t = Conv(ConvLSTM(R_{t-1}, CNNs(R_{t-1}))), t = 1, ..., T. \quad (4)$$

At the first iteration, the recursive block takes the feature maps generated by the front layer as inputs. During looping, the CNNs takes outputs of the last iteration as inputs, and the convlutional LSTM block uses outputs of the last iteration and the CNNs as inputs. Note that outputs of CNNs are split equally as the input of the convlutional LSTM block as [26] does. In summary, the recursive block achieves the deblurring task by reusing the three components $T$ times. According to our experiments, the recursive block achieves the best performance when $T$ is equal to 4.

### C. Temporal Loss Function

Unlike existing methods which only concern with the temporal information of inputs, the temporal consistency of generated frames is also a concern in this study. Inspired by perceptual loss which computes differences between feature maps [17], [51], we propose a temporal loss function to ensure the temporal consistency of outputs. The temporal loss function is defined as the following equation:

$$\mathcal{L}_T = \frac{1}{CWH} \sum_{c=1}^{C} \sum_{w=1}^{W} \sum_{h=1}^{H} (f^{sharp} - f^{output})^2 \quad (5)$$

where $f$ refers to the feature maps generated by the non-local block. $f^{sharp}$ and $f^{output}$ are the generated feature maps when the non-local block takes sharp frames and generated frames as input. $C, W$ and $H$ are the channel, width and height of $f$.

By comparing mechanisms of perceptual loss function and the proposed one, it can be found that the proposed temporal loss function has several advantages: a) compared with the perceptual loss function, the temporal loss function is more sensitive to temporal information among output frames; b) because the temporal loss function is based on components of RVD (*i.e.*, the proposed non-local block), it can enforce RVD to fast converge; c) the temporal loss function does not introduce additional parameters, while the perceptual loss function relies on pretrained VGG; d) the temporal loss function can be easily enhanced by designing more effective modules to capture temporal information, while improving the perceptual loss function needs systematically explore high-dimensional features of VGG.

TABLE III
ABLATION STUDY SETTING OF SUB NETWORKS WITH VARIANT RECURSIVE BLOCKS. THE BLOCK COLUMN REFERS TO HOW MANY RECURSIVE BLOCKS ARE USED. THE ITERATION COLUMN REPRESENTS HOW MANY TIMES THIS BLOCK IS REUSED.

| Networks | Block | Iteration | PSNR | SSIM |
|---|---|---|---|---|
| RNNs [26] | 4 | 0 | 30.05 | 0.92 |
| Recursive-1 | 1 | 2 | 29.86 | 0.89 |
| Recursive-2 | 1 | 6 | 29.63 | 0.88 |
| Recursive-3 | 2 | 2 | 30.06 | 0.89 |
| Recursive-4 | 2 | 4 | 29.69 | 0.88 |
| RVD | 1 | 4 | 30.66 | 0.91 |

TABLE IV
METRICS OF SUB NETWORKS WITH DIFFERENT VALUES OF $\alpha$. BOTH OF THE THREE SUB NETWORKS HAVE THE SAME ARCHITECTURE AS RVD.

| Networks | PSNR | SSIM |
|---|---|---|
| Perceptual | 30.07 | 0.89 |
| RVD | 30.66 | 0.91 |

### D. Overall Architecture

As presented in Fig. 2, the proposed network (RVD) mainly contains a non-local block, a recursive block and four convolution layers. Following [4], the inputs are five consecutive frames chosen from the same video, and the outputs are corresponding deblurred frames.

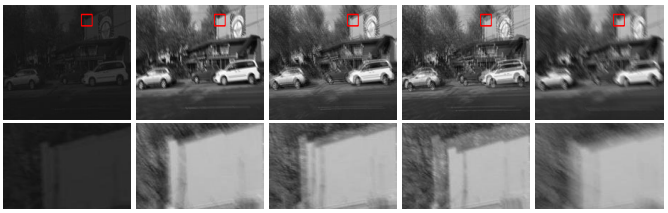
In the non-local block, the kernel size and padding of both $Conv_4$ and $GRU$ are $3 \times 3$ and 1. For $Conv_1$, $Conv_2$ and $Conv_3$, the kernel size is set to $1 \times 1$, and the padding is set to 0. The output channel of convolution layers in the non-local block is set to 32, while the output channel of $Conv_4$ is 64. In the recursive block, all convolution layers in the CNNs and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2020.3035722, IEEE Transactions on Circuits and Systems for Video Technology



(a) Features Generated by $Conv_3$ (b) Standard Deviation of Frame T-2 and T-1 (c) Standard Deviation of Frame T-2 and T (d) Standard Deviation of Frame T-2 and T+2 (e) Standard Deviation of Five Frames

Fig. 6. Visualizations about the inter-frame difference.



(a) Features Generated by GRU (b) Mean Value of Frame T-2 and T-1 (c) Mean Value of Frame T-2 and T (d) Mean Value of Frame T-2 and T+2 (e) Mean Value of Five Frames

Fig. 7. Visualizations about the inter-frame similarity.

convolutional LSTM block have a kernel size of $3 \times 3$ and padding of 1. In Fig. 2, the kernel size, padding and output channel of the leftmost convolution layer are $3 \times 3$, 1 and 32, respectively. For the second convolution layer, the three parameters are $4 \times 4$, 1 and 32. At the third convolution layer, the kernel size and padding are set to 9 and 4. A bilinear up-sampling layer, which restores feature maps to the original spatial size, is adopted after the third convolution layer. Finally, the fourth convolution layer, with a kernel size equals to 3 and padding equals to 1, reduces the channel to 5.

Overall loss function of the proposed network contains the Mean Squared Error (MSE) and the temporal loss function. Therefore, the overall loss function can be represented as

$$\mathcal{L} = \mathcal{L}_{MSE} + \alpha \mathcal{L}_T \tag{6}$$

where $\alpha$ is a hyper parameter that balances MSE and the temporal loss function. As our experiments demonstrate, the proposed network achieves the best performance when $\alpha$ is equal to 1e-6.

## IV. Experiments

In this section, we present several experiments to demonstrate the effectiveness of our network. First, we introduce the public benchmark dataset for the video deblurring task. Next, we implement training details of the proposed network. Then, we discuss the effectiveness of different parts in our network. Finally, we compare the proposed network with other video deblurring methods.

### A. Dataset

Su *et al.* [3] propose a benchmark dataset for the video deblurring task. Videos in this dataset are captured by hand-held devices such as iPhone 6s, GoPro Hero 4 black and Canon 7D. The DeepVideoDeblurring dataset contains two

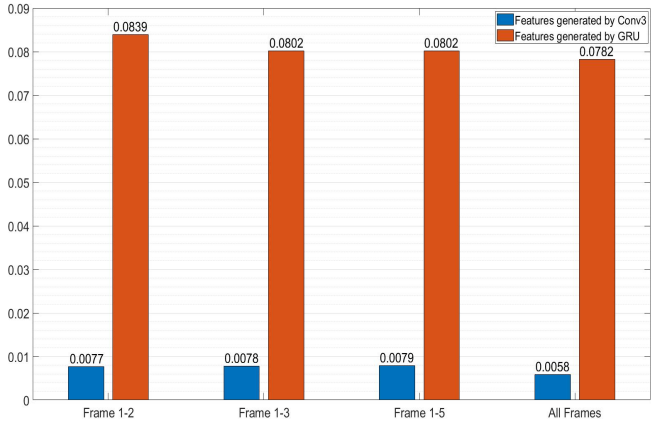



Fig. 8. The mean squared error (MSE) between feature maps generated by the non-local block and criteria of inter-frame similarity and difference.

subsets: the quantitative subset and the qualitative subset. In the quantitative dataset, there are 71 videos which are composed of 100 frames that are $1280 \times 720$ in size. In the qualitative dataset, videos are obtained from 22 different scenes without ground truth data. As existing method does [3], [4], the quantitative dataset is split into a training set and a testing set. The training set contains 61 videos, while the testing set contains 10 videos.

### B. Implementation Details

The proposed network is implemented by the PyTorch framework and an NVIDIA GTX 1080ti GPU. For data augmentation, we randomly crop $128 \times 128$ patch from any location of inputs. This method provides at least 712,193 pairs of blurry frames and sharp frames. Moreover, all frames are transformed into YCbCr space. The Y channel is used as input, and the corresponding Cb, Cr channels are used to restore the generated frame to RGB space. For network optimization, Adam with $\beta_1 = 0.5$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$ is used. The learning rate is set to $0.0001$, and the batch size is set to 4. According to our experiments, 3,000 epochs are sufficient for the proposed network to converge.

### C. Model Analysis

*1) Effectiveness of Non-local Block.:* To demonstrate the effectiveness of the proposed non-local block, we compare the proposed network (RVD) to three sub networks with variants of the non-local block. The configurations of the two sub networks are presented in Table II. Specially, the network w/o Non is obtained by replacing the non-local block with an equal number of stacked convolution layers. Then, the scale recurrent scheme [14] and the original non-local operation [48] are employed into w/o Non to obtain the SR Non and Original Non, respectively. The temporal part of Non-1 can be considered as the 3D convolution layer [4], as both of them take three frames as inputs and use the generated feature maps as the input of the following layers.

The quantitative results are shown in Table II, from which we obtain some important observations. First, as RVD achieves better performance than SR Non and Original Non, we can

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2020.3035722, IEEE Transactions on Circuits and Systems for Video Technology



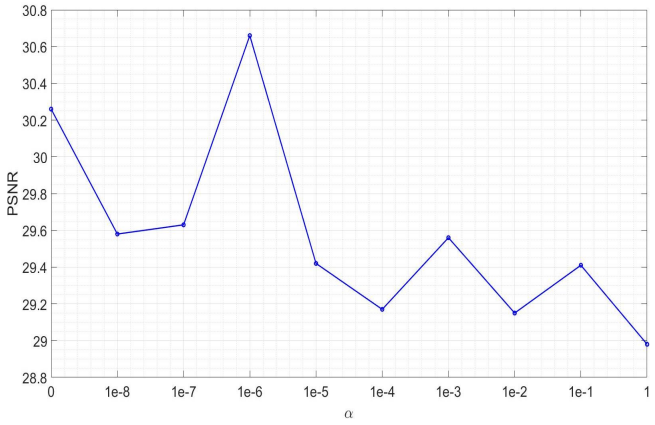
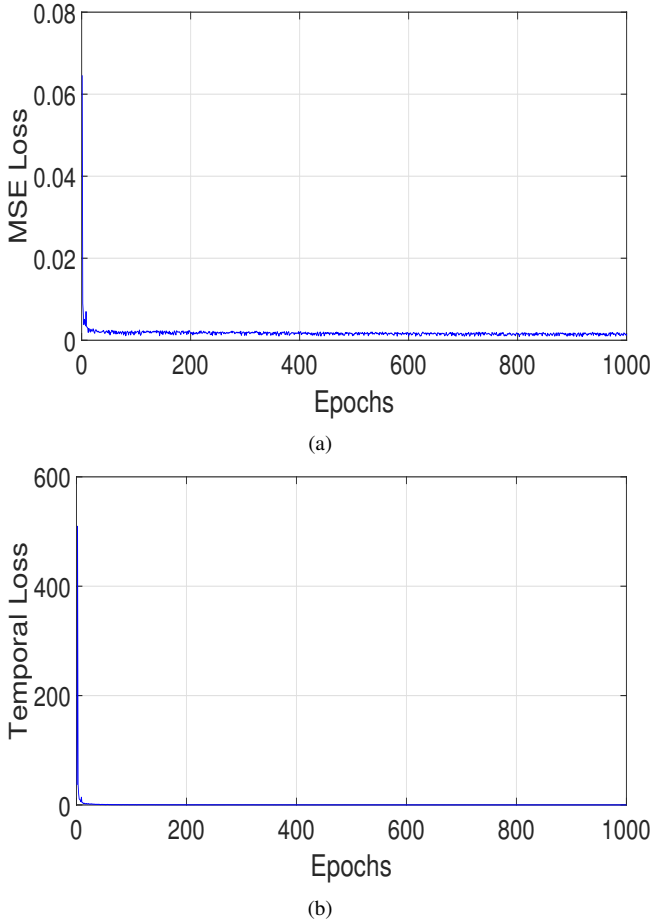



Fig. 9. Influence of the hyper-parameter $\alpha$.

find that the non-local block is sufficient to model the temporal information. Second, by comparing the w/o Non, Non-1, Non-2, and RVD, we find that both the Non-1, Non-2, and w/o Non perform worse than RVD. Moreover, the PSNR and SSIM of the Non-1 and Non-2 are higher than those of w/o Non. It can be inferred that, the proposed non-local block is beneficial to the overall performance. However, because the central frame is usually the most representative frame of the five frames, taking more frames as inputs of $Conv_1$ makes the calculation of inter-frame similarity difficult. On the other hand, since $Conv_3$ estimates differences among several frames, reducing its inputs degrades the representation ability of estimated inter-frame difference.

After demonstrating that non-local block is beneficial to estimate temporal relationship, we are still curious about the learned inter-frame similarity and difference. Therefore, some visual presentations about them are shown in Fig. 6 and Fig. 7. We first present visualization results of the learned inter-frame difference, which is learned by the $Conv_3$. For this purpose, standard deviations among input frames are computed as the criterion of inter-frame difference. In detail, given five input frames that are denoted as T-2, T-1, T, T+1, T+2, standard deviations between T-2 and T-1, T-2 and T, T-2 and T+2, all frames are visualized and presented in Fig. 6. Then, features generated by the $Conv_3$ are visualized and compared with these criterion. It can be seen that features generated by $Conv_3$ are similar to the standard deviation among five input frames (*e.g.*, edges of the car and tyre). In addition, by observing edges of the car, it can be found that $Conv_3$ can not only learn local difference among T-2, T-1 and T, but also global difference between T-2 and T+2.

Next, we validate the learned inter-frame similarity. For making visual comparison, mean values of input frames are taken as criterion of the inter-frame similarity. Then, we also visualize features generated by $GRU$ and show these visualization results in Fig. 7. As can be seen from rectangles in Fig. 7, features generated by $GRU$ can clearly represent similar parts among other images. Specially, by observing that learned features can still indicate similar parts between images of (c) and (d) columns, it is demonstrated that $GRU$ is able to model long-term similarities.




(a)

(b)

Fig. 10. (a) Values of the MSE loss function. (b) Values of the temporal loss function.

Finally, we present a statistical result to further validate the learned similarity and difference. In detail, the statistic is calculated after transforming all pixel values into [0, 1], and the mean squared error (MSE) is taken to represent such result. As Fig. 8 indicates, both learned similarity and difference are close to their corresponding criterion. However, it can also be found that learned similarity is relatively far from mean values between T-2 and T-1. This is because that the non-local block is supervised to learn global temporal information among all input frames, thus it may not be effective enough to model all short-term relationships.

*2) Effectiveness of Recursive Block.:* To verify the effectiveness of the recursive block, we also make a comparison between RVD and four sub networks. The configurations of the four sub networks are presented in Table III. In the second column, we present how many recursive blocks are used, and we present the looping times at the third column.

Quantitative results are presented in Table III. By comparing these networks, we can find that RVD achieves the best performance. Among these blocks, the RNNs has four blocks with different parameters, while other models do not. Therefore, compared with other models, RNNs can achieve comparable performance. However, as Table VI indicates, RNNs has much more parameters (about 9.3 MB and 1.9 M in terms of model size and parameter numbers) than the proposed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2020.3035722, IEEE Transactions on Circuits and Systems for Video Technology



TABLE V
QUANTITATIVE COMPARISON WITH PSDEBLUR, DEBLURGAN [52], MSCNN [25], WFA [10], DBN [3], STAN(M/A_A) [6], RNNS[26] AND SRN[14] ON THE VIDEODEBLURRING [3] DATASET. AVERAGE(PSNR) IS COMPUTED IN THE TEN TESTING VIDEOS. — MEANS CORRESPONDING DATA IS NOT REPORTED IN RELATED PAPERS.

| Method | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | Average (PSNR) |
|---|---|---|---|---|---|---|---|---|---|---|---|
| INPUT | 24.14 | 30.52 | 28.38 | 27.31 | 22.60 | 29.31 | 27.74 | 23.86 | 30.59 | 26.98 | 27.14 |
| PSDEBLUR | 24.42 | 28.77 | 25.15 | 27.77 | 22.02 | 25.74 | 26.11 | 19.71 | 26.48 | 24.62 | 25.08 |
| DeblurGAN [52] | 25.23 | 29.17 | 27.82 | 27.51 | 22.58 | 28.83 | 26.83 | 23.84 | 31.04 | 26.18 | 26.90 |
| MSCNN [25] | 26.84 | 31.56 | 29.29 | 29.46 | 24.19 | 29.94 | 28.50 | 25.18 | 32.07 | 27.89 | 28.49 |
| WFA [10] | 25.89 | 32.33 | 28.97 | 28.36 | 23.99 | 31.09 | 28.58 | 24.78 | 31.30 | 28.20 | 28.35 |
| DBN (single) [3] | 25.75 | 31.15 | 29.30 | 28.38 | 23.63 | 30.70 | 29.23 | 25.62 | 31.92 | 28.06 | 28.37 |
| DBN (noalign) [3] | 27.83 | 33.11 | 31.29 | 29.73 | 25.12 | 32.52 | 30.80 | 27.28 | 33.32 | 29.51 | 30.05 |
| DBN (flow) [3] | 28.31 | 33.14 | 30.92 | 29.99 | 25.58 | 32.39 | 30.56 | 27.15 | 32.95 | 29.53 | 30.05 |
| STAN (M/A_A) [6] | 28.73 | 33.34 | 31.21 | 30.77 | 25.33 | 32.56 | 30.11 | 27.07 | 34.13 | 29.62 | 30.29 |
| RNNs [26] | - | - | - | - | - | - | - | - | - | - | 30.05 |
| SRN [14] | - | - | - | - | - | - | - | - | - | - | 29.97 |
| Ours | 28.31 | 34.60 | 31.33 | 31.23 | 25.11 | 32.22 | 30.37 | 27.78 | 36.15 | 29.51 | 30.66 |

TABLE VI
MODEL SIZES OF SOME RELATED METHODS. AS [53], [54] DID, THE MODEL SIZE IS EXPRESSED IN THE FILE SIZE AND PARAMETER NUMBERS OF EACH MODEL.

| Models | Su *et al.* [3] | Sun *et al.* [8] | Nah *et al.* [25] | Tao *et al.* [14] | Zhang *et al.* [26] | Gong *et al.* [55] | VMPHN [53] | Ours |
|---|---|---|---|---|---|---|---|---|
| File Size (MB) | - | 54.1 | 303.6 | 33.6 | 37.1 | 41.2 | 43.4 | 27.8 |
| Parameters (M) | 16.7 | 7.3 | 11.7 | 8.1 | 9.2 | 10.3 | - | 7.3 |

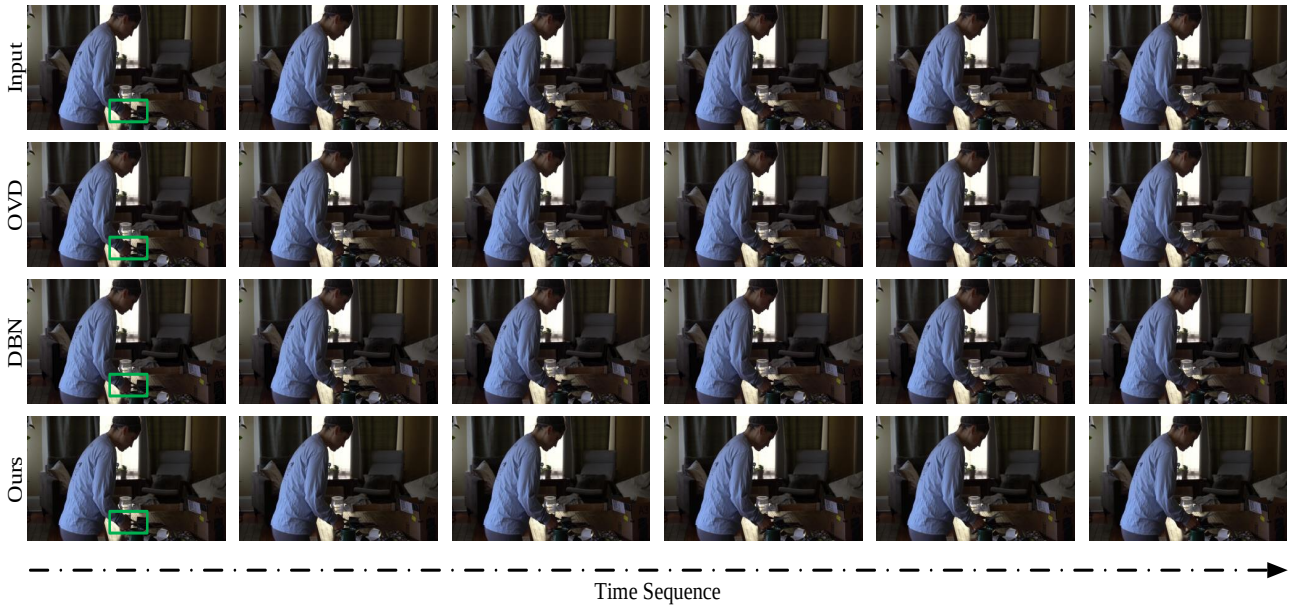



Fig. 11. Visual comparison on the qualitative subset. From left to right, there are six frames that change across time. From top to bottom, we present blurry inputs, deblurred results generated by OVD [56], DBN [3], and the proposed RVD.

RVD, which is designed to conduct video deblurring task with few parameters. In addition, by comparing the Recursive-1 and the Recursive-3, the Recursive-4 and RVD, we can find that reducing the number of recursive blocks benefits the network performance if the number of iterations is large enough. Performances of the Recursive-1, the Recursive-2 and RVD demonstrate that appropriate iteration number allows the recursive block to work better.

*3) Effectiveness of Temporal Loss.:* The performance of the temporal loss function is also an important consideration. Empirically, we use the hyper parameter $\alpha$ to balance the MSE and the temporal loss function [4]. Thus, values of $\alpha$ inevitably influence effectiveness of the temporal loss function and the overall performance. Therefore, Fig. 9 is presented to explore such influence. It can be found that within the range of [0, 1], reducing $\alpha$ does not always improve the PSNR, and the best value of $\alpha$ is 1e-6. This is because that values of the temporal loss function is about 1e+5 bigger than that of MSE (see Fig. 10). In addition, we also make a comparison between the perceptual loss function and the proposed temporal loss function, since both of them are computed in the feature space. As Table IV shows, the proposed RVD trained with the temporal loss function outperform the Perceptual about 0.59 and 0.03 in terms of PSNR and SSIM. This indicates that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2020.3035722, IEEE Transactions on Circuits and Systems for Video Technology



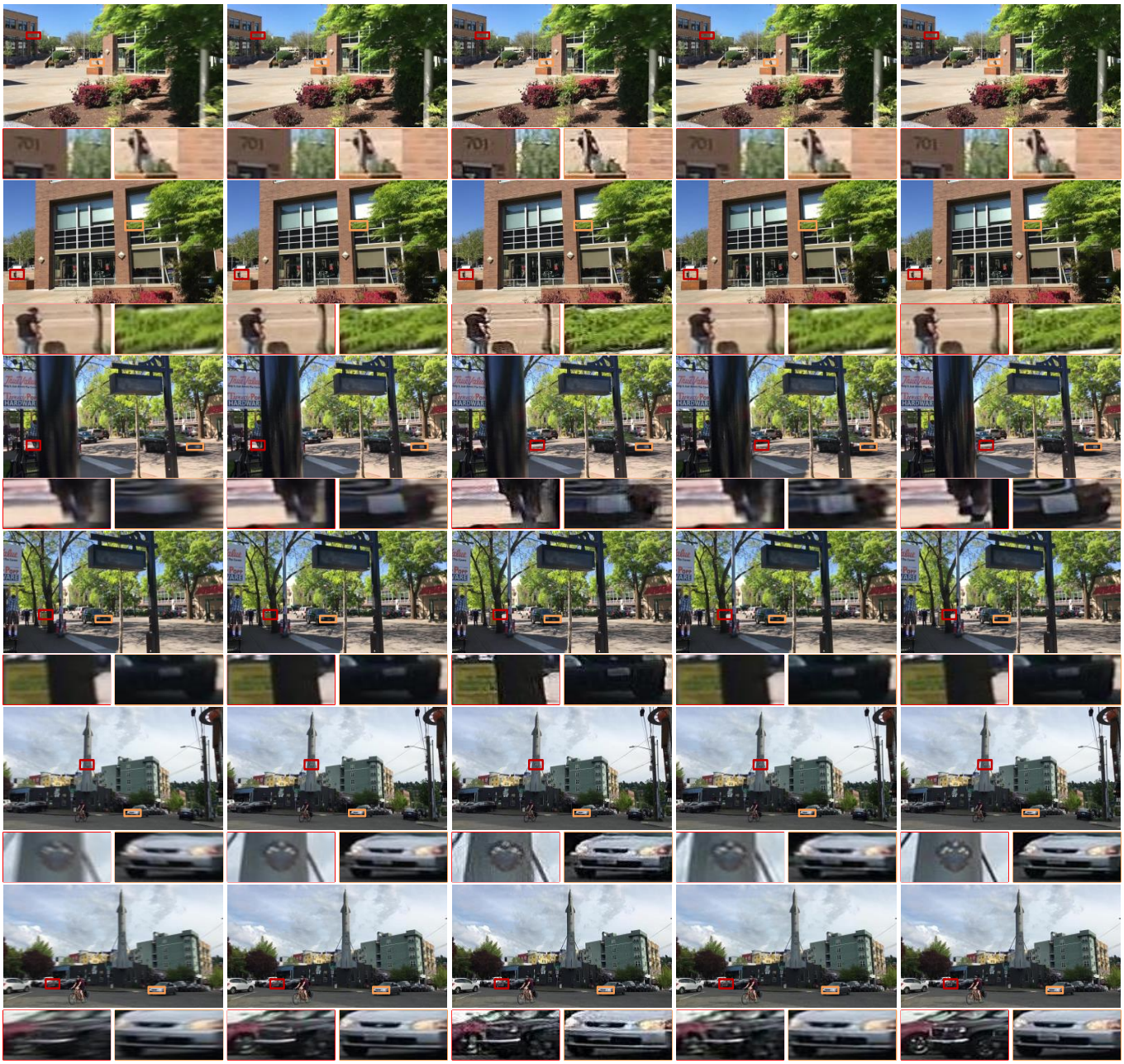


Fig. 12. Visual comparison on the quantitative subset. From left to right, we show inputs, results of [14], [52], ours and ground truth images. Best view on screen.

with the temporal loss function, the proposed RVD is more sensitive to temporal information among restored frames, and is supervised to generate more consistent sequences.

### *D. Comparison*

We compare our network with seven state-of-the-art methods on the testing set of the DeepVideoDeblurring dataset [3]. As shown in Table V, the compared methods are PSDEBLUR, DeblurGAN [52], MSCNN [25], WFA [10], DBN [3], STAN (M/A_A) [6], RNNs [26] and SRN [14]. PSDEBLUR represents the deblurred result of Photoshop. WFA stacks several frames to generate a deblurred frame. DeblurGAN, which is based on generative adversarial learning, generates one frame at each time. DBN (single), DBN (noalign) and DBN (flow) are three variants of DBN, which models the temporal information via the 2D convolution layer [3]. RNNs [26] uses a spatially variant neural network to conduct the deblurring task. SRN also takes convolutional LSTM as basic components, and takes the scale-recurrent scheme to capture spatial topology of a single image.

As exhibited in Table V and VI, the proposed network (RVD) achieves the best result of video deblurring in terms of PSNR and model efficiency. Compared with RNNs [26], our method improves the average values of PSNR to 30.66. The three variations of DBN [3] (DBN (single), DBN (noalign), DBN (flow)) are all worse than our proposed method. The

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2020.3035722, IEEE Transactions on Circuits and Systems for Video Technology



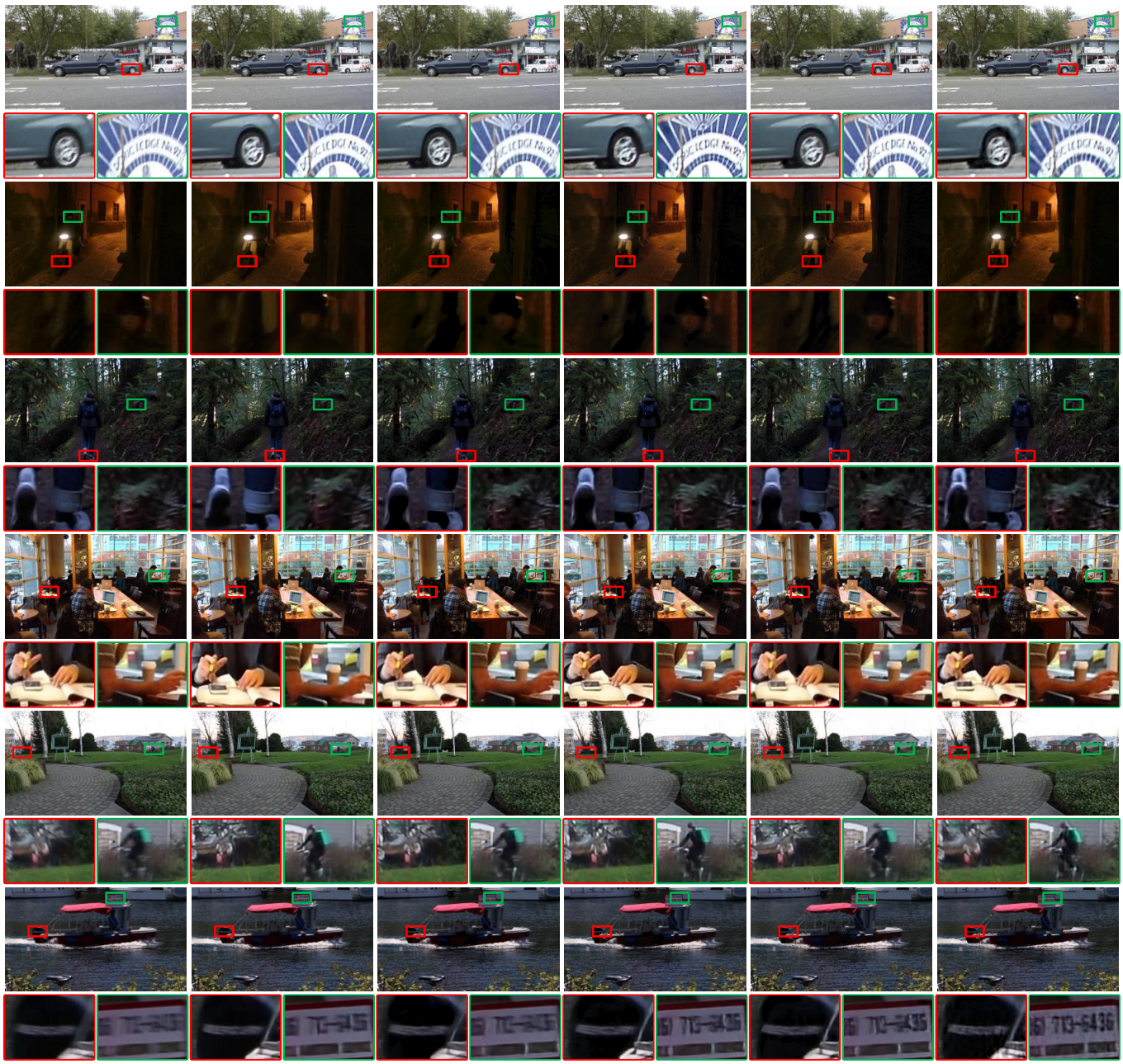


Fig. 13. Visual comparison on the qualitative subset. From left to right, we present blurry inputs, deblurred results generated by OVD [56], DBN(single), DBN(noalign), DBN(flow) and the proposed method.

DeblurGAN, which is based on adversarial learning, is worse than our method at 3.76 dB in term of PSNR. Compared with recent video deblurring methods MSCNN and WFA, our model outperforms them by about 7.6%. Moreover, we make three visual comparisons on the quantitative subset and the qualitative subset, which are presented at Fig. 11, Fig. 12, and Fig. 13. As is evident from Fig. 11, the frames generated by our method are temporally consistent and perceptually pleasing. In Fig. 12, our method successfully restores more recognizable details. In Fig. 13, six frames from different real cases are presented for further demonstrating the generalization ability of the proposed RVD.

### *E. Limitation and Possible Improvements*

As experimental results indicate, the proposed non-local block and temporal loss function can respectively model temporal relationships among input and output frames, and the proposed recursive block can effectively model the spatial characteristic with little parameters. However, as can be seen from Fig. 14, both the proposed method and existing ones cannot effectively restore tiny and distant objects. To be specific, compared with DBN and OVD, although the proposed method can recover the level of the paddle, the appearance of the man is still unrealistic. This is because that for a tiny and distant object, its appearance is reflected by few pixels, whose values always seriously change along with time

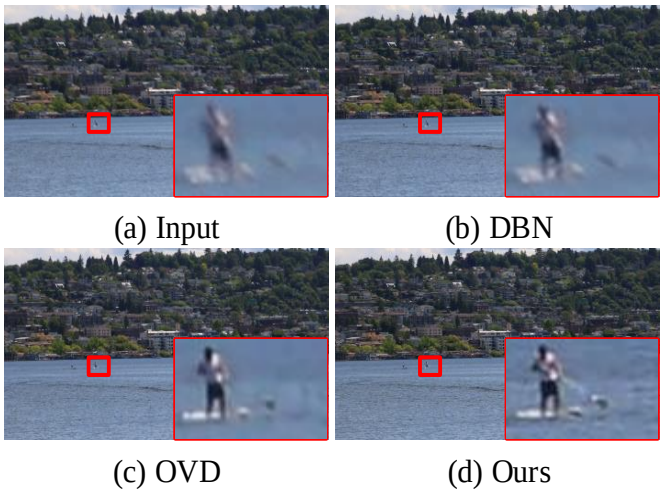



Fig. 14. An example of restoring tiny and distant objects.

sequences. Therefore, it is difficult to model similarities of these pixels in a time sequence, leading to the difficulty of estimating pixel-level temporal relationships. In addition, as the above figure illustrates, existing methods such as DBN and OVD also suffer from the limitation of calculating pixel-level relationships. Thus, addressing the above limitation is essential for developing effective video deblurring methods. To this end, here we provide a possible methodology. As discussed in [40], most blur artifacts are caused by events such as people moving or car moving. Therefore, by explicitly understanding events in consecutive frames, video deblurring methods may be able to reason these pixel-level relationships and restore appearances of such tiny and distant objects.

## V. Conclusion

In this work, we have analyzed the non-local operation and proposed a non-local block to model the temporal information. We have also proposed a novel recursive block for restoring sharp frames by iteratively refine feature maps generated at the last iteration. Besides, we have introduced a temporal loss function to ensure the temporal consistency of the generated frames. By adopting the non-local block, the recursive block and the temporal loss function, the proposed network achieves state-of-the-art performance.

## VI. Acknowledgment

This work was supported in part by the National Natural Science Foundation of China [grant no. 61922064], in part by the Zhejiang Provincial Natural Science Foundation [grant nos. LR17F030001, LQ19F020005], in part by the Project of science and technology plans of Wenzhou City [grant nos. C20170008, G20150017, ZG2017016].

## References

[1] H. S. Lee, J. Kwon, and K. M. Lee, "Simultaneous localization, mapping and deblurring," in *International Conference on Computer Vision*, pp. 1203–1210, 2011.

[2] X. Zhang, W. Hu, N. Xie, H. Bao, and S. Maybank, "A robust tracking system for low frame rate video," *International Journal of Computer Vision*, vol. 115, pp. 279–304, 2015.

[3] S. Su, M. Delbracio, J. Wang, G. Sapiro, W. Heidrich, and O. Wang, "Deep video deblurring for hand-held cameras," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1279–1288, 2017.

[4] K. Zhang, W. Luo, Y. Zhong, L. Ma, W. Liu, and H. Li, "Adversarial spatio-temporal learning for video deblurring," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 291–301, 2018.

[5] S. Nah, S. Son, and K. M. Lee, "Recurrent neural networks with intra-frame iterations for video deblurring," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8102–8111, 2019.

[6] Z. Zhan, X. Yang, Y. Li, and C. Pang, "Video deblurring via motion compensation and adaptive information fusion," in *Neurocomputing*, vol. 341, 03 2019.

[7] Y. Li, "Video forecasting with forward-backward-net: Delving deeper into spatiotemporal consistency," in *ACM Multimedia Conference on Multimedia Conference*, pp. 211–219, 2018.

[8] J. Sun, W. Cao, Z. Xu, and J. Ponce, "Learning a convolutional neural network for non-uniform motion blur removal," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 769–777, 2015.

[9] J. Pan, J. Dong, Y. W. Tai, Z. Su, and M. H. Yang, "Learning discriminative data fitting functions for blind image deblurring," in *The IEEE International Conference on Computer Vision*, pp. 1068–1076, 2017.

[10] M. Delbracio and G. Sapiro, "Burst deblurring: Removing camera shake through fourier burst accumulation." in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2385–2393, 2015.

[11] X. Zhang, D. Wang, Z. Zhou, and Y. Ma, "Robust low-rank tensor recovery with rectification and alignment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[12] D. Liu, B. Wen, Y. Fan, C. C. Loy, and T. S. Huang, "Non-local recurrent network for image restoration," in *Advances in Neural Information Processing Systems*, pp. 1673–1682, 2018.

[13] Q. Guo, Z. Yu, Y. Wu, D. Liang, H. Qin, and J. Yan, "Dynamic recursive neural network," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5147–5156, 2019.

[14] X. Tao, H. Gao, X. Shen, J. Wang, and J. Jia, "Scale-recurrent network for deep image deblurring," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8174–8182, 2018.

[15] H. Huang, H. Wang, W. Luo, L. Ma, W. Jiang, X. Zhu, Z. Li, and W. Liu, "Real-time neural style transfer for videos," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 783–791, 2017.

[16] N. Bonneel, J. Tompkin, K. Sunkavalli, D. Sun, S. Paris, and H. Pfister, "Blind video temporal consistency," *ACM Transactions on Graphics*, vol. 34, no. 6, p. 196, 2015.

[17] J. Johnson, A. Alahi, and F. F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, pp. 694–711, 2016.

[18] M. Jin, S. Roth, and P. Favaro, "Noise-blind image deblurring," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3510–3518, 2017.

[19] J. Zhang, J. Pan, W. S. Lai, R. W. Lau, and M. H. Yang, "Learning fully convolutional networks for iterative non-blind deconvolution," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3817–3825, 2017.

[20] M. G. Jin, S. Roth, and P. Favaro, "Normalized blind deconvolution," in *European Conference on Computer Vision*, pp. 694–711, 2018.

[21] S. Liu, H. Wang, J. Wang, and C. Pan, "Blur-kernel bound estimation from pyramid statistics," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 26, no. 5, pp. 1012–1016, 2015.

[22] L. Zhang, L. Zhou, and H. Huang, "Bundled kernels for nonuniform blind video deblurring," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 27, no. 9, pp. 1882–1894, 2016.

[23] O. Whyte, J. Sivic, A. Zisserman, and J. Ponce, "Non-uniform deblurring for shaken images," *International Journal of Computer Vision*, vol. 98, no. 2, pp. 168–186, 2012.

[24] M. Noroozi, P. Chandramouli, and P. Favaro, "Motion deblurring in the wild," in *German Vonference on Pattern Recognition*, pp. 65–77, 2017.

[25] S. Nah, K. T. Hyun, and M. L. Kyoung, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3883–3891, 2017.

[26] J. Zhang, J. Pan, J. Ren, Y. Song, L. Bao, R. W. Lau, and M. H. Yang, "Dynamic scene deblurring using spatially variant recurrent neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2521–2529, 2018.

[27] H. Gao, X. Tao, X. Shen, and J. Jia, "Dynamic scene deblurring with parameter selective sharing and nested skip connections," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3848–3856, 2019.

[28] Y. Li, S. B. Kang, N. Joshi, S. M. Seitz, and D. P. Huttenlocher, "Generating sharp panoramas from motion-blurred videos," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2424–2431, 2010.

[29] C. Paramanand and A. N. Rajagopalan, "Non-uniform motion deblurring for bilayer scenes," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1115–1122, 2013.

[30] X. Lin, J. Suo, and Q. Dai, "Extracting depth and radiance from a defocused video pair," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 557–569, 2014.

[31] K. T. Hyun and L. K. Mu, "Generalized video deblurring for dynamic scenes," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5426–5434, 2015.

[32] F. Klose, O. Wang, J. C. Bazin, M. Magnor, and H. A. Sorkine, "Sampling based scene-space video processing," *ACM Transactions on Graphics*, vol. 34, no. 4, p. 67, 2015.

[33] P. Wieschollek, M. Hirsch, B. Scholkopf, and H. Lensch, "Learning blind motion deblurring," in *The IEEE International Conference on Computer Vision*, pp. 231–240, 2017.

[34] F. Tan, S. Liu, L. Zeng, and B. Zeng, "Kernel-free video deblurring via synthesis." in *IEEE International Conference on Image Processing*, pp. 2683–2687, 2016.

[35] K. Zhang, W. Luo, Y. Zhong, L. Ma, B. Stenger, W. Liu, and H. Li, "Deblurring by realistic blurring," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2737–2746, 2020.

[36] J. Pan, H. Bai, and J. Tang, "Cascaded deep video deblurring using temporal sharpness prior," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3043–3051, 2020.

[37] X. Wang, K. C. Chan, K. Yu, C. Dong, and C. Change Loy, "Edvr: Video restoration with enhanced deformable convolutional networks," in *The IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 0–0, 2019.

[38] Z. Zhong, Y. Gao, Y. Zheng, and B. Zheng, "Efficient spatio-temporal recurrent neural network for video deblurring," 2020.

[39] 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," *arXiv preprint arXiv:1911.07410*, 2019.

[40] S. Lin, J. Zhang, J. Pan, Z. Jiang, D. Zou, Y. Wang, J. Chen, and J. Ren, "Learning event-driven video deblurring and interpolation," 2019.

[41] X. Zhang, R. Jiang, T. Wang, P. Huang, and L. Zhao, "Attention-based interpolation network for video deblurring," *Neurocomputing*, 2020.

[42] R. Jiang, L. Zhao, T. Wang, J. Wang, and X. Zhang, "Video deblurring via temporally and spatially variant recurrent neural network," *IEEE Access*, vol. 8, pp. 7587–7597, 2019.

[43] D. Eigen, J. Rolfe, R. Fergus, and Y. LeCun, "Understanding deep architectures using a recursive convolutional network," *arXiv preprint arXiv:1312.1847*, 2013.

[44] R. Socher, B. Huval, B. Bath, C. D. Manning, and A. Y. Ng, "Convolutional-recursive deep learning for 3d object classification," in *Advances in Neural Information Processing Systems*, pp. 656–664, 2012.

[45] J. Kim, L. J. Kwon, and M. L. Kyoung, "Deeply-recursive convolutional network for image super-resolution," in *The IEEE conference on Computer Vision and Pattern Recognition*, pp. 1637–1645, 2016.

[46] M. Liang and X. Hu, "Recurrent convolutional neural network for object recognition," in *The IEEE conference on Computer Vision and Pattern Recognition*, pp. 3367–3375, 2015.

[47] S. Zhang, Z. Fan, N. Ling, and M. Jiang, "Recursive residual convolutional neural network-based in-loop filtering for intra frames," *IEEE Transactions on Circuits and Systems for Video Technology*, 2019.

[48] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7794–7803, 2018.

[49] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," *arXiv preprint arXiv:1511.06432*, 2015.

[50] X. J. Shen, Z. R. Chen, H. Wang, D. Y. Yeung, W. K. Wong, and W. c. Woo, "Convolutional lstm network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems*, pp. 802–810, 2015.

[51] X. Zhang, T. Wang, J. Wang, G. Tang, and L. Zhao, "Pyramid channel-based feature attention network for image dehazing," *Computer Vision and Image Understanding*, vol. 2197-198, p. 103003, 2020.

[52] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 8183–8192, 2018.

[53] H. Zhang, Y. Dai, H. Li, and P. Koniusz, "Deep stacked hierarchical multi-patch network for image deblurring," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 5978–5986, 2019.

[54] S. Zhou, J. Zhang, J. Pan, H. Xie, W. Zuo, and J. Ren, "Spatio-temporal filter adaptive network for video deblurring," *arXiv preprint arXiv:1904.12257*, 2019.

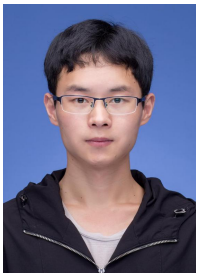
[55] D. Gong, J. Yang, L. Liu, Y. Zhang, I. Reid, C. Shen, A. Van Den Hengel, and Q. Shi, "From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur," in *The IEEE Conference on Computer Vision and Pattern Recognition*, pp. 2319–2328, 2017.

[56] K. T. Hyun, L. K. Mu, B. Scholkopf, and M. Hirsch, "Online video deblurring via dynamic temporal blending network," in *The IEEE International Conference on Computer Vision*, pp. 4038–4047, 2017.


**Xiaoqin Zhang** received the B.Sc. degree in electronic information science and technology from Central South University, China, in 2005 and Ph.D. degree in pattern recognition and intelligent system from the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, China, in 2010. He is currently a professor in Wenzhou University, China. His research interests are in pattern recognition, computer vision and machine learning. He has published more than 100 papers in international and national journals, and international conferences, including IEEE T-PAMI, IJCV, IEEE T-IP, IEEE T-IE, IEEE T-C, ICCV, CVPR, NIPS, IJCAI, AAAI, and among others.

**Runhua Jiang** is currently a graduate student majoring in computer software and theory at College of Computer Science and Artificial Intelligence, Wenzhou University, China. He received his B.Sc. degree in department of information science at Tianjin University of Finance and Economy, China. His research interests include image and video processing, pattern recognition and machine learning.

**Tao Wang** is currently a graduate student at College of Computer Science and Artificial Intelligence, Wenzhou University, China. He received the B.Sc. degree in information and computing science from Hainan Normal University, China, in 2018. His research interests include several topics in computer vision and machine learning, such as object tracking/detection, image/video quality restoration, adversarial learning, image-to-image translation and reinforcement learning.



**Jinxin Wang** is currently a graduate student at College of Computer Science and Artificial Intelligence, Wenzhou University, China. He received his bachelor's degree in information and computing science at Wenzhou University. His research interests include reinforcement learning, image generation and deep learning.