Efficient Spatio-Temporal Recurrent Neural Network for Video Deblurring

Zhihang Zhong[1], Ye Gao[2], Yinqiang Zheng[3⋆], and Bo Zheng[2]

[1] The University of Tokyo, Tokyo 113-8656, Japan
zhong@race.t.u-tokyo.ac.jp
[2] Tokyo Research Center, Huawei
{jeremy.gao, bozheng.jp}@huawei.com
[3] National Institute of Informatics, Tokyo 101-8430, Japan
yqzheng@nii.ac.jp

Abstract. Real-time video deblurring still remains a challenging task due to the complexity of spatially and temporally varying blur itself and the requirement of low computational cost. To improve the network efficiency, we adopt residual dense blocks into RNN cells, so as to efficiently extract the spatial features of the current frame. Furthermore, a global spatio-temporal attention module is proposed to fuse the effective hierarchical features from past and future frames to help better deblur the current frame. For evaluation, we also collect a novel dataset with paired blurry/sharp video clips by using a co-axis beam splitter system. Through experiments on synthetic and realistic datasets, we show that our proposed method can achieve better deblurring performance both quantitatively and qualitatively with less computational cost against state-of-the-art video deblurring methods.

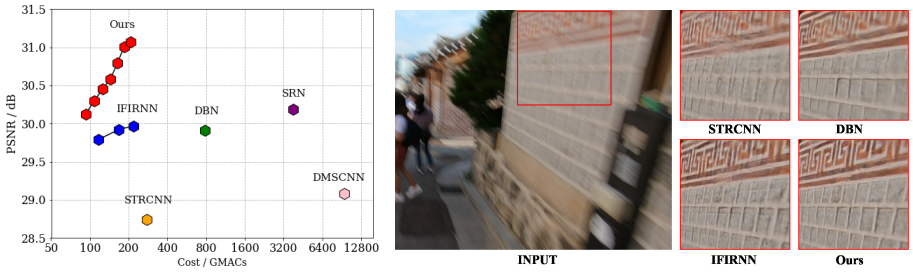
Keywords: Video deblurring, RNN, Network efficiency, Attention, Dataset

1 Introduction

Nowadays, video recording usually suffers from the quality issues caused by motion blur. This is especially true in poorly illuminated environment, where one has to lengthen the exposure time for sufficient brightness. A great variety of video deblurring methods have been proposed, which have to deal with two competing goals, i.e., to improve the deblurring quality and to reduce the computational cost. The latter is of critical importance for low-power mobile devices, like smartphones.

To properly make use of the spatio-temporal correlation of the video signal is the key to achieve better performance on video deblurring. The CNN-based methods [30] [34] make an inference of the deblurred frame by stacking neighboring frames with current frame as input to the CNN framework. The RNN-based methods, like [35] [13] [43] [23], employ recurrent neural network architecture to

⋆ Yinqiang Zheng is the corresponding author.



(a) (b)

Fig. 1: A comparison of network efficiency on video deblurring. SRN[32], DMSCNN[22] are state-of-the-art (SoTA) methods for image deblurring, and STRCNN[30], DBN[13], IFIRNN[23] are SoTA methods for video deblurring. (a) shows the computational cost required for processing a frame of 720P(1280×720) video and the corresponding performance of each model on GOPRO[22] dataset in terms of GMACs and PSNR, respectively. (b) shows the deblurred image generated by SoTA video deblurring methods and ours.

transfer the effective information frame by frame for deblurring. However, how to utilize spatio-temporal dependency of video for deblurring more efficiently still needs to be explored. The CNN-based methods are usually cumbersome (繁重) in dealing with spatio-temporal dependency of concatenated neighboring frames, and the existing RNN-based methods have limited capacity to transfer the effective information temporally. Thus, they suffer from either the huge computational cost, or the ineffectiveness of deblurring.

In this work, we propose an efficient spatio-temporal recurrent neural network (denoted as ESTRNN) to solve the above issues. We mainly focus on the network efficiency of video deblurring methods, which directly reflects on the deblurring performance of the method under the limited computational resources, as Fig. 1a. It shows that our method can achieve much better performance with less computational cost against SoTA deblurring methods. Due to making full use of spatio-temporal dependency of the video signal, our method is exceptionally good at restoring the details of the blurry frame compared with SoTA video deblurring methods, as shown in Fig 1b.

To make a more computational efficient video deblurring method, we develop our method through amelioration of basic RNN architecture from three aspects: 1) In temporal domain, the high-level features generated by RNN cell are more informative, which are more suitable for temporal feature fusion (see Fig.2) than using channel-concatenated neighboring frames as input. Another advantage of using neighboring high-level features for temporal fusion is that it reuses the intermediate results of deblurring process of other frames, which helps to improve the overall network efficiency; 2) it is obvious that not all high-level features from neighboring frames are beneficial to deblurring of the current frame. Thus, it is worth designing an attention mechanism [1] that allows the method to

focus on more informative part of high-level features from other frames. To this end, we propose a novel global spatio-temporal attention module (see Sec.3.3) for efficiently temporal feature fusion; 3) Regarding the spatial domain, how to extract the spatial features from the current frame will affect the quality of information transmitted in temporal domain. In other words, well generated spatial features of each frame are a prerequisite for ensuring good temporal feature fusion. Therefore, we integrate the residual dense blocks (RDB [41]) as backbone into RNN cell to construct our RDB cell (see Sec.3.2). The high-level hierarchical features generated by RDB cell is more computationally efficient with richer spatial information.

Our contributions in this work are summarized as follows:

- To the best of our knowledge, this is the first work making use of the high-level features of RNN cell from future and past frames for deblurring the current frame in the video.
- To efficiently utilize the high-level features from neighboring frames, we propose a global spatio-temporal attention module for temporal feature fusion.
- To improve the efficiency of extracting spatial features from the current frame, we adopt residual dense blocks into our RNN cell to generate more informative hierarchical features.
- Besides the conventional synthetic video deblurring dataset, such as REDS [21] and GOPRO [22], we also use a beam splitter system [14] to capture realistic blurry/sharp video pairs for evaluation. Our realistic dataset will be released to facilitate further researches.
- The experimental results demonstrate that our method achieves better deblurring performance both quantitatively and qualitatively than SoTA video deblurring methods with less computational cost.

2 Related Works

2.1 Video Deblurring

In recent years, video deblurring technologies become significant for daily life media editing and for advanced processing such as SLAM [17], 3D reconstruction [29] and visual tracking [36]. Research focus starts to shift from early single non-blind image deblurring [44] [28] [31] and single blind image deblurring [38] [6] [20] [3] [25] to the more challenging video deblurring task.

Typically, the blur in a video has different sizes and intensities in different position of each frame. In the early work of video deblurring, [18] and [2] attempt to automatically segment a moving blurred object from the background and assume a uniform blur model for them. Then, in view of the different kinds of blur in different regions of an image, [37] tries to segment an image into various layers and generate segment-wise blur kernels for deblurring. More recently, there are some researches that estimate pixel-wise blur kernel with segmentation [27], or without segmentation [11] [12]. However, these kernel based methods are quite

expensive in computation and usually rely on human knowledge. An inaccurate blur kernel will result in severe artifacts in deblurred image.

To overcome the above issues, recently researchers start to work on deep learning methods for video deblurring. CNN-based methods are used to handle the inter-frame relationship of video signal, such as [30], which makes the estimation of deblurred frame by using channel-concatenated neighboring frames. Usually, alignment of neighboring frames is required for these methods, which is quite computationally expensive. They realize it by traditional way like optical flow [26] or the network itself such as using deformable convolutional operation [4] in [34]. Some researchers tend to focus on RNN-based methods because of their excellent performance for time-series signal. RNN-based methods do not need to perform the explicit alignment and the model could manage it implicitly through hidden states. For example, [35] employs RNN architecture to reuse the features extracted from the past frame, and [13] improves the performance of deblurring by blending the hidden states in temporal domain. Then, [23] iteratively updates the hidden state via reusing RNN cell parameters and achieves SoTA video deblurring performance while operating in real time.

In this paper, we adopt a RNN framework similar to [23]. Our method is different from [23] in that we integrate RDB into the RNN cell in order to exploit the potential of the RNN cell through feature reusing and generating hierarchical features for the current frame. Furthermore, we propose a GSA module to selectively merge effective hierarchical features from both past and future frames, which enables our model to utilize the spatio-temporal information more efficiently.

2.2 Attention Mechanism

Allocating more computational resources towards the most informative components of a signal is a wise strategy to enhance system performance under the situation of limited resources. Such a selectively focusing mechanism originated from natural language processing (NLP) is named as attention mechanism [1] [39] [33], which has demonstrated to be very effective in many areas including image restoration task [40] [34]. Inspired by the success of attention mechanism in image restoration task, [34] proposed their attention module to assign pixel-level aggregation weights on each neighboring frame for video deblurring. The principle of their attention module is the same as the original idea from NLP that a neighboring frame that is more similar to the reference one in an embedding space should be paid more attention. However, we believe that considering the situation of video deblurring, the method should pay attention to the useful features (the lost information), rather than the similar features from neighboring features. Therefore, we propose a global spatio-temporal attention module, which allows our method to efficiently fuse the effective features from both past and future frames for deblurring the current frame.

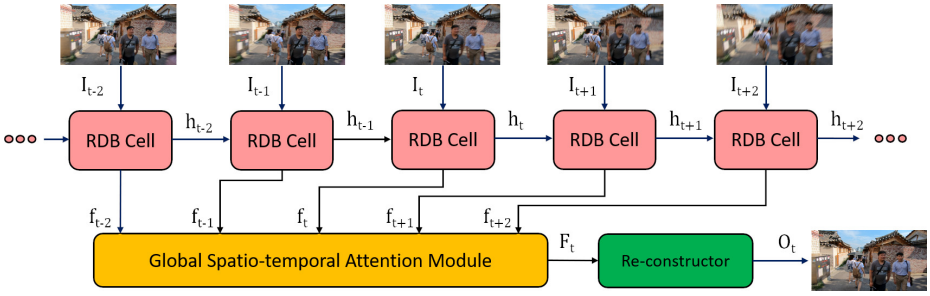


Fig. 2: Framework of proposed efficient spatio-temporal recurrent neural network. I_t refers to the t^{th} input blurry frame; h_t and f_t refer to the extracted hidden state and hierarchical features of RDB-based RNN cell (see Sec. 3.2) from t^{th} frame; F_t refers to the fused features generated by GSA module (see Sec. 3.3) for t^{th} frame; O_t refers to the t^{th} deblurred frame by the proposed method.

3 Proposed Method

In this section, we will first give an overview of the proposed method first in Sec. 3.1. Then we will go into details of RDB cell and GSA module in Sec. 3.2 and Sec. 3.3, respectively.

3.1 Overview

According to the characteristics of blur in the video, it may keep varying temporally and spatially, which makes deblurring problem intractable. In turn, it is possible that the blurred information in the current frame is relatively clear and complete in the past frames and future frames. When using RNN-based method to implement video deblurring, high-level features of the current frame will be generated to make the inference of deblurred image. Actually, some parts of the high-level features are worth saving and reusing for making up the loss information for other frames. Therefore, distributing part of computational resources to fuse informative features in past and future frames could be a method to effectively improve the efficiency of the neural network. Furthermore, how to improve RNN cell itself to extract high-level features with better spatial structure is critical to enhancing the efficiency of the neural network. Starting from the above viewpoints, we integrate multiple residual dense blocks into RNN cell to generate hierarchical features and propose a global spatio-temporal attention module for feature fusion of neighboring frames.

The whole video deblurring process of our method is shown as Fig. 2. We denote the input frames of blurry video and corresponding output frames as $\{I_t\}$ and $\{O_t\}$ respectively, where $t \in \{1 \cdots T\}$. Through RDB-based RNN cell, the model could get hierarchical features for each frame as $\{f_t\}$. To get the inference of latent frame O_t, the global spatio-temporal attention module takes current hierarchical feature f_t with two past and two future features $(f_{t-2}, f_{t-1}, f_{t+1}, f_{t+2})$

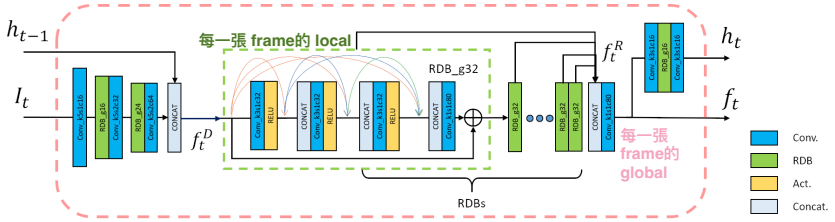


Fig. 3: The structure of RDB-based RNN cell. h_t and h_{t-1} refer to the hidden state of past frame and current frame, respectively; I_t refers to the input blurry frame; f_t^D refers to the features after downsampling module; f_t^R refers to the feature set generated by series of RDB modules; f_t refers to the hierarchical features generated by the RDB cell; As for the details of each layer and RDB module, k, s, c and g denote kernel size, stride, channels and growth rate, respectively.

as input to perform feature fusion and generate F_t as output. Finally, through re-constructor module, the model can get the latent frame O_t.

3.2 RDB Cell: RDB-based RNN Cell

We adopt residual dense block (RDB) [41] [42] into the RNN cell, which is named as RDB cell. The dense connections of RDB inherited from dense block (DB) [10] let each layer receive feature maps from all the previous layers by concatenating them together in channels. The output channels of each layer in RDB will keep the same size, which allows collective features to be reused and save the computational resources. Moreover, through local feature fusion, RDB could generate hierarchical features from convolutional layers in different depth with different size of receptive fields, which could provide better information for image reconstruction.

The structure of RDB-based RNN cell is shown as Fig. 3. First, the current input frame I_t will be downsampled and concatenated with last hidden state h_{t-1} to get shallow feature maps f_t^D as

$$f_t^D = CAT(DS(I_t), h_{t-1}) \tag{1}$$

where $CAT(\cdot)$ refers to concatenation operation; $DS(\cdot)$ refers to downsampling operation in the cell which consists of 5×5 convolutional layers and RDB module. Then, f_t^D will be fed into a series of RDB modules. For each RDB module, the output is represented as $f_t^R = \{f_t^{R_1}, \cdots, f_t^{R_N}\}$, where N refers to the number of RDB modules. RDB cell could get the global hierarchical features f_t by fusing the concatenation of local hierarchical features f_t^R with 1×1 convolutional layer as follows:

$$f_t = Conv(CAT(f_t^R)) \tag{2}$$

where $Conv(\cdot)$ refers to convolutional operation. Then, the hidden state h_t could be updated as follows:

$$h_t = H(f_t) \tag{3}$$

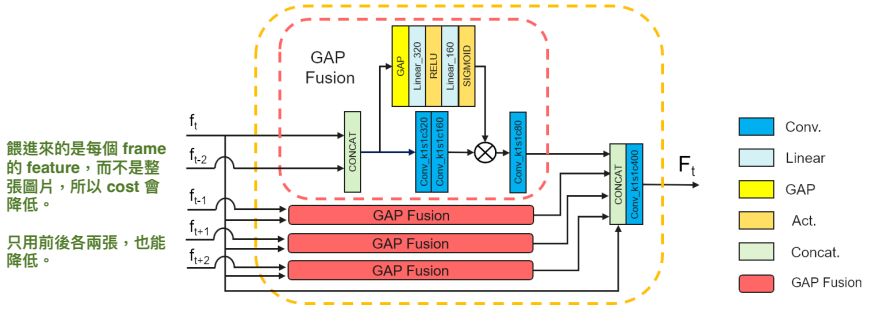


Fig. 4: The structure of global spatio-temporal attention module. f_{t-2}, f_{t-1}, f_{t+1}, f_{t+2} and f_t refer to the hierarchical features of corresponding neighboring frames in the past or future and the current frame, respectively; *linear* refers to fully convolutional layer; GAP refers to global average pooling layer; F_t refers to the output of the GSA module, integrating the effective components of hierarchical features from each past and future frame by GAP Fusion module

where H refers to the hidden state generation function, consisting of 3×3 convolutional layer and RDB module. In short, while processing each frame in the video, the inputs of RDB cell are current blur frame and previous hidden state. Then, RDB cell will generate the hierarchical features of this frame and update the hidden state as well.

3.3 GSA: Global Spatio-temporal Attention Module

The structure of GSA module is shown as Fig. 4. This module aims to extract and fuse the effective components of hierarchical features from future and past frames. Intuitively, the frames which are closer to the current frame in time domain are more likely to have useful information for deblurring of current frame. In the situation of real-time video deblurring, considering that the requirement of low computational cost for each output frame, the number of neighboring hierarchical features that will be fused into current frame should be limited. Furthermore, considering that delaying output by only several frames is usually acceptable, the hierarchical features from the future frames are available for the feature fusion. Therefore, the input of GSA will be hierarchical features of two frames before and two frames after the current frame as $\{f_{t-2}, f_{t-1}, f_t, f_{t+1}, f_{t+2}\}$. Inspired by Squeeze-and-Excitation (SE) block in [9], a submodule named global averaging pooling fusion is proposed, which takes features of current frame and a neighboring frame as input to filter out effective hierarchical features f^e_{t+i} from the neighboring frame as follows:

$$f^c_{t+i} = CAT(f_t, f_{t+i}) \tag{4}$$

$$f^e_{t+i} = L(GAP(f^c_{t+i})) \otimes P(f^c_{t+i}) \tag{5}$$

where $i \in \{-2, -1, 1, 2\}$; $GAP(\cdot)$ refers to global averaging pooling [19]; L refers to a series of linear transformation with activation function as $ReLU$ [24] and

Sigmoid for channel weight generation; P refers to a series of 1×1 convolutional operations for feature fusion. Finally, GSA module will fuse the f_t with all effective hierarchical features from neighboring frames to get the output F_t as follows:

$$F_t = Conv(CAT(f^e_{t-2}, f^e_{t-1}, f^e_{t+1}, f^e_{t+2}, f_t)) \tag{6}$$

The output F_t of GSA module will be upsampled by deconvolutional layers [5] in re-constructor module for generating latent image for the current frame.

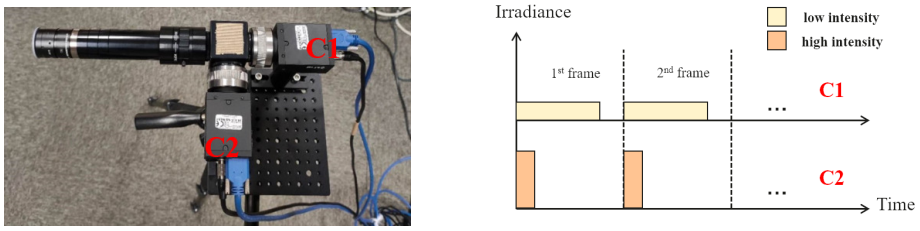
4 Experiment Results

4.1 Implementation Details

Synthesized Dataset We test our model ESTRNN on two public datasets that made by averaging high-FPS video as GOPRO [22] and REDS [21]. We choose the same GOPRO version as [23]. There are 22 training sequences and 11 evaluation sequences in GOPRO with 2103 training samples and 1111 evaluation samples respectively. As for REDS, there are 240 training sequences and 30 evaluation sequences with 100 frames for each sequence. Due to the huge size of REDS dataset and limited computational resources, we train our model and other SoTA models only on first-half training sequences of REDS (120 sequences) for comparison.

Beam-Splitter Dataset (BSD) At present, there are still very limited methods for building a video deblurring dataset. The mainstream way is to average several consecutive short-exposure images in order to mimic the phenomenon of blur caused by relatively long exposure time [15]. This kind of method requires a high-speed camera to capture high-FPS video and then synthesizes pairs of sharp and blurry videos based on the high-FPS video. Video deblurring datasets such as DVD [30], GOPRO [22] and REDS [21] were born by the above method. However, it is questionable whether such a synthetic way truly reflects the blur in real scenarios. In here, we provide a new solution for building video deblurring dataset by using a beam splitter system with two synchronized cameras, as shown in Fig. 5. In our solution, by controlling the length of exposure time and strength of exposure intensity during video shooting as shown in Fig. 5b, the system could obtain a pair of sharp and blurry video samples by shooting video one time.

In this work, we captured beam-splitter datasets (BSD) with two different recording frequency as 15fps and 30fps, respectively. For each frequency, there are 24 sequences of short video with 50 frames for each. We let the exposure time of camera $C1$ and camera $C2$ as 16ms and 2ms to capture blurry and sharp videos, respectively. The intensity of irradiance of $C1$ is $\frac{1}{8}$ of $C2$, in order to keep the total irradiance intensity equalized. This is physically implemented by inserting a 12.5% neutral density filter in the front of $C1$. The video resolution is $720P$ (1280×720). 75% sequences (18) will be randomly selected for training, and the rest sequences (6) will be used for testing.



(a) (b)

Fig. 5: A beam splitter system for building video deblurring dataset. (a) is the profile of our beam splitter system. $C1$ and $C2$ refer to two cameras with same configurations for generating blur and sharp videos, respectively; (b) shows the exposure scheme of $C1$ and $C2$ to generate blurry/sharp video pairs.

Training setting To be fair, we try our best to keep the hyper-parameters as same for each model. We train each model for 500 epochs by ADAM optimizer [16] ($\beta_1 = 0.9, \beta_2 = 0.999$) with initial learning rate as 10^{-4} (decay rate as 0.5, decay step as 200 epochs). We use RGB patches of size 256×256 in subsequence of 10 frames as input to train the models. Also, same data augmentation processes are taken for each model, including 90°, horizontal and vertical flips. Mini-batch size is set to 4 for single GPU (V100) training. For synthentic dataset GOPRO and REDS, the loss function is uniformly defined as $\mathcal{L}2$ loss; while for the proposed dataset BSD, we use $\mathcal{L}1$ for each model as follows:

$$\mathcal{L}2 = \frac{1}{TCHW} \sum_{t}^{T} \|O_t - O_t^{GT}\|_2^2, \tag{7}$$

$$\mathcal{L}1 = \frac{1}{TCHW} \sum_{t}^{T} \|O_t - O_t^{GT}\|_1 \tag{8}$$

where T, C, H, W denote the number of frames and the number of channel, height, width for each frame; O_t^{GT} refers to the ground truth of the t^{th} frame. Source code and dataset will be released on `https://github.com/zzh-tech/ESTRNN`.

4.2 Results

GOPRO First, we compare our method with the SoTA video deblurring methods on GOPRO dataset. We implement 7 variants of our model with different computational cost by modifying the number of channels ($C_\#$) of base model and keeping the number of RDB blocks ($B_\#$) as 9. The larger $C_\#$ is, the higher computational cost it needs. We report the deblurring performance and the corresponding computational cost for processing one frame in the video of all compared models in terms of PSNR [8], SSIM and GMACs, respectively, in Table 1. From the perspective of quantitative analysis, it is clear that our model

Table 1: Quantitative results on both GOPRO and REDS datasets. Cost refers to the computational cost of the model for deblurring one frame of HD(720P) video in terms of GMACs. The meaning of cost is same for other tables and figures in this paper. For our model, $B_{\#}$ and $C_{\#}$ denote the # of RDB blocks in RDB cell and the # of channels for each RDB block, respectively

Model	GOPRO		REDS		Cost
	PSNR	SSIM	PSNR	SSIM	
STRCNN [13]	28.74	0.8465	30.23	0.8708	276.20
DBN [30]	29.91	0.8823	31.55	0.8960	784.75
IFIRNN ($c2h1$) [23]	29.79	0.8817	31.29	0.8913	116.29
IFIRNN ($c2h2$) [23]	29.92	0.8838	31.35	0.8929	167.09
IFIRNN ($c2h3$) [23]	29.97	0.8859	31.36	0.8942	217.89
ESTRNN (B_9C_{60})	30.12	0.8837	31.64	0.8930	92.57
ESTRNN (B_9C_{65})	30.30	0.8892	31.63	0.8965	108.20
ESTRNN (B_9C_{70})	30.45	0.8909	31.94	0.8968	125.55
ESTRNN (B_9C_{75})	30.58	0.8923	32.06	0.9022	143.71
ESTRNN (B_9C_{80})	30.79	0.9016	32.33	0.9060	163.61
ESTRNN (B_9C_{85})	31.01	0.9013	32.34	0.9074	184.25
ESTRNN (B_9C_{90})	31.07	0.9023	32.63	0.9110	206.70

Table 2: Quantitative results on BSD dataset

Model	BSD	PSNR	SSIM	Cost
IFIRNN ($c2h3$)	15fps	34.50	0.8703	217.89
ESTRNN (B_9C_{80})	15fps	35.06	0.8739	206.70
IFIRNN ($c2h3$)	30fps	34.28	0.8796	217.89
ESTRNN (B_9C_{80})	30fps	34.80	0.8835	206.70

can achieve higher PSNR and SSIM value with less computational cost, which means our model has higher network efficiency. To further validate the deblurring performance of proposed model, we also show the deblurred image generated by each model, as illustrated in Fig. 6. We can see the proposed model can restore sharper image with more details, such as the textures of tiles on the path and the characters on the poster.

REDS We also do the comparison on REDS, which has more diverse scenes from different places. From Table 1, we can see our model B_9C_{90} achieves best results as 32.63 PSNR with only around 200 GMACs computational cost for one $720P$ frame. Even our small model B_9C_{60} with cost less than 100 GMACs can achieve same level performance as $c2h3$ of IFIRNN, the computational cost of which is as twise as the former. On the qualitative results as Fig. 7, the proposed model can significantly reduce ambiguous parts for the deblurred frame and the

(a) Blur (b) Deblur (Ours)

(c) Blur (d) STRCNN (e) DBN (f) IFIRNN (g) Ours (h) GT

Fig. 6: Visual comparisons on testing dataset of GOPRO [22].

restored details such as the texture of the wall, characters and human body are closer to the ground truth.

BSD We further compare our model B_9C_{80} with IFIRNN $c2h3$ on our beam splitter video deblurring dataset, shown as Table. 2. The proposed model is superior to the SoTA with around $0.5dB$ more gain under same level of computational cost, on both $15fps$ and $30fps$ BSD dataset. The deblurring results are shown in Fig.8, which proves the effectiveness of our method on realistic video deblurring dataset.

Network Efficiency Analysis We collect the computational cost for one frame as well as the performance (PSNR) of the SoTA lightweight image [32] and video deblurring models on GOPRO dataset, as shown in Fig. 1. The proposed

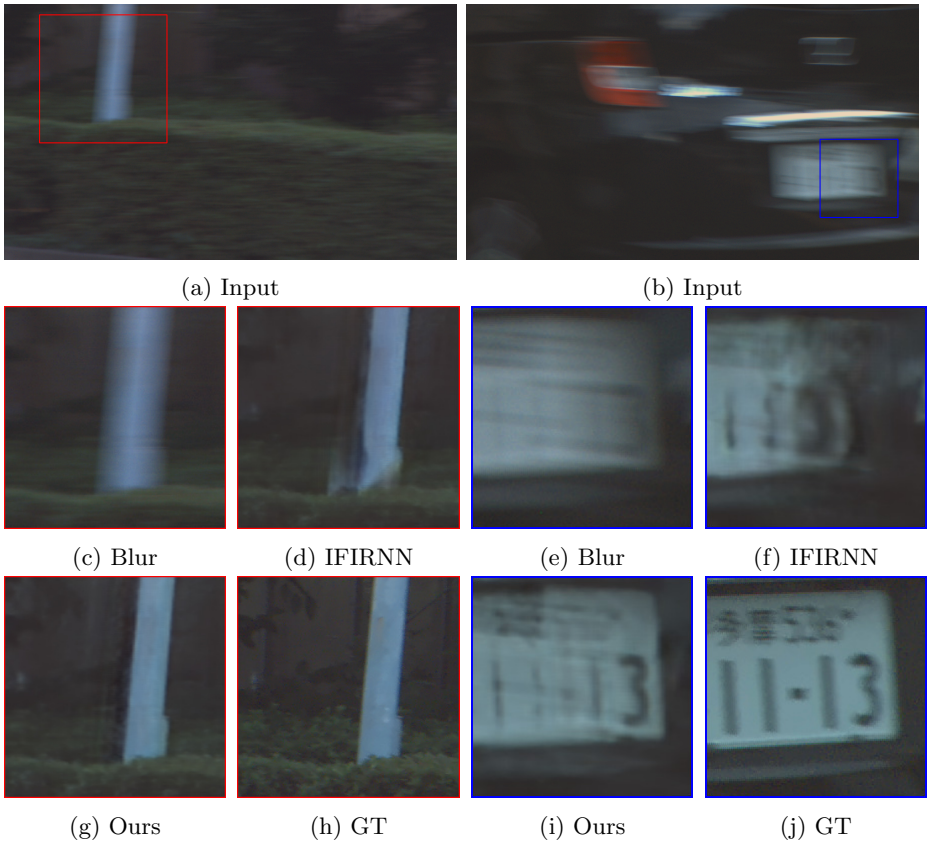
(a) Blur (b) Deblur (Ours)

(c) Blur (d) STRCNN (e) DBN (f) IFIRNN (g) Ours (h) GT

Fig. 7: Visual comparisons on testing dataset of REDS [21].

model includes 7 red nodes that represent different variants of our ESTRNN from B_9C_{60} to B_9C90 in Table. 1. Also, the three blue nodes represent different variants of IFIRNN as $c2h1$, $c2h2$ and $c2h3$. Because the computational cost of different models varies drastically, we take log_{10}(GMACs) as abscissa unit to better display the results. An ideal model with high network efficiency will locate at upper-left corner of the coordinate. The proposed models are closer to the upper-left corner than the existing image or video deblurring models, which reflects the high network efficiency of our model.

Ablation Study We conduct an ablation study to demonstrate the effectiveness of the high-level feature fusion strategy, RDB cell, as well as GSA module, as shown in Table 3. When ablating the modules, we keep the computational cost almost unchanged by adjusting the number of channels ($C_\#$) for fair com-



(a) Input (b) Input

(c) Blur (d) IFIRNN (e) Blur (f) IFIRNN

(g) Ours (h) GT (i) Ours (j) GT

Fig. 8: Visual comparisons on testing dataset of our BSD.

Table 3: Ablation study of ESTRNN. Fusion refers to the fusion strategy that utilizes the high level features from neighboring frames

Model	Fusion	RDB Cell	GSA	PSNR	Cost
B_9C_{110}	×	×	×	30.29	163.48
B_9C_{100}	✓	×	×	30.46	165.59
B_9C_{100}	×	✓	×	30.51	168.56
B_9C_{90}	✓	✓	×	30.55	161.28
B_9C_{85}	×	✓	✓	30.69	162.69
B_9C_{80}	✓	✓	✓	30.79	163.61

parison. Specifically, without using fusion strategy means that the model directly reconstructs the result according to high-level features only from current frame; without RDB cell, the model will use residual block [7] instead, in the same way as [22] does; without GSA module, high-level features will be directly concate-

Table 4: Effectiveness of # of RDB blocks

	B_3C_{80}	B_6C_{80}	B_9C_{80}	$B_{12}C_{80}$	$B_{15}C_{80}$
PSNR	29.74	30.31	30.79	31.03	31.27
Cost	123.03	143.32	163.31	183.90	204.19

Table 5: Effectiveness of # of neighboring frames used by GSA module. $F_{\#}$ and $P_{\#}$ refers to the number of future and past frames used by the model. The base model is B_9C_{80}

	F_0P_1	F_0P_2	F_0P_3	F_1P_1	F_2P_2	F_3P_3
PSNR	30.54	30.57	30.69	30.58	30.79	30.82
Cost	119.93	133.75	148.31	133.75	163.61	196.42

nated in channel dimension. The results clearly demonstrate that each module or design can improve the deblurring efficiency, because each module can improve the overall performance of model when the computational cost keeps unchanged.

We further explore the effectiveness of the number of RDB blocks and the number of past and future frames used by the model as Table. 4 and Table. 5, respectively. First, from the perspective of the number of RDB blocks, this is intuitive that more blocks which means more computational cost will achieve better performance. If we compare the variant $B_{15}C_{80}$ with variant B_9C_{90} in Table. 1 which has almost same computational cost, we can find that it is better to increase the number of RDB blocks rather than the channels, when the number of channels is relatively enough. As for the number of neighboring frames, Table 5 shows that, considering the increased computational cost, the benefit of using more neighboring frames as F_3P_3 is relatively small. Besides, the results of F_0P_1, F_0P_2 and F_0P3 show that the proposed model can still achieve comparative good results even without high-level features borrowed from future frames.

5 Conclusions

In this paper, we proposed a novel RNN-based method for more computational efficient video deblurring. Residual dense block was adopted to the RNN cell to generate hierarchical features from current frame for better restoration. Moreover, to make full use of the spatio-temporal correlation, our model utilized the global spatio-temporal fusion module for fusing the effective components of hierarchical features from past and future frames. The experimental results show that our model is more computational efficient for video deblurring, which can achieve much better performance with less computational cost. Furthermore, we also propose a new method for generating more realistic video deblurring dataset by using a beam splitter based capture system.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
2. Bar, L., Berkels, B., Rumpf, M., Sapiro, G.: A variational framework for simultaneous motion estimation and restoration of motion-blurred video. In: 2007 IEEE 11th International Conference on Computer Vision. pp. 1–8. IEEE (2007)
3. Chakrabarti, A.: A neural approach to blind motion deblurring. In: European conference on computer vision. pp. 221–235. Springer (2016)
4. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision. pp. 764–773 (2017)
5. Dumoulin, V., Visin, F.: A guide to convolution arithmetic for deep learning. ArXiv e-prints (mar 2016)
6. Goldstein, A., Fattal, R.: Blur-kernel estimation from spectral irregularities. In: European Conference on Computer Vision. pp. 622–635. Springer (2012)
7. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
8. Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: 2010 20th International Conference on Pattern Recognition. pp. 2366–2369. IEEE (2010)
9. Hu, J., Shen, L., Sun, G.: Squeeze-and-excitation networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 7132–7141 (2018)
10. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4700–4708 (2017)
11. Hyun Kim, T., Mu Lee, K.: Segmentation-free dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2766–2773 (2014)
12. Hyun Kim, T., Mu Lee, K.: Generalized video deblurring for dynamic scenes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5426–5434 (2015)
13. Hyun Kim, T., Mu Lee, K., Scholkopf, B., Hirsch, M.: Online video deblurring via dynamic temporal blending network. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4038–4047 (2017)
14. Jiang, H., Zheng, Y.: Learning to see moving objects in the dark. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 7324–7333 (2019)
15. Kim, T.H., Nah, S., Lee, K.M.: Dynamic scene deblurring using a locally adaptive linear blur model. arXiv preprint arXiv:1603.04265 (2016)
16. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
17. Lee, H.S., Kwon, J., Lee, K.M.: Simultaneous localization, mapping and deblurring. In: 2011 International Conference on Computer Vision. pp. 1203–1210. IEEE (2011)
18. Levin, A.: Blind motion deblurring using image statistics. In: Advances in Neural Information Processing Systems. pp. 841–848 (2007)
19. Lin, M., Chen, Q., Yan, S.: Network in network. arXiv preprint arXiv:1312.4400 (2013)
20. Michaeli, T., Irani, M.: Blind deblurring using internal patch recurrence. In: European Conference on Computer Vision. pp. 783–798. Springer (2014)

21. Nah, S., Baik, S., Hong, S., Moon, G., Son, S., Timofte, R., Mu Lee, K.: Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
22. Nah, S., Hyun Kim, T., Mu Lee, K.: Deep multi-scale convolutional neural network for dynamic scene deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3883–3891 (2017)
23. Nah, S., Son, S., Lee, K.M.: Recurrent neural networks with intra-frame iterations for video deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8102–8111 (2019)
24. Nair, V., Hinton, G.E.: Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML-10). pp. 807–814 (2010)
25. Nimisha, T.M., Kumar Singh, A., Rajagopalan, A.N.: Blur-invariant deep learning for blind-deblurring. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4752–4760 (2017)
26. Pérez, J.S., Meinhardt-Llopis, E., Facciolo, G.: Tv-l1 optical flow estimation. Image Processing On Line **2013**, 137–150 (2013)
27. Ren, W., Pan, J., Cao, X., Yang, M.H.: Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 1077–1085 (2017)
28. Schuler, C.J., Christopher Burger, H., Harmeling, S., Scholkopf, B.: A machine learning approach for non-blind image deconvolution. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1067–1074 (2013)
29. Seok Lee, H., Mu Lee, K.: Dense 3d reconstruction from severely blurred images using a single moving camera. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 273–280 (2013)
30. Su, S., Delbracio, M., Wang, J., Sapiro, G., Heidrich, W., Wang, O.: Deep video deblurring for hand-held cameras. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1279–1288 (2017)
31. Sun, L., Cho, S., Wang, J., Hays, J.: Good image priors for non-blind deconvolution. In: European Conference on Computer Vision. pp. 231–246. Springer (2014)
32. Tao, X., Gao, H., Shen, X., Wang, J., Jia, J.: Scale-recurrent network for deep image deblurring. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8174–8182 (2018)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)
34. Wang, X., Chan, K.C., Yu, K., Dong, C., Change Loy, C.: Edvr: Video restoration with enhanced deformable convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 0–0 (2019)
35. Wieschollek, P., Hirsch, M., Scholkopf, B., Lensch, H.: Learning blind motion deblurring. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 231–240 (2017)
36. Wu, Y., Ling, H., Yu, J., Li, F., Mei, X., Cheng, E.: Blurred target tracking by blur-driven tracker. In: 2011 International Conference on Computer Vision. pp. 1100–1107. IEEE (2011)
37. Wulff, J., Black, M.J.: Modeling blurred video with layers. In: European Conference on Computer Vision. pp. 236–252. Springer (2014)

38. Xu, L., Jia, J.: Two-phase kernel estimation for robust motion deblurring. In: European conference on computer vision. pp. 157–170. Springer (2010)
39. Yang, J., Nguyen, M.N., San, P.P., Li, X.L., Krishnaswamy, S.: Deep convolutional neural networks on multichannel time series for human activity recognition. In: Twenty-Fourth International Joint Conference on Artificial Intelligence (2015)
40. Zhang, Y., Li, K., Li, K., Wang, L., Zhong, B., Fu, Y.: Image super-resolution using very deep residual channel attention networks. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 286–301 (2018)
41. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image super-resolution. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2472–2481 (2018)
42. Zhang, Y., Tian, Y., Kong, Y., Zhong, B., Fu, Y.: Residual dense network for image restoration. IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)
43. 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 International Conference on Computer Vision. pp. 2482–2491 (2019)
44. Zoran, D., Weiss, Y.: From learning models of natural image patches to whole image restoration. In: 2011 International Conference on Computer Vision. pp. 479–486. IEEE (2011)