Erase or Fill? Deep Joint Recurrent Rain Removal and Reconstruction in Videos (Supplementary Material)

Jiaying Liu, Wenhan Yang, Shuai Yang, Zongming Guo,
Institute of Computer Science and Technology, Peking University, Beijing, P.R. China

Abstract

This supplementary material provides the network configuration of J4R network, details of the training process, evaluations in synthesized training data from single images and more qualitative evaluation results with more metrics. More visual results are provided in the attached video.

1. Network Configuration

We provide the specific network configurations in Tables 1 and 2.

Table 1. Architecture of the J4R network. (Part1)

Module	Layer and Output Name	Type	Kernel	Output Channels	Inputs
–	Conv1	Conv.	3×3	64	Input
	ReLU1	ReLU	–	64	Conv1
CNN Extractor	Conv2	Conv.	3×3	64	ReLU1
	ReLU2	ReLU	–	64	Conv2
	Conv3	Conv.	3×3	64	ReLU2
	Sum1	Sum	–	64	ReLU1, Conv3
	ReLU3	ReLU	–	64	Sum1
	Conv4	Conv.	3×3	64	ReLU3
	ReLU4	ReLU	–	64	Conv4
	Conv5	Conv.	3×3	64	ReLU4
	Sum2	Sum	–	64	ReLU3, Conv5
	ReLU5	ReLU	–	64	Sum2
	Conv6	Conv.	3×3	64	ReLU5
	ReLU6	ReLU	–	64	Conv6
	Conv7	Conv.	3×3	64	ReLU6
	Sum3	Sum	–	64	ReLU5, Conv7
	ReLU7 ($\mathbf{F}_t$)	ReLU	–	64	Sum3
D-Net	Conv8	Conv.	3×3	64	ReLU7
	Conv9	Conv.	3×3	2	Conv8
	Softmax1	Softmax	–	2	Conv9
	Conv10	Conv.	3×3	64	Softmax1
	Conv11	Conv.	3×3	64	Conv10

Table 2. Architecture of the J4R network. (Part2)

Module	Layer and Output Name	Type	Kernel	Output Channels	Inputs
F-Net	Conv12	Conv.	3×3	64	$\mathbf{H}_{t-1}$, $\mathbf{F}_t$, Conv11
	ReLU8 ($\mathbf{r}_t$)	ReLU	–	64	Conv12
	Conv13	Conv.	3×3	64	$\mathbf{H}_{t-1}$, $\mathbf{F}_t$, Conv11
	Sigmoid1 ($\mathbf{Z}_t$)	Sigmoid	–	64	Conv13
	Multiply1	Multiply	–	64	ReLU8, $\mathbf{H}_{t-1}$
	Conv14	Conv.	3×3	64	Multiply1,Sum3
	TanH1	TanH	–	64	Conv14
	Weighting1 ($\mathbf{H}_t$)	Weight	–	64	Sigmoid1, TanH1, ReLU7
C-Net	C_Conv1	Conv.	3×3	64	$\mathbf{H}_t$
	C_ReLU1	ReLU	–	64	C_Conv1
	C_Conv2	Conv.	3×3	64	C_ReLU1
	C_Sum1	Sum	–	64	C_Conv2, Weight1
	C_ReLU2	ReLU	–	64	C_Sum1
	C_Conv3	Conv.	3×3	64	C_ReLU2
	C_ReLU3	ReLU	–	64	C_Conv3
	C_Conv4	Conv.	3×3	64	C_ReLU3
	C_Sum2	Sum	–	64	C_Conv4, C_ReLU2
	C_ReLU4	ReLU	–	64	C_Sum2
R-Net	R_Conv1	Conv.	3×3	64	$\mathbf{H}_t$
	R_ReLU1	ReLU	–	64	R_Conv1
	R_Conv2	Conv.	3×3	64	R_ReLU1
	R_Sum1	Sum	–	64	R_Conv2, Weight1
	R_ReLU2	ReLU	–	64	R_Sum1
	R_Conv3	Conv.	3×3	64	R_ReLU2
	R_ReLU3	ReLU	–	64	R_Conv3
	R_Conv4	Conv.	3×3	64	R_ReLU3
	R_Sum2	Sum	–	64	R_Conv4, R_ReLU2
	R_ReLU4	ReLU	–	64	R_Sum2
JRC-Net	J_Conv1	Conv.	3×3	64	$\mathbf{H}_t$
	J_ReLU1	ReLU	–	64	J_Conv1
	J_Conv2	Conv.	3×3	64	J_ReLU1
	J_Sum1	Sum	–	64	J_Conv2, Weight1
	J_ReLU2	ReLU	–	64	J_Sum1
	J_Conv3	Conv.	3×3	64	J_ReLU2
	J_ReLU3	ReLU	–	64	J_Conv3
	J_Conv4	Conv.	3×3	64	J_ReLU3
	J_Sum2	Sum	–	64	J_Conv4, J_ReLU2
	J_ReLU4	ReLU	–	64	J_Sum2
–	C_Sum2	Conv.	3×3	64	$E(\mathbf{B}_t)$
	R_Sum2	Conv.	3×3	64	$\mathbf{S}_t$
	J_Sum2	Conv.	3×3	64	$\mathbf{B}_t$

2. Training Details

2.1. Single-Frame Rain Removal Pretraining

We first pre-train the rain streak removal network. The pretrained single-frame rain removal network includes a subset of the components in J4R: the first convolutional layer, CNN extractor, R-Net, and the last two convolutions connected to R-Net. The rain streaks are synthesized by non-rain sequences with the rain streak generated by the probabilistic model [1], sharp line streaks [7] and sparkle noises. The training and evaluation sets of *BSD500* [4] are used as the background frames. These images are cropped into 32×32 patches and in total 270,000 non-overlapping patches.

We use Adam optimizer [3] for network training. The initial learning rate is set to 0.001. After 200 epochs, the learning rate is changed to 0.0001. After another 50 epochs, the learning rate is changed to 0.00001. When a total of 300 epochs is reached, the training is stopped. The batch size and weight decay are set to 64 and 0.0001, respectively. The network is initialized by MSRA algorithm [2].

2.2. Multi-Frame Rain Removal Finetuning

After training the rain streak removal network, we fine-tine the whole J4R network on synthesized videos. The background frames of these videos are from two sources: 1) standard testing sequences, including CIF testing sequences, HDTV sequences[1] and HEVC standard testing sequences[2]; 2) synthesized videos from the images in BSD500 [4] with artificially simulated motions, including rescaling and displacement operations. The training videos are cropped into $32 \times 32 \times 9$ cubes. We obtain in total 20,000 cubes from videos and 30,000 cubes from images.

We also use Adam optimizer [3] for fine-tuning. The learning rate of the whole network is set to 0.001. The learning rates of the components that have been pretrained in the first stage are multiplied by a factor of 0.01. A step-wise learning rate decay policy is employed. After 80 epochs, the learning rate is changed to 0.0001. After another 20 epochs, the learning rate is changed to 0.00001. When a total of 120 epochs is reached, the training is stopped.

[1] https://media.xiph.org/video/derf/

[2] http://ftp.kw.bbc.co.uk/hevc/hm-10.0-anchors/bitstreams/

3. Evaluations in Synthesized Training Data from Single Images

We compare the evaluation performance on *RainSynComplex25* of training with and without the augmented video clips from single images. The evaluation performance curves in the training phase are presented in Fig. 1. It is observed that, using augmented video clips significantly boosts the performance.

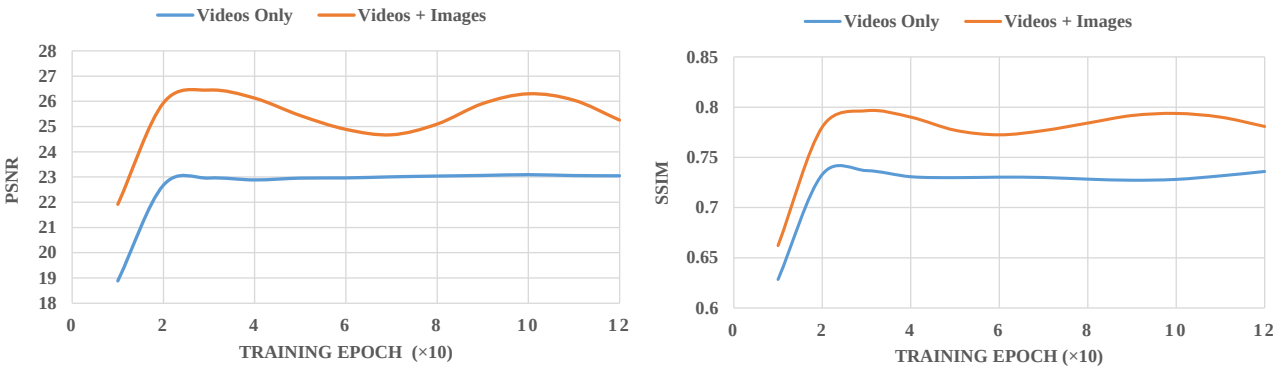

Figure 1. PSRN and SSIM results in the multi-frame finetuing stage.

4. Quantitative Evaluations

Besides PSNR and SSIM, we also use three metrics, VIF [5], FSIM [8], and UQI [6], to compare the results of different methods as shown in Table 3.

Table 3. Performance comparison of different methods on *RainSynLight25* and *RainSynComplex25* in items of VIF, FSIM and UQI.

Dataset	Rain Images			DetailNet		
Metrics	VIF	FSIM	UQI	VIF	FSIM	UQI
RainSynLight25	0.4184	0.8440	0.9845	0.4225	0.8848	0.9882
RainSynComplex25	0.2001	0.6450	0.8467	0.2180	0.7012	0.8695
Dataset	LP			DSC		
Metrics	VIF	FSIM	UQI	VIF	FSIM	UQI
RainSynLight25	0.5135	0.8908	0.9922	0.4293	0.8736	0.9889
RainSynComplex25	0.2478	0.7030	0.8878	0.2109	0.6765	0.9058
Dataset	TCLRM			JORDER		
Metrics	VIF	FSIM	UQI	VIF	FSIM	UQI
RainSynLight25	0.4714	0.9216	0.9960	0.5124	0.9171	0.9932
RainSynComplex25	0.1807	0.6916	0.8862	0.2460	0.7419	0.9560
Dataset	SE			J4R-Net		
Metrics	VIF	FSIM	UQI	VIF	FSIM	UQI
RainSynLight25	0.3851	0.8819	0.9878	**0.6555**	**0.9660**	**0.9985**
RainSynComplex25	0.2264	0.7052	0.8736	**0.3432**	**0.8544**	**0.9765**

References

[1] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *ACM Trans. Graphics*, volume 25, pages 996–1002, 2006. 3

[2] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. IEEE Int'l Conf. Computer Vision*, 2015. 3

[3] D. P. Kingma and J. Ba. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations*, December 2015. 3

[4] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. IEEE Int'l Conf. Computer Vision*, volume 2, pages 416–423, July 2001. 3

[5] H. R. Sheikh and A. C. Bovik. Image information and visual quality. *IEEE Trans. on Image Processing*, 15(2):430–444, Feb 2006. 4

[6] Z. Wang and A. C. Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, March 2002. 4

[7] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, July 2017. 3

[8] L. Zhang, L. Zhang, X. Mou, and D. Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Trans. on Image Processing*, 20(8):2378–2386, Aug 2011. 4