Latency-Constrained Spatial-Temporal Aggregated Architecture Search for Video Deraining

Zhu Liu[1], Long Ma[1], Risheng Liu[2], Xin Fan[2], Zhongxuan Luo[2(✉)], and Yuduo Zhang[3]

[1] School of Software, Dalian University of Technology, Dalian 116024, China
[2] DUT-RU International School of Information Science and Engineering, Dalian University of Technology, Dalian 116024, China
zxluo@dlut.edu.cn
[3] Dalian Minzu University, Dalian 116024, China

Abstract. Existing deep learning-based video deraining techniques have achieved remarkable processes. However, there exist some fundamental issues including plentiful engineering experiences for architecture design and slow hardware-insensitive inference speed. To settle these issues, we develop a highly efficient spatial-temporal aggregated video deraining architecture, derived from the architecture search procedure under a newly-defined flexible search space and latency-constrained search strategy. To be specific, we establish an inter-frame aggregation module to fully integrate temporal correlation according to a set division perspective. Subsequently, we construct an intra-frame enhancement module to eliminate the residual rain streaks by introducing rain kernels that characterize the rain locations. A flexible search space for defining architectures of these two modules is built to avert the demand for expensive engineering skills. Further, we design a latency-constrained differentiable search strategy to automatically discover a hardware-sensitive high-efficient video deraining architecture. Extensive experiments demonstrate that our method can obtain best performance against other state-of-the-art methods.

Keywords: Latency-constrained neural architecture search · Spatial-temporal aggregation · Video deraining

1 Introduction

The degradation of rain streaks is a common imaging factor of severe weather, which leads to the visual-unpleasant quality for human visual system and brings

This work is partially supported by the National Natural Science Foundation of China (Nos. 61922019, 61733002, and 61672125), LiaoNing Revitalization Talents Program (XLYC1807088), and the Fundamental Research Funds for the Central Universities. The first author is a student.

H. Ma et al. (Eds.): PRCV 2021, LNCS 13021, pp. 16–28, 2021.
https://doi.org/10.1007/978-3-030-88010-1_2

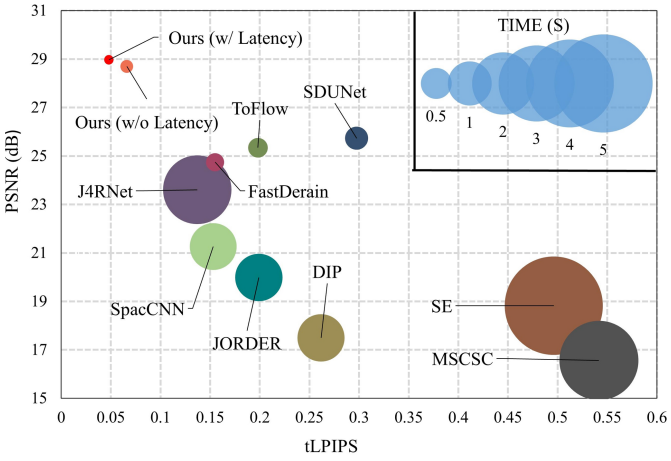

Fig. 1. Numerical performance, temporal consistency and inference speed comparisons on the *RainSynComplex25* [12]. We plotted our schemes (i.e., w/ and w/o latency versions) with other video deraining approaches including JORDER [28], SE [25], DIP [6], MSCSC [9], ToFlow [26], SDUNet [27], FastDerain [7], SpacCNN [1] and J4RNet [12]. We can illustrate the superiority in the aspects of visual quality, temporal preservation and inference speed.

occlusions with blurred objects for a series of high-level vision tasks. Therefore, the extreme urgency of removing rain streaks accurately has been recognized in recent years [13,18,19]. To recover the clear background from rain corrupted observations, numerous methods have been proposed in past decades. We can roughly divide these methods into single image deraining and video deraining.

Recently, single image deraining methods have attracted widespread attentions. Extracting the video as successive frames, these methods can be applied for the video deraining task. The basic formulation of rainy images can be considered as the superimposition of rain streaks and clear background. Based on this principle, conventional model-based schemes were proposed to characterize the rain streaks by exploiting inhere features. For instance, sparse coding methods [9] utilize the high frequency features to learn the rain streaks. A great pile of prior-based methods construct prior knowledge measures such as low rank representation [8], dictionary learning [5], guided filters [33] and Gaussian mixture model [10] to restore the rain-free images. These model-driven methods achieve comparable deraining performance. However, these schemes have high computational burdens and are time-consuming. With the emergence of CNN-based methods, plentiful handcrafted CNN architectures [14,15,17] have been designed for single image deraining. For example, Yang *et al.* [28] proposed a dilated convolution network to capture different properties of rain streaks. Furthermore, attention mechanisms [24] are introduced for image deraining.

In contrast to the single-image deraining schemes, video sequences can provide more contextual compensation and more effective information to discover

and eliminate rain streaks from temporal correlated frames. The classic methods exploited the intrinsic temporal and photometric characteristics of videos to estimate rain streaks. For instance, the directional property of rain streaks [6], the correlation between spatial and temporal information in local patches [2], the shape and size of rain streaks [23] have been investigated widely. The inhere prior knowledge are formulated by Gaussian mixture models, low-rank regularization, sparse coding and tensor models to explore the property of rain streaks for rain detection and removal. Lately, CNN-based schemes have achieved remarkable performances to address the video deraining task. Specifically, a sparse coding method with multi-scale convolution [9] is proposed. Recently, a semi-supervised method [29] is proposed to construct a dynamic generator of rain streaks to explore insightful characteristics of frames.

Different from aforementioned handcrafted architecture construction schemes, Neural Architecture Search (NAS) methodology provides an insightful viewpoint to discover the desired architecture automatically. Especially, differentiable gradient-based schemes [11] utilize the continuous weight relaxation to composite a super-network, reducing the search time effectively. Various gradient-based schemes are performed for low-level vision tasks. In details, Zhang *et al.* [30] proposed a hierarchical search strategy with channel width for image denosing using a series of primitive operators (e.g., 3×3 convolution). Liu *et al.* [16] presented a collaborative search strategy by unrolling architecture for image enhancement. However, the ignorance of exploring task characteristics (e.g., degradation formation) creates limitations of the flexibility for addressing different scenarios. Actually, current CNN-based video deraining methods produce clearer backgrounds since the relevant temporal information from consecutive frames can help the rain removal implicitly. However, there exist several fundamental issues in these existing approaches. Firstly, the temporal correlated information is leveraged as one part of black-box networks. Secondly, the network architectures are based on manual design and relied on heuristic architecture engineering, which needs abundant handcrafted experiences and dedicated adjustment of hyper-meters (e.g., setting different convolution layers and connections). Last but not least, most of existing CNN methods for video deraining do not consider the deployment on hardware, which have huge computation costs.

To mitigate the above issues, we first formulate the video deraining task via investigating the temporal and spatial characteristics aggregation. In detail, we analyze the inner relationships between consecutive frames from the perspective of set division. Based on this principle, we propose an inter-frame aggregation module to fully integrate temporal correlation explicitly for initial rain streak estimation, that breaks down the black-box of temporal information utilization. Furthermore, we construct an intra-frame enhancement module to further eliminate the rain streaks and enhance the details, assisted by one group of learnable rain kernels. Subsequently, we introduce the latency-constrained architecture search strategy to discover the entire structure to avoid lots of manual labor for designing networks. Targeting to establish different principled modules, we introduce diverse operators to construct the specific search space. Constrained by the

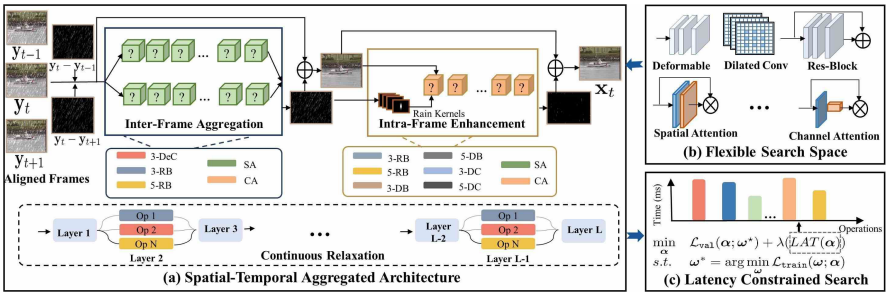

Fig. 2. Overview of main components in the proposed method. We first illustrate the entire architecture that is comprised by the inter-frame aggregation module and intra-frame enhancement module in the subfigure (a). We also illustrate the corresponding search spaces and relaxation formulation. We then demonstrate the concrete details of these operators and the latency constrained search in subfigure (b) and (c).

latency of hardware devices, we can obtain an effective architecture with fast inference speed. The superiority of our method can be demonstrated in Fig. 1. In brief, our contributions can be summarized as three-folds:

- Different from manually designed network-based video deraining schemes, we propose a latency constrained search strategy and a flexible task-specific search space to discover an efficient video deraining architecture.
- We fully explore the intrinsic characteristics of video deraining from the temporal aggregation and spatial enhancement perspectives, to establish a search-based spatial-temporal aggregated macro-structure.
- Comprehensive experiments compared with various state-of-the-art methods on three benchmarks fully reveal the superiority of our method. A series of evaluative experiments demonstrate the effectiveness of our scheme.

2 The Proposed Method

2.1 Spatial-Temporal Aggregated Architecture

Inter-Frame Aggregation Module. We first propose an Inter-Frame Aggregation Module (IFAM) to estimate the major rain streaks by investigating the explicit temporal information from set division perspective. In detail, the current rainy frame (denoted as $\mathbf{y}_t$) can be considered as the union set based on background set ($\Phi_{\mathbf{x}}$) and rain set ($\Phi_{\mathbf{r}}$), i.e., $\Phi_{\mathbf{y}_t} = \Phi_{\mathbf{x}_t} \cup \Phi_{\mathbf{r}_t}$ and $\Phi_{\mathbf{x}_t} \cap \Phi_{\mathbf{r}_t} = \emptyset$, where Φ denotes the set of pixel locations. Leveraging aligned consecutive frames, we can decouple the rain streaks into two parts, $\Phi_{\mathbf{r}_t} = (\sum_{i,i\neq t} \Phi_{\mathbf{r}_t} \cap \Phi_{\mathbf{r}_i}) \cup \widehat{\Phi}_{\mathbf{r}_t}$[1]. $\sum_{i,i\neq t} \Phi_{\mathbf{r}_t} \cap \Phi_{\mathbf{r}_i}$ denotes the rain streaks that contain in the adjacent frames and $\widehat{\Phi}_{\mathbf{r}_t}$ denotes the unique rain streaks generated in current t-th frame or the

[1] We leverage the latest optical flow method RAFT [22] to align the frames.

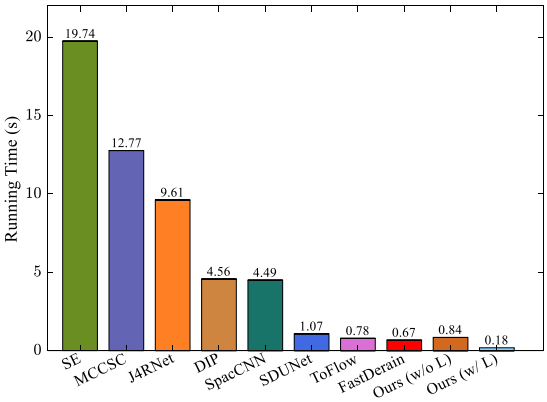

Fig. 3. The average running time calculated on *RainSynComplex25*. "L" denotes the latency constraint.

residual moved rain streaks. In other words, the rain streaks $\mathbf{r}_t$ can be captured by the shared regions in other frames and residue streaks in current frames.

The mentioned intrinsic principle motivates us to design one module for capturing the main rain streaks by utilizing the concatenation of current frames and the coarse rain streaks contained in $\mathbf{y}_t - \mathbf{y}_{t\pm i}$. In this paper, we only utilize three-frames temporal information as the inputs. As shown in the subfigure (a) of Fig. 2, we extract the rain streaks based on two parallel layers with $\mathbf{N}_T$ blocks and obtain the final streaks by the 3×3 convolution.

Intra-Frame Enhancement Module. Subsequently, the residue rain streaks $\widehat{\Phi}$ cannot be removed exactly based on the above formulation. Thus, we propose an Intra-Frame Enhancement Module (IFEM) to perform single-frame deraining and estimate the partial rain streaks. To enhance the spatial structure, we first introduce the successive architecture with $\mathbf{N}_S$ blocks. Then, aiming to focus on the location of rain streaks, we introduce the convolutional dictionary learning mechanism to learn a series of rain kernels (denoted as $\mathbf{C}$). Based on this mechanism, we can obtain the accurate locations and size of residual rain streaks with obvious rain region, i.e., $\widehat{\mathbf{r}}_t = \mathbf{C} \otimes \mathbf{r}_t$. Then we concatenate $\widehat{\mathbf{r}}_t$ and estimated frame as the inputs. The whole module aims to learn the residue streaks.

2.2 Architecture Search

Flexible Search Space. Establishing a task-specific search space is the vital component to perform architecture search. In contrast to adopting the primitive operators (e.g., separable *conv* 3×3) directly, which maybe not suitable for video deraining tasks. Therefore we explore more effective operations to composite our search space. Thus, we list the ten operators that are included in the search space in following: 3×3 Deformable Convolution (i.e., 3-DeC) [4],

$3\times3/5\times5$ Dilated Convolution with dilation rate of 2 (3-DC, 5-DC), $3\times3/5\times5$ Residual Blocks (3-RB, 5-RB), $3\times3/5\times5$ Dense Blocks (3-DB, 5-DB), Channel Attention (CA) and Spatial Attention (SA). The structure details of some operators are shown in subfigure (b) of Fig. 2. Considering the different peculiarity of blocks, we divide these operators into two sub search spaces for each module. In detail, only deformable convolution, residual blocks and attention mechanisms are constituted as the search space of IFAM, which have the better ability to extract or align the features. For instance, deforamble convolution blocks are used widely to represent the shared information across aligned frames. On the other hand, the dilated convolution, residual blocks, dense blocks and attentions are considered in the search of IFEM. These operators are widely used for deraining tasks. Furthermore, we remove the pooling and skip connections from the search space. In order to keep the fair comparison, each operation has three layers of convolutions. Constructing this flexible search space, the performance of searched architecture can be guaranteed.

Latency-Constrained Search. In order to speed up of the inference time on diverse hardware scenarios, we introduce the hardware latency as a regularization term, aiming to discover a architecture with low latency. The search process based on the differentiable search framework [11] can be formulated as:

$$\begin{aligned} &\min_{\boldsymbol{\alpha}} \mathcal{L}_{\mathtt{val}}(\boldsymbol{\alpha}; \boldsymbol{\omega}^{\star}) + \lambda(LAT(\boldsymbol{\alpha})) \\ &s.t.\ \ \boldsymbol{\omega}^{*} = \arg\min_{\boldsymbol{\omega}} \mathcal{L}_{\mathtt{train}}(\boldsymbol{\omega}; \boldsymbol{\alpha}), \end{aligned} \tag{1}$$

where $\boldsymbol{\alpha}$ and $\boldsymbol{\omega}$ denote the architecture and network weights. The formulation of super-net (i.e., continuous relaxation by $\boldsymbol{\alpha}$) is shown in the bottom row of subfigure (a) in Fig. 2. More concretely, the LAT term can be obtained by the weighted linear sum of operations:

$$LAT(\boldsymbol{\alpha}) = \sum_{\mathtt{k}} \sum_{\mathtt{i}} \alpha_{\mathtt{i}}^{\mathtt{k}} LAT(op_{\mathtt{i}}), op_{\mathtt{i}} \in \mathcal{O}, \tag{2}$$

where we denote that $\alpha_{\mathtt{i}}^{\mathtt{k}}$ is the $\mathtt{i}$-th operation weight of $\mathtt{k}$-th cell and $\mathcal{O}$ is the search space. In this manuscript, we only calculate the inference time on GPU.

3 Experimental Results

3.1 Experiment Preparation

Datasets. We utilize three mainstream benchmarks to evaluate our method with various state-of-the-arts. *RainSynLight25*, *RainSynComplex25* are two synthetic datasets, proposed in [12], used to simulate the light rain and heavy rain scenarios of real world. *NTURain* datasets [1] includes the synthetic and real rainy videos. Additionally, several real-world rainy videos are collected from Youtube and mixkit[2] websites for the evaluation.

[2] https://www.youtube.com, https://mixkit.co.

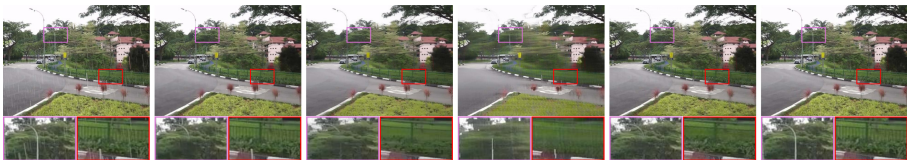

Fig. 4. Visual Comparison among various remarkable approaches on the *NTU* dataset.

Baselines and Metrics. We compare our method with competitive deraining methods: three single frame deraining methods (JORDER [28], SE [25] and SPANet [24]) and eight multi-frame methods (DIP [6], MSCSC [9], ToFlow [26], SDUNet [27], FastDerain [7], SpacCNN [1] and J4RNet [12]). Two widely used numerical metrics, Peak Signal-to-Noise Ratio (PSNR) and Structure Similarity Index (SSIM) are calculated as the criterion terms. Several perceptual metrics are also introduced: Visual Information Fidelity (VIF) [21], Feature SIMilarity (FSIM) [31], Natural Image Quality Evaluator (NIQE) [20], Learned Perceptual Image Patch Similarity (LPIPS) [32] and temporal LPIPS (tLPIPS) [3].

Search Configurations. We define the basic hyper-parameters empirically in our search process. The supernet have only one IFAM and one IFEM. Each of modules has four candidate layers (i.e., $\mathbf{N}_T = \mathbf{N}_S = 4$). We randomly selecte ten video sequences from *RainSynComplex25* to composite the dataset in search phase. We divide it equally to be used for updating the network weights and architecture weights. The loss term is composited by two parts, which is leveraged for the training and validation:

$$\mathcal{L} = \mathcal{L}_{L1}(\mathbf{x}_t, \mathbf{x}_{gt}) + \mathcal{L}_{SSIM}(\mathbf{x}_t, \mathbf{x}_{gt}) + \gamma \mathcal{L}_{L1}(\mathbf{x}_a, \mathbf{x}_{gt}), \tag{3}$$

where the first part composited by previous two term is to restraint the final output $\mathbf{x}_t$. The last term is utilized to restraint the output $\mathbf{x}_a$ of IFAM. We set the γ as 0.1 in our search and training phase. We utilize SGD optimizer with the cosine annealing strategy to perform the search with 150 epochs, where the initial learning rate is 0.0005 and $\lambda = 0.05$. We only reserve the layers with the maximum values in $\boldsymbol{\alpha}$. Derived from the search phase, we can obtain the final architecture. IFAM consists of 5×5 residual block, deformable block, channel attention and 3×3 residual block. IFEM consists of 3×3 dilated convolution, 3×3 residual block, 3×3 residual block and channel attention.

Training Configurations. We propose a stage-wise training strategy to train our searched architecture, rather than adopting end-to-end training straightforwardly. At the first stage, we first train the IFAM with 50 epochs, using L1 and SSIM losses to enforce for utilizing reasonable temporal information and generating rain streaks exactly. Then at the second stage, we train the entire

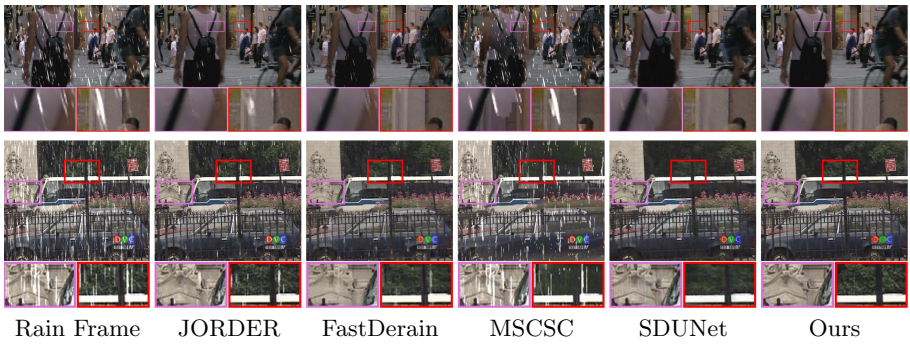

Fig. 5. Visual Comparison among deraining schemes on the synthetic datasets.

architecture (including IFEM) end to end with 200 epochs. We leverage the similar training losses of search phase to constrain the whole training process. Data augmentation, such as the random flipping and shape changing are performed in our training phase. We use Adam as the optimizer and set β_1, β_2 as 0.9, 0.999 respectively. Furthermore, we set the initial learning rate as 0.0005 and perform the cosine annealing strategy to decay the learning rates. Our method is based on the PyTorch framework and runs on a NVIDIA GTX1070 GPU.

3.2 Running Time Evaluation

Figure 3 reports the average running time of various multi-frame deraining methods, which were calculated on *RainSynComplex25*. In detail, we plot the concrete inference speed in Fig. 3 and make a bubble diagram (Fig. 1) to show the performance and inference time simultaneously. Compared with deep learning based methods, our approach significantly reduce the inference time. At the same time, our method also can guarantee the best performance on the challenging *RainSynLight25*. On the other hand, we can obtain faster inference time than existing fastest multi-frame video deraining methods. Both two figures demonstrated the superiority of our method, which obtain the comparable inference time and remarkable performance improvement.

3.3 Quantitative Comparison

We compare our method with a series of remarkable deraining schemes on three mainstream datasets in Table 1. We can observe the remarkable improvement to previous schemes. It is worth noting that we only trained our models on *RainSynLight25* and *RainSynComplex25* and used the model for light video raining to solve the *NTURain* dataset. Compared with S2VD, which is the latest method training on *NTURain*, our method gain 0.69 dB in PSNR and 0.0057 in SSIM on this dataset. Compared on *RainSynLight25* and *RainSynComplex25*, we can observe the consistent improvement than either single-frame deraining

schemes, low-rank based methods or deep learning based multi-frame deraining approaches. Furthermore, compared with multi-frame video deraining methods (e.g., SDUNet and J4RNet), which utilizes the temporal information implicitly, we can gain 4.47 dB and 5.46 dB improvements. That also verifies the effectiveness of our proposed intra-frame aggregation.

Table 1. Quantitative comparison with a series of single image deraining and video deraining methods on *RainSynLight25*, *RainSynComplex25* and *NTURain* benchmarks.

Methods	*RainSynLight25*		*RainSynComplex25*		*NTURain*	
	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
JORDER	31.0298	0.9142	19.9904	0.6085	33.3922	0.9410
SPANet	27.3924	0.8816	18.1857	0.5819	31.6594	0.9388
DIP	28.0723	0.8526	17.8974	0.5316	30.7583	0.8970
SE	25.4435	0.7550	18.8258	0.5897	25.4151	0.7548
MSCSC	24.4424	0.7310	16.5653	0.4810	26.1984	0.7630
ToFlow	32.3821	0.9208	25.3418	0.7695	35.1296	0.9636
FastDerain	29.2038	0.8745	24.7442	0.7434	29.5142	0.9303
SDUNet	29.8117	0.8803	25.7357	0.7594	26.5602	0.8604
SpacCNN	31.6704	0.8997	21.2634	0.5863	33.0235	0.9465
J4RNet	30.5339	0.9071	23.6075	0.7506	31.0203	0.9373
Ours	**35.2668**	**0.9540**	**28.6975**	**0.8660**	**36.7337**	**0.9698**

In Table 2, we also report the perceptual quality evaluation on the *RainSynComplex25* benchmark, using various perceptual metrics. VIF and FSIM are two essential metrics to measure the perceptual quality for human visual system by the low-level features and information fidelity. We can obtain the best results on the both reference-based metrics. Our method has the smallest value under the NIQE metric, which measures the distance to natural images. Moreover, LPIPS and tLPIPS are constructed by the feature distances of AlexNet. tLPIPS

Table 2. Perceptual quality comparison and temporal consistency evaluation on the *RainSynComplex25* benchmark.

Methods	VIF	FSIM	NIQE	LPIPS	tLPIPS
JORDER	0.242	0.738	5.270	0.407	0.199
SpacCNN	0.198	0.759	4.933	0.386	0.153
FastDeRain	0.335	0.861	8.708	0.454	0.155
J4RNet	0.275	0.824	3.804	0.274	0.137
Ours	**0.485**	**0.915**	**3.181**	**0.188**	**0.066**

measures the inter-frame temporal loss. Similarly, the results verify the excellent performance of our scheme for human visual system and temporal consistency.

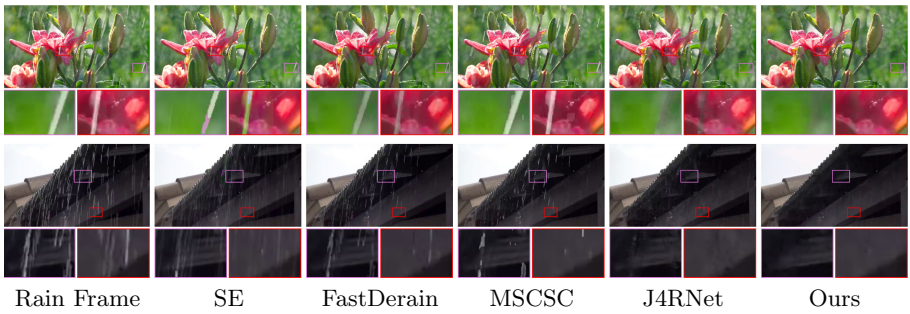

Fig. 6. Visual comparison on two real-world rainy videos.

Table 3. Ablation study on the *RainSynComplex25*. The underlined results are generated by the final configuration of this manuscript.

λ	w/IFAM	w/IFEM	3 frames	5 frames	7 frames	PSNR	Time (s)
0.05	✓	✓	✓	✗	✗	<u>28.70</u>	<u>0.18</u>
0.05	✓	✗	✓	✗	✗	27.46	0.12
0	✓	✓	✓	✗	✗	28.57	0.84
0.5	✓	✓	✓	✗	✗	25.17	0.15
0.05	✓	✓	✗	✓	✗	29.04	0.34
0.05	✓	✓	✗	✗	✓	26.23	0.76

3.4 Qualitative Comparison

We also carry out the qualitative experiment to evaluate the visual quality from the subjective perspective. As for the synthesized videos, we conduct the visual comparisons in Fig. 4 and Fig. 5. As shown in Fig. 4, one can see that our method preserves more texture details (e.g., the streetlight and fences). Obviously, other methods may consider the streetlight as rain streaks and eliminate it wrongly. MSCSC introduces much blurred objects in the frame. Compared on the heavy rainy video, shown in Fig. 5, our method removes the most of rain streaks and keeps regionally consistent with rich details. We also collected two real world rainy videos to evaluate the generation ability for challenging scenarios, which is shown in Fig. 6. The result in the top row depicts the effectiveness of our method to remove the long rain streaks. While other methods still remain the residue

rain to some extent or are failed to remove this kind of rain streaks. As shown in the last row, this frame also contains different types of rain streaks. Our method can preserve the structure well and remove all types of rain streaks.

3.5 Ablation Study

The results of a series of ablation study are reported in Table 3. First, we verify the role of proposed modules respectively. We can conclude that IFAM plays the essential role to estimate the rain streaks and IFEM can remove the residual rain streaks effectively. Moreover, with the increase of λ, the latency can be reduced. However, we need to adjust λ carefully to improve performance and reduce latency. Three-frame video deraining schemes obtain the best balance between numerical results and inference time, which is shown in the first row. Large frames cannot obtain the best numerical results. The possible reason is that the fast movement of rain streaks cannot be captured entirely and the temporal information cannot be utilized sufficiently in seven frames.

4 Conclusions

In this paper, we settled the video deraining by investigating the inhere characteristics from temporal correlation and spatial structure perspectives. A novel temporal and spatial aggregation architecture was proposed and constructed by the automatic architecture search. Leveraging an efficient and compact search space and coupling with the hardware constraint, the architecture can guarantee outstanding performance for video deraining and fast inference time. Consistent improvements of numerical and visual performances demonstrate the superiority of our method against various state-of-the-art deraining schemes.

References

1. Chen, J., Tan, C.H., Hou, J., Chau, L.P., Li, H.: Robust video content alignment and compensation for rain removal in a cnn framework. In: CVPR (2018)
2. Chen, Y.L., Hsu, C.T.: A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In: IEEE ICCV, pp. 1968–1975 (2013)
3. Chu, M., Xie, Y., Mayer, J., Leal-Taixé, L., Thuerey, N.: Learning temporal coherence via self-supervision for gan-based video generation. ACM TOG **39**(4), 75–1 (2020)
4. Dai, J., et al.: Deformable convolutional networks. In: CVPR (2017)
5. Deng, L.J., Huang, T.Z., Zhao, X.L., Jiang, T.X.: A directional global sparse model for single image rain removal. Appl. Math. Model. **59**, 662–679 (2018)
6. Jiang, T.X., Huang, T.Z., Zhao, X.L., Deng, L.J., Wang, Y.: A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In: IEEE CVPR, pp. 4057–4066 (2017)
7. Jiang, T., Huang, T., Zhao, X., Deng, L., Wang, Y.: Fastderain: a novel video rain streak removal method using directional gradient priors. IEEE TIP **28**(4), 2089–2102 (2019)

8. Kim, J.H., Sim, J.Y., Kim, C.S.: Video deraining and desnowing using temporal correlation and low-rank matrix completion. IEEE Trans. Image Process. **24**(9), 2658–2670 (2015)
9. Li, M., et al.: Video rain streak removal by multiscale convolutional sparse coding. In: IEEE CVPR, pp. 6644–6653 (2018)
10. Li, Y., Tan, R.T., Guo, X., Lu, J., Brown, M.S.: Rain streak removal using layer priors. In: IEEE CVPR, pp. 2736–2744 (2016)
11. Liu, H., Simonyan, K., Yang, Y.: Darts: differentiable architecture search. In: ICLR (2019)
12. Liu, J., Yang, W., Yang, S., Guo, Z.: Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In: IEEE CVPR, pp. 3233–3242 (2018)
13. Liu, R., Cheng, S., He, Y., Fan, X., Lin, Z., Luo, Z.: On the convergence of learning-based iterative methods for nonconvex inverse problems. IEEE TPAMI (2019)
14. Liu, R., Liu, J., Jiang, Z., Fan, X., Luo, Z.: A bilevel integrated model with data-driven layer ensemble for multi-modality image fusion. IEEE Trans, Image Process (2020)
15. Liu, R., Liu, X., Yuan, X., Zeng, S., Zhang, J.: A hessian-free interior-point method for non-convex bilevel optimization. In: ICML (2021)
16. Liu, R., Ma, L., Zhang, J., Fan, X., Luo, Z.: Retinex-inspired unrolling with cooperative prior architecture search for low-light image enhancement. In: IEEE CVPR (2021)
17. Liu, R., Mu, P., Chen, J., Fan, X., Luo, Z.: Investigating task-driven latent feasibility for nonconvex image modeling. IEEE TIP **29**, 7629–7640 (2020)
18. Liu, R., Mu, P., Yuan, X., Zeng, S., Zhang, J.: A generic first-order algorithmic framework for bi-level programming beyond lower-level singleton. In: ICML (2020)
19. Liu, R., Zhong, G., Cao, J., Lin, Z., Shan, S., Luo, Z.: Learning to diffuse: A new perspective to design pdes for visual analysis. IEEE TPAMI (2016)
20. Mittal, A., Soundararajan, R., Bovik, A.C.: Making a "completely blind" image quality analyzer. IEEE SPL, pp. 209–212 (2012)
21. Sheikh, H.R., Bovik, A.C.: Image information and visual quality. IEEE TIP
22. Teed, Z., Deng, J.: Raft: Recurrent all-pairs field transforms for optical flow. IEEE ECCV (2020)
23. Wang, H., Xie, Q., Zhao, Q., Meng, D.: A model-driven deep neural network for single image rain removal. In: IEEE CVPR, pp. 3103–3112 (2020)
24. Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., Lau, R.W.: Spatial attentive single-image deraining with a high quality real rain dataset. In: IEEE CVPR, pp. 12270–12279 (2019)
25. Wei, W., Yi, L., Xie, Q., Zhao, Q., Meng, D., Xu, Z.: Should we encode rain streaks in video as deterministic or stochastic? In: IEEE ICCV, pp. 2516–2525 (2017)
26. Xue, T., Chen, B., Wu, J., Wei, D., Freeman, W.T.: Video enhancement with task-oriented flow. IJCV **127**(8), 1106–1125 (2019)
27. Xue, X., Ding, Y., Mu, P., Ma, L., Liu, R., Fan, X.: Sequential deep unrolling with flow priors for robust video deraining. In: IEEE ICASSP, pp. 1813–1817 (2020)
28. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Deep joint rain detection and removal from a single image. IEEE CVPR, pp. 1685–1694 (2017)
29. Yue, Z., Xie, J., Zhao, Q., Meng, D.: Semi-supervised video deraining with dynamic rain generator. CVPR (2021)
30. Zhang, H., Li, Y., Chen, H., Shen, C.: Memory-efficient hierarchical neural architecture search for image denoising. In: IEEE CVPR, pp. 3657–3666 (2020)

31. Zhang, L., Zhang, L., Mou, X., Zhang, D.: Fsim: a feature similarity index for image quality assessment. IEEE TIP
32. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: IEEE CVPR, pp. 586–595 (2018)
33. Zheng, X., Liao, Y., Guo, W., Fu, X., Ding, X.: Single-Image-Based Rain and Snow Removal Using Multi-guided Filter. In: Lee, M., Hirose, A., Hou, Z.-G., Kil, R.M. (eds.) ICONIP 2013. LNCS, vol. 8228, pp. 258–265. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-42051-1_33