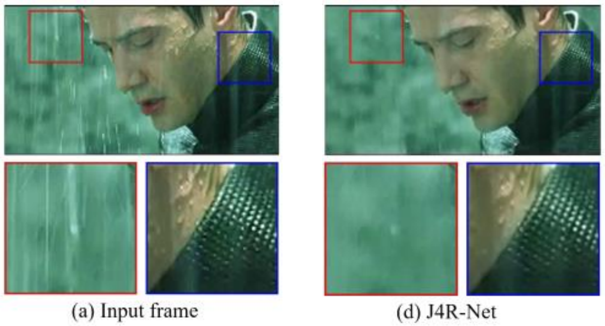
Erase or Fill? Deep Joint Recurrent Rain Removal and Reconstruction in Videos

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



CVPR 2018

Slides compiled by Lars Gjesteby

October 3, 2018

Motivation

- Rain streaks obstruct and blur scenes in videos, which hinder outdoor computer vision applications
- Practical scenarios may be more complex than the *additive rain model*: $\mathbf{O} = \mathbf{B} + \mathbf{S}$
- Low light transmittance occludes corresponding background regions
- Spatial and temporal redundancies should be considered together
- Paired videos for training are difficult to obtain

Approach

- Hybrid rain model
- Joint Recurrent Rain Removal and Reconstruction Network (J4R-Net)
- Integrate degradation classification, spatial texture-based rain removal, and temporal coherence-based background detail reconstruction
- Use synthetic videos from natural images with artificially simulated motions to train de-raining networks

Hybrid Rain Model

$$\mathbf{O}_t = (1 - \alpha_t)(\mathbf{B}_t + \mathbf{S}_t) + \alpha_t \mathbf{A}_t$$

$$\alpha_t(i,j) = \begin{cases} 1, & \text{if } (i,j) \in \Omega_{\mathbf{S}} \\ 0, & \text{if } (i,j) \notin \Omega_{\mathbf{S}} \end{cases}$$

$\mathbf{O}_t$ - Captured image

$\mathbf{B}_t$ - Background without streaks

$\mathbf{A}_t$ - Rain reliance map

$\mathbf{S}_t$ - Streak image

$\Omega_{\mathbf{S}}$ - Rain occlusion region (where light transmittance is low)

Formulation

$$\mathbf{O}(i,j,t) = (1-\alpha(i,j,t))\left(\mathbf{B}(i,j,t)+\mathbf{S}(i,j,t)\right) + \alpha(i,j,t)\mathbf{A}(i,j,t),$$

- Goal: Recover $\mathbf{B}_t$ given $\mathbf{O}_t$
- First, learn a mapping for α_t:

$$\hat{\alpha}(i,j,t) = F_{\alpha}\left(\{\mathbf{O}(x,y,z) | (x,y,z) \in \epsilon\,(i,j,t)\}\right)$$

- Then $\mathbf{B}_t$ can be derived for two cases:
  - $\alpha_t = 0$

$$\hat{\mathbf{S}}(i,j,t) = F_{\mathbf{S}}\left(\{\mathbf{O}(x,y,z) | (x,y,z) \in \epsilon\,(i,j,t)\}\right)$$

  - $\alpha_t = 1$ (In this case, $\mathbf{O}_t$ contains no information about $\mathbf{B}_t$, so the missing data is inferred from neighboring pixels)

$$\hat{\mathbf{A}}(i,j,t) = F_{\mathbf{A}}\left(\{\mathbf{O}(x,y,z) | (x,y,z) \in \epsilon\,(i,j,t)\}\right)$$

Reconstruction

- $\alpha_t = 0$

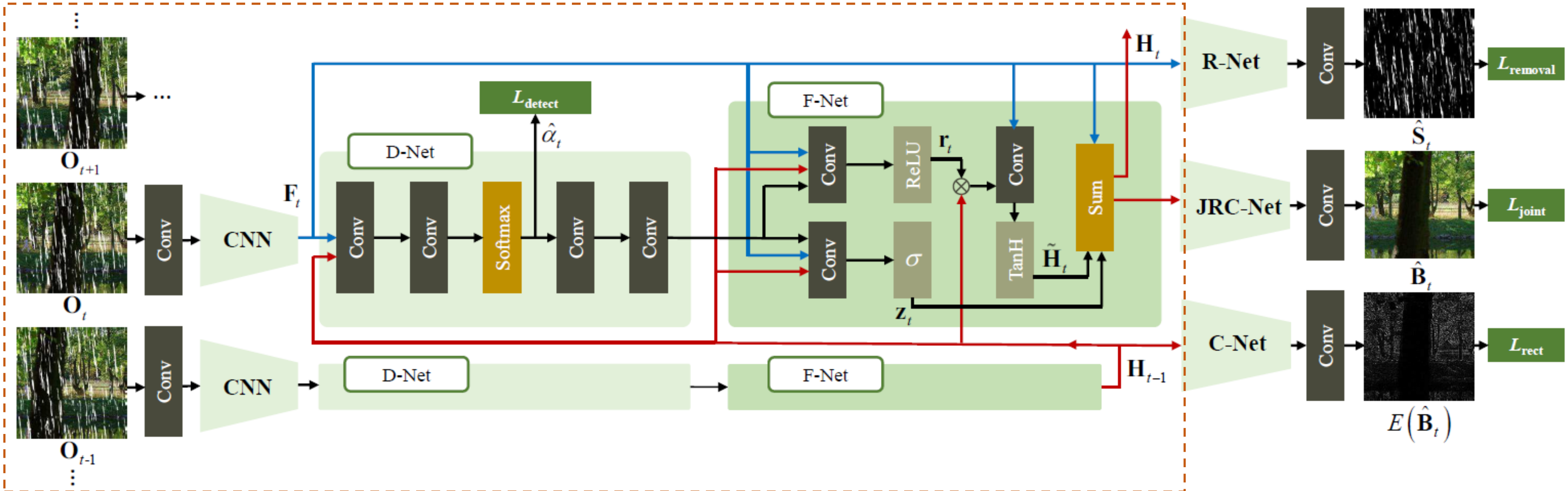
$$\hat{\mathbf{B}}(i,j,t) = \mathbf{O}(i,j,t) - \hat{\mathbf{S}}(i,j,t)$$

- $\alpha_t = 1$

$$\hat{\mathbf{B}}(i,j,t) = F_{\mathbf{B}}\left(\{\mathbf{O}(x,y,z) | (x,y,z) \in \epsilon^{\alpha_0}(i,j,t)\} \right.$$
$$\left. \hat{\mathbf{A}}(i,j,t) \right)$$

$\epsilon^{\alpha_0}(i,j,t)$ - Neighboring pixels in non-occlusion regions whose $\hat{\alpha}$ value is zero. Approximated by temporally aggregated features from adjacent frames

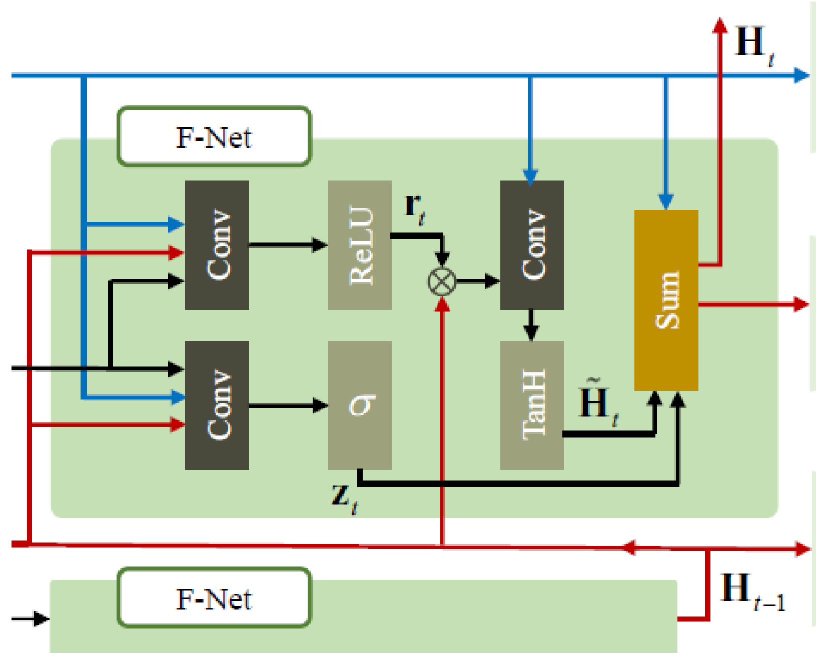
J4R-Net Structure



- CNN: 2-layer feature extractor to estimate single-frame rain streaks ($\mathbf{F}_t$)
- Degradation Classification Net (D-Net): Predicts rain degradation map (α_t) from $\mathbf{F}_t$ and aggregate memory from previous video frame ($\mathbf{H}_{t-1}$)
- Fusion Net (F-Net): Gated RNN (GRU) to generate aggregate memory ($\mathbf{H}_t$) from spatial features, temporal features, and degradation-dependent features

Gated Recurrent Unit (GRU)

- Similar to LSTM, but combines input and forget gates into “update” gate



$$\mathbf{H}_t^j = \left(1 - \mathbf{z}_t^j\right) \mathbf{H}_{t-1}^j + \mathbf{z}_t^j \widetilde{\mathbf{H}}_t^j,$$

Output aggregate memory

$$\widetilde{\mathbf{H}}_t^j = \tanh\left(\mathbf{W}_h \mathbf{F}_t + \mathbf{U}_h\left(\mathbf{r}_t^j \odot \mathbf{H}_{t-1}\right)\right)^j,$$

New information generated

$$\mathbf{z}_t^j = \sigma\left(\mathbf{W}_z \mathbf{F}_t + \mathbf{U}_z \mathbf{H}_{t-1}^j + \mathbf{V}_z \mathbf{f}_{t,4}^d\right)^j,$$

Update gate

$$\mathbf{r}_t^j = \mathrm{ReLU}\left(\mathbf{W}_r \mathbf{F}_t + \mathbf{U}_r \mathbf{H}_{t-1} + \mathbf{V}_r \mathbf{f}_{t,4}^d\right)^j$$

Read gate

Closer Look at F-Net Limits

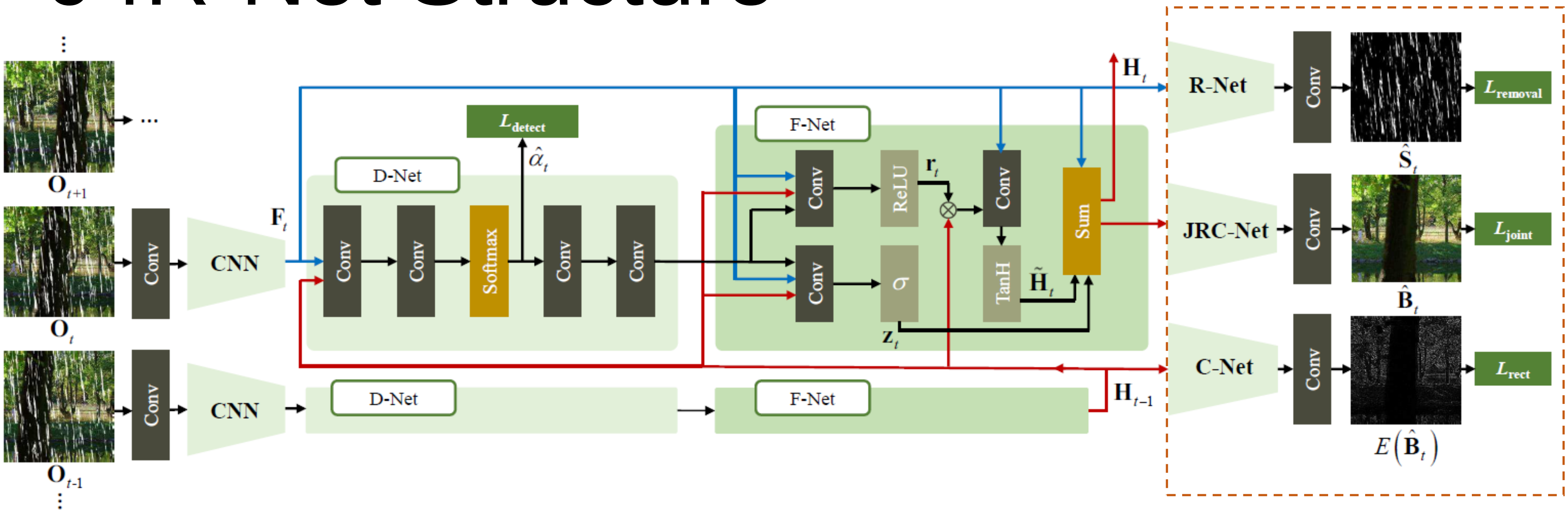
- If “read gate” is 0 and “update gate” is 1, the network ignores accumulated memory from previous time-steps and just focuses on the current frame:

$$\mathbf{H}_t^j = \tanh\left(\mathbf{W}_h \mathbf{F}_t\right)^j$$

- If “read gate” is large and “update gate” is 0, then the output depends more on accumulated memory of previous frames:

$$\mathbf{H}_t^j = \tanh\left(\mathbf{U}_h\left(\mathbf{r}_t^j \odot \mathbf{H}_{t-1}\right)\right)^j$$

J4R-Net Structure



- Rain Removal Net (R-Net): Takes $\mathbf{F}_t$ as input to estimate the rain streaks based on spatial appearances
- Reconstruction Net (C-Net): Fills in missing rain occlusion regions with structural details (high-pass filter) based on temporal redundancy ($\mathbf{H}_{t-1}$).
- Joint Rain Removal and Reconstruction Net (JRC-Net): Final output estimates background image from both information types

Loss Functions

$$l_{\text{all}} = l_{\text{joint}} + \lambda_d l_{\text{detect}} + \lambda_c l_{\text{rect}} + \lambda_r l_{\text{removal}},$$

$$l_{\text{joint}} = \left\| \hat{\mathbf{B}}_t - \mathbf{b}_t \right\|_2^2,$$

$$l_{\text{detect}} = \log\left(\sum_{k=1,2} \exp\left(\mathbf{f}_{t,2}^{d}\left(k\right) \right) \right) - \alpha_t,$$

$$l_{\text{rect}} = \left\| E\left(\hat{\mathbf{B}}_t \right) - E\left(\mathbf{b}_t \right) \right\|_2^2,$$

$$l_{\text{removal}} = \left\| \hat{\mathbf{S}}_t - \mathbf{s}_t \right\|_2^2,$$

where $E(\cdot)$ is a high-pass filter. λ_d, λ_c, and λ_r are set to 0.001, 0.0001, and 0.0001, respectively.

Training

- Pre-train single-frame rain removal network using components of J4R-Net: the first convolutional layer, CNN extractor, R-Net, and last two convolutions connected to R-Net
  - 32x32 image patches (270k)
  - Learning rate decays from 10^{-3} to 10^{-5}
  - 300 epochs
- J4R-Net is then fine-tuned
  - Videos cropped to 32x32x9 (20-30k)
  - Learning rate decays from 10^{-3} to 10^{-5}
  - 120 epochs

Evaluation Methods

- Compare with six state-of-the-art methods:
  - Discriminative sparse coding (DSC) - **single-frame**
  - Layer priors (LP) - **single-frame**
  - Joint rain detection and removal (JORDER) - **single-frame, deep learning-based**
  - Deep detail network (DetailNet) - **single-frame, deep learning-based**
  - Stochastic encoding (SE) - **video**
  - Temporal correlation and low-rank matrix completion (TCLRM) - **video**
- Datasets:
  - *RainSynLight25* – synthesized by non-rain sequences with rain streaks generated by the probabilistic model
  - *RainSynComplex25* – synthesized by non-rain sequences with rain streaks generated by the probabilistic model, sharp line streaks, and sparkle noises
  - *RainPractical10* – ten rain video sequences collected from practical scenes from Youtube, GIPHY, and movie clips

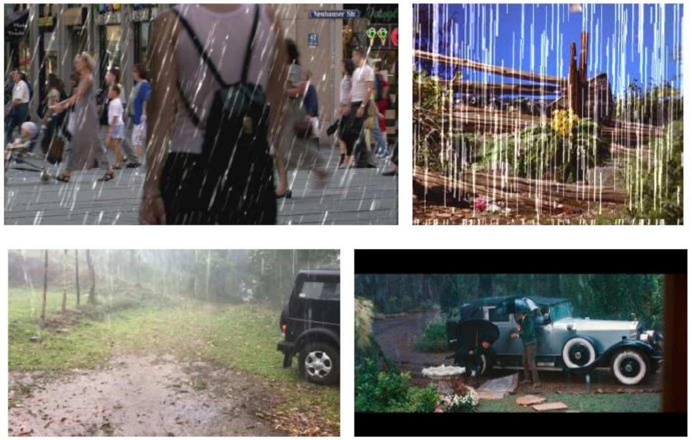


Figure 6. Top left panel: one example of *RainSynLight25*. Top right panel: one example of *RainSynComplex25*. Bottom panel: two examples of *RainPractical10*.

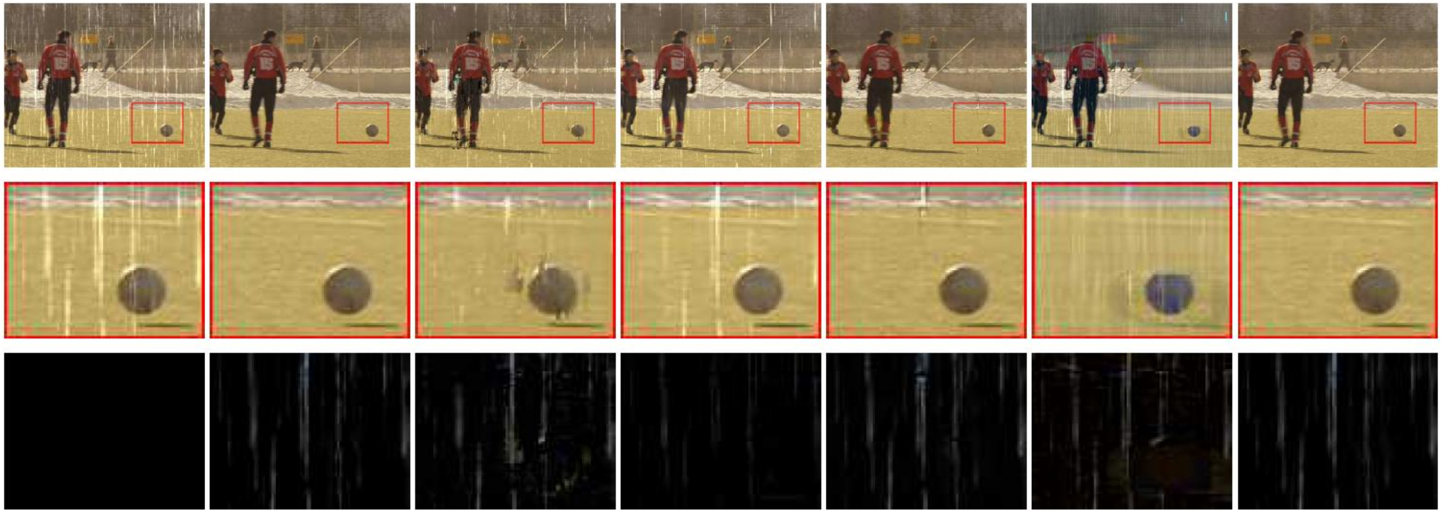
Quantitative Evaluation

Table 1. PSNR and SSIM results among different rain streak removal methods on *RainSynLight25* and *RainSynComplex25*.

Dataset	Rain Images		DetailNet		TCLRM		JORDER	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Light	23.69	0.8058	25.72	0.8572	28.77	0.8693	30.37	0.9235
Heavy	14.67	0.4563	16.50	0.5441	17.31	0.4956	20.20	0.6335
Dataset	LP		DSC		SE		Ours	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Light	27.09	0.8566	25.63	0.8328	26.56	0.8006	**32.96**	**0.9434**
Heavy	17.65	0.5364	17.33	0.5036	16.76	0.5293	**24.13**	**0.7163**

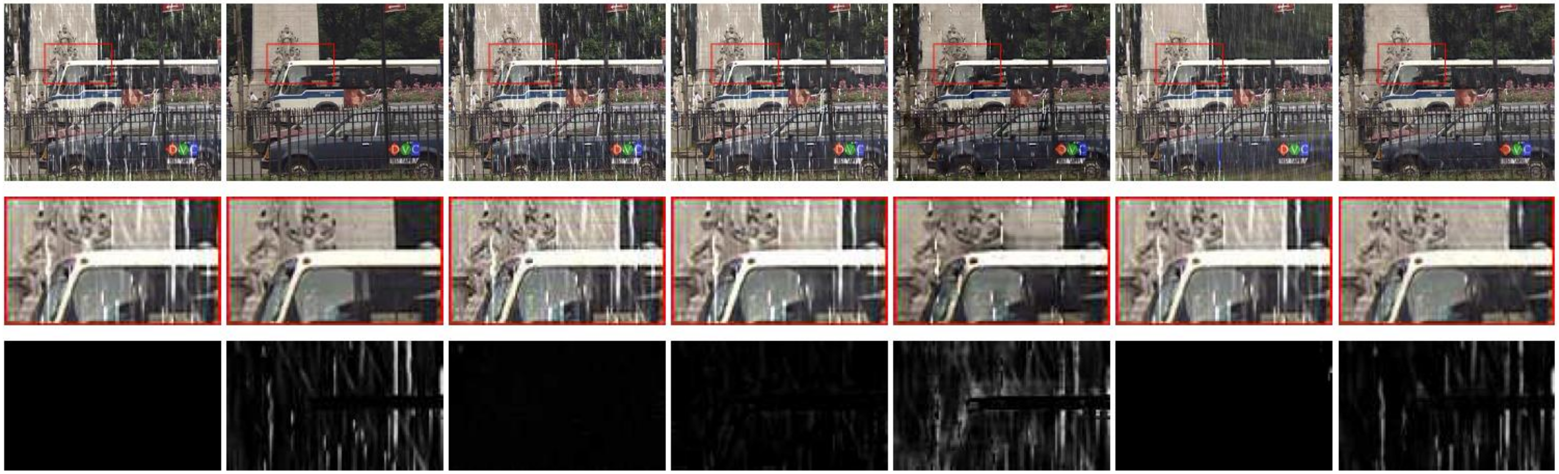
- J4R-Net outperforms all prior algorithms on PSNR and SSIM metrics, including JORDER, the state-of-the-art single-image rain removal method

Results on Synthetic Datasets



(a) Rain image (b) Ground truth (c) TCLRM (d) DetailNet (e) JORDER (f) SE (g) J4R-Net

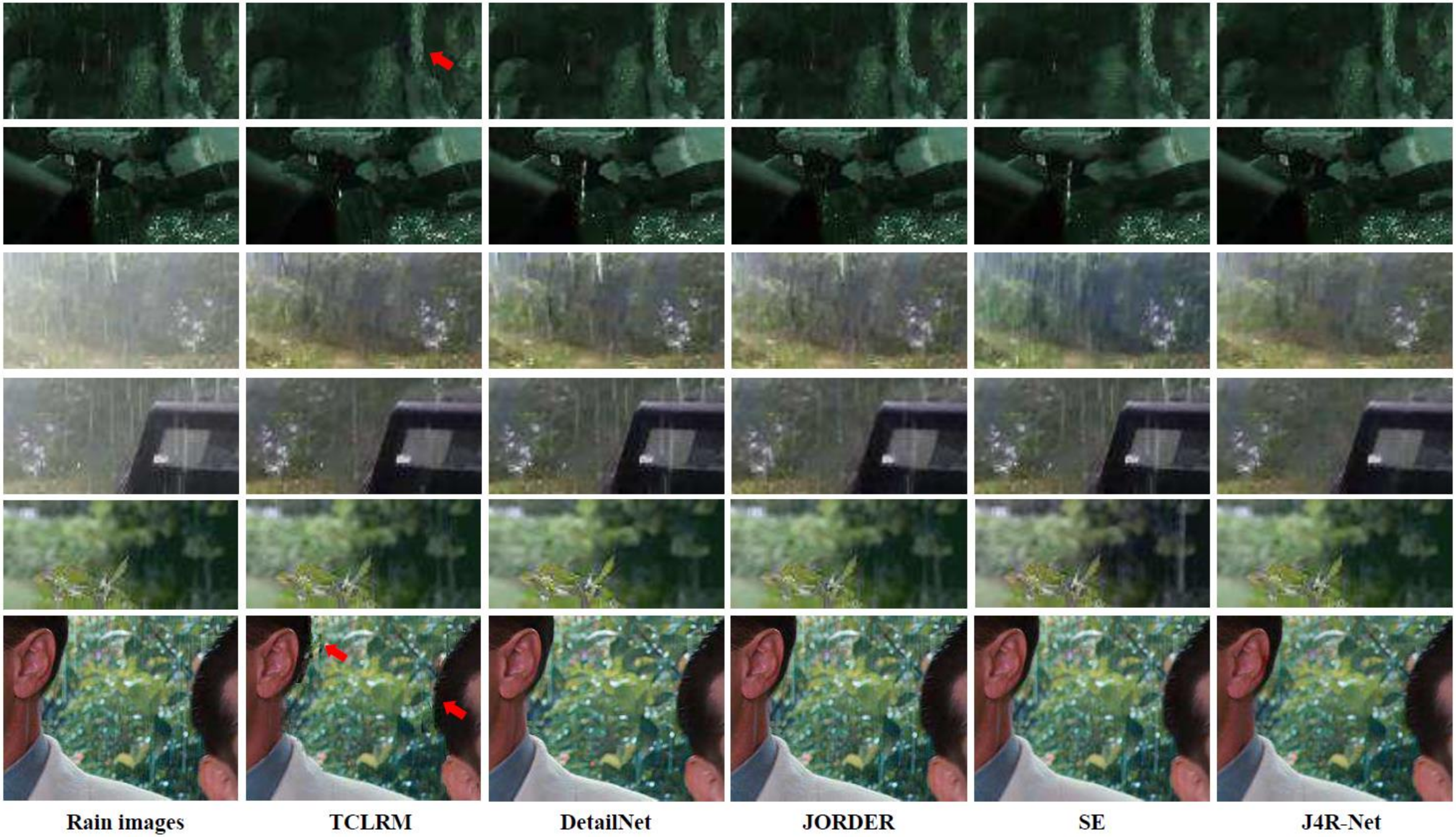
Figure 7. Results of different methods on an example of *RainSynLight25*. From top to down: whole image, local regions of the estimated background layer, and local regions of the estimated rain streak layer.



(a) Rain image (b) Ground truth (c) TCLRM (d) DetailNet (e) JORDER (f) SE (g) J4R-Net

Figure 8. Results of different methods on an example of *RainSynComplex25*. From top to down: whole image, local regions of the estimated background layer, and local regions of the estimated rain streak layer.

Results on Practical Images/Frames



Paper Conclusions

- Hybrid rain model predicts both rain streaks and occlusions
- Spatial, temporal, and degradation information used in parallel
- RNN gate makes trade-off between removing rain streak removal and reconstructing background details