

Rain Removal Collections

Zhi Li

June 17, 2019

1 Single Image Based Methods

1.1 SPANet

Spatial Attentive Single-Image Deraining with a High Quality Real Rain Dataset

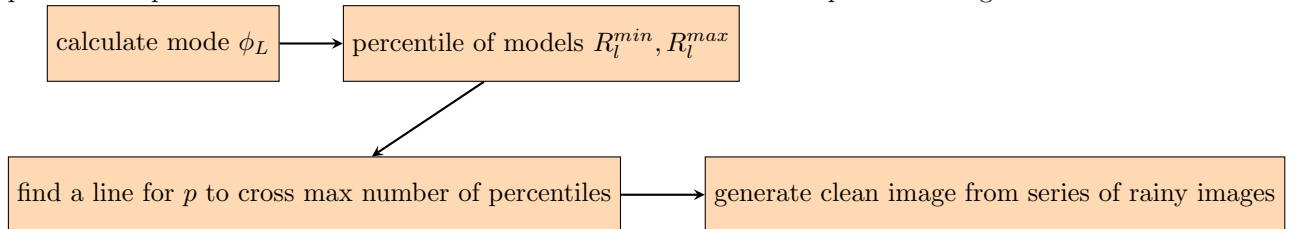
1.1.1 Key Points

- Created Real Images
- Network (Attentive Network with four-directional feature map)

1.1.2 Create real images

since previous works are highly dependent on synthetic dataset to train the model, which is an ill-posed question. When their model is performed on real-world situation, they tend to be illy behaved, and of course not scaleable.

Author generates series of real world images to train. He extracts successive images at one rainfall event, and then depicts histogram of every pixel, followed by manually inspection whether this pixel is occupied with rainfall or not because we understand that rain pixels are brighten.



1.1.3 Network

“Recurrent neural networks with ReLU and identity matrix initialization (IRNN) for natural language processing have shown to be easy to train, good at long range dependencies as well as efficient. When applied to computer vision problems, their key advantage is that information can be efficiently propagated across the entire image to accumulate long range varying contextual information.”

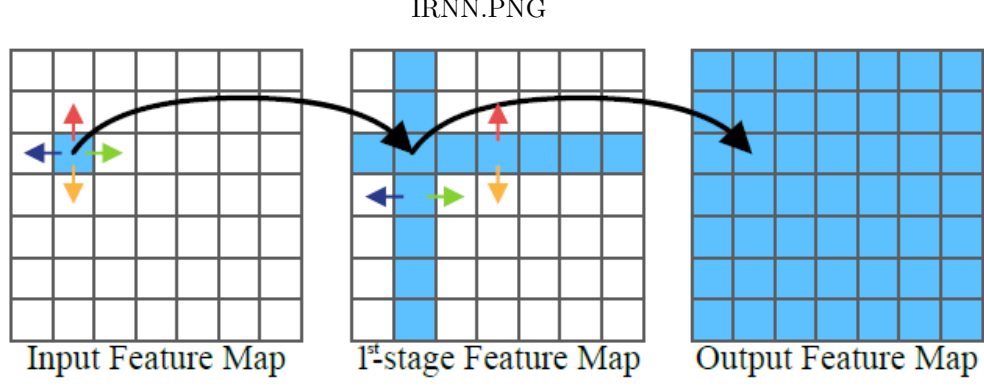


Figure 1: two-round four-directional IRNN schemetization

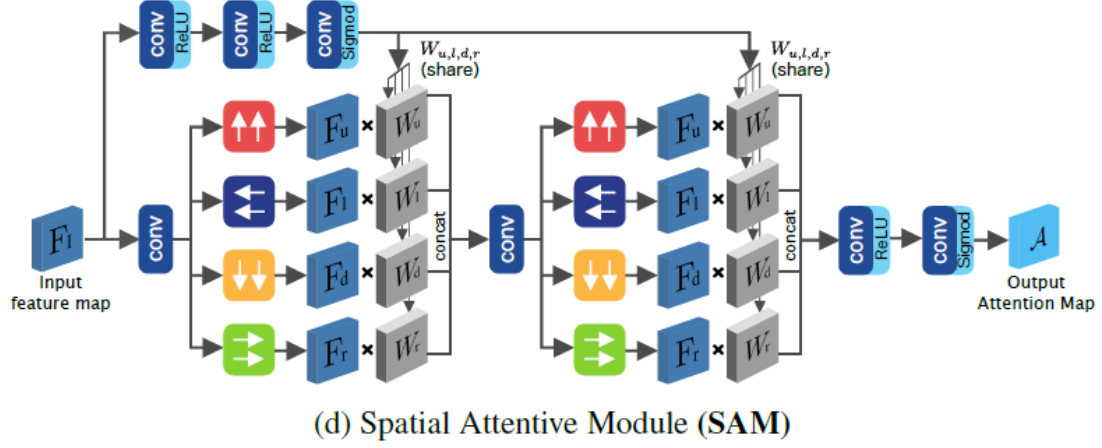


Figure 2: Network description

1.1.4 Loss Function

$$L_{total} = L_1 + L_{SSIM} + L_{Att} \quad (1)$$

$$L_{Att} = \|A - M\|_2 \quad (2)$$

where A is the attention map from the first SAM in the network and M is the binary map of the rain streaks, which is computed by thresholding the difference between the rain image and clean image. In this binary map, a 1 indicates that the pixel is covered by rain and 0 otherwise.

1.2 JORDER

Deep Joint Rain Detection and Removal from a Single Image

1.2.1 Key Points

- separate background learning and rain streak learning
- network representing rainfall accumulation
- recurrent rain detection and removal network

1.2.2 Binary Rain Streak Map

$$O = B + SR \quad (3)$$

SR is binary values, where 1 indicates rain regions and 0 indicates non-rain regions.”(1) it gives additional information for the network to learn about rain streak regions. (2) it allows a new rain removal pipeline to detect rain regions first, and the to operate differently on rain-streak and non-rain-streak regions, preserving background details.”

1.2.3 Rain Accumulation and Heavy Rain

$$O = \alpha(B + \sum_{t=1}^s S_t R) + (1 - \alpha) * A \quad (4)$$

t : overlapping streak number

s : number of shape and direction

A : global atmospheric light

α : scene transmission

1.2.4 Network

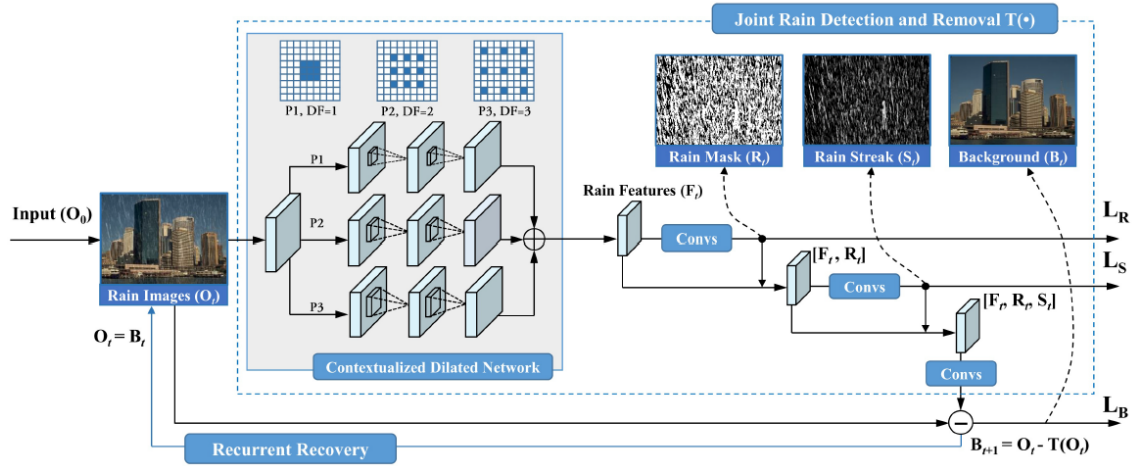


Figure 3: JORDEN network schematization

For the training part, the author selected set of corresponding rain images, background images, rain regions maps and rain streak maps for training.

1.2.5 Loss Function

Multi-Task Objection Function

$$\operatorname{argmin}_{B,S,R} \|O - B - SR\|_2^2 + P_b(B) + P_s(S) + P_r(R) \quad (5)$$

$$L(\theta) = \frac{1}{n} (\|F_{rs}(o_i; \theta) - s_i\|^2 + \lambda_1 \|F_{bg}(o_i; \theta) - g_i\|^2 - \lambda_2 (\log \hat{r}_{i,1} r_{i,1} + \log(1 - \hat{r}_{i,2})(1 - r_{i,2}))) \quad (6)$$

with

$$\hat{r}_{i,j} = \frac{\exp F_{rs}(o_i; \theta)}{\sum_{k=1}^2 \exp F_{rs,k}(o_i; \theta)}, j \in \{1, 2\} \quad (7)$$

Metrics Function: PSNR, SSIM

1.3 Semi-supervised Transfer Learning for Single Image Rain Removal

Semi-supervised Transfer Learning for Single Image Rain Removal attacks to provide two approaches, one supervised for paired no-rain and synthetic rain images, and the other unsupervised for real-world rainy images.

1.3.1 Model Formulation

Firstly, the author expresses the rain as Gaussian mixture model (GMM), because "GMM can be universal approximations to any continuous functions if the parameters are learned appropriately."

$$R \sim \sum_{k=1}^K \pi_k N(R | \mu_k, \Sigma_k)$$

The negative log likelihood function imposed on these unsupervised samples can be written as

$$L_{unsupervised}(R; \Pi, \Sigma) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k N(R_n | 0, \Sigma_k) \quad (8)$$

1.3.2 Learning Diagram

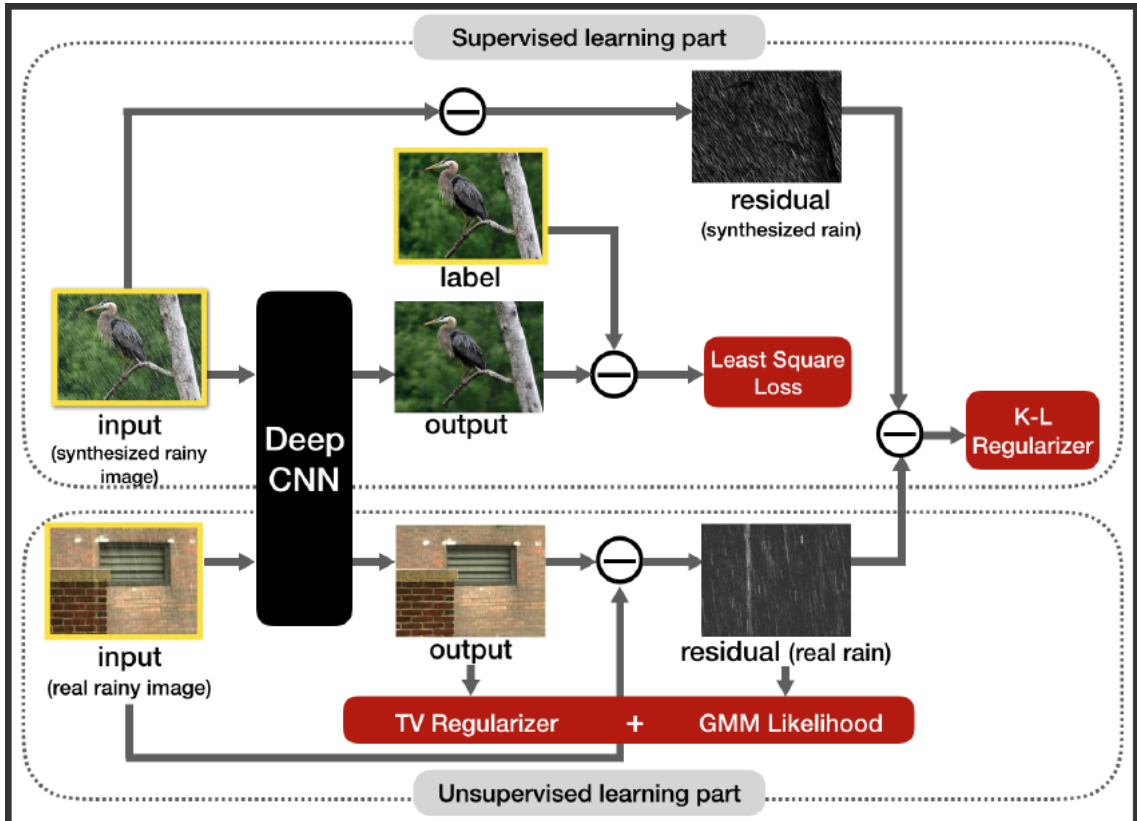


Figure 4: schematic representation of the framework

1.3.3 Loss Function

For supervised paired loss, it is simple to adapt any norm loss like l1 loss or l2 loss. Hence, it can be formulated as follows.

$$L_{supervised} = \sum_{i=1}^N ||f_w(x_i) - y_i||_F^2$$

However, for unlabelled learning, we should constrain the data distributions so that force generated domain not far from the real rain.

$$D_{KL}(G_{syn}||GMM_{real}) \simeq \min_k D_{KL}(G_{syn}||GMM_{real}^k) \quad (9)$$

Overall, the entire objective function to train the network is formulated as:

$$L(w, \Pi, \Omega) = \sum_{i=1}^{N_1} \|f_w(x_i) - y_i\|_F^2 + \alpha \sum_{n=1}^{N_2} \|f_w(\hat{x}_n)\|_{TV} + \beta D_{KL}(G_x||GMM_{\hat{x}}) - \lambda \sum_{n=1}^{N_2} \log \sum_{k=1}^K \pi_k N(\hat{x}_n - f_w(\hat{x})_n | 0, \Omega_k) \quad (10)$$

1.4 D3RNet

Dynamic Rounting Residue Recurrent Network for Video Rain Removal

1.4.1 Key Points

- Dynamic Framework for video rain removal
- Recurrent units to handle temporal fusion
- Residual unit to compensate for negative rain features

1.5 Model context

1.5.1 Occlusion-Aware Rain Model

To address the occlusion effect caused by rain streaks, the author proposed a hybrid rain model which is adaptive to mdoel rain occlusions. All pixels in rain frames are classified into two groups: 1) the ones following the additive rain model the first equation; 2) the others whose pixel values are just equal to the rain reliance.

$$O_t = (1 - \alpha)(B_t + S_t) + \alpha_t A_t \quad (11)$$

$$\alpha_t(i, j) = \begin{cases} 1 & \text{if } (i, j) \in \Omega_S \\ 0 & \text{if } (i, j) \notin \Omega_S \end{cases} \quad (12)$$

1.5.2 Rain Removal Context

"In our work, the difference of adjacent frames are used as a standard to classify motion regions. For groundtruth background frames, if the square of the difference of two adjacent frames is greater than 0.01, the region is denoted as motion regions."

1.6 Network

1.6.1 Spatial-Temporal Residue Recurrent Network

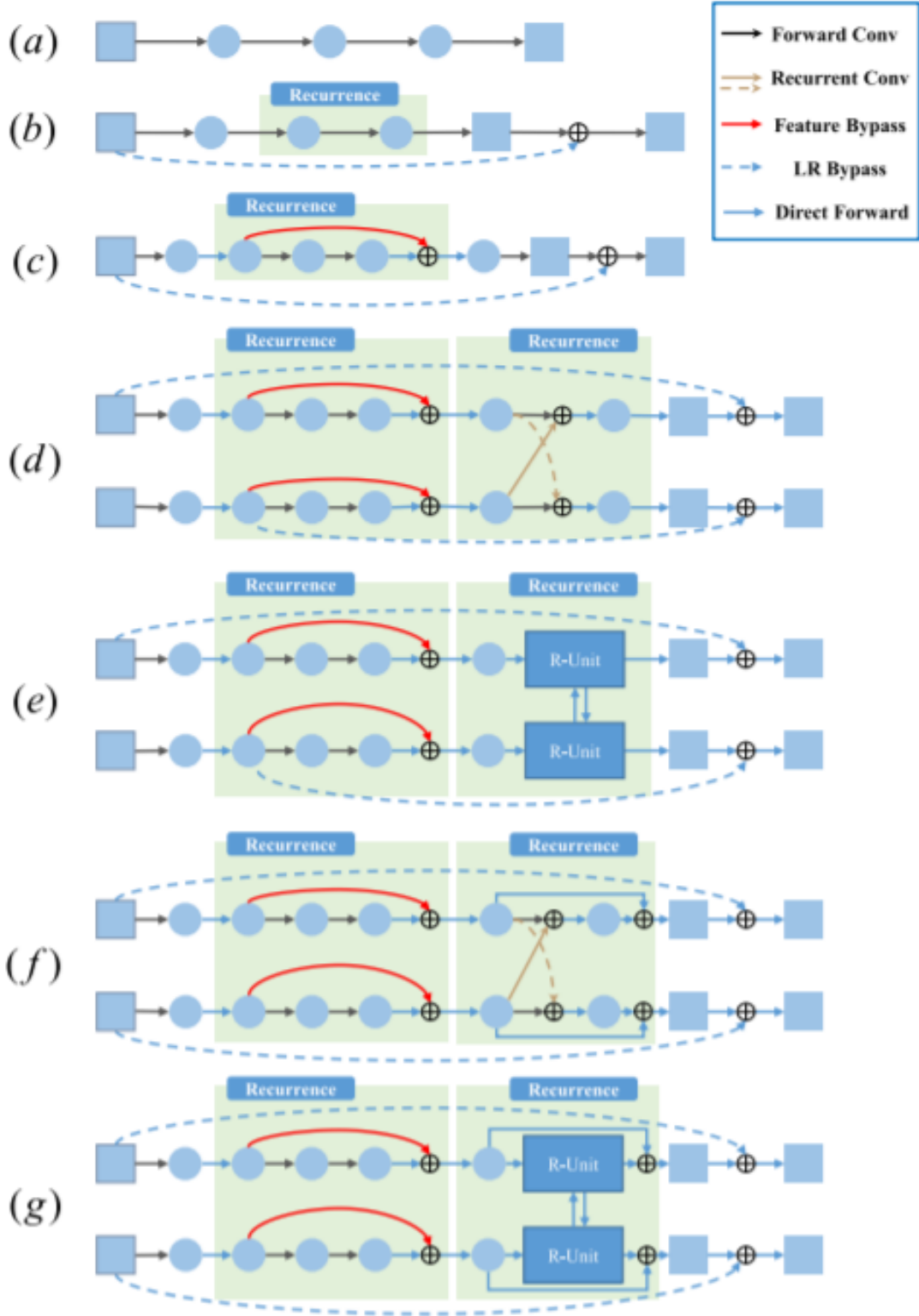


Figure 5: Network architecture evolved from vanilla convolutional neural network to our proposed spatial-temporal residue recurrent network

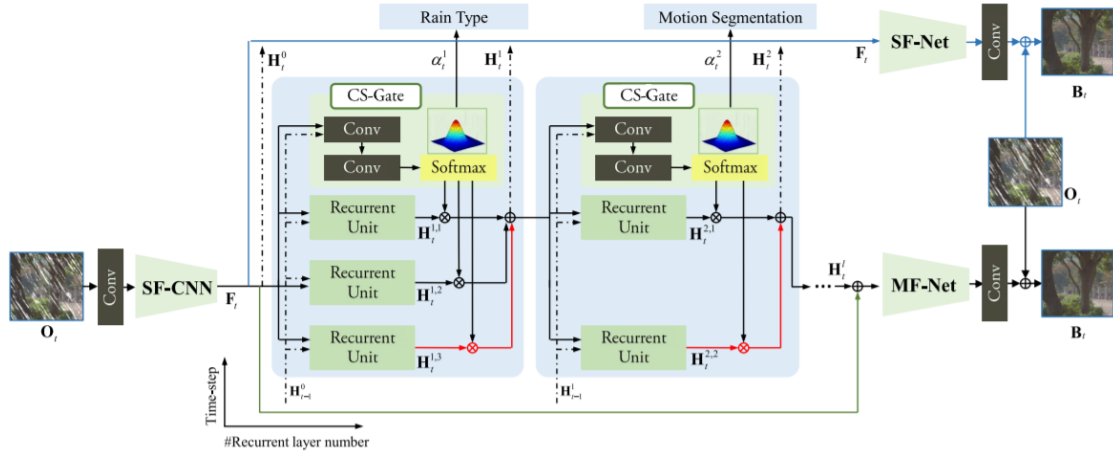


Figure 6: D3RNet framework

1.7 ReMAEN

Single Image Deraining Using a Recurrent Multi-scale Aggregation and Enhancement Network

1.7.1 Key Points

- Recurrent blocks with shared channel attention
- Detect multi-scale rain details
- Edge loss is applied to preserve the background texture

1.7.2 Network

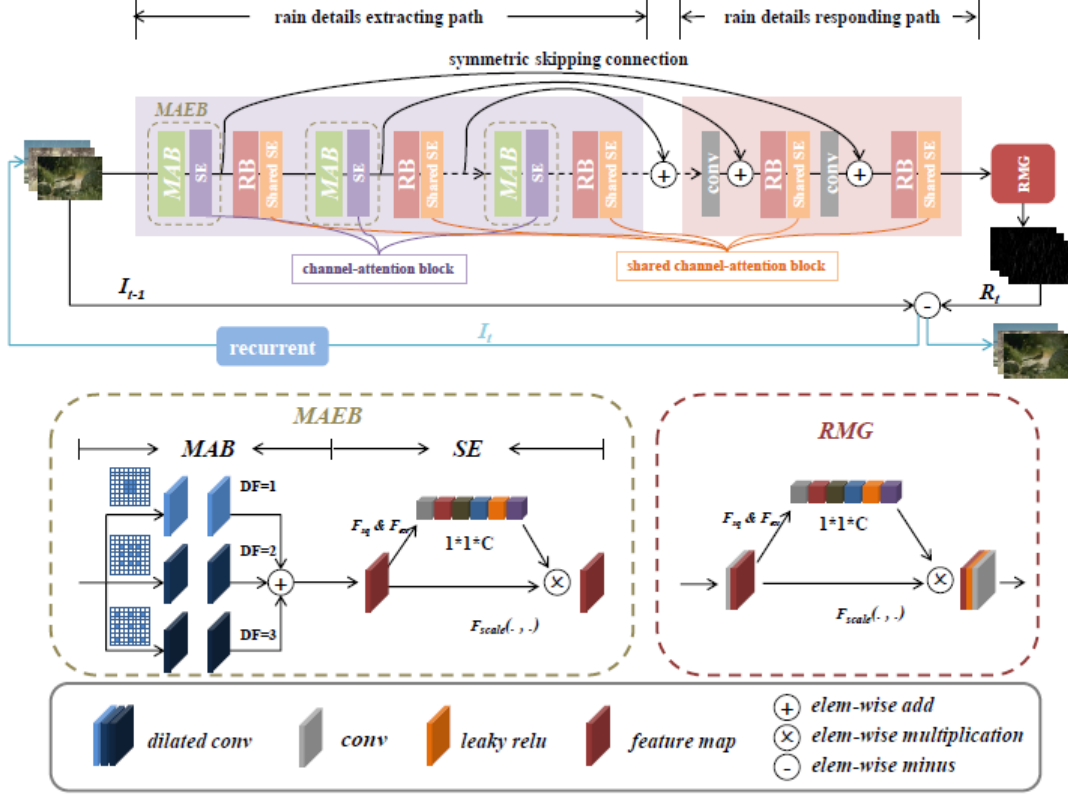


Figure 7: Schematic Network description

$$\begin{aligned} R_t &= f_{ReMAEN}(I_{t-1}) \\ I_t &= I_{t-1} - R_t \end{aligned} \quad (13)$$

1.7.3 Loss Function

- MSE Loss

No need to describe MSE loss here but the reason author applies more than MSE loss is because that MSE tends to blur out the image, and thus some details will be lost.

- Edge Loss

$$L_{edge} = \frac{1}{HWC} \sum_{x=1}^H \sum_{y=1}^W \sum_{z=1}^C \|f(I_T)^{x,y,z} - f(I_{gt})^{x,y,z}\|^2 \quad (14)$$

Where the author defines the f function as a edge detection kernel, see as follows,

$$K = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix} \quad (15)$$

1.8 PReNet

Progressive Image Deraining Networks: A Better and Simpler Baseline

2 Video based Methods

2.1 Detection of rainfall using general-purpose surveillance cameras

Is it raining outside? Detection of rainfall using general-purpose surveillance cameras

2.1.1 Key Points

- A new publicly available rain dataset, consisting of surveillance videos, rainfall data from tipping bucket and disdrometer
- 3D CNN architecture
- open-source implementation of the rain level detection algorithm of Bossu

2.1.2 Pipelines and model description

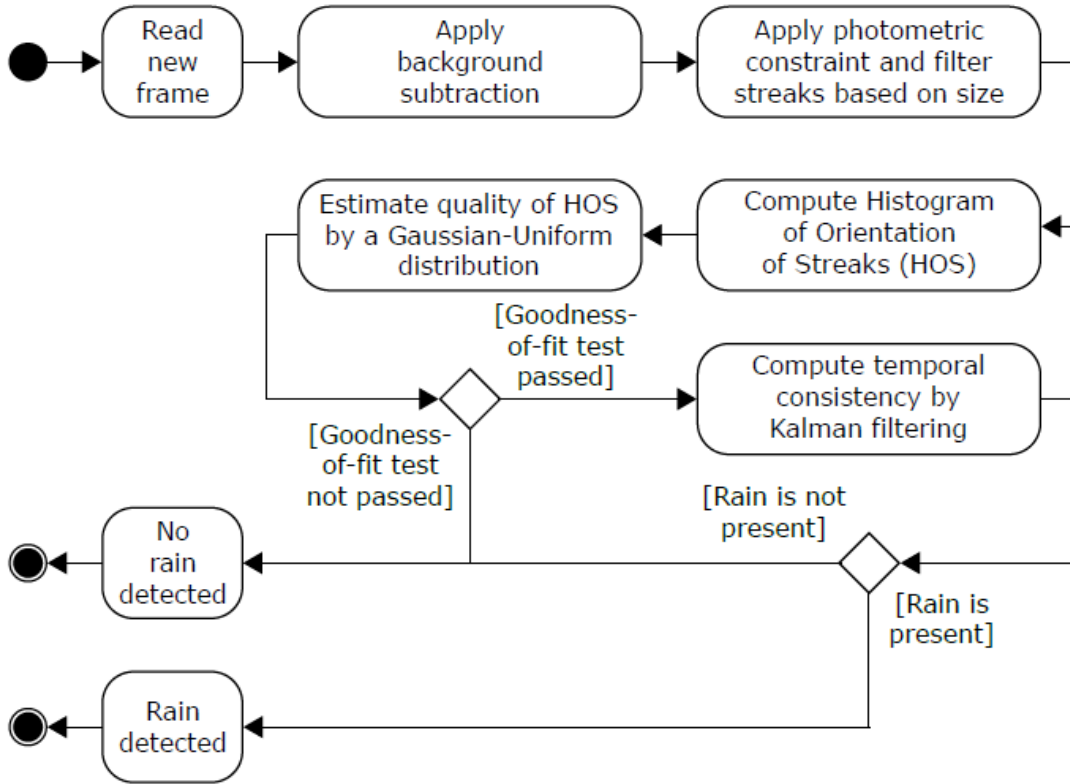


Figure 8: Activity diagram of the rain detection algorithm by Bossu et al.

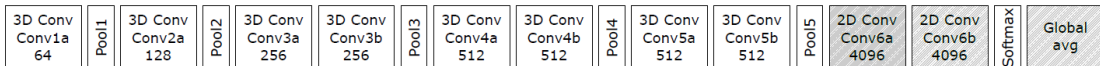


Figure 9: Overview of C3DCNN model structure

2.1.3 results

In summary, for the rainfall detection, C3DCNN with full-connected layer turns to be the most accurate method. see full results from the paper mentioned above.