

Single Image Deraining Network with Rain Embedding Consistency and Layered LSTM

Yizhou Li¹ Yusuke Monno² Masatoshi Okutomi³

Tokyo Institute of Technology, Japan

{yli¹, ymonno²}@ok.sc.e.titech.ac.jp, mxo@ctrl.titech.ac.jp³

Abstract

Single image deraining is typically addressed as residual learning to predict the rain layer from an input rainy image. For this purpose, an encoder-decoder network draws wide attention, where the encoder is required to encode a high-quality rain embedding which determines the performance of the subsequent decoding stage to reconstruct the rain layer. However, most of existing studies ignore the significance of rain embedding quality, thus leading to limited performance with over/under-deraining. In this paper, with our observation of the high rain layer reconstruction performance by an rain-to-rain autoencoder, we introduce the idea of “Rain Embedding Consistency” by regarding the encoded embedding by the autoencoder as an ideal rain embedding and aim at enhancing the deraining performance by improving the consistency between the ideal rain embedding and the rain embedding derived by the encoder of the deraining network. To achieve this, a Rain Embedding Loss is applied to directly supervise the encoding process, with a Rectified Local Contrast Normalization (RLCN) as the guide that effectively extracts the candidate rain pixels. We also propose Layered LSTM for recurrent deraining and fine-grained encoder feature refinement considering different scales. Qualitative and quantitative experiments demonstrate that our proposed method outperforms previous state-of-the-art methods particularly on a real-world dataset. Our source code is available at <http://www.ok.sc.e.titech.ac.jp/res/SIR/>.

1. Introduction

Single image deraining is the task to remove the rain from a single rainy image and has been treated as a significant process, since the images captured under rainy conditions are heavily degraded, which may disturb many high-level computer vision tasks, such as object detection [20], video surveillance [22], and autonomous driving [2]. Single image deraining is very challenging since it is a highly

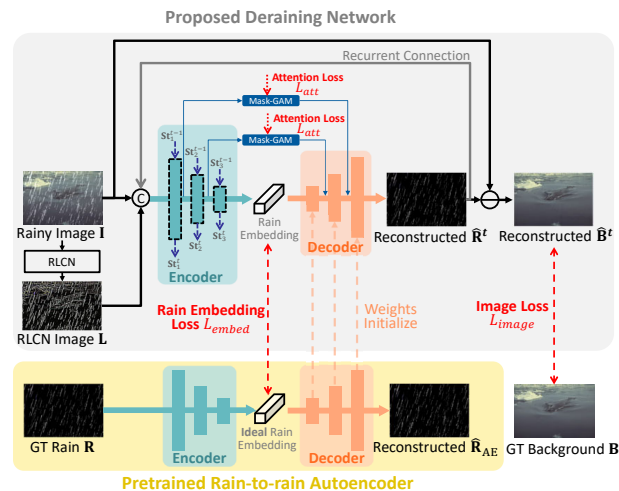


Figure 1. An overview of the proposed network. \mathbf{St} denotes cross-stage states for LSTM with hidden states \mathbf{H} and cell states \mathbf{C} .

ill-posed problem to decompose an observed rainy image into the rain layer and the rain-free background image.

Various deraining methods based on a physical or a subjective prior on rain streaks or background have been proposed [4, 19, 44]. However, these prior-based methods have limited applicability for the images captured under complex rainy situations. On the other hand, data-driven learning-based deraining methods based on a Convolutional Neural Network (CNN) have recently demonstrated their superior performance on different benchmark datasets (see [16, 17, 40] for reviews).

An encoder-decoder model is one of widely applied network architectures for single image deraining [15, 25, 31, 41], where the typical target of the encoder-decoder is to predict the residuals $\mathbf{R} = \mathbf{I} - \mathbf{B}$ between the input rainy image \mathbf{I} and the rain-free background image \mathbf{B} [9]. Since the residuals correspond to the rain layer, the predicted rain-free image $\hat{\mathbf{B}}$ is derived by subtracting the predicted residuals $\hat{\mathbf{R}}$ from the input rainy image \mathbf{I} as $\hat{\mathbf{B}} = \mathbf{I} - \hat{\mathbf{R}}$. In this residual deraining encoder-decoder, the encoder is required to encode a latent high-quality rain embedding well

representing the rain layer information, which is a base for high-performance rain layer generation by the subsequent decoding process by the decoder [1]. However, the deraining performance is often limited due to the learned low-quality rain embedding as the encoder fails to decompose the rain features with the input rainy image where the rain and the background are tightly coupled. This may cause insufficient rain removal (under-deraining) or lacked textures (over-deraining).

In this paper, we introduce the idea of ‘‘Rain Embedding Consistency.’’ To explain this, we consider a rain-to-rain autoencoder as shown in the bottom part of Fig. 1, whose target is to reconstruct a rain layer $\hat{\mathbf{R}}_{AE}$ to be as close as its input Ground-truth (GT) rain layer \mathbf{R} . Our observation is that the autoencoder derives a superior reconstruction performance with Peak Signal-to-noise Ratio (PSNR) larger than 50 by comparing the GT background image \mathbf{B} and predicted background image $\hat{\mathbf{B}}_{AE} = \mathbf{I} - \hat{\mathbf{R}}_{AE}$, implying that the autoencoder succeeds to encode an ideal rain embedding sufficiently representing the rain layer. This motivates us to improve the consistency between the ideal rain embedding by the autoencoder and the rain embedding predicted by the encoder of the deraining network.

To incorporate the Rain Embedding Consistency, we first propose to simply take a L1 loss function between the two embeddings, where we name it as Rain Embedding Loss as shown in Fig. 1, to supervise and constraint the encoding process of the deraining network. We also propose to apply a Rectified Local Contrast Normalization (RLCN) as an encoder input (as shown in Fig. 1) to guide the encoding process, which extracts the pixels having a higher pixel intensity than the average pixel intensity of its neighborhood pixels. We find that, since rain streaks usually show a higher pixel intensity, the extracted pixels by RLCN cover almost all the rain candidates including weak rain streaks. Thus, RLCN can be used as a rain location prior to guide the encoder to learn better convolutional filters focusing on extracting rain features. We also propose a Mask Guided Attention Module (Mask-GAM) to replace general skip connection from encoder to decoder [28], which further augments the decoding phase by intensifying rain-related part of encoder features with an accurate learned rain location map before feeding to decoder.

With the above proposed components, we construct our proposed deraining network with Rain Embedding Consistency, as illustrated in Fig. 1. To further improve the performance, we also propose a Recurrent Neural Network (RNN) framework with Long Short-term Memory (LSTM), which have shown great potential in single image deraining task [18, 26, 27]. Different from previous methods that apply LSTM on a single scale [26, 27] of the network, we propose Layered LSTM that applies LSTM to every scale of the encoder, and let the encoded features at

each scale flow to the subsequent stages as shown in Fig. 1, where the later stages can selectively compensate the features that effectively represent the rain layer, and forget the non-related features not representing the rain layer. With this stage-by-stage and scale-by-scale fine-grained rain feature refinement using Layered LSTM, our proposed network can approach the Rain Embedding Consistency better.

Through experiments, we demonstrate that our proposed deraining network with Rain Embedding Consistency and Layered LSTM outperform state-of-the-art deraining methods by a large margin for a real-world benchmark dataset, while keeping computational efficiency. We also show the experiments on several synthetic datasets, further showing the superiority of the proposed methods, which has good resistance to over/under-deraining issues with its robust rain embedding encoding.

2. Related Work

We here briefly review existing single image deraining methods. Please refer to the survey papers [16, 17, 40] for other classes of the methods, such as video-based methods and raindrop removal methods.

2.1. Prior-based Methods

Many traditional approaches treat the single image deraining as an image decomposition problem, where a rainy image is commonly modeled as an addition of a rain layer and a rain-free background layer [4, 13, 19, 35, 44]. To regularize the decomposition problem, various priors have been proposed to model rain streaks with different directions and scales, such as Gaussian mixture model-based prior [19], sparsity-based prior [4, 13, 21, 44], and high-frequency component-based prior [13, 35]. A more complex and realistic image formation model, such as a screen-blend model [21], has also been studied.

Although these traditional methods based on a physical or subjective prior demonstrate favorable performance on specific situations, they fail to remove complex rain patterns of real-world scenes where the prior does not hold.

2.2. Learning-based Methods

Because of the strong representation capability of CNN, CNN-based deraining methods have been extensively studied in recent years. Residual learning is commonly applied, where the network is trained to predict the residuals between the rainy and the rain-free background images, which correspond to the rain layer [9]. Based on the residual learning, different kinds of network architecture, such as multi-scale fusion modules [5, 12, 29, 30, 38, 39, 42], a density-label-guide network [42], a residual-guide network [8], a confidence-measure-guided network [41], a non-locally enhanced network [15], have been proposed to boost the capacity of the network. Generative adversarial networks [43]

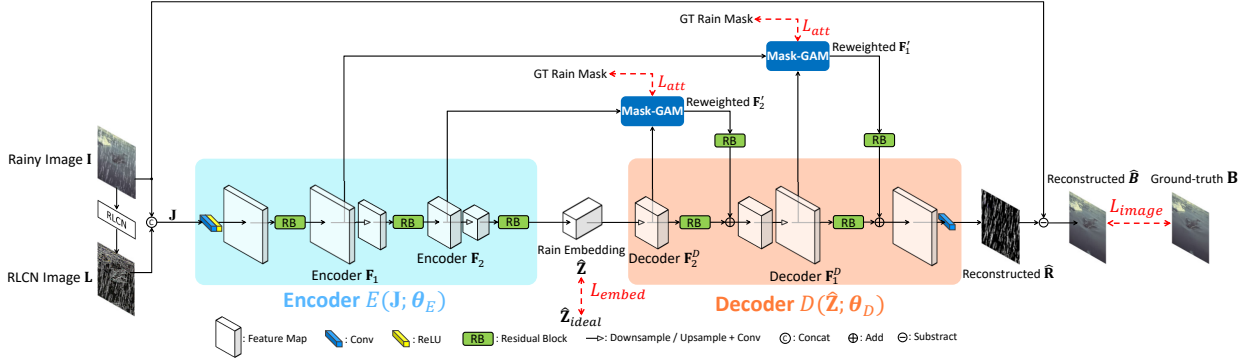


Figure 2. The overall architecture of proposed ECNet. It includes four scales where the number of channels for each scale is 32, 64, 128, and 256, respectively. For simplicity of illustration, we here show the three scales version of the network architecture.

and variational auto-encoders [6, 7] have also been introduced to capture the visual characteristics of the rain streaks. These methods have achieved convincing performance with their carefully designed network architectures. However, the complex deraining process is supervised with only the image-level loss function in terms of deraining output, which is difficult for the model to learn optimal latent rain features and thus limiting the deraining performance.

The methods based on an RNN is becoming a new trend to progressively remove the rain by dividing the whole deraining process into multiple stages, where LSTM [37] or GRU [3] is applied to selectively forget noise features and refine task-related features stage-by-stage. The study [18] proposes to apply an RNN with GRU [3] and channel-wise attention to predict the rain layer and share the information across recurrent stages in feature levels. The study [27] applies recursive residual blocks with LSTM [37] to build a simple and fast baseline for the deraining. The study [26] proposes the bilateral LSTM for the information sharing between the rain layer and the background layer. However, these methods apply the LSTM or GRU to a single scale of the whole deraining pipeline, without considering the potential of addressing the feature refinement on different scales separately, thus showing limited refinement performance.

3. Proposed Method

3.1. Overall Architecture

As shown in Fig. 1, we use a rain-to-rain autoencoder for rain embedding supervision and our main deraining network, for both of which we adopt the same encoder-decoder architecture as shown in the encoder-decoder part of Fig. 2, where Residual Blocks (RBs) [11] are used as feature extractors for both the encoder and the decoder.

To train the autoencoder, the ground-truth rain layer is generated as $\mathbf{R} = \mathbf{I} - \mathbf{B}$, where \mathbf{I} and \mathbf{B} are a pair of rainy image \mathbf{I} and ground-truth background image \mathbf{B} sam-

pled from training data. The autoencoder takes the ground-truth rain layer as the input and encodes ideal rain embedding as $\hat{\mathbf{Z}}_{ideal} = E_{AE}(\mathbf{R}; \theta_E^{AE})$, where E_{AE} represents the encoder of the autoencoder with input \mathbf{R} and parameters θ_E^{AE} . The decoder then reconstructs the rain layer from the rain embedding as $\hat{\mathbf{R}}_{AE} = D_{AE}(\hat{\mathbf{Z}}_{ideal}; \theta_D^{AE})$, where D_{AE} represents the decoder of the autoencoder with input $\hat{\mathbf{Z}}_{ideal}$ and parameters θ_D^{AE} . As the purpose of the autoencoder is to make reconstructed rain layer $\hat{\mathbf{R}}_{AE}$ to be as close as its ground-truth input \mathbf{R} , we simply pre-train the autoencoder using a self-supervision loss $L_{self} = \|\mathbf{R} - \hat{\mathbf{R}}_{AE}\|_2$.

Fig. 2 illustrates the overall architecture of our proposed deraining network with Rain Embedding Consistency, which we call Embedding Consistency Network (ECNet). Our target is to learn the encoder-decoder network to make the predicted rain layer $\hat{\mathbf{R}}$ that is as close as ground-truth rain layer \mathbf{R} , such that an accurate rain-free background prediction $\hat{\mathbf{B}} = \mathbf{I} - \hat{\mathbf{R}}$ can be derived.

The architecture of our network is explained as follows. We first calculate the RLCN image \mathbf{L} from the input rainy image \mathbf{I} to extract the candidate pixels suggesting rain regions. The calculation of the RLCN image will be detailed in Sec. 3.2. Then, the channel-wise concatenation of \mathbf{I} and \mathbf{L} is used as the network input, such that $\mathbf{J} = \text{concat}(\mathbf{I}, \mathbf{L})$. The network encodes and derives a predicted rain embedding by $\hat{\mathbf{Z}} = E(\mathbf{J}; \theta_E)$, where E represents the encoder with input \mathbf{J} and parameters θ_E . Then, the rain layer is reconstructed from the rain embedding $\hat{\mathbf{Z}}$ by the decoder D with parameters θ_D as $\hat{\mathbf{R}} = D(\hat{\mathbf{Z}}; \theta_D)$. Instead of applying the skip connection directly passing the encoder feature \mathbf{F} to the decoder, we propose Mask-GAM to generate an accurate rain attention map $\hat{\mathbf{M}}$ to derive rain-attentive features $\mathbf{F}' = \hat{\mathbf{M}} \circ \mathbf{F}$, which is passed to the decoder. The details of Mask-GAM will be introduced in Sec. 3.3.

The loss function to training the network will be the weighted summation of three terms as

$$L = \lambda_{embed} L_{embed} + \lambda_{att} L_{att} + \lambda_{image} L_{image}, \quad (1)$$

where λ_{embed} , λ_{att} and λ_{image} are coefficients to balance each loss term. The first loss L_{embed} is our proposed Rain Embedding Loss that uses the encoded ideal rain embedding $\hat{\mathbf{Z}}_{ideal} = E_{AE}(\mathbf{R})$ by the autoencoder to train the encoder with Rain Embedding Consistency. The second loss L_{att} is Attention Loss to supervise the learning of the attention map in our proposed Mask-GAM. The third loss is an Image Loss, for which we use negative SSIM loss [26], which can be represented by $L_{image} = -\text{SSIM}(\mathbf{B}, \hat{\mathbf{B}})$ to evaluate the difference between the ground-truth and reconstructed background images.

We detail each proposed component in Sec 3.2 and Sec 3.3 which is utilized to construct the proposed ECNet. Furthermore, we introduce our proposed Layered LSTM to further improve the performance in Sec 3.4, where we call the resultant network ECNet+LL.

3.2. Rain Embedding Consistency with RLCN

As we introduced in Section 1, the observation of high rain layer reconstruction performance by the rain-to-rain autoencoder implies that the autoencoder can derive an ideal rain embedding $\hat{\mathbf{Z}}_{ideal} = E_{AE}(\mathbf{R})$ by its encoder, where its decoder is able to decode and derive the rain layer $\hat{\mathbf{R}}_{AE}$ which is very close to its input $\hat{\mathbf{R}}$. This motivates us to reuse the pretrained weights θ_D^{AE} from D_{AE} to initialize the decoder D 's weights θ_D , which enables us to focus on training the encoder E to improve the Rain Embedding Consistency between the autoencoder's ideal rain embedding $\hat{\mathbf{Z}}_{ideal}$ and predicted rain embedding by the encoder of the proposed deraining network $\hat{\mathbf{Z}} = E(\mathbf{J})$. To approach the Rain Embedding Consistency, we propose to use a feature-level Rain Embedding Loss expressed as

$$L_{embed} = \|\hat{\mathbf{Z}}_{ideal} - \hat{\mathbf{Z}}\|_1, \quad (2)$$

to directly supervise the encoder in terms of the feature space and force the encoder's output embedding $\hat{\mathbf{Z}}$ to approach the ideal rain embedding $\hat{\mathbf{Z}}_{ideal}$.

To further regularize the rain embedding learning, we proposed to apply RLCN to the input rainy image as

$$\mathbf{L}(i, j, c) = \frac{\phi_{rec}(\mathbf{I}_c(i, j) - \mu_{\mathbf{I}_c}(i, j))}{\sigma_{\mathbf{I}_c}(i, j) + \epsilon}, \quad (3)$$

where \mathbf{L} is the output RLCN image, (i, j) denotes the pixel coordinate, $c \in (R, G, B)$ represents the color channel, \mathbf{I}_c is c -th channel of the input rainy image, $\mu_{\mathbf{I}_c}(i, j)$ denotes the mean pixel intensity within a local square window centered at the pixel (i, j) , $\sigma_{\mathbf{I}_c}(i, j)$ denotes the standard deviation of the pixel intensities within the local window, ϵ is a small value for numerical stability, and $\phi_{rec}(\cdot) = \max(\cdot, 0)$ is the rectification function that outputs the max value compared with zero.

In Eq. (3), the subtracted value, $\mathbf{I}_c(i, j) - \mu_{\mathbf{I}_c}(i, j)$, takes a high absolute value if the pixel intensity of the pixel (i, j) is

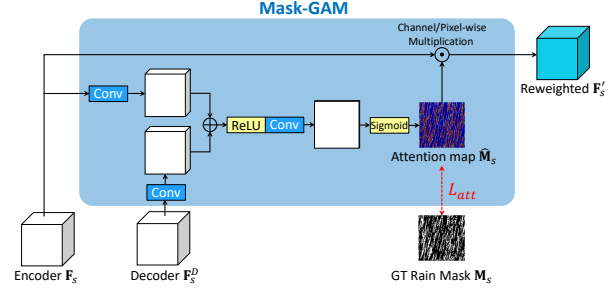


Figure 3. Architecture of proposed Mask-GAM.

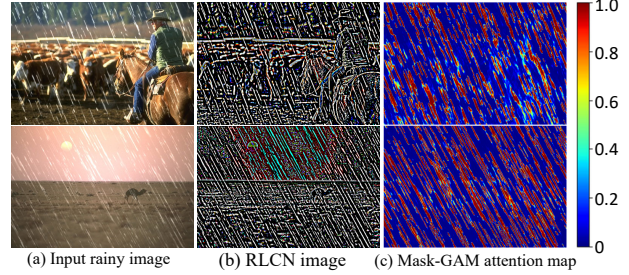


Figure 4. Two examples of (a) input rainy images, (b) our proposed RLCN images, (c) our proposed Mask-GAM attention maps.

significantly higher or lower than the intensities of the other pixels within the local window. Because we focus on the rain, which usually shows a higher pixel intensity, the max function is applied to perform the rectification and filter out negative values. The positive value after the rectification is further normalized by the local standard deviation $\sigma_{\mathbf{I}_c}(i, j)$, which corresponds to local contrast, to enable the extraction of the pixels for weak rain regions as well as strong rain regions. As shown in Fig. 4, the derived RLCN image can cover most of the rain regions and thus can be used as a guide for the rain embedding encoding, which helps the encoder to learn rain-focus convolutional filters, thus improving the Rain Embedding Consistency performance by the encoder.

3.3. Mask Guided Attention Module (Mask-GAM)

Skip connection is widely applied in the encoder-decoder to reuse the encoder features to prevent the over-smoothing of the decoding. However, as RLCN image contains some additional edges regarding the non-rain regions as shown in Fig. 4, there exist unexpected features non-related to rain included in the encoder features. Therefore, passing the features directly from the encoder to the decoder could disturb the decoding phase. To address this issue, Mask-GAM of Fig. 3 is proposed to generate a pixel-level attention map to more accurately regard the rain location as shown in Fig. 4. The attention map is used to reweight the encoder features to remove the unexpected features and highlight the rain-related features.

Following the mathematical symbol notations in Fig. 3,

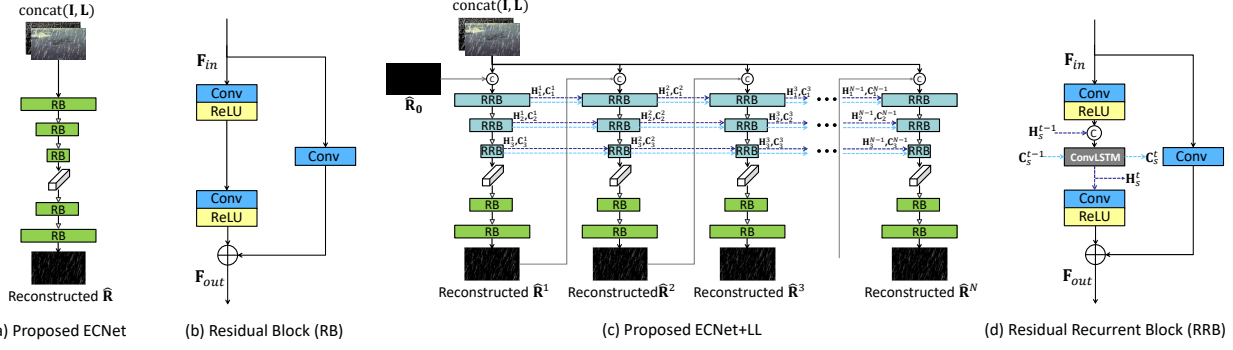


Figure 5. Our proposed recurrent framework. For simplicity the Mask-GAM is ignored in this illustration. Based on proposed ECNet (a) which uses RBs (b), we construct ECNet+LL (c) by introducing the recurrent framework with Layered LSTM that replaces RBs in the encoder with the proposed RRBs (d).

Mask-GAM is expressed as

$$\hat{\mathbf{M}}_s = \phi_2(\mathbf{W}_s^A * (\phi_1(\mathbf{W}_s * \mathbf{F}_s + \mathbf{W}_s^D * \mathbf{F}_s^D))), \quad (4)$$

where $\hat{\mathbf{M}}_s$ is the attention map for scale s of the encoder-decoder and \mathbf{F}_s and \mathbf{F}_s^D represent the features from the encoder and the decoder, respectively. \mathbf{W}_s and \mathbf{W}_s^D denote the weights for 1×1 convolutional layers with n filters, where we set n as the half of the channel number of \mathbf{F}_s , which is equal to the channel number of \mathbf{F}_s^D . The convolution operation, which is expressed as $*$, is performed to \mathbf{F}_s and \mathbf{F}_s^D , respectively, and the resultant features are combined by the element-wise summation. Then, the Rectified Linear Unit (ReLU) [10] function ϕ_1 is applied to the combined features, which followed by a one-channel 1×1 convolutional layer with the weight \mathbf{W}_s^A to sum up the n -channel feature maps. The sigmoid activation function σ_2 is then applied to derive the attention map within the range of $[0, 1]$. Finally, using the obtained attention map, the re-weighted features \mathbf{F}'_s is derived by $\mathbf{F}'_s = \hat{\mathbf{M}}_s \circ \mathbf{F}_s$ as rain-attentive features and then sent to the decoder, where \circ denotes the channel and pixel-wise multiplication. Then, to supervise the learning of the attention map, we use the ground-truth rain mask \mathbf{M} derived by thresholding the ground-truth rain layer \mathbf{R} to construct a loss function as

$$L_{att} = \frac{\sum_{s=1}^S \|\hat{\mathbf{M}}_s - \mathbf{M}\|_2}{S}, \quad (5)$$

where S is the scale number of the encoder-decoder and s is the scale index. This attention module not only excludes the non-related features from the encoded features, but also can be seen as a tiny additional branch of the proposed encoder-decoder network whose output target is the rain regions, thus forcing the network to derive features that can represent the information inside rain regions.

3.4. Recurrent Framework with Layered LSTM

In this section, we introduce the proposed ECNet+LL network which is built based on the proposed ECNet.

Firstly, we reformulate the ECNet network of Fig. 5(a) into a recurrent style as shown in Fig. 5(c), where the output of the decoder corresponds to the rain layer $\hat{\mathbf{R}}^t$ of current stage t . Then, the current output $\hat{\mathbf{R}}^t$ is used as the input of the next stage $t+1$ and the overall network flow is repeated until the iteration reaches the maximum recurrent time N . Here we set $\hat{\mathbf{R}}_0 = \mathbf{0}$. For every recurrent stage, the network parameters are shared and optimized during the training phase.

Then, we introduce a Residual Recurrent Block (RRB) with LSTM to each scale of the encoder as shown in Fig. 5(c). As they let the features flow across successive recurrent stages with a scale-by-scale style, we name this architecture as Layered LSTM (LL). As shown in Fig. 5(d), a convolutional LSTM layer is added inside RB of Fig. 5(b) to construct the RRB. As the information flow of LSTM, the hidden and cell state output of stage $t-1$, which are denoted as \mathbf{H}_s^{t-1} and \mathbf{C}_s^{t-1} respectively with s describing the scale index, and the features from the last ReLU layer are sent to the LSTM layer, where the gating signal inside LSTM decides which part of the information is preserved or thrown away. Then, the LSTM outputs the hidden and cell state of current stage t , which is denoted as \mathbf{H}_s^t and \mathbf{C}_s^t , where \mathbf{H}_s^t is used as the input of the second convolutional layer and both of \mathbf{H}_s^t and \mathbf{C}_s^t are sent to the RRB of the next stage $t+1$. By applying the recurrent framework with Layered LSTM, the network is able to conduct a scale-by-scale fine-grained feature refinement for encoding, where the subsequent stages of encoding are able to retain and refine the rain-related features in terms of each scale separately, and forget the features non-related to rain, thus boosting the network to improve Rain Embedding Consistency.

As for the loss function after introducing the recurrent framework, we calculate the loss L for each stage t separately, which is denoted as L_t , and recursive supervision is introduced by adding them together as the final loss function. This can be represented as

$$L_{recur} = \sum_{t=1}^N \lambda_t L_t \quad (6)$$

where λ_t is used to balance the loss of each stage t .

4. Experimental Results

4.1. Implementation Details

Our ECNet and ECNet+LL networks were implemented using Pytorch [24] and trained using one NVIDIA Tesla P100 GPU. For all the experiments, the networks share the same training settings. The size of the local window for the RLCN image calculation is set to 9×9 and the recurrent time N of the proposed network is set to 6. We set $\lambda_{embed} = 0.02$, $\lambda_{att} = 0.1$, $\lambda_{image} = 1$ to balance the influence of each loss, and the λ_t is set to 0.5 for $t \leq 5$, and 1.5 for final stage $t = 6$. The patch size of 96×96 is used for the training, with the batch size set to 16. ReLU [10] is used as the activation function and the gradient clipping of 5 is applied to stabilize the network training. Adam optimizer [14] is used to train the network with an initial learning rate $1e^{-3}$. We trained the network for 100 epochs while decaying the learning rate when reaching 25, 50, 75 epochs by multiplying 0.2.

4.2. Study of Proposed Components

The studies to confirm the effectiveness of each proposed component was conducted using real-world SPA-Data dataset [33]. SPA-Data contains high-quality 638,492 training images and 1,000 test images generated using video redundancy and human supervision. For time efficiency, we randomly selected 6,385 images (1%) from the original training data to train the networks and tested on the full 1,000 test images.

Effects of Rain Embedding Consistency with RLCN: To explore the effectiveness of the proposed method with Rain Embedding Consistency with the guide of the RLCN image, the following four networks are evaluated in Table 1. ED is the basic encoder-decoder that applies only the encoder-decoder part of the proposed network and uses only a rainy image as the network input, and it is trained using only Image Loss L_{image} . ED + EmbedLoss means Rain Embedding Consistency is considered by training the network with Rain Embedding Loss L_{embed} . ED + RLCN means that the RLCN image is used as the input for the encoder. ED + EmbedLoss + RLCN means that both the Rain Embedding Loss and the RLCN guide are applied. Table 1 shows that the training with L_{embed} actually benefit the network performance and also the RLCN guide remarkably improves the network performance. When combining L_{embed} and the RLCN guide, the network achieves the best results. Hereafter, we use ED + EmbedLoss + RLCN as the base of the proposed method (Proposed-Base) for the next experiment.

Effects of Mask-GAM: To explore the effectiveness of the proposed Mask-GAM, Proposed-Base and the following three networks are evaluated in Table 2. Proposed-

Table 1. Effects of Rain Embedding Consistency with RLCN.

Methods	PSNR	SSIM
ED	40.01	0.978
ED + EmbedLoss	40.38	0.979
ED + RLCN	40.73	0.980
ED + EmbedLoss + RLCN (Proposed-Base)	40.85	0.980

Table 2. Effects of Mask Guided Attention Module.

Methods	PSNR	SSIM
Proposed-Base	40.85	0.980
Proposed-Base + Skip	41.35	0.986
Proposed-Base + Self-attention	41.58	0.987
Proposed-Base + Mask-GAM (ECNet)	41.83	0.987

Table 3. Effects of Recurrent Framework with Layered LSTM.

Methods	PSNR	SSIM
ECNet	41.83	0.987
ECNet + LSTM	41.90	0.987
ECNet + LayeredLSTM (ECNet+LL)	42.49	0.988

Base + Skip applies the direct skip connection between the encoder and the decoder. Proposed-Base + Self-attention means that self-attention [23] is utilized to reweight the encoder features before feeding to the decoder. Proposed-Base + Mask-GAM means that our proposed Mask-GAM with the Attention Loss L_{att} is applied to supervise the attention learning and this method corresponds to the proposed ECNet. Table 2 shows that the direct skip connection provides limited improvements as it may introduce the features non-related to rain from the encoder to the decoder. With self-attention, this phenomenon is relieved to some extent. And with the proposed Mask-GAM, the network achieves the best performance as it explicitly guides the learning of the attention and leaves rain-related features only.

Effects of Recurrent Framework with Layered LSTM: To explore the effectiveness of the proposed Layered LSTM, proposed ECNet and the following two networks are evaluated in Table 3. ECNet + LSTM denotes that the network is first formulated to a recurrent style, and only the RB of the last scale of the encoder is changed to RRB, as performed in [34]. ECNet + LayeredLSTM further replaces the RBs in every scale of the encoder with RRBs, denoting the proposed Layered LSTM. This method corresponds to the full version of our method, i.e., ECNet+LL. As we can see in Table 3, only applying LSTM to the last scale of the encoder almost does not benefit the network performance, while the proposed Layered LSTM shows substantial performance boost as LSTM is applied to every scale of the encoder to conduct fine-grained rain feature refinement scale-by-scale.

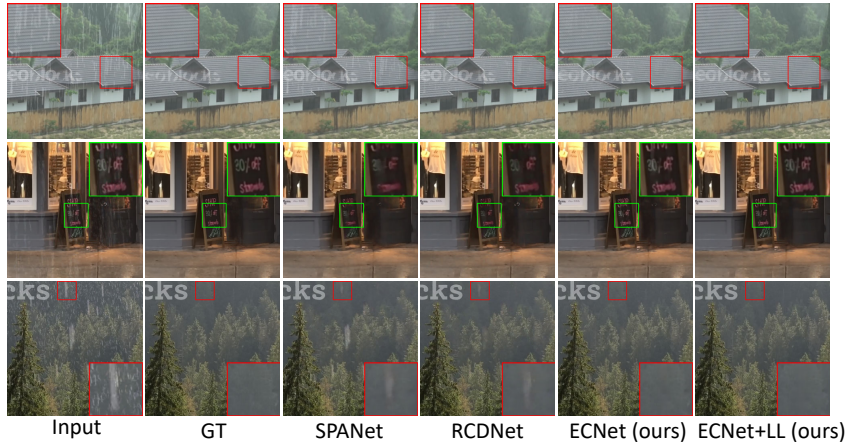


Figure 6. Qualitative comparison on real-world SPA-Data

Table 4. Quantitative comparison on real-world SPA-Data (Red: rank 1st; Blue: rank 2nd) with different amount of training data from SPA-Data.

Methods	SPANet [33]		RCDNet [32]		ECNet (ours)		ECNet+LL (ours)	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
6385 (1%)	36.37	0.971	39.75	0.982	41.83	0.987	42.49	0.988
Full	40.04	0.984	41.05	0.985	43.62	0.990	44.32	0.991

4.3. Comparison on Real-world Dataset

Compared methods: We evaluate the deraining performance of proposed ECNet and ECNet+LL for real-world images of SPA-Data [33]. We trained the proposed networks twice using the full training set (638,492 images) and 1% of the training set (6,385 images), to show the learning performance with a small amount of data, respectively. Since the full dataset is very large and retraining takes a huge amount of time, we chose two previous state-of-the-art methods, SPANet [33] and RCDNet [32], for comparison, which provide a public pretrained model on the full training set. SPANet is the method that proposes SPA-Data and RCDNet is a recent state-of-the-art method. For the 1% training set, we retrained the models for SPANet and RCDNet as no pretrained model is provided for the 1% set and retraining can be done with reasonable training time.

Results: Table 4 shows the PSNR and the SSIM results. We can see that proposed ECNet and ECNet+LL provide remarkably better results for real-world rainy images, showing the strong applicability to real-world situations. Also, both of proposed methods derives better performance even when they are trained on only 1% of the training data, compared with SPANet and RCDNet trained on the full training data, showing that the proposed methods considering Rain Embedding Consistency can achieve high performance with only a small amount of training data. Fig. 6 shows the qualitative results of each method trained on the full training set. For samples on the first and the third rows, both of

proposed ECNet and ECNet+LL can remove the rain more completely compared with the other methods. For samples on the second row, our methods remove the rain effectively and succeed to maintain the background textures of a black board, while SPANet and RCDNet fails to reproduce them. These results validate that proposed ECNet and ECNet+LL are able to better deal with varied rain conditions containing different intensities and shapes that generally appear in real-world scenarios, and can address under/over-deraining issues more robustly. More deraining samples are shown in supplementary material.

4.4. Comparison on Synthetic Datasets

Compared methods: We evaluate proposed ECNet and ECNet+LL on five synthetic benchmark datasets, Rain100H [39], Rain200H [39], Rain100L [39], Rain200L [39], Rain800 [43]. Rain100H/200H synthesize the heavy rain conditions, Rain100L/200L synthesize the light rain conditions, and Rain800 synthesizes both heavy and light rainy images. We compare the proposed networks with state-of-the-art methods including semi-supervised SIRR [36], rain-mask-based JORDER-E [38], model-driven RCDNet [32], and RNN-based RESCAN [18], PRNet [27], and BRN [26]. RCDNet and BRN are recent state-of-the-art methods. We used the pretrained models of the above methods for Rain100H, Rain200H and Rain100L, and retrain models for Rain200L and Rain800 under the original settings of each method as no pretrained model is provided except BRN for Rain200L.

Results: Table 5 reports the PSNR and the SSIM results, where proposed ECNet+LL provides the best average performance compared with the other methods, showing its strong adaptability under different rain conditions. Proposed ECNet also derives fairly good performance considering its light-weightness as will be discussed in Section 4.5. Fig. 7 shows two qualitative samples on the

Table 5. Quantitative comparison on synthetic datasets (Red: rank 1st; Blue: rank 2nd)

Datasets	Rain100H		Rain200H		Rain100L		Rain200L		Rain800		Average	
Metrics	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
RESCAN [18]	28.82	0.867	27.95	0.862	38.09	0.980	38.43	0.982	28.36	0.872	32.33	0.913
SIRR [36]	22.03	0.714	22.17	0.726	32.31	0.926	32.21	0.931	22.73	0.762	26.29	0.812
PReNet [27]	30.31	0.910	29.47	0.907	37.21	0.978	37.93	0.983	26.82	0.888	32.35	0.933
JORDER-E [38]	30.22	0.898	29.23	0.894	39.36	0.985	39.13	0.985	27.92	0.883	33.17	0.929
RCDNet [32]	31.26	0.912	30.18	0.909	39.76	0.986	39.49	0.986	28.66	0.893	33.87	0.937
BRN [26]	31.32	0.924	30.27	0.919	38.16	0.982	38.86	0.985	28.31	0.896	33.38	0.941
ECNet (ours)	29.80	0.903	28.54	0.893	38.21	0.981	38.37	0.983	28.80	0.901	32.74	0.932
ECNet+LL (ours)	31.43	0.921	30.22	0.912	39.66	0.986	39.72	0.987	29.26	0.905	34.06	0.942

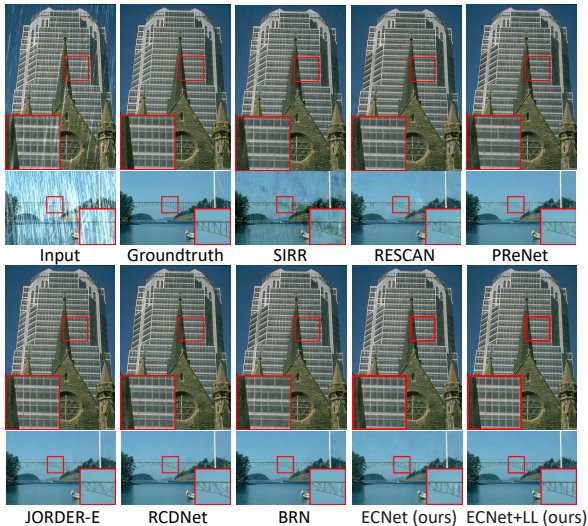


Figure 7. Qualitative comparison on synthetic datasets

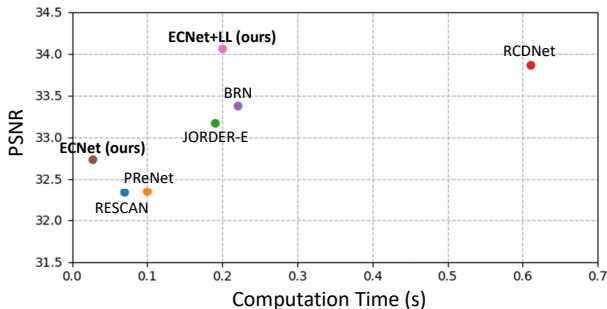


Figure 8. Computational time vs. PSNR plot for average PSNR on all the five synthetic datasets

Rain100L dataset and the Rain100H dataset. For the samples on the first row, we can see that both of ECNet and ECNet+LL can maintain the window frames of the building that have similar appearance to the rain streaks, while other methods show apparent over-deraining as they also remove background textures. Similarly, for the heavy rain samples on the second row, both of our methods reproduce the bridge textures more effectively. These results show the strength of our methods to deal with under/over-deraining issues.

4.5. Computational Time

Figure 8 shows the computational time versus PSNR plot for average PSNR on all the five synthetic datasets (except SIRR because of its lower PSNR). The running time is recorded with one NVIDIA P100 GPU for the image whose resolution is 480×320 . As shown in Fig. 8, proposed ECNet+LL achieves the best PSNR result with almost the same computation time as BRN and JORDER-E and much less computational time than RCDNet, showing the better performance-speed balance compared with other competing methods. Also, proposed ECNet shows the best performance among the methods with computation time that is no more than 0.1s. Proposed ECNet can achieve a real-time inference with 37FPS (0.027s per inference) on a NVIDIA P100 GPU, which could be meaningful to real-world applications where computational time is an important factor.

5. Conclusion

In this paper, we have proposed a single image deraining network considering Rain Embedding Consistency with an ideal rain embedding encoded by the rain-to-rain auto-encoder to address the difficulty to learn a better rain embedding. First, we have introduced the idea of Rain Embedding Consistency and train the network with a Rain Embedding Loss to incorporate the Rain Embedding Consistency, with an RLCN image as the guide for the encoder to learn better rain-focused convolutional filters. Next, we have proposed Mask-GAM with ground-truth rain mask supervision to reweight the encoder features to leave only rain-attentive features which are passed to the decoder. Finally, we have proposed a recurrent framework with Layered LSTM to form the encoding into a recurrent process with LSTM on each scale of the encoder to iteratively refine the encoded rain features with a fine-grained scale-by-scale style. The quantitative and qualitative evaluations using representative deraining benchmark datasets have demonstrated that our proposed method outperforms existing state-of-the-art deraining methods, where it is particularly noteworthy that our method clearly achieves the best performance on a real-world dataset.

References

- [1] Chenghao Chen and Hao Li. Robust representation learning with feedback for single image deraining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7742–7751, 2021.
- [2] Chenyi Chen, Ari Seff, Alain Kornhauser, and Jianxiang Xiao. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2722–2730, 2015.
- [3] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.
- [4] Liang-Jian Deng, Ting-Zhu Huang, Xi-Le Zhao, and Tai-Xiang Jiang. A directional global sparse model for single image rain removal. *Applied Mathematical Modelling*, 59:662–679, 2018.
- [5] Sen Deng, Mingqiang Wei, Jun Wang, Yidan Feng, Luming Liang, Haoran Xie, Fu Lee Wang, and Meng Wang. Detail-recovery image deraining via context aggregation networks. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 14560–14569, 2020.
- [6] Yingjun Du, Jun Xu, Qiang Qiu, Xiantong Zhen, and Lei Zhang. Variational image deraining. In *Proc. of IEEE Winter Conf. on Applications of Computer Vision (WACV)*, pages 2406–2415, 2020.
- [7] Yingjun Du, Jun Xu, Xiantong Zhen, Ming-Ming Cheng, and Ling Shao. Conditional variational image deraining. *IEEE Trans. on Image Processing*, 29:6288–6301, 2020.
- [8] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Huang, and Xinghao Ding. Residual-guide network for single image deraining. In *Proc. of ACM Int. Conf. on Multimedia (MM)*, pages 1751–1759, 2018.
- [9] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3855–3863, 2017.
- [10] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proc. of Int. Conf. on Artificial Intelligence and Statistics*, pages 315–323, 2011.
- [11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [12] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 8346–8355, 2020.
- [13] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans. on Image Processing*, 21(4):1742–1755, 2011.
- [14] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *Int. Conf. on Learning Representations (ICLR)*, 2015.
- [15] Guanbin Li, Xiang He, Wei Zhang, Huiyou Chang, Le Dong, and Liang Lin. Non-locally enhanced encoder-decoder network for single image de-raining. In *Proc. of ACM Int. Conf. on Multimedia (MM)*, pages 1056–1064, 2018.
- [16] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: A comprehensive benchmark analysis. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3838–3847, 2019.
- [17] Siyuan Li, Wenqi Ren, Feng Wang, Iago Breno Araujo, Eric K Tokuda, Roberto Hirata Junior, Roberto M Cesar-Jr, Zhangyang Wang, and Xiaochun Cao. A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives. *International Journal of Computer Vision*, 129(4):1301–1322, 2021.
- [18] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 254–269, 2018.
- [19] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2736–2744, 2016.
- [20] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.
- [21] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pages 3397–3405, 2015.
- [22] Khan Muhammad, Jamil Ahmad, Zhihan Lv, Paolo Bellavista, Po Yang, and Sung Wook Baik. Efficient deep cnn-based fire detection and localization in video surveillance applications. *IEEE Trans. on Systems, Man, and Cybernetics*, 49(7):1419–1434, 2018.
- [23] Ozan Oktay, Jo Schlemper, Loic Le Folgoc, Matthew Lee, Mattias Heinrich, Kazunari Misawa, Kensaku Mori, Steven McDonagh, Nils Y Hammerla, Bernhard Kainz, Ben Glocker, and Daniel Rueckert. Attention U-net: Learning where to look for the pancreas. In *Proc. of Conf. on Medical Imaging with Deep Learning (MIDL)*, pages 1–10, 2018.
- [24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *Proc. of Advances in Neural Information Processing Systems (NeurIPS)*, pages 8026–8037, 2019.
- [25] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for rain-drop removal from a single image. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 2482–2491, 2018.

- [26] Dongwei Ren, Wei Shang, Pengfei Zhu, Qinghua Hu, Deyu Meng, and Wangmeng Zuo. Single image deraining using bilateral recurrent network. *IEEE Trans. on Image Processing*, 29:6852–6863, 2020.
- [27] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3937–3946, 2019.
- [28] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proc. of Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 234–241, 2015.
- [29] Cong Wang, Yutong Wu, Zhixun Su, and Junyang Chen. Joint self-attention and scale-aggregation for self-calibrated deraining network. In *Proc. of ACM Int. Conf. on Multimedia (MM)*, pages 2517–2525, 2020.
- [30] Cong Wang, Xiaoying Xing, Yutong Wu, Zhixun Su, and Junyang Chen. Dcsfn: Deep cross-scale fusion network for single image rain removal. In *Proc. of ACM Int. Conf. on Multimedia (MM)*, pages 1643–1651, 2020.
- [31] Guoqing Wang, Changming Sun, and Arcot Sowmya. Erl-net: Entangled representation learning for single image deraining. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5644–5652, 2019.
- [32] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3103–3112, 2020.
- [33] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 12270–12279, 2019.
- [34] Wei Wang, Kaicheng Yu, Joachim Hugonot, Pascal Fua, and Mathieu Salzmann. Recurrent u-net for resource-constrained segmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2142–2151, 2019.
- [35] Yinglong Wang, Shuaicheng Liu, Chen Chen, and Bing Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE Trans. on Image Processing*, 26(8):3936–3950, 2017.
- [36] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 3877–3886, 2019.
- [37] SHI Xingjian, Zhouong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Proc. of Advances in Neural Information Processing Systems (NIPS)*, pages 802–810, 2015.
- [38] Wenhan Yang, Robby T Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 42(6):1377–1393, 2019.
- [39] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1357–1366, 2017.
- [40] Wenhan Yang, Robby T Tan, Shiqi Wang, Yuming Fang, and Jiaying Liu. Single image deraining: From model-based to data-driven and beyond. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2020.
- [41] Rajeev Yasarla and Vishal M Patel. Confidence measure guided single image de-raining. *IEEE Trans. on Image Processing*, 29:4544–4555, 2020.
- [42] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, pages 695–704, 2018.
- [43] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE Trans. on Circuits and Systems for Video Technology*, 30(11):3943–3956, 2019.
- [44] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *Proc. of IEEE Int. Conf. on Computer Vision (ICCV)*, pages 2526–2534, 2017.