An Effective Two-Branch Model-Based Deep Network for Single Image Deraining

Yinglong Wang†, Dong Gong‡, Jie Yang‡, Qinfeng Shi‡, Anton van den Hengel‡, Dehua Xie†, Bing Zeng†

†School of Information and Communication Engineering, University of Electronic Science and Technology of China
‡School of Computer Science, The University of Adelaide

Abstract

Removing rain effects from an image automatically has many applications such as autonomous driving, drone pilot- ing and photo editing and still draws the attention of many people. Traditional methods use heuristics to handcraft various priors to remove or separate the rain effects from an image. Recently end-to-end deep learning based deraining methods have been proposed to offer more flexibility and effectiveness. However, they tend not to obtain good visual effect when encountered images with heavy rain. Heavy rain brings not only rain streaks but also haze-like effect which is caused by the accumulation of tiny raindrops. Different from previous deraining methods, in this paper we model rainy images with a new rain model to remove not only rain streaks but also haze-like effect. Guided by our model, we design a two-branch network to learn its parameters. Then, an SPP structure is jointly trained to refine the results of our model to control the degree of removing the haze-like effect flexibly. Besides, a subnetwork which can localize the rainy pixels is proposed to guide the training of our network. Extensive experiments on several datasets show that our method outperforms the state-of-the-art in both objectives assessments and visual quality.

1. Introduction

The prevalence of rain, particularly in some locations, not only seriously reduces the images quality captured by cameras, but more importantly impacts negatively upon the robustness of devices and/or algorithms that must operate continuously irrespective of the weather. For example, the inability of driverless cars to operate in the rain has become a notorious issue. Most of the early attempts use videos [11, 36, 4, 3, 1, 2, 29] as utilising temporal correlation helps to improve the results. Not until recently, single image based deraining methods achieve impressive results and gain popularity [14, 10, 20, 5, 18, 17, 32, 31, 9, 34, 35, 25].

We mainly focus on single image based deraining. Conventional methods can be categorised into three categories. The first is filtering-based, where a nonlocal mean filter or guided filter [14] is used [7]. One limitation is that the filter can adapt to neither the raindrop sizes nor its locations. The second category uses dictionary learning to decompose a rain-affected image so that the dictionary elements corresponding to the rain might be separable from those associated with the image content, and their effects can be removed accordingly [10, 20, 5, 18, 17, 32, 31]. This kind of approaches are suitable for removing rain streaks with clear edges, but for heavy rain, they often leads to undesirable visual artifacts. The third category builds models for rain streaks [6, 27, 26]. These models attempt to discriminate

Figure 1. An example of a real-world rainy image and the deraining results. Our method removes the obvious heavy rain streaks and recovers the colors of the scene by removing the haze-like effect.
rain streaks from the background. However, these methods tend to misidentify fine image details as rain streaks, thus falsely removing desirable fine details.

Very recently, deep learning has also been applied to rain removal and achieved remarkable results [9, 34, 35, 8, 22]. They either train a network to estimate a clear (rain-free) image $B$ from a rainy image $I$ directly, or to estimate a residual $R$ to model the rain layer and $B$ can be obtained by $B = I - R$. Learning to estimate the clear image directly often ignores the forming process of the rainy image, thus requires a large dataset to learn the impact of every possible rain type, at every scale, orientation, and illumination in nearly all scenes to obtain better rain-removed results. Learning the residual processes rainy image as a simple summation of the background image and a rain layer. As rain layer is always easier to learn compared to the background with complex textures, these networks are faster to converge and also obtain remarkable rain-removed effects.

However, both of these two types of networks cannot remove rain effect completely, such as Figure 1. Under the rainy condition, not only rain streaks will be imaged, but also the colors of objects will become shallow or gray as if they are covered by a layer of haze which is more apparent when encountered heavy rain (e.g. Figure 1). This phenomenon is caused by the scattering of accumulated tiny raindrops to light which leads to the color shift [28]. There is no standard term for this phenomenon, so we call it haze-like effect (maybe not accurate) for convenience in our paper. Besides, the edges of rain streaks tend to be blur under this situation, which also enhances the difficulties in the rain-removing task.

To deal with these problems and remove rain effect more deeply and completely, we take the scattering of accumulated tiny raindrops to light into consideration to model a rainy image $I$ as: $I = T \circ B + R$ ($\circ$ is pixel-wise multiplication) to describe the relationship of $I$ and $B$. Compared with the commonly-used linear rain model $I = B + R$, we also use $R$ to model apparent rain streaks, but an enhanced coefficient $T$ is used to model the transmission of accumulated tiny raindrops (haze-like effect) which is always ignored by using $R$ only. Instead of solely training an end-to-end network for rain removal, considering that the rain model itself contains a more specific structure of the deraining task, we integrate the rain model into the design of the deep neural network. Specifically, we propose a two-branch model-based deep network for single image deraining via rAin Model Parameter Estimation, referred to as AMPE-Net. The network framework is shown in Figure 2.

Rain streaks and haze-like effect can be removed cleanly by the trained two-branch subnetworks. However, people may think the haze-like effect is over-removed (the colors of rain-removed results are brighter than the colors in original rainy images) by our method. After all, this is a subjective assessment, different people have different opinions. We will analyze the colors of our results in the later section. However, in order to offer more flexible results to meet the requirements of different people, we utilize a spatial pyramid pooling (SPP) module [16] in our network to refine our results by extracting multi-scale features. Besides, a parameter $\alpha$ is set to control the degree of removing the haze-like effect. Every user of our code will obtain their favourite results. Because the processing of network for rainy pixels is different from that for non-rainy pixels, the location of rain pixels which is always ignored by many other methods can play a positive role in the rain removal performances. We design a LocNet to learn the location of rain to guide the training of subsequent networks. Our main contributions are:

- We use a new rain model which takes the scattering of accumulated tiny raindrops into consideration to model rainy images more completely, so that not only rain streaks but also haze-like effect can be removed to obtain a clearer image.

- We build a two-branch network which is guided by our rain model to jointly learn the model parameters. A SPP module is used to refine our rain-removed results to control the degree of removing haze-like effect. Besides, we design a simple but effective deep convolutional network to identify the location of rain. Rain location provides more information about rainy images which will also help other peer workers in many other rain-removing methods.

- Our method removes full range of rain effects, from large raindrops to haze-like effect. It can also flexibly control the haze-like effect to produce different visual effects. Hence, we provide a more complete solution to rain-removing task. We also simply show the potentials of our model and network in dehazing task.

2. Related Work

Conventional Methods Rain streaks were detected and removed initially in videos. For example, temporal correlation and motion blur were explored to model the photometry of rain in order to detect and remove the rain effect from videos [11]. Same authors extended the work by considering how to render rain streaks as realistic as possible and then using which to remove the rain effect [12]. Similarly, temporal and chromatic characteristics of rain streaks were both explored to detect and remove rain streaks in [36]. The distribution of orientation of streaks was exploited in [3] to remove rain streaks. Apart from exploring the properties in the temporal and spatial domain, characteristics of rain in
frequency domain was also investigated, using which rain streaks were removed based on the frequency [1, 2].

Single-image rain removal has gained much success and popularity recently. There are many attempts using dictionary learning to remove rain streaks by their shape or color characteristics [10, 20, 5, 18, 17, 32, 27]. However this type of methods tend to work well on the rainy images with apparent streaks, and dictionary learning is often time-consuming. To avoid rain pixel detection and time-consuming dictionary learning stage, a low-rank appearance model was proposed in [6] to capture the spatio-temporally correlated rain streaks to remove rain streaks in images (and videos). Instead of learning a dictionary or imposing a low rank structure, simple patch-based priors (using Gaussian mixture models) were proposed to model both the background and rain layers [26]. These priors accommodate multiple orientations and scales of the rain streaks and remove rain streaks well in some cases. However, the resulting images sometimes lose fine details.

Deep Learning Based Methods Very recently, deep learning has been used in rain removal. Inspired by deep residual network (ResNet) [15], a deep detail network was proposed to reduce the mapping range from input to output [9], to make the learning process easier. Moreover, they use priori image domain knowledge by focusing on high frequency detail during training, which can remove the interference from the background to a degree. They extended the work by decomposing the rain image into low and high frequency components, and extract image details from the high-frequency component [8]. These two methods are particularly good for removing light rain, but have issues of removing bright or blur rain streaks. To handle bright rain streaks better, Yang et al. add a binary map to locate rain streak. They create a new model to represent rain streak accumulation (where individual streaks cannot be seen, and they appear as mist or haze instead), and various shapes and directions of overlapping rain streaks [34]. Their method is very good for removing bright rain streaks, but often fails for removing blur rain streaks. To handle diverse rain effects, Zhang et al. propose a multi-stream dense network that can automatically determine the rain-density information and thus can efficiently remove the corresponding rain-streaks according to the estimated rain-density label [35]. This method can handle a diverse range of rainy images, but sometimes causes blur in image details. To model and remove rain streaks of various size (and distance among them) and the veiling effect, a multi-stage network consisting of several parallel sub-networks was designed, each of which models a different scale of rain streaks [23]. Li et al. remove the rain streaks via multiple stages and use recurrent neural network to exchange information across stages [25]. Unlike cascaded multi-stage learning scheme, a non-locally enhanced encoder-decoder network framework is proposed, which captures long-range spatial dependencies via skip-connections and learns increasingly abstract feature representation while preserving the image detail by pooling indices guided decoding [22].

3. The Proposed Method

Given an observed image $I$ contaminated by rain, the goal of deraining is to recover a clean image $B$. Our target is to train a neural network (i.e. AMPE-Net) to estimate $B$ from $I$ by:

$$B = \mathcal{U}(I),$$

(1)

where $\mathcal{U}(\cdot)$ denotes the proposed AMPE-Net. Unlike the commonly-used rain model $I = B + R$, we propose to integrate our rain model into the neural network to guide the estimation of parameters. Before introducing the network $\mathcal{U}(\cdot)$, we will first remodel a rainy image $I$ to express the
relationship between $I$ and $B$ more completely (in Section 3.1).

3.1. Rainy Image Modeling

Existing deep learning based rain-removing works produce good rain-removed performances for many rainy images [35, 9]. Majority of them model a rainy image $I$ as a simple summation of the background $B$ and the rain layer $R$:

$$I = B + R. \quad (2)$$

However, in some rainy scenes, not only large raindrops but also accumulated tiny raindrops will be imaged by a camera [34]. Large raindrops tend to be imaged as apparent rain streaks which are considered to be sparse and have similar falling directions and shapes [25]. Single tiny raindrops cannot be seen, but can impair an image by accumulating together and occluding the propagation of light through scattering. When imaged, they always look like a layer of haze (haze-like effect) which can shallow the colors of background and lead to a low image contrast. In this case, the edges of rain streaks also become blurry and merged into the haze-like effect, which enhances difficulties in deraining task further (e.g. Figure 1).

When encountered rainy images with apparent haze-like effect, especially under heavy rain condition, model (2) tends not to obtain satisfactory results, haze-like effect and some rain streaks which have blurry edges still remain in the deraining results (e.g. Figure 1). To handle above problems, we take the scattering of accumulated tiny raindrops to light into consideration and add a variable $T$ to describe the influence of accumulated tiny raindrops on the background:

$$I = T \circ B + R, \quad (3)$$

where $T$ is the transmission of the accumulated tiny raindrops to model haze-like effect and its value is greater than 1 as haze-like effect always enhance the intensity of pixels [13], and $R$ models the apparent rain streaks with values in range $[0, 1]$.

However, the ground-truth of $T$ and $R$ are difficult to acquire. That is why we design a two-branch network which is trained jointly to evaluate $T$ and $R$ together. We let network itself to determine the optimal parameters to fit our rain model and remove rain effect (including rain streaks and haze-like effect).

3.2. The Proposed AMPE-Net for Deraining

According to our rain model in Eq. (3), given a rainy image $I$, if we can obtain the corresponding parameters $T$ and $R$, the clean image $B$ can be predicted through:

$$\hat{B}_m = (I - R) \odot T, \quad (4)$$

where $\hat{B}_m$ denotes the estimation of the background $B$ by our rain model, and $\odot$ is the point-wise division. However, estimating $T$ and $R$ from $I$ is non-trivial, and different rainy images have different values. We thus consider estimating $T$ and $R$ by learning a jointly-trained networks.

There is no ground truth for $T$ and $R$, hence we cannot implement complete supervision training to the network to simultaneously estimate two unknown variables $T$ and $R$ only under the supervision of the ground truth for background $B$. In our work, we utilize incomplete supervision (not semi-supervision, in our paper ‘incomplete supervision’ means the number of unknown variables is larger than the number of variables which have ground truth during the training) to train a parallel two-branch network which is guided by our new rain model to estimate $T$ and $R$ simultaneously.

As shown in Figure 2, the two subnetworks EstNet-T and EstNet-R which are concatenated in parallel to form a two-branch network to estimate $T$ and $R$, then the estimation $\hat{B}_m$ of background $B$ is calculated further by Eq. (4). As Eq. (4) is differentiable to $T$ and $R$, EstNet-T and EstNet-R can be updated simultaneously to find the optimal $T$ and $R$ to remove rain more completely, which is also the way our model guides the training of two-branch network. Compared with complete supervision training and rain model $I = B + R$, our network and model have more flexibility and capabilities to remove rain effect.

After using Eq. (4), we can obtain a clear image in which rain streaks are removed and the colors of objects are also recovered (e.g. Figure 1(f)). However, people may think the haze-like effect is over-removed (the recovered colors are too bright). We will analyze the color of our results in the experiment section. As visual effect of an image is after all a subjective assessment, different people have different views. In our work, we design a network (RefNet) which are trained jointly with the two-branch EstNet-T and EstNet-R to refine our results and use an averaged weighted combination to control the degree of removing haze-like effect. We will show the results later.

Considering that network has different treating to rainy and non-rainy pixels, we propose to estimate rain location map as a guide. The proposed AMPE-Net consists of three major components: a subnetwork for estimating a rough location map ($\text{LocNet}$), a two-branch subnetwork for estimating $T$ and $R$ ($\text{EstNet-T}$ and $\text{EstNet-R}$) and a SPP module ($\text{RefNet}$) to refine our rain-removed result. In the following parts, we name the two-branch subnetwork of EstNet-T and EstNet-R as two-branch unit for description convenience.

LocNet It takes the rainy image $I$ as input and estimates the location information $L$ of rain pixels in $I$:

$$\hat{L} = \mathcal{H}(I), \quad (5)$$

where $\mathcal{H}(\cdot)$ denotes the mapping of LocNet. $\hat{L}$ is the es-
Raindrops always have different sizes, only using convolutional kernels with single size cannot always extract features of rain completely. Inspired by [35], three densely connected convolutional networks [19] are first utilized in LocNet (Figure 2) to extract multi-scale shallow features of I. The kernel sizes of the three densely-connected blocks are $7 \times 7$, $5 \times 5$ and $3 \times 3$ respectively. We concatenate the obtained features with I to form the shallow-layer feature $f_t$ after a Conv layer.

The core part of our LocNet composes of downsampling, details extraction and up-sampling operations with four different scale factors $(16, 8, 4, 2$ respectively), which is a bit similar to pyramid pooling module in [37]. The reason why we utilize four scale factors to extract the deep features is also the different size and shape of rain. We use 5 ResNet blocks [15] to deepen the features, then up-sample the obtained features to original size to form the deep-layer features $f_d$. After concatenation of the shallow and deep features, a Conv is used to fuse the combined features. At last, we utilize Softmax to estimate the location map $\hat{L}$. Some location maps of real-world rainy images are shown in Figure 3. Not only rain streaks but also pixels covered by haze-like effect will be localised, hence for some images, like Figure 1, majority part will be identified as rain and have high values in $\hat{L}$.

3.4. Network Structure of EstNet-T $F(\cdot)$ and EstNet-R $G(\cdot)$

In two-branch unit, a $9 \times 9$ Conv is firstly utilized to extract the features of the guided input. To eliminate inaccuracy near image edges, the input is reflection-padded. As rain streaks are various in shape and size, then two downsamplings are implemented to suit to the variety. Furthermore, we use five ResBlocks to extract deeper features. Then, two times up-samplings are used to recover the original input size. In order to avoid checkerboard artifacts, we up-sample directly then followed by a Conv to substitute traditional Dconv. After up-sampling the feature map, we concatenate it with the output of the first Conv. At last, $T$ is estimated by a composition of Conv and ReLu operation. Except for the last Softmax, EstNet-R is the same as EstNet-T.

3.5. Network Structure of RefNet $R(\cdot)$

As the rain-removed result by Eq. (4) has already been very close to the ground truth, we do not need high-level features to refine its colors. In $R(\cdot)$, only two Conv layers are first used to extract low-level features. Then SPP which is originally used to improve the recognition accuracy [16] is utilized to obtain multi-scale low-level features. The scale factors are $4, 8, 16$ and $32$ respectively. For every feature with different size, we adopt pointwise convolution [30] to reduce their channels and up-sample them by nearest interpolation method to original size. The refined result is obtained by implementing Conv and Tanh activation function on the concatenated multi-scale features successively.
At last average weighted combination is utilized to integrate refined result and the result before being refined to obtain our final rain removed results.

### 3.6. Training Loss

Our AMPE-Net is trained by two steps. At first, given the training samples \( \{(I_t, L_t)\}_{t=1}^N \), we learn the mapping \( \mathcal{H}(\cdot) \) from \( I \) to \( L \). Then the three sub-networks \( \mathcal{F}(\cdot) \) and \( \mathcal{G}(\cdot) \) in two-branch unit and \( \mathcal{R}(\cdot) \) are trained jointly on the training samples \( \{(I_t, B_t)\}_{t=1}^M \).

**Training loss on location map** LocNet is trained to provide more information for the subsequent networks. Because we apply Softmax activation to approximate the binary location map, the MSE loss is used:

\[
L_L = \sum_{t=1}^N \left\| \mathcal{H}(I_t) - L_t \right\|_F^2. \tag{10}
\]

**Training loss for rain-removal** To fully use the constraints of our rain model to optimize the parameters of two-branch unit, we minimize the following two MSE loss functions which are related to Eq. (3) and (4):

\[
L_1 = \sum_{t=1}^M \left\| I_t - \mathcal{T} \circ B_t - \hat{R} \right\|_F^2, \tag{11}
\]

\[
L_2 = \sum_{t=1}^M \left\| B_t - (I_t - \hat{R}) \circ \mathcal{T} \right\|_F^2. \tag{12}
\]

By these two losses, we can obtain good estimation of \( I_t \) from \( B_t \), and vice versa. Hence, the trained two-branch unit is more robust to remove rain than only using \( L_2 \), the results will be shown in Section 4.5. In the training process, these two loss functions are utilized alternatively with different batches of training samples. Hence, the loss function \( L_m \) to optimize \( T \) and \( R \) and further obtain the rain-removed result \( \hat{B}_m \) by our model is:

\[
L_m = \begin{cases} 
L_1, & i = 1, 3, 5, \ldots \\
L_2, & i = 2, 4, 6, \ldots 
\end{cases} \tag{13}
\]

where \( i \) is the batch index during training. \( \hat{B}_m \) is refined by \( \mathcal{R}(\cdot) \) and average weighted combination is utilized to control the degree of removing haze-like effect to obtain the final rain-removed result \( \hat{B} \). For \( \hat{B} \) and the ground truth \( B \), we also can obtain a loss \( L_r \):

\[
L_r = \sum_{t=1}^M \left\| \alpha \hat{B}_m + (1 - \alpha) \mathcal{R}(\hat{B}_m) - B_t \right\|_F^2. \tag{14}
\]

Our final loss function is:

\[
L = L_m + L_r, \tag{15}
\]

where \( \alpha \) is the parameter to control the degree of removing haze-like effect. Larger \( \alpha \) removes more haze-like effect.

### 4. Experiments

To assess the performances of our method quantitatively, we utilize PSNR and SSIM [33] as evaluation metrics. For real-world images, we only evaluate the performance visually. As the author of [22] is not convenient to release their code, another four state-of-the-art works [9, 34, 35, 25] are selected to make comparisons.

#### 4.1. Datasets

**Training and testing dataset** Li et al. [24] synthesized a rain dataset that includes 20800 training pairs. Zhang et al. [35] synthesized 12000 pairs of training samples. We randomly select half of these datasets respectively to constitute our training dataset. For our LocNet, we utilize the dataset of [34], which includes 2000 pairs of samples. Besides, we randomly select 100 testing sample pairs from the testing datasets of [9, 25, 35] respectively to constitute a 300-image dataset Rain-I as one of our testing datasets. For our LocNet, we utilize the dataset of [34], which includes 2000 pairs of samples. Besides, we randomly select 100 testing sample pairs from the testing datasets of [9, 25, 35] respectively to constitute a 300-image dataset Rain-II as one of our testing datasets. Hence, the trained two-branch unit is more robust to remove rain than only using \( L_2 \), the results will be shown in Section 4.5. In the training process, these two loss functions are utilized alternatively with different batches of training samples. Hence, the loss function \( L_m \) to optimize \( T \) and \( R \) and further obtain the rain-removed result \( \hat{B}_m \) by our model is:

\[
L_m = \begin{cases} 
L_1, & i = 1, 3, 5, \ldots \\
L_2, & i = 2, 4, 6, \ldots 
\end{cases} \tag{13}
\]

where \( i \) is the batch index during training. \( \hat{B}_m \) is refined by \( \mathcal{R}(\cdot) \) and average weighted combination is utilized to control the degree of removing haze-like effect to obtain the final rain-removed result \( \hat{B} \). For \( \hat{B} \) and the ground truth \( B \), we also can obtain a loss \( L_r \):

\[
L_r = \sum_{t=1}^M \left\| \alpha \hat{B}_m + (1 - \alpha) \mathcal{R}(\hat{B}_m) - B_t \right\|_F^2. \tag{14}
\]

Our final loss function is:

\[
L = L_m + L_r, \tag{15}
\]

where \( \alpha \) is the parameter to control the degree of removing haze-like effect. Larger \( \alpha \) removes more haze-like effect.

Figure 4. (a) Rainy images, (b) \( 1 - T \), (c) parameter \( R \), (d) results of \( I - R \), (e) deraining results \( (I - R) \circ T \). \( T \) and \( R \) are normalized to \([0, 1]\) to show clearly.

#### 4.2. Studies of the Behaviors of the Proposed Model

As we said in Section 3.1, we jointly train the two-branch unit, and let network itself learn the most proper parameters \( T \) and \( R \) to remove rain effect. In Figure 4, we show

\[\text{http://www.photoshopessentials.com/photo-effects/rain/}\]
Figure 5. Rain-removed results for synthetic rainy images, GT is short for ground truth, the different values of $\alpha$ are 1.0, 0.6, 0.3, 0.0.

Table 1. PSNR/SSIM of selected methods and our method with different $\alpha$. Please pay attention to the role of different $\alpha$.

| Metric | Rain-I | Rain-II |
|--------|--------|---------|
| [9]    | 29.22  | 0.867   |
| [34]   | 27.22  | 0.832   |
| [35]   | 25.93  | 0.865   |
| [25]   | 27.38  | 0.881   |
| Ours ($\alpha = 1.0$) | 28.90 | 0.853   |
| Ours ($\alpha = 0.6$) | 30.13 | 0.887   |
| Ours ($\alpha = 0.3$) | 31.03 | 0.903   |
| Ours ($\alpha = 0.0$) | **31.65** | **0.905** |

The learned parameters $T$ and $R$ for two randomly-selected real-world rainy images. We can see that nearly all rain streaks are in the parameter $R$, and haze-like effect is included in $T$. From the results of $I - R$ in Figure 4(d), we can see nearly all rain streaks disappear. But some slight haze-like effect and some trace of rain streaks still remain. In Figure 4(e), better deraining results are obtained by the revision of $T$.

4.3. Quantitative Evaluation on Synthetic Datasets

Table 1 shows the PSNR/SSIM values of different methods on the testing datasets Rain-I and Rain-II. When $\alpha = 1$ (without RefNet), our method has comparable PSNR/SSIM with the other methods. After RefNet is added, our PSNR/SSIM surpass them. The reason is some ground truths are taken in slight-haze environment, our method ($\alpha = 1$) will regard haze as haze-like effect and remove it to obtain a more clear images (e.g., the first one in Figure 5) which causes the difference between ground truths and our results and leads to relatively lower objective index. Please take a close look at our results in the first line of Figure 5 with different $\alpha$ controlling the degree of haze-like effect.

Figure 7. Ablation studies on synthetic rainy images: (a) Ground truth, (b) synthetic rainy image, (c-f) results of $H + G (L_1 + L_2)$, $F + G (L_2)$, $H + F + G (L_2)$, $H + F + G (L_1 + L_2)$. The first part of the notation is the used sub-networks, e.g., $H + G$ means $H(\cdot)$ and $G(\cdot)$ are used, the second part in parentheses is the used loss, $L_1 + L_2$ means both $L_1$ and $L_2$ are used in the training.
Table 2. Average running time on $512 \times 512$ image of different methods on our testing datasets.

| Methods    | [9] (CPU) | [34] (GPU) | [35] (GPU) | [25] (GPU) | Ours (GPU) |
|------------|-----------|------------|------------|------------|------------|
| Time       | 4.08s     | 1.40s      | 0.06s      | 0.50s      | 0.05s      |

Figure 8. Ablation studies on real-world rainy images: (a) rainy images, (b-e) results of $\mathcal{H} + \mathcal{G} (\mathcal{L}_1 + \mathcal{L}_2)$, $\mathcal{F} + \mathcal{G} (\mathcal{L}_1 + \mathcal{L}_2)$, $\mathcal{H} + \mathcal{F} + \mathcal{G} (\mathcal{L}_1 + \mathcal{L}_2)$. The notations have the same meanings as in Figure 7.

Table 3. PSNR/SSIM of the variants of our AMPE-Net

| $\mathcal{H}$ | $\mathcal{F}$ | $\mathcal{G}$ | $\mathcal{L}_1$ | $\mathcal{L}_2$ | Rain-I  | Rain-II |
|------------|------------|------------|------------|------------|---------|---------|
| √          | √          | √          | √          | 26.42/0.794 | 28.58/0.888 |
| √          | √          | √          | √          | 27.80/0.830 | 29.67/0.914 |
| √          | √          | √          | √          | 28.63/0.849 | 30.37/0.923 |
| √          | √          | √          | √          | 28.90/0.853 | 30.45/0.925 |

The ground truth in the second line (from the testing dataset by Yang et al.) are clear, so the results obtain by different $\alpha$ are almost the same and our method obtain better rain-removed results. In Table 2, we show the averaged running time consumed by selected methods on our testing datasets. We can see our method is faster.

4.4. Qualitative Evaluation on Real-World Images

In this section, we show some results of real-world image in Figure 6. We can see that our method outperforms the state-of-the-art. Rain streaks and haze-like effect are both removed. Figure 6(f) is the result of our two-branch unit without $\mathcal{R}(\cdot)$ ($\alpha = 1$), in which object colors become brighter than in rainy images. This is the result of removing haze-like effect, not unnatural hue. The color of letters in the second line images is gray which is kept in our result in Figure 6(f), hence our method do not introduce abnormal hue in the rain-removed results. The results of tuning the degree of removing haze-like effect are shown in Figure 6(g)(h)(i) with selected $\alpha$. We can see that the degree of removing haze-like effect can be controlled by $\alpha$ flexibly. Our results are more closer to the reality and the image details are also preserved better. Selected methods cannot remove haze-like effect well and some apparent rain streaks with blurry edges still remain in the final results for some images.

4.5. Ablation Studies

To verify the roles of different parts in our AMPE-Net, we do some ablation experiments. PSNR/SSIM of different variants of our network are shown in Table 3. Some visual results on synthetic and real-world images are shown in Figure 7 and 8 respectively. In our ablation study, we do not include $\mathcal{R}(\cdot)$ whose role has been shown above and all the experiments in this subsection are done with $\alpha = 1$. We can see that the guide of $\mathcal{H}(\cdot)$ (LocNet) to the subsequent network is important and it boosts performances apparently. When removing $\mathcal{F}(\cdot)$ (EstNet-T), our model degrades into $\mathbf{I} = \mathbf{B} + \mathbf{R}$, the PSNR/SSIM decrease most seriously. Besides, the performance of removing haze-like effect is also lower than other cases which proves the role of rain model further (Figure 8). The loss $\mathcal{L}_1$ does not make too many differences.

4.6. Potentials of our Model and Network

Our model can be easily extended to other weather conditions, such as haze and snow. For haze, we randomly select 5000 training samples from [21] to dehaze with our model and networks (note that the LocNet will not be used in haze condition). Here, we only show some dehazing results for real-world hazy images in Figure 9. For more complete dehazing results including comparisons with state-of-the-art works we will show in our extended journal paper. We will try to deal with snowy images in the future. Besides, our model has the potential to deal with deblur task. The reason is that the blurry image can be modelled by convolutions with different masks, and the operation of convolution can be rewrite as a linear model.

5. Conclusion

In this paper, we utilized a new model to describe rainy images more completely. In order to remove rain effect more completely, we proposed a two-branch network to jointly learn the parameters in our rain model. Two invertible loss functions are utilized to optimize the two-branch unit alternatively to better fit our model. To control the degree of removing haze-like effect, an average weighted
combination and a SPP structure was utilized to refine our rain-removed results. Besides, a location map of rain was also learned to guide the training of our network. Compared with several state-of-the-art deep learning works, our method outperforms these methods objectively and subjectively, and our work can handle more kinds of rainy images, including removing haze-like effect to recover the original color of degraded images.

References

[1] P. Barnum, T. Kanade, and S. Narasimhan. Spatiotemporal frequency analysis for removing rain and snow from videos. In International Workshop on Photometric Analysis For Computer Vision (PACV 2007), pages 8–p, Rio de Janeiro, Brazil, Oct. 2007. INRIA. 1, 3

[2] P. C. Barnum, S. Narasimhan, and T. Kanade. Analysis of rain and snow in frequency space. International Journal of Computer Vision, 86(2):256–274, Jan. 2010. 1, 3

[3] J. Bossu, N. Hautiere, and J. P. Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. International Journal of Computer Vision, 93(3):348–367, July 2011. 1, 2

[4] N. Brewer and N. Liu. Using the shape characteristics of rain to identify and remove rain from video. In Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, volume 5342, pages 451–458, Olando, USA, Dec. 2008. 1

[5] D. Y. Chen, C. C. Chen, and L. W. Kang. Visual depth guided color image rain streaks removal using sparse coding. IEEE Transactions on Circuits and Systems for Video Technology, 24(8):1430–1455, Aug. 2014. 1, 3

[6] Y. L. Chen and C. T. Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In IEEE International Conference on Computer Vision (ICCV 2013), pages 1968–1975, Sydney, Australia, Dec. 2013. IEEE. 1, 3

[7] X. H. Ding, L. Q. Chen, X. H. Zheng, Y. Huang, and D. L. Zeng. Single image rain and snow removal via guided 10 smoothing filter. Multimedia Tools and Applications, 24(8):1–16, Jun. 2015. 1

[8] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the skies: a deep network architecture for single-image rain removal. IEEE Transactions on Image Processing, 26(6):2944–2956, July 2017. 2, 3

[9] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2017), pages 1715–1723, Honolulu, HI, USA, July 2017. IEEE. 1, 2, 3, 4, 6, 7, 8

[10] Y. H. Fu, L. W. Kang, C. W. Lin, and C. T. Hsu. Single-frame-based rain removal via image decomposition. In IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011), pages 1453–1456, Prague, Czech Republic, May 2011. IEEE. 1, 3

[11] K. Garg and S. K. Nayar. Detection and removal of rain from videos. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2004), volume 1, pages 528–535, Washington DC, USA, Jun. 2004. IEEE. 1, 2

[12] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. ACM Transactions on Graphics, 25(3):996–1002, Jul. 2006. 2

[13] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(12):2341–2353, Dec. 2011. 4

[14] K. He, J. Sun, and X. Tang. Guided image filtering. IEEE Transactions on Pattern Analysis and Machine Intelligence, 35(6):1397–1409, June 2013. 1

[15] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2017). IEEE, July 2015. 3, 5

[16] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. arXiv:1406.4729, 2015. 2, 5

[17] D. A. Huang, L. W. Kang, Y. C. F. Wang, and C. W. Lin. Self-learning based image decomposition with applications to single image denoising. IEEE Transactions on Multimedia, 16(1):83–93, Jan. 2014. 1, 3

[18] D. A. Huang, L. W. Kang, M. C. Yang, C. W. Lin, and Y. C. F. Wang. Context-aware single image rain removal. In IEEE International Conference on Multimedia and Expo (ICME-2012), pages 164–169, Melbourne, Australia, July 2012. IEEE. 1, 3

[19] G. Huang, Z. Liu, and L. Maaten. Densely connected convolutional networks. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2017), pages 4700–4708, Honolulu, HI, USA, July 2017. IEEE. 5

[20] L. W. Kang, C. W. Lin, and Y. H. Fu. Automatic single-image-based rain streaks removal via image decomposition. IEEE Transactions on Image Processing, 21(4):1742–1755, Apr. 2012. 1, 3

[21] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang. Benchmarking single-image dehazing
and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2019.

[22] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin. Non-locally enhanced encoder-decoder network for single image de-raining. In *ACM Multimedia (MM-2018)*, Seoul, Republic of Korea, Oct. 2018. ACM. 2, 3, 6

[23] R. Li, L. F. Cheong, and R. T. Tan. Single image de-raining using scale-aware multi-stage recurrent network. *arXiv:1712.06830*, 2017. 3

[24] S. Li, W. Ren, J. Zhang, J. Yu, and X. Guo. Fast single image rain removal via a deep decomposition-composition network. *arXiv:1804.02688*, 2018. 6

[25] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *European Conference on Computer Vision (ECCV-2018)*, Munich, Germany, Sep. 2018. IEEE. 1, 3, 4, 6, 7, 8

[26] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016)*, pages 2736–2744, Las Vegas, Nevada, USA, June 2016. 1, 3

[27] Y. Luo, X. Yong, and J. Hui. Removing rain from a single image via discriminative sparse coding. In *IEEE International Conference on Computer Vision (ICCV 2015)*, pages 3397–3405, Boston, MA, USA, Dec. 2015. 1, 3

[28] S. G. Narasimhan and S. K. Nayar. Vision and the atmosphere. *International journal of computer vision*, (3):233–254, Dec. 2002. 2

[29] M. Roser and A. Geiger. Video-based raindrop detection for improved image registration. In *IEEE International Conference on Computer Vision Workshops (ICCV Workshops 2009)*, pages 570–577, Kyoto, Japan, Sep. 2009. IEEE. 1

[30] L. Sifre. Rigid-motion scattering for image classification. *Ph.D thesis*, 2014. 5

[31] Y. Wang, S. Liu, C. Chen, and B. Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE Transactions on Image Processing*, 26(8):3936–3950, August 2017. 1

[32] Y. L. Wang, C. Chen, S. Y. Zhu, and B. Zeng. A framework of single-image deraining method based on analysis of rain characteristics. In *IEEE International Conference on Image Processing (ICIP 2013)*, pages 4087 – 4091, Phoenix, USA, Sep. 2016. IEEE. 1, 3

[33] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004. 6

[34] W. Yang, R. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep joint rain detection and removal from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2017)*, pages 1685–1694, Honolulu, HI, USA, July 2017. IEEE. 1, 2, 3, 4, 6, 7, 8

[35] H. Zhang and V. Patel. Density-aware single image deraining using a multi-stream dense network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2018)*, pages 1685–1694, Salt Lake City, UT, July 2018. IEEE. 1, 2, 3, 4, 5, 6, 7, 8

[36] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng. Rain removal in video by combining temporal and chromatic properties. In *IEEE International Conference on Multimedia and Expo (ICME 2006)*, volume 1, pages 461–464, Toronto, Ontario, Canada, July 2006. IEEE. 1, 2

[37] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia. Pyramid scene parsing network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR-2017)*, pages 2881–2890, Honolulu, HI, USA, July 2017. IEEE. 5