Deep Learning for Seeing Through Window With Raindrops

Yuhui Quan[1], Shijie Deng[1], Yixin Chen[1], Hui Ji[2]
[1]School of Computer Science & Engineering, South China University of Technology, Guangzhou 510006, China
[2]Department of Mathematics, National University of Singapore, Singapore 119076
{csyhquan@scut.edu.cn; shijie.deng.cs@foxmail.com; yx.chen.cs@foxmail.com; matjh@nus.edu.sg} *

Abstract

When taking pictures through glass window in rainy day, the images are comprised and corrupted by the raindrops adhered to glass surfaces. It is a challenging problem to remove the effect of raindrops from an image. The key task is how to accurately and robustly identify the raindrop regions in an image. This paper develops a convolutional neural network (CNN) for removing the effect of raindrops from an image. In the proposed CNN, we introduce a double attention mechanism that concurrently guides the CNN using shape-driven attention and channel re-calibration. The shape-driven attention exploits physical shape priors of raindrops, i.e. *convexness and contour closedness, to accurately locate raindrops, and the channel re-calibration improves the robustness when processing raindrops with varying appearances. The experimental results show that the proposed CNN outperforms the state-of-the-art approaches in terms of both quantitative metrics and visual quality.*

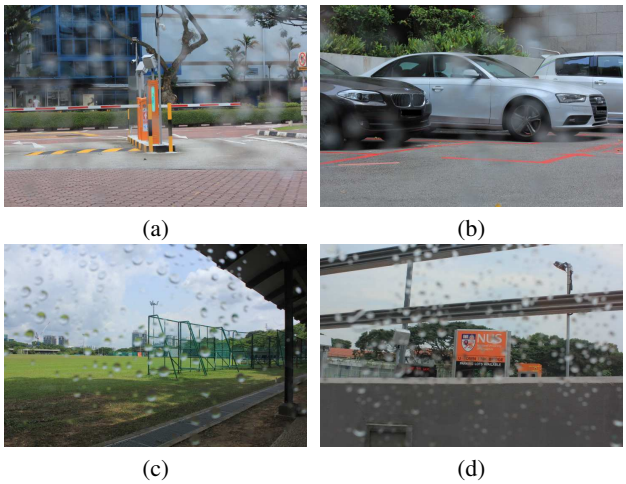
1. Introduction

In rainy weather, there are many situations that one need to take pictures of outside scenes through glass windows or window shields. As the glass window is covered by raindrops in such scenarios, the captured images are comprised by the effect caused by these raindrops. The same phenomenon happens when camera lens are covered by raindrops when taking pictures outdoor in rainy day. The effect of raindrops on an image can significantly degrade its visibility. See Fig. 1 for the illustration of real images taken through windows with raindrops. In addition to visual quality degradation, the raindrops also have negative impacts on the performance of outdoor computer vision systems, particularly the ones used in safety driving, outdoor surveillance, intelligent vehicle systems and camera drones; see *e.g.* [12,23]. There is certainly a need for developing computational methods that can effectively remove raindrops from an image, *i.e.* restore the visual distortions caused by the raindrops on images.

The optical model of an image taken through window with raindrops is quite complex. For our purpose, we consider a simple linear model [23]:

$$\boldsymbol{I} = (1 - \boldsymbol{A}) \odot \boldsymbol{L} + \boldsymbol{A} \odot \boldsymbol{R}, \tag{1}$$

where $\odot$ denotes element-wise multiplication, and $\boldsymbol{I}, \boldsymbol{L}, \boldsymbol{R} \in \mathbb{R}^{C\times M\times N}$ denote the image with raindrops, the latent raindrop-free layer, and the raindrop layer respectively. The matrix $\boldsymbol{A} \in [0,1]^{C\times M\times N}$ denotes the transparency matrix. Each entry of $\boldsymbol{A}$ represents the percentage of the light path covered by raindrops for the corresponding pixel. As the latent image layer $\boldsymbol{L}$ refers to the scene behind the glass, we also refer to $\boldsymbol{L}$ as background image/layer.



(a) (b) (c) (d)

Figure 1: Examples of images with raindrop effects.

Different values of $\boldsymbol{A}$ will lead to different types of visual distortions. Consider a pixel indexed by r. When

*This research was supported by the National Natural Science Foundation of China (61602184, 61872151), Natural Science Foundation of Guangdong Province (2017A030313376), Fundamental Research Funds for Central Universities of China (x2js-D2181690), and Singapore MOE AcRF (R146000229114, MOE2017-T2-2-156).

$\boldsymbol{A}(r) = 0$, the pixel belongs to the background image and has no visual effect of raindrop; when $0 < \boldsymbol{A}(r) < 1$, the pixel only contains partial information of the background (*e.g.* Fig. 1 (a)–(b)); and when $\boldsymbol{A}(r) = 1$, the background part is completely occluded by raindrops (*e.g.* Fig. 1 (c)–(d)). In the last case, there might be more complex effects on the appearance of raindrops due to the light reflection on the glass.

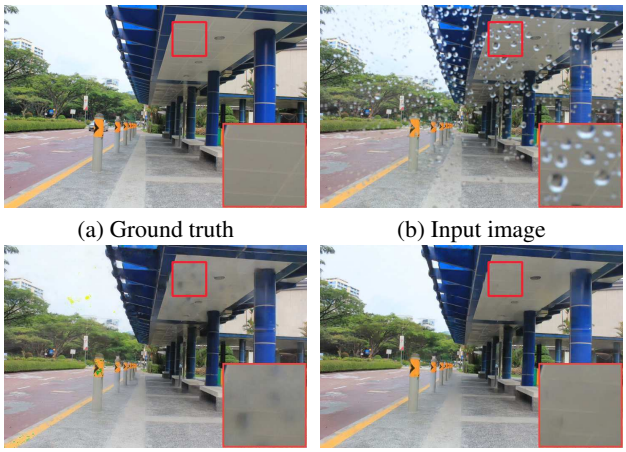
It is a challenging problem to solve (1) for removing raindrops from a single image. With totally three unknowns $\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{A}$ to be estimated, the problem is highly ill-posed with an infinite number of mathematically sound solutions. Some approaches utilize specific imaging hardware, such as multiple cameras [20], pan-tilt camera [19], stereo camera [21] and video camera [14, 23], to provide multiple images so that more information can be used to resolve the ambiguities. Such approaches have limitations on the applicability in practice.

Without additional sources, strong priors need to be imposed on the raindrop layer $\boldsymbol{R}$ so that the raindrops can be accurately detected. Then, one might be able to separate the raindrop layer and the background layer from the input. Unfortunately, owing to the significant variations of raindrops on size, shape and reflection, it is very difficult to have a universal characterization on the appearance of raindrops. It is noted that raindrops are harder to identify than rain streaks, as rain streaks have simpler reflection effect and have quite straightforward priors on its orientation and needle-type (*e.g.* [2, 10]).

1.1. Related work

Based on the availability of sources, the existing methods for raindrop removal can be categorized as multi-image methods and single-image methods.

Multi-image methods. Taking multiple images as input, multi-image methods [14, 19–21, 23] exploit the strong correlations among multiple images for jointly eliminating the raindrops of multiple images. Yamashita *et al.* [20] worked on the images of the same scene taken by a multi-camera system. It is further extended in [19] to work on a single-camera system, which simulates a multi-camera system by adjusting the camera to different angles for generating additional images. Later, Yamashita *et al.* [21] also worked on the images produced by a stereo camera system. The methods above request specific hardware, which limits their applicability in practice. There are a few methods [8, 14, 23] working on video data and utilizing the contextual information provided by adjacent frames. Kurihata *et al.* [8] proposed to learn the spatio-temporal features of raindrops using PCA and then detect the raindrops by template matching with learned raindrop features. Roser *et al.* [15] proposed to recognize raindrops using Bezier curve fitting. These two methods mainly focus on raindrop detection, not removal. Roser and Geiger [14] proposed a method that detects raindrops using a photometric raindrop model and restores occluded regions by fusing the intensity information from adjacent image frames. You *et al.* [23] exploited the difference on the motion speed as well as the difference on the scale of temporal intensity change between raindrop pixels and latent image pixels for separating the raindrops from a video.



(a) Ground truth (b) Input image

(c) Qian *et al.*'s [12] (d) Ours

Figure 2: Raindrop removal using different methods.

Single-image methods. Instead of using multiple images, single-image methods only take one image as input. The problem of removing raindrops from a single image is much more challenging. The progress along this line has been stagnated until the rapid progress of deep learning in recent years. Eigen *et al.* [1] used a shallow convolutional neural network (CNN) with only three layers for the task. This method does not work well on the images with large and dense raindrops, as shown in [12].

Recently, Qian *et al.* [12] proposed a generative adversarial network (GAN) for raindrop removal. They trained a recurrent network with spatial attention mechanism for generating attentive raindrop masks from the input image. The generated raindrop mask is then used to help restoring the latent image using a CNN with a discrimination loss. The performance of Qian *et al.*'s method is highly dependent on the availability of the high-quality raindrop masks that are used as the ground-truths for training the neural network (NN). Nevertheless, it is very difficult to prepare such training data. A dataset for training NN-based raindrop removers is also presented by Qian *et al.* [12], whose ground-truth raindrop masks are not very accurate owing to several factors, *e.g.* the binarization of the mask which leads to accuracy loss, and the imperfect alignment of rainy image and raindrop-free image. See Fig. 2 for an illustration of the results from existing methods. There is certainly a lot of room for further improvement. It is also shown in [12] that

the general learning of image-to-image mapping by CNNs with GANs, *e.g*. Pix2Pix [6], does not perform well either. In comparison to the success of deep learning on removing rain streaks from images [2–4, 9, 22], the deep learning for raindrop removal is still in its infancy stage.

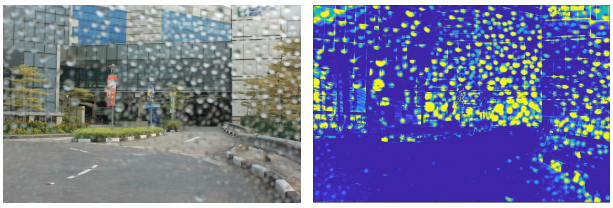
1.2. Motivation and Contributions

Inspired by recent success of deep learning in image processing, this paper proposes a CNN-based method for removing raindrops from a single image. Particularly, we build an attention mechanism into the CNN which exploits both shape-driven attention and channel re-calibration for effectively removing the raindrops from a single image.

Shape-driven attention of raindrop regions. As the transparency effect caused by raindrops is spatially-varying, the transparency degree varies at different locations. Recall that the transparency degree is related to the magnitude of the mask matrix $\boldsymbol{A}$ in Eq. (1). Then $\boldsymbol{A}$ can be used as an attention mask which guides the detection of raindrop regions in the CNN. A recurrent sub-network is trained in Qian *et al*.'s approach [12], under the supervision of truth masks in binary form. Indeed, the binary form of truth masks is oversimplified. Together with the challenges on building high-quality training data of truth masks, the results could contain noticeable artifacts in many cases; see *e.g*. Fig. 2. Such an approach for detecting raindrop regions omits the inherent shape property of raindrops, *e.g*. plumpness (or convexness), non-anisotropy, and contour closedness. These physical properties indeed provide very informative shape priors for identifying raindrop regions, which should be exploited in the NN for facilitating the estimation of the attention mask associated with the raindrop regions.

In this paper, we propose a shape-driven module to help determining the attention mask. It is noted that the shape of most raindrops in images are consistent on the geometrical properties of convexness and contour closedness. Moreover, the raindrops tend to have roundness [23]. Such contours can be well modeled by ellipses or mixtures of ellipses; see *e.g*. Fig. 3. Motivated by such an observation, we propose a measurement of local image regions with the following two properties: (i) it can be easily implemented in the CNN with learnable parameters, and (ii) it measures the probability of a patch embracing a raindrop, in terms of the fitness of the alignment of elliptic iso-contours with the contour edges in the patch. Such a shape-driven attention mechanism can be painlessly plugged into the CNN to guide the detection of raindrops.

Channel re-calibration of CNN features. The problem of removing raindrops in Eq. (1) can be viewed as a layer separation problem. One key to layer separation is the discriminative depiction of two layers, which can be improved by refining the discriminability of features or the discriminability of feature responses; see *e.g*. [10]. A CNN for image raindrop removal often encodes a huge number of image features in its intermediate results, including the features relevant/irrelevant to image/raindrop. It is very helpful to have a mechanism, during the separation process, to separate features of natural images and raindrop-related features in the CNN. More importantly, given a test image/patch, not all the features in the CNN are equivalently useful. It is better to selectively use the features that are more related to the natural image contents as well as the raindrop types in the given image/patch, which benefits the artifact suppression.



(a) Image with raindrops (b) Raindrop map

Figure 3: A rainy image and its raindrop map generated by the difference between the rainy image and the truth one. Blue denotes zero and red denotes the largest value.

In the context of the channels of a CNN, different channels are about different types of image features or raindrop features, which cover a wide range of local image patterns (*e.g*. edges with different orientations) as well as raindrop patterns (*e.g*. different sizes and appearance) for restoration. Then, the feature selection in a CNN is about assigning different weights to different feature channels. This motivates us to incorporate a channel attention/re-calibration mechanism into the CNN. The channel attention mechanism can estimate the contribution of each feature channel based on the inter-dependency among the channels, by which the CNN can automatically include relevant features or exclude irrelevant features for a given image during restoration.

Joint attention. Combining the proposed shape-driven attention module and the channel attention module, we propose a CNN with a built-in joint attention mechanism for single-image raindrop removal. The results show that such an attention mechanism brings noticeable performance improvement. Compared to Qian *et al*.'s approach [12], our attention is free from the supervised training with high-quality masks which can be troublesome to construct, and has better robustness in recovery with the help of channel re-calibration. In the experiments, our approach outperformed existing state-of-the-art ones.

2. Measure of Roundness and Closedness

The raindrops in images show two geometric properties that are quite consistent over most scenarios: one is convexness and the other is contour closedness. Built upon the image torque operator [11, 16, 18] that captures the concept

of closed Gestalt-like contours with robustness to noises, we propose a measure, implemented in the CNN with learnable parameters, on the convexness and closedness of local image contours.

More specifically, considering that the shapes of raindrops tend to have roundness [23] which can be modeled by ellipses and mixture of ellipses, we generalize the torque operator to handle elliptic contours and develop a calculation scheme based on the convolution operations. Note that there are other powerful ellipse detectors for handling complex variations of regions (*e.g.* texture regions). These methods cannot be implemented using simple convolution operations and they are overkill for our case with only flat or smooth variations on raindrop regions.

Recall that the iso-contours of ellipses are of the function

$$f(x,y) = \frac{x^2}{a^2} + \frac{y^2}{b^2}, \tag{2}$$

where a, b are the semi-major and semi-minor axes respectively. The tangent lines of these iso-contours, denoted by $\boldsymbol{g}(x,y)$, are perpendicular to the gradient field $\nabla f = [\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}]$ and thus are given by

$$\boldsymbol{g}(x,y) = [-\frac{\partial f}{\partial y}, \frac{\partial f}{\partial x}] = [-\frac{y}{b^2}, \frac{x}{a^2}]. \tag{3}$$

Given an input test patch modeled as a 2D function $P(x,y)$ with its domain denoted by $\mathcal{D}(P)$, we first determine its gradient field, denoted as $\nabla P = [\frac{\partial P}{\partial x}, \frac{\partial P}{\partial y}]$, and their edges (tangent vectors), denoted as $\boldsymbol{e}(x,y) = [-\frac{\partial P}{\partial y}, \frac{\partial P}{\partial x}]$. If an elliptic pattern exists in P, then the edges $\boldsymbol{e}(x,y)$ must align well with the tangent vectors $\boldsymbol{g}(x,y)$. A simple measure of alignment for a point $(x_0, y_0) \in \mathcal{D}(P)$ is the inner product between $\boldsymbol{e}(x_0,y_0)$ and $\boldsymbol{g}(x_0,y_0)$:

$$\begin{aligned}\beta(x_0,y_0) &= \boldsymbol{e}(x_0,y_0) \cdot \boldsymbol{g}(x_0,y_0)^\top \\ &= [-\frac{\partial P}{\partial y}(x_0,y_0), \frac{\partial P}{\partial x}(x_0,y_0)] \cdot [-\frac{y_0}{b^2}, \frac{x_0}{a^2}]^\top.\end{aligned}$$

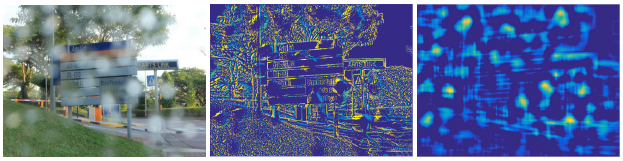
Then the torque on the patch P, measuring how likely the P embraces a complete ellipse, is given by

$$\tau(P) = \sum_{(x_0,y_0)\in\mathcal{D}(P)} \beta(x_0,y_0)/\#\mathcal{D}(P), \tag{4}$$

where $\#\mathcal{D}(P)$ is the number of pixels in P to make $\tau(P)$ invariant to the scale of P. In other words, $\tau(P)$ measures the roundness of the shape embraced by P. Given an image $\boldsymbol{I}$, $\tau(\cdot)$ can be calculated on all its image patches with size $S \times S$. Then, the torque operator, denoted by $\mu_S(\boldsymbol{I})$, which indicates the likeliness for each pixel of $\boldsymbol{I}$ that the pixel is contained in an ellipse with major-axis $2a$ and minor-axis $2b$:

$$\mu_S(\boldsymbol{I})(i,j) = \tau(P_{(i,j),S}), \tag{5}$$

where $P_{(i,j),S}$ denotes the patch centered at the position (i,j) with the size $S \times S$. Since raindrops can be well modeled by ellipses or mixtures of ellipses with closedness, the operator $\mu_S(\boldsymbol{I})(i,j)$ can be used to capture the raindrops in an image. See Fig. 4 (c) for an example.



(a) Input image (b) Edge map (c) Measure

Figure 4: Torque-based measure on an image with raindrops. Blue denotes zero and red denotes the largest value.

The calculation of $\mu_S(\boldsymbol{I})$ can be implemented based on convolution. Let $P_1, \cdots, P_N$ denote all patches of size $S \times S$ of $\boldsymbol{I}$. Define $\mathcal{D}(P_n) = \{(x_i, y_j)\}_{i,j=1}^S$. We rewrite $\tau(P_n)$ as

$$\tau(P_n) = \tau_1(P_n) + \tau_2(P_n),$$

where

$$\tau_1(P_n) = \sum_{(x_i,y_j)\in\mathcal{D}(P_n)} \frac{x_i}{a^2}\frac{\partial P_n}{\partial x}(x_i,y_j)/\#\mathcal{D}(P), \tag{6}$$

$$\tau_2(P_n) = \sum_{(x_i,y_j)\in\mathcal{D}(P_n)} \frac{y_j}{b^2}\frac{\partial P_n}{\partial y}(x_i,y_j)/\#\mathcal{D}(P). \tag{7}$$

Let $\boldsymbol{E}_1^{(n)} \in \mathbb{R}^{S\times S}, \boldsymbol{E}_2^{(n)} \in \mathbb{R}^{S\times S}$ denote two matrices such that $\boldsymbol{E}_1^{(n)}(i,j) = \frac{\partial P_n}{\partial x}(x_i,y_j), \boldsymbol{E}_2^{(n)}(i,j) = \frac{\partial P_n}{\partial y}(x_i,y_j)$, which are the n-th patch of the edge maps $\boldsymbol{B}_1 = \nabla_x \boldsymbol{I}, \boldsymbol{B}_2 = \nabla_y \boldsymbol{I}$ respectively. Then $\tau_1(P_n)$ is equivalent to the inner product of $\boldsymbol{E}_1^{(n)}$ and $\boldsymbol{H}'$ where

$$\boldsymbol{H}' = \frac{1}{S^2a^2}[x_1, \cdots, x_S]^\top \underbrace{[1, \cdots, 1]}_{S}. \tag{8}$$

The inner product of $\boldsymbol{E}_1^{(n)}$ and $\boldsymbol{H}'$ over all n is the same as the convolution of $\boldsymbol{B}_1$ with the kernel $\boldsymbol{H} = \text{flip}(\boldsymbol{H}')$. Similarly, the inner product of $\boldsymbol{E}_2^{(n)}$ and $\boldsymbol{V}'$ over all n is the same as the convolution of $\boldsymbol{B}_2$ with the kernel $\boldsymbol{V} = \text{flip}(\boldsymbol{V}')$ where

$$\boldsymbol{V}' = \frac{1}{S^2b^2}\underbrace{[1, \cdots, 1]}_{S}^\top [y_1, \cdots, y_S]. \tag{9}$$

Thus, we have

$$\mu_S(\boldsymbol{I}) = \boldsymbol{B}_1 \otimes \boldsymbol{H}_{a,S} + \boldsymbol{B}_2 \otimes \boldsymbol{V}_{b,S}, \tag{10}$$

where $\otimes$ denotes the convolution, and $\boldsymbol{H}_{a,S}, \boldsymbol{V}_{b,S}$ are of the form $\boldsymbol{H}, \boldsymbol{V}$ with their parameters as subscripts. See supplementary materials for some demonstrations of our proposed measure.

Robust edge detector. The edge maps $\boldsymbol{B}_1, \boldsymbol{B}_2$ computed

with image gradients are not robust to noises and make the measure μ vulnerable to local intensity variations. A robust edge detector is implemented to compute $\boldsymbol{B}_1, \boldsymbol{B}_2$. The edge detector identifies image edges as follows. Firstly, find the difference magnitude between neighboring pixels offset by given step (set to 2 by default) for vertical pairs and horizontal pairs (a cross of four segment). Secondly, find the maximum/minimum within the crosses of four pairs. Lastly, an edge is identified if the maximum of a cross, is the minimum in a neighbor cross. Let $\theta(x, y)$ denote the gradient orientation of the identified edge point at the position (x, y). Then we compute $\boldsymbol{B}_1(x, y) = \cos(\theta(x, y)), \boldsymbol{B}_2(x, y) = \sin(\theta(x, y))$. For the non-edge pixels, we set the corresponding elements in $\boldsymbol{B}_1, \boldsymbol{B}_2$ to zero. See Fig. 4(b) for an example of such an edge detector and supplementary materials for more examples.

3. Proposed Neural Network

3.1. Network architecture

The proposed CNN for raindrop removal is built upon the encoder-decoder architecture [13] which has been widely used in image processing, including image deraining [2, 12]. See Fig. 5 (a) for the framework of the proposed CNN. The CNN takes as input an image with raindrops, denoted by $\boldsymbol{I}_{\text{r}} \in \mathbb{R}^{C\times H\times W}$, and two edge maps of the image computed by the aforementioned robust edge detector, denoted by $\boldsymbol{B}_1, \boldsymbol{B}_2 \in \mathbb{R}^{1\times H\times W}$. The output is a raindrop-free image, denoted by $\boldsymbol{I}_{\text{c}} \in \mathbb{R}^{C\times H\times W}$:

$$f : (\boldsymbol{I}_{\text{r}}, \boldsymbol{B}_1, \boldsymbol{B}_2) \to \boldsymbol{I}_{\text{c}}. \tag{11}$$

The image is first concatenated with the edge maps for additional high-frequency information, which forms a tensor in $\mathbb{R}^{(C+2)\times H\times W}$. Then the tensor is sequentially passed to a convolutional (Conv) layer with rectified linear unit (ReLU), an encoder, a decoder and a convolutional layer. The encoder contains nine residual blocks (ResBlks) which are divided into three groups. As shown in Fig. 5 (b), each ResBlk sequentially connects Conv/ReLU/Conv layers and a joint physical shape + channel attention (JPCA) module, with a skip connection from the front to the end. The JPCA module additionally requires the input edge maps $\boldsymbol{B}_1, \boldsymbol{B}_2$ as input. The numbers of convolution kernels of the three ResBlk groups (from left to right) are 32, 64 and 128 respectively. The down-sampling layer is inserted before the last two ResBlk groups. As for the decoder, its structure is symmetric to the encoder, except that the downsampling layers are replaced with the upsampling layers. The downsampling layers are implemented by the convolution with stride 2, and the upsampling layers are implemented by the transposed convolution. The sizes of all convolution kernels in the CNN, if not specified, are set to 5×5. For better optimization as well as better preserving image details [5], the long skip connections are added to connect the corresponding ResBlks in the encoder/decoder.

3.2. Joint attention module

Given the intermediate feature maps $\boldsymbol{F} \in \mathbb{R}^{C\times H\times W}$ and the edge maps $\boldsymbol{B}_1, \boldsymbol{B}_2$ of the input image, our JPCA module outputs the re-calibrated feature maps $\boldsymbol{F}' \in \mathbb{R}^{C\times H\times W}$ based on attention:

$$\text{JPCA} : (\boldsymbol{F}, \boldsymbol{B}_1, \boldsymbol{B}_2) \to \boldsymbol{F}'. \tag{12}$$

The structure of the JPCA module is illustrated in Fig. 5 (c). Firstly, the JPCA module infers a 2D spatial attention map $\boldsymbol{A}^{\text{sp}} \in \mathbb{R}^{1\times H\times W}$ and an 1D channel attention map $\boldsymbol{A}^{\text{ch}} \in \mathbb{R}^{C\times 1\times 1}$. Then, the JPCA map, denoted by $\boldsymbol{A} \in \mathbb{R}^{C\times H\times W}$, is constructed by the tensor product of $\boldsymbol{A}^{\text{ch}}$ and $\boldsymbol{A}^{\text{sp}}$. The final output of the JPCA module is calculated by the following attention process:

$$\boldsymbol{F}' = \boldsymbol{F} \odot \boldsymbol{A}. \tag{13}$$

There are mainly two sub-modules: physical attention (PA) module driven by shape that calculates $\boldsymbol{A}^{\text{sp}}$ and channel attention (CA) module that calculates $\boldsymbol{A}^{\text{ch}}$.

Shape-driven attention module. The PA module is built upon the torque-based measure proposed in Section 2. The edge maps $\boldsymbol{B}_1, \boldsymbol{B}_2$ (with possible downsampling to match the size of the intermediate results in the encoder/decoder) are used as the input of each PA module. To simulate the process of (10) using K patch scales and M elliptic shapes, we construct two sets of convolution kernels $\{\boldsymbol{H}_{a_i,S_j}\}_{i,j=1}^{M,K}$ and $\{\boldsymbol{V}_{b_i,S_j}\}_{i,j=1}^{M,K}$ of the forms (8) and (9) respectively, where $\{a_i, b_i\}_{i=1}^{M}$ are learnable parameters for the shape of raindrops, and $\{S_j\}_{j=1}^{K}$ are some predefined scales. According to the scale, we group these kernels into $\{\boldsymbol{H}_{a_i,S_1}\}_{i=1}^{M}, \cdots, \{\boldsymbol{H}_{a_i,S_K}\}_{i=1}^{M}$, $\{\boldsymbol{V}_{b_i,S_1}\}_{i=1}^{M}, \cdots, \{\boldsymbol{V}_{b_i,S_K}\}_{i=1}^{M}$. Based on these sets of kernels, we build up a series of convolutional layers $Conv_k^{\text{H}}, Conv_k^{\text{V}} \in \mathbb{R}^{M\times H\times W}$:

$$Conv_k^{\text{H}} : \boldsymbol{X} \to [\boldsymbol{H}_{a_1,S_k} * \boldsymbol{X}; \cdots; \boldsymbol{H}_{a_M,S_k} * \boldsymbol{X}],$$
$$Conv_k^{\text{V}} : \boldsymbol{X} \to [\boldsymbol{V}_{b_1,S_k} * \boldsymbol{X}; \cdots; \boldsymbol{V}_{b_M,S_k} * \boldsymbol{X}],$$

for $k = 1, \cdots, K$. Then we compute

$$\boldsymbol{U}_k = Conv_k^{\text{H}}(\boldsymbol{B}_1) + Conv_k^{\text{V}}(\boldsymbol{B}_2), \tag{14}$$

for all k. The $\boldsymbol{U}_k$s can be viewed as the roundness measure of (10) computed on a series of scales with multiple learnable shape parameters. Next, the shape-driven attention map $\boldsymbol{A}^{\text{sp}}$ is defined as

$$\boldsymbol{A}^{\text{sp}}(h, w) = \max_{k,m} \boldsymbol{U}_k(m, h, w). \tag{15}$$

In other words, the attention $\boldsymbol{A}^{\text{sp}}$ takes the maximal torque-based measure values across different image scales and dif-

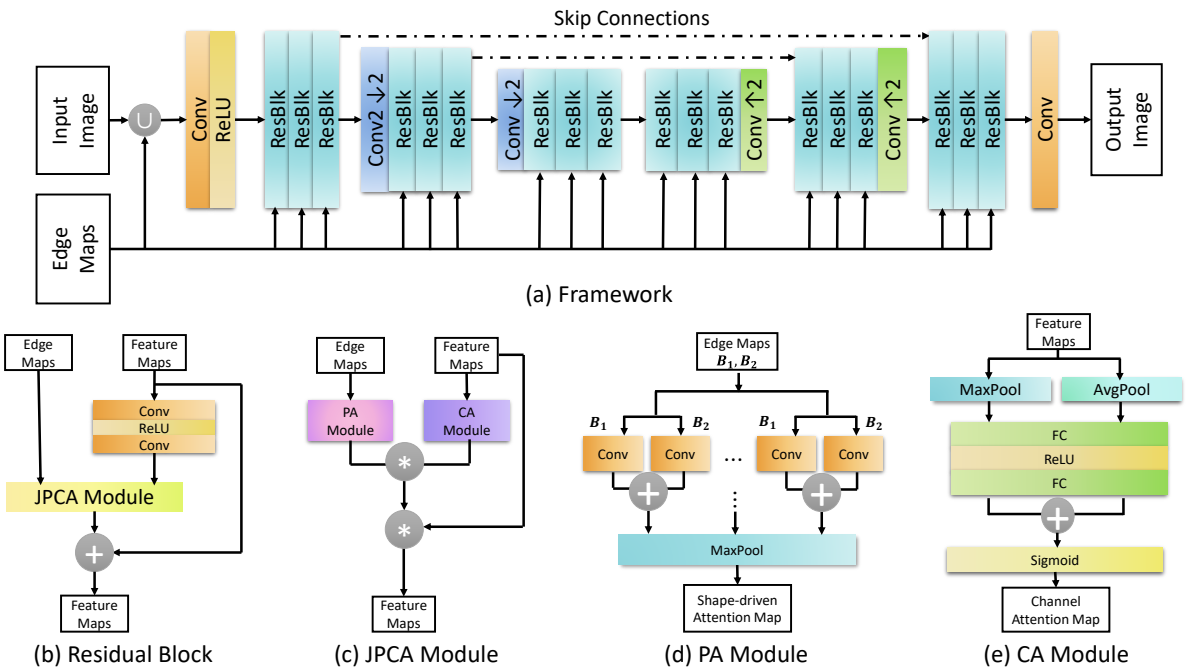


Figure 5: Framework and modules of proposed CNN for image raindrop removal.

ferent elliptic shapes. The structure of the PA module is shown in Fig. 5 (d).

Channel attention module. We implement the CA module as the one proposed in [17], whose structure is shown in Fig. 5 (e). Firstly, the spatial information of the feature map is aggregated into two descriptors by using the average pooling and max-pooling operations respectively. To capture the relationship among channels, both the descriptors are forwarded to a shared multi-layer perceptron (MLP). The MLP has one hidden layer, *i.e.* two fully-connected (FC) layers, with the hidden state size set to $C/8 \times 1 \times 1$ and uses ReLU as the activation function. Finally, the channel attention map is generated by summing up the outputs of the MLP and passing it through the sigmoid activation function.

3.3. Loss function

Given the training image pairs $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$ for $i = 1, \cdots, N$, where $\boldsymbol{Y}_i$ is the image corrupted by raindrops and $\boldsymbol{X}_i$ is the corresponding raindrop-free image. Let $\hat{\boldsymbol{X}}_i$ denote the output of our network when using $(\boldsymbol{X}_i, \boldsymbol{Y}_i)$ as the input. The loss function for training our network is defined as follows:

$$L = \frac{1}{N} \sum_{i=1}^{N} \|\hat{\boldsymbol{X}}_i - \boldsymbol{X}_i\|_1. \tag{16}$$

where $\|\cdot\|_1$ denotes the ℓ_1 distance.

4. Experiments

4.1. Experiment setup

Data preparation. The dataset used in the experiments is based on the one released by Qian *et al.* [12]. The images with raindrops in the dataset [12] are created by the same way as Eigen *et al.* [1]: taking photos before/after spraying water on the glass. Qian *et al.*'s dataset totally contains 1110 corrupted/clean image pairs. Following the same strategy as [12], we used 861 image pairs for training. Image patches of size 256×256 were randomly cropped from the training images as the input/truth data. In the remaining images of the dataset, there are 58 well-aligned image pairs that were picked out for test (same as [12]). Two operations were used for 4x data augmentation: (i) flipping each image horizontally and (ii) resizing each image with ratio 0.6.

Implementation and training details. The proposed CNN is implemented using TensorFlow. All the experiments were run on a PC with Intel(R) Core(TM) i7-8600 CPU and NVIDIA GeForce GTX 2080Ti. Regarding the PA module, we set the patch scales (*i.e.* sizes of convolution kernels) in calculating the torque-based measure to $15\times15, 20\times20, \cdots 40\times40$. At each scale, we used 8 elliptic parameters. In training, we randomly cropped the patches of size 256×256 as the inputs of the CNN. The batch size was set to 10 and the number of epochs was set to 1000. The learning rate was initialized with 10^{-4} and using polynomial decay with power of 0.3. For the optimization of the training loss, we used Adam [7] with default parameters.

Benchmark. The quantitative evaluation is done in terms of three metrics: PSNR and SSIM calculated on the processed image, and PSNR on the luminance (Y channel of YUV space). The last metric is used in [1]. There are few published works for single-image raindrop removal. We compare the proposed one with two methods: Eigen *et al.*'s [1] and AttentGAN [12]. We also include the results, quoted from [12], a general image-to-image translation approach called Pix2Pix [6].

4.2. Results

Quantitative results. Table 1 lists the results of quantitative evaluation. It can be seen that ours outperformed both Eigen *et al.*'s method and Pix2Pix in terms of PSNR and SSIM. Eigen *et al.*'s method performs much worse than other compared methods, as it uses a shallow CNN with only 3 layers. Pix2Pix uses a similar encoder-decoder structure to ours, but without any attention mechanism. It can be seen that our CNN outperforms Pix2Pix with a large margin. Compared to AttentGAN, our method yielded slightly lower PSNR on luminance channel but with both higher PSNR and SSIM on the RGB channel. This implies that our method can have better recovery on both image color and image details. See supplementary materials for an illustration of estimated attention maps.

Qualitative results. See Fig. 6 for the visualization of some results. Limited by its network capacity, Eigen *et al.*'s NN failed to remove raindrops. Pix2Pix removed the raindrops with a moderate amount, but it produced many artifacts on the recovered images. AttentGAN removed most raindrops but it tended to generate artifacts on raindrop regions. In comparison, our method can remove the raindrops with less artifacts, and obtain the best visual quality among the compared methods. As a simple user study, we invited 24 students for the study and 22 of them chose our recovery results as their preferred ones.

Table 1: Results of tested methods on Qian *et al.*'s dataset.

Metric	Eigen *et al.* [1]	Pix2Pix [6]	AttentGAN [12]	Ours
PSNR(dB)	23.74	28.15	30.55	**30.86**
PSNR-L(dB)	23.61	28.02	**31.57**	31.44
SSIM	0.7884	0.8547	0.9023	**0.9263**

Results on real images. We also collected some real-world images via mobile phones for evaluation. The results are shown in Fig. 7 and supplementary materials. It can be seen that our method also works well on real images. Similar to the results of previous experiments, Eigen *et al.* 's method cannot remove raindrops well and even caused image blurring. The images generated by Pix2Pix and AttentGAN are not as clean as ours, and are likely to have more artifacts on raindrop regions. In short, our method outperformend the existing ones in terms of visual quality.

4.3. Ablation study

To validate the effectiveness of the attention mechanism in our CNN, we constructed three baseline models:

- Base-1: Removing the JPCA module from the CNN;
- Base-2: Removing the CA module from the CNN (*i.e.* fixing the output of CA module to be 0.5);
- Base-3: Removing the PA module from the CNN (*i.e.* fixing the output of PA module as well as the input edge maps to be 0.5).

These baseline models were retrained as the original one, with parameters tuned up for fair comparison.

Table 2 shows the comparison of the proposed CNN with the baseline models on Qian *et al.*'s dataset. The rank of PSNR/SSIM is ours, Base-3, Base-2, Base-1. By comparing the Base-1 with ours, we can find that, adding the JPCA module to network can bring noticeable performance improvement, *e.g.* the PSNR improvement of 0.95dB is observed. It can be also seen that both the PA module and the CA module have contributions to the performance improvement of the network. The improvement of CA module is larger than the PA module. Such results have demonstrated the effectiveness of each module in our network. We also tested the performance of our CNN while fixing the shape parameters to be ones. The performance was decreased by 0.21dB. This demonstrated that making the shape parameters learnable and adaptive to data can benefit the results.

Table 2: Results of baseline models on Qian *et al.*'s dataset.

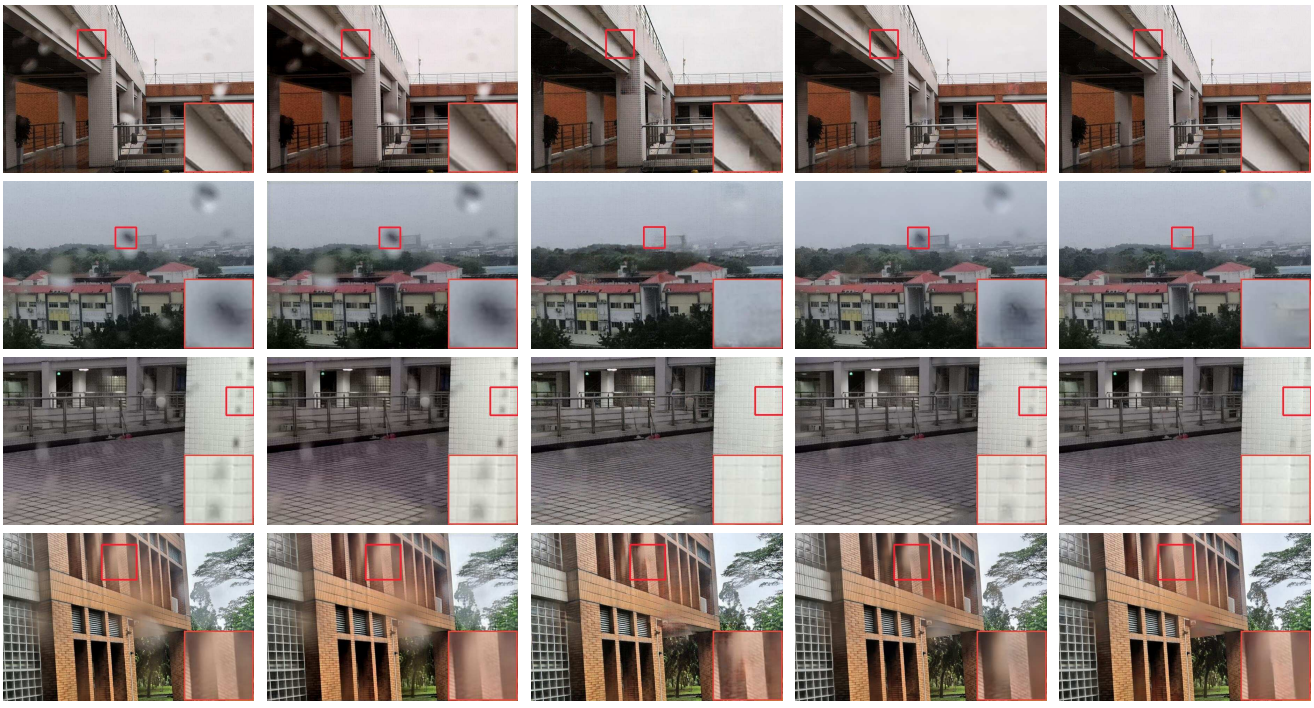
Method	PA	CA	PSNR(dB)	SSIM
Base-1	×	×	29.91	0.8967
Base-2	✓	×	30.08	0.9154
Base-3	×	✓	30.58	0.9224
Ours	✓	✓	**30.86**	**0.9263**

5. Summary

A CNN-based method is proposed in this paper for solving one challenging problem arising from computer vision in bad weather, *i.e.*, how to restore an image taken through glass window in rainy weather. The proposed method introduced a joint shape-channel attention mechanism, in which the shape-driven attention exploits the physical shape properties of raindrops, including closedness and roundness, and the channel attention refines the features relevant to the background layer or the raindrop layer. The experiments showed that the proposed method outperformed the existing ones in terms of both quantitative metrics and qualitative inspections.

(a) Input image (b) Truth image (c) Eigen *et al.* [1] (d) Pix2Pix [6] (e) AttentGAN [12] (f) Ours

Figure 6: Visualization of the results of different methods on Qian *et al.*'s dataset.



(a) Input image (b) Eigen *et al.* [1] (c) Pix2Pix [6] (d) AttentGAN [12] (e) Ours

Figure 7: Visualization of the results of different methods on real-world images.

References

[1] David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proc. Int. Conf. Comput. Vision*, pages 633–640, 2013. 2, 6, 7, 8

[2] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Huang, and Xinghao Ding. Residual-guide network for single image deraining. In *Proc. ACM Multimedia Conf.*, pages 1751–1759. ACM, 2018. 2, 3, 5

[3] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Trans. Image Proc.*, 26(6):2944–2956, 2017. 3

[4] 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. IEEE Conf. Comput. Vision Pattern Recognition*, pages 1715–1723, 2017. 3

[5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pages 770–778, 2016. 5

[6] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A. Efros. Image-to-image translation with conditional adversarial networks. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pages 5967–5976. IEEE, 2017. 3, 7, 8

[7] Diederik P. Kingma and Jimmy Ba. Adam: a method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[8] Hiroyuki Kurihata, Tomokazu Takahashi, Ichiro Ide, Yoshito Mekada, Hiroshi Murase, Yukimasa Tamatsu, and Takayuki Miyahara. Rainy weather recognition from in-vehicle camera images for driver assistance. In *Proc. Int. Intell. Vehicles Symp.*, pages 205–210. IEEE, 2005. 2

[9] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proc. European Conf. Comput. Vision*, pages 254–269, 2018. 3

[10] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *Proc. Int. Conf. Comput. Vision*, pages 3397–3405, 2015. 2, 3

[11] Morimichi Nishigaki, Cornelia Fermüller, and Daniel Dementhon. The image torque operator: A new tool for mid-level vision. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pages 502–509. IEEE, 2012. 3

[12] Rui Qian, Robby T. Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pages 2482–2491, 2018. 1, 2, 3, 5, 6, 7, 8

[13] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: convolutional networks for biomedical image segmentation. In *Proc. Int. Conf. Medical Image Compt. & Computer-Assisted Intervention*, pages 234–241. Springer, 2015. 5

[14] Martin Roser and Andreas Geiger. Video-based raindrop detection for improved image registration. In *Workshop Int. Conf. Comput. Vision*. IEEE, 2009. 2

[15] Martin Roser, Julian Kurz, and Andreas Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. In *Proc. Asian Conf. Comput. Vision*, pages 235–244. Springer, 2010. 2

[16] Ching Teo, Cornelia Fermuller, and Yiannis Aloimonos. Fast 2d border ownership assignment. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pages 5117–5125, 2015. 3

[17] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: convolutional block attention module. In *Proc. European Conf. Comput. Vision*, pages 3–19, 2018. 6

[18] Yong Xu, Yuhui Quan, Zhuming Zhang, Hui Ji, Cornelia Fermüller, Morimichi Nishigaki, and Daniel Dementhon. Contour-based recognition. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pages 3402–3409. IEEE, 2012. 3

[19] A. Yamashita, T. Harada, T. Kaneko, and K. T. Miura. Removal of adherent noises from images of dynamic scenes by using a pan-tilt camera. In *Proc. Int. Conf. Intell. Robots Syst.* IEEE, 2004. 2

[20] A. Yamashita, M. Kuramoto, T. Kaneko, and K. T. Miura. A virtual wiper - restoration of deteriorated images by using multiple cameras. In *Proc. Int. Conf. Intell. Robots Syst.* IEEE, 2003. 2

[21] A. Yamashita, Y. Tanaka, and T. Kaneko. Removal of adherent waterdrops from images acquired with stereo camera. In *Proc. Int. Conf. Intell. Robots Syst.* IEEE, 2005. 2

[22] 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. IEEE Conf. Comput. Vision Pattern Recognition*, pages 1357–1366, 2017. 3

[23] Shaodi You, Robby T. Tan, Rei Kawakami, and Katsushi Ikeuchi. Adherent raindrop detection and removal in video. In *Proc. IEEE Conf. Comput. Vision Pattern Recognition*, pages 1035–1042, 2013. 1, 2, 3, 4