Image Desnowing via Deep Invertible Separation

Yuhui Quan, Xiaoheng Tan, Yan Huang*, Yong Xu and Hui Ji

***Abstract*—Images taken on snowy days often suffer from severe negative visual effects caused by snowflakes. The task of removing snowflakes from a snowy image is known as image desnowing, which is challenging as image details are easily mistakenly treated and thus may be significantly lost during snowflake removal. Leveraging invertible neural networks (INNs), this paper presents a deep learning-based method for single image desnowing, which can remove snowflakes accurately while preserving image details well. Interpreting desnowing as an image decomposition problem, we propose an INN composed of two asymmetric interactive paths for predicting a latent image and a snowflake layer respectively. Such an INN is able to progressively refine the features of both latent images and snowflake layers for disentanglement, while retaining all information possibly relevant to latent image reconstruction. In addition, an attentive coupling layer supervised by snowflake masks is introduced to enhance feature dismantlement and a coupling-in-coupling structure is developed for further improvement. Extensive experiments show that, the proposed method outperforms existing ones on three benchmark datasets of synthetic and real-world images, and meanwhile it also shows advantages in terms of model size and computational efficiency.**

***Index Terms*—Image desnowing, Invertible neural networks, Image separation, Deep learning.**

I. Introduction

SNOW is a type of bad weather often seen in cold regions at high altitudes and latitudes. Images taken on snowy days are usually corrupted by falling snow in the form of snowflakes; see Fig. 1 for some examples. The visual effects caused by snowflakes not only affect the visual aesthetics and perception of an image negatively, but also decrease the performance of many downstream computer vision systems that are sensitive to occlusion, *e.g.*, object tracking [1] and video surveillance [2]. Image desnowing aims at removing snowflakes from a snowy image while preserving all image details. Such a technique can see its applications in digital photography and computer vision.

The formation of an image with snowflakes is complicated, and there have been different formation models proposed in existing studies; see *e.g.* [3]–[6]. In general, an observed snowflake-corrupted image $\boldsymbol{I}_{\mathrm{s}}$ is composed of the latent image $\boldsymbol{I}_{\mathrm{c}}$ desired for applications and the corresponding snowflake layer $\boldsymbol{S}$ containing all intensity (color) fluctuations caused by snowflakes [3]. Single image desnowing is about recovering $\boldsymbol{I}_{\mathrm{c}}$ given only $\boldsymbol{I}_{\mathrm{s}}$. It is a challenging inverse problem which needs to locate all snowflakes accurately and recover all those pixels occluded by snowflakes, while no other cues except the single input snowy image itself can be exploited.

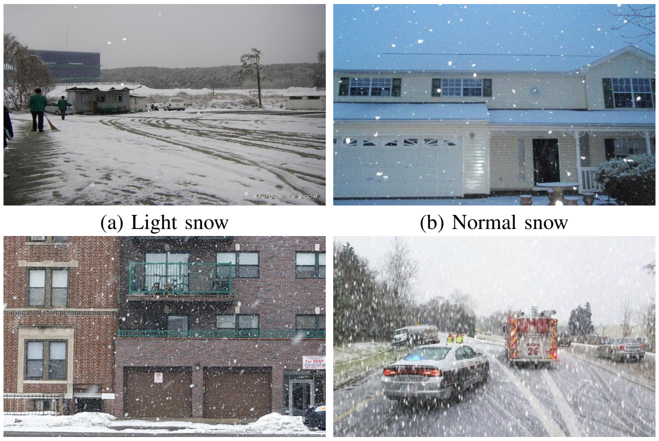

(a) Light snow
(b) Normal snow
(c) Dense snow
(d) Heavy snow

Fig. 1. Examples of images taken on snowy days.

In the past, many methods (*e.g.* [7]–[11]) have been proposed for single image desnowing, which handcrafted some priors of snowflakes for snowflake detection and imposed priors on latent images for recovering occluded image pixels. As shown in Fig. 1, the physical properties of snowflakes vary greatly in density, size, shape and translucency from image to image, and even within the same image. For instance, some snowflakes in motion even look more like streaks than flakes. In addition, images of natural scenes may differ significantly in appearance. Therefore, pre-defined priors on snowflakes or images are usually too simple to handle snowy images of complex scenes, and not adaptive to different images either.

In recent years, deep learning has become one promising tool for single image desnowing; see *e.g.* [4]–[6], [12]–[15]. These deep learning-based methods train an end-to-end deep neural network (DNN) that maps an image with snowflakes to its corresponding snowflakes-free latent image. Although these methods bring significant performance gains over traditional methods, there is still much room for further improvement. See Fig. 2 for an illustration. We can observe that it is still difficult for existing methods to eliminate all visual effects of snowflakes while preserving all image details. There is still the need for developing desnowing methods that can win the both in snowflake removal and image detail preservation.

A. Motivation and Main Idea

One approach to designing a DNN for single image desnowing, *e.g.* [14], is to follow a pipeline of snowflake detection and removal. However, the interaction between snowflake detection

Y. Quan, X. Tan and Y. Xu are with School of Computer Science and Engineering at South China University of Technology, Guangzhou 510000, China, as well as with Pazhou Laboratory, Guagnzhou 510335, China.

Y. Huang is with School of Computer Science and Engineering at South China University of Technology, Guangzhou 510000, China.

H. Ji is with Department of Mathematics at National University of Singapore 119076, Singapore.

*Corresponding author: Yan Huang (email: aihuangy@scut.edu.cn).

This work was supported in part by National Natural Science Foundation of China under Grants 61902130 and 61872151, in part by Natural Science Foundation of Guangdong Province under Grant 2022A1515011755, and in part by Singapore MOE AcRF under Grant R-146-000-315-114.

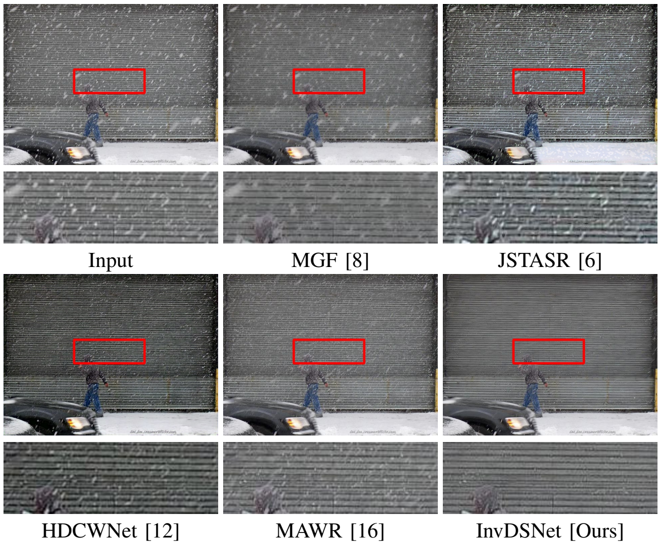


Fig. 2. Image desnowing results of different methods.

and snowflake removal is limited in such a two-stage scheme. A better approach is to interpret image desnowing as an end-to-end image separation procedure, *i.e.*, separating the desired snowflake-less image from the input one. Existing studies such as [4], [5], [13] usually implemented such an image separation process by progressively encoding an input image into latent image-related features while discarding snowflake-related features, and then reconstructing the latent image by decoding its corresponding features. Though the typical DNNs used in existing works (*e.g.* U-Net [4], [5] and ResNet [13]) may be able to extract rich features related to latent images, there is no guarantee that these features provide complete information for recovering all image details. As a result, the possible loss of information in the encoding stage will have negative impacts on the reconstruction in the decoding stage. Even for the decoding stage, it may also omit important image features for reconstruction.

To address the issues above, this paper interprets single image desnowing as a progressive disentanglement and decomposition process and leverages invertible coupling layers [17] which guarantee that no information will be lost during feature extraction and disentanglement. Based on coupling layers, we propose a dual-path invertible neural network (INN), where one path is used for snowflake layer extraction and the other for latent image prediction; see Fig. 3 for an illustration. In comparison to existing DNNs for image desnowing, using coupling layers as the building blocks in our method not only ensures the information-losslessness of feature extraction to benefit the reconstruction process, but also enables rich interactions between the two paths for feature disentanglement. In addition, since an INN in its reverse mode can replicate its input using its output, our approach can be viewed as implicitly learning a formation model for snowflake-corrupted images. Together with a reverse reconstruction loss, such an implicitly learned formation model can bring further regularization.

There are another two techniques introduced in the proposed dual-path INN for further performance improvement. One is a coupling-in-coupling structure where coupling layers are not only between the two paths in the INN, but also used inside the image path for better extracting image features. The other is an attentive coupling layer supervised by snow masks, which provides additional guidance for the identification of regions of snowflakes for better feature disentanglement. Benefiting from all its techniques, the proposed INN provides a computationally efficient, light-weight, yet effective solution to single image desnowing, with better performance than existing ones on three datasets of both synthetic and real-world snowy images. See Fig. 2 for visual illustration of one example.

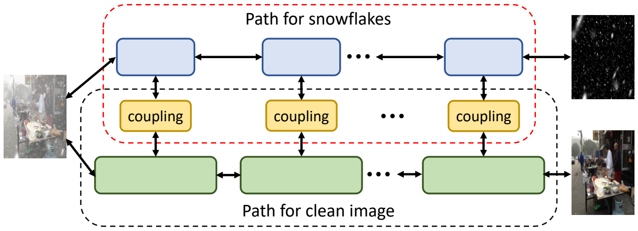


Fig. 3. Basic idea of proposed INN for single image desnowing. The two paths are asymmetric and interacting via feature fusion with a coupling process.

B. Contributions

To summarize, the contributions of this paper are three-fold.

- We exploit deep invertible representation for single image desnowing, by which information losslessness is achieved during feature disentanglement for better preserving image details during snowflake removal.
- A dual-path attentive INN with a coupling-in-coupling structure is proposed for image desnowing, which shows improvement over both standard INNs for image processing and existing non-invertible DNNs for desnowing.
- An effective solution to image desnowing with state-of-the-art performance, which shows its advantages over existing ones in visual quality, quantitative performance, and computational efficiency.

II. Related Work

A. Image Desnowing with Handcrafted Filters

Traditional methods for single image desnowing are usually based on a handcrafted filtering process. Xu *et al.* [18], [19] applied the filtering guided by an edge map. Zheng *et al.* [8] improved guided filtering using multiple clues for guidance. Wang *et al.* [10] improved guided filtering based on the principal direction of a patch and the sensitivity of variance of color channels. Sparse coding is another often-used technique for desnowing. Rajderkar *et al.* [20] modeled a latent image and a snowflake layer by sparse representation on two dictionaries respectively. The latent image is recovered via sparse coding on the high-frequency image parts extracted by bilateral filtering. Ding *et al.* [9] proposed to obtain a coarse desnowed image via a guided ℓ_0 smoothing filter and refine it via sparse regularization. Some methods use a two-stage scheme: snowflake detection followed by image inpainting (or called image error concealment [21]). Pei *et al.* [7] proposed to detect snowflakes via thresholding high-frequency image parts and remove the snowflakes via inpainting occluded pixels. Huang *et al.* [11] proposed to detect snowflakes with adaptive median filtering. The latent image is then recovered via sparse coding with spatially-varying reconstruction tolerances estimated by a particle swarm algorithm.

All above methods are based on certain predefined priors which are not adaptive to input images. Therefore, they do not perform satisfactorily when there are large variations in the appearance of snowflakes and the contents of natural scenes.

B. Image Desnowing via Deep Learning

In recent years, deep learning has been exploited for image desnowing. One early work is Liu *et al.* [4] which proposed a multi-stage DNN with two sequential modules: one for recovering the areas obscured by translucent snowflakes, and the other for estimating the remaining parts occluded by opaque snowflakes. In addition, Liu *et al.* [4] released a dataset for performance benchmarking. Li *et al.* [14] also built up a dataset, and they constructed a DNN by sequentially connecting a multi-scale convolutional neural network (CNN) used for feature extraction and two stacked modified DenseNets [22] used for snowflake detection and removal respectively. Li *et al.* [15] proposed a generative adversarial network (GAN) composed of a generator for latent images, a generator for snowflake layers, and a discriminator for adversarial training. Jaw *et al.* [13] also proposed a GAN for desnowing, where the generator is a pyramidal hierarchical CNN with lateral connections across different scales for more propositional information and less computational time. Chen *et al.* [23] proposed a DNN consisting of three parts: one part to predict a snowflake mask by self-pixel and cross-pixel attention, one part to remove snowflakes by the guidance of the predicted snow mask, and one part to remove veiling effects.

Chen *et al.* [6] considered veiling effects in the formation model of snowy images, based on which they designed a GAN with three stages: removing veiling effects using an atmospheric light prediction module motivated from dark channel prior [24], identifying snowflakes using three different-scale sub-networks with multi-scale convolutional layers, and removing snowflakes with partial convolutions. Further, they contributed a benchmark dataset using their proposed image formation model. More recently, Chen *et al.* [12] proposed another dataset by considering the effects of streak-like snowflakes. They proposed a hierarchical DNN [12] based on dual-tree complex wavelet representation, with a contradictory channel loss related to dark channel prior [24] and bright channel prior [25] for image desnowing.

C. Video Desnowing and Related Techniques

In addition to single image desnowing, there are also some works on desnowing a sequence of video frames; see *e.g.* [2], [3], [26]–[30]. Video desnowing is relatively easier, as the spatial-temporal redundancy among video frames provides more cues for detecting snowflakes and for recovering latent frames. Exploitation of such spatial-temporal cues is the focus of video desnowing. For instance, the spatial-temporal redundancy is exploited via statistical appearance models of snow-streaks or snowflakes in [26], [27], [30], sparse representation of consecutive frames in [2], [3], and low-rank approximation of adjacent clean image layers in [28], [29]. These methods cannot be directly called for single image desnowing. It is worth mentioning that an asymmetric dual-path structure is also used in the Slowfast network [31], a non-invertible DNN to extract information from different temporal speeds for video processing, where one path captures semantic information and the other captures motion information. In comparison, our INN extracts different layers from the image, with one path for the snowflake layer and the other for the latent image layer.

D. Recovering Images Taken under Other Bad Weathers

In addition to image desnowing, there are extensive studies on recovering images taken under other bad weathers. Deep learning also has been widely used for such image recovery tasks; see *e.g.* [32]–[39]. Among them, the task of image raindrop removal from an image [33]–[36] shares similarities with snowflake removal from an image. There are also some studies on developing a universal model for recovering images taken under different bad weathers, *e.g.* [5], [16]. While these methods can be applied for image desnowing with minor modifications, they are not specifically designed for single image desnowing. As a result, they do not perform as well as the ones specifically designed for image desnowing, as shown in existing studies (*e.g.* [12]). In addition, compared to the DNNs used in these existing works which do not guarantee the information-losslessness over different network stages, ours is invertible with no information loss across network stages. Such a property allows our DNN to remove snowflakes more effectively while preserving image details better than these DNNs used in other tasks.

E. Invertible Neural Networks for Image Processing

INNs have been exploited in many image processing tasks; see *e.g.*, image rescaling [40], hiding [41], denoising [42], super-resolution [43], inpainting [44], smoothing [45], decolorization [46], low-light enhancement [47], and high dynamic range reconstruction [48]. To the best of our knowledge, this work is the first one to exploit INNs for removing particles from images taken under bad weathers.

In most existing works, the INNs for image recovery are used as either a generative model in normalizing flow (*e.g.*, [44]) or a shared-weight auto-encoder with invertibility (*e.g.*, [40], [42], [45]). The latter one is related to our method. The pipeline of these methods usually first applies a forward pass as an encoder to disentangle layers in a latent code space, then removes undesired layers by zeroing their latent codes, and lastly applies a backward process as a decoder to obtain the desired image. Note that once the latent code is mistakenly zeroed, the related important image information cannot be retrieved back in the backward process. In comparison, the INN we propose for desnowing does not explicitly decompose latent codes at its intermediate output, but directly maps an input image to two layers. From the perspective of DNN design, such an architecture is more efficient in feature expression. In addition, some improved INN blocks are proposed in our method for further improvement.

III. Basic Building Blocks

This section describes the basic building blocks employed in the proposed INN for image desnowing. INNs are first proposed for unsupervised learning to model complex data distributions by simple mean-field distributions of exponential family under some transform [17], [49]. An INN $f:\mathbb{R}^N\rightarrow\mathbb{R}^N$ can be viewed as a deep bijective (*i.e.* one-to-one) mapping whose inverse $f^{-1}:\mathbb{R}^N\rightarrow\mathbb{R}^N$ can be directly called by a backward process. In other words, an INN provides a forward process to transform an input to some output, which can replicate the input via a reverse mode of the network.

There are different architectures for an INN. In this paper, we consider the so-called coupling layers [17] for constructing the INNs used for image desnowing. See Fig. 4 for the diagram of a standard coupling layer, which can be used in either the forward mode or the reverse mode. In the forward mode, the coupling layer first splits the input data $\boldsymbol{X}$ into two parts: $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, then transforms them to $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ respectively, by the learned functions $\mathcal{F}$, $\mathcal{G}$, $\mathcal{H}$ in a coupling way, and finally concatenates $\boldsymbol{Y}_1$ and $\boldsymbol{Y}_2$ as its output $\boldsymbol{Y}$. Such a forward process can be expressed by

$$[\boldsymbol{X}_1,\boldsymbol{X}_2]=\mathrm{split}(\boldsymbol{X}), \tag{1}$$

$$\boldsymbol{Y}_1=\boldsymbol{X}_1+\mathcal{F}(\boldsymbol{X}_2), \tag{2}$$

$$\boldsymbol{Y}_2=\boldsymbol{X}_2\odot\exp(\mathcal{G}(\boldsymbol{Y}_1))+\mathcal{H}(\boldsymbol{Y}_1), \tag{3}$$

$$\boldsymbol{Y}=\mathrm{concat}(\boldsymbol{Y}_1,\boldsymbol{Y}_2), \tag{4}$$

where both the split and concatenation operations are done along the channel dimension. The functions $\mathcal{F},\mathcal{G},\mathcal{H}$ above can be implemented by arbitrary DNN modules for effective feature processing, and the invertibility of the coupling layer is independent from their definitions. The role of (2) and (3) is to introduce complicated nonlinear transforms defined by $\mathcal{F},\mathcal{G},\mathcal{H}$ for generating effective deep features, while guaranteeing the invertibility of the whole process to keep all information. The invertibility of the process comes from that, at each step we keep a copy of one split of input, and then form a new representation by combining its non-linearly transformed result to the other split via a linear form. Then, the other split can be recovered by solving the linear equations with the given split copy. Concretely, the coupling layer in its inverse mode can revert the above process as follows:

$$[\boldsymbol{Y}_1,\boldsymbol{Y}_2]=\mathrm{split}(\boldsymbol{Y}), \tag{5}$$

$$\boldsymbol{X}_2=(\boldsymbol{Y}_2-\mathcal{H}(\boldsymbol{Y}_1))\oslash\exp(\mathcal{G}(\boldsymbol{Y}_1)), \tag{6}$$

$$\boldsymbol{X}_1=\boldsymbol{Y}_1-\mathcal{F}(\boldsymbol{X}_2), \tag{7}$$

$$\boldsymbol{X}=\mathrm{concat}(\boldsymbol{X}_1,\boldsymbol{X}_2), \tag{8}$$

where $\oslash$ denotes element-wise division.

For invertible down-sampling layers, Haar Transform (HT) is an often-used one in INNs. A standard HT layer reduces the spatial size of an input image while increasing the channel number by a stride-2 convolution with four kernels including $\boldsymbol{L}\boldsymbol{L}^\top,\boldsymbol{H}\boldsymbol{L}^\top,\boldsymbol{L}\boldsymbol{H}^\top,\boldsymbol{H}\boldsymbol{H}^\top$, where $\boldsymbol{L}=\frac{1}{\sqrt{2}}[1,1]^\top$ and $\boldsymbol{H}=\frac{1}{\sqrt{2}}[1,-1]^\top$. The low-pass filter $\boldsymbol{L}\boldsymbol{L}^\top$ acts as the average pooling on feature maps while the three high-pass filters capture edge-like information with different orientations. The reverse mode of a HT layer is conducted via Inverse Haar Transformation (IHT), implemented by transposed convolution with the aforementioned four kernels of HT.

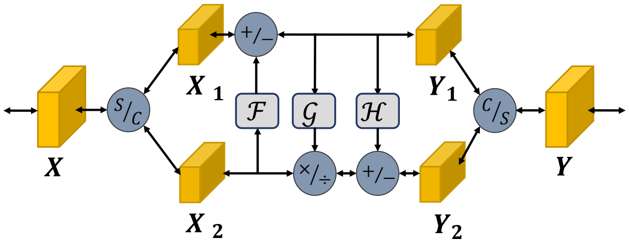


Fig. 4. Structure of a standard coupling layer.

For an INN constructed by coupling layers and HT layers, its forward pass can be done by enabling the forward modes of all these layers. Similarly, its reverse pass can be done by just switching their reverse modes on.

IV. InvDSNet for Single Image desnowing

A. Outline of Architecture

The proposed INN for single image desnowing, named as InvDSNet for convenience, aims at decomposing a snowy image $\boldsymbol{I}_\mathrm{s}\in\mathbb{R}^{H\times W\times 3}$ into the desired latent image $\boldsymbol{I}_\mathrm{c}\in\mathbb{R}^{H\times W\times 3}$ and snowflake layer $\boldsymbol{S}\in\mathbb{R}^{H\times W\times 3}$. See Fig. 5(a) for the outline of our InvDSNet. As an INN is an one-to-one mapping, the InvDSNet takes two copies of the snowy image as its input, making the input and output have the same size.

The InvDSNet is composed of five blocks: two Coupling-in-Coupling (CIC) encoding blocks (EBs), an Attentive Coupling Block (ACB), and two CIC decoding blocks (DBs). That is,

$$\text{CIC-DB}_{\times 2}(\text{ACB}(\text{CIC-EB}_{\times 2}([\boldsymbol{I}_\mathrm{s},\boldsymbol{I}_\mathrm{s}])))\rightarrow[\boldsymbol{I}_\mathrm{c},\boldsymbol{S}], \tag{9}$$

where $_{\times K}$ denotes the repeat of a block for K times. The reverse process is simply done by a backward process:

$$\text{CIC-EB}_{\times 2}(\text{ACB}(\text{CIC-DB}_{\times 2}([\boldsymbol{I}_\mathrm{c},\boldsymbol{S}])))\rightarrow[\boldsymbol{I}_\mathrm{s},\boldsymbol{I}_\mathrm{s}], \tag{10}$$

with the reverse modes on all modules turned on.

Each block in InvDSNet has two inputs and also two outputs, with two processing paths. In other words, the InvDSNet is a dual-path DNN with two paths for separating the latent image and the snowflake layer respectively from the snowy image. Based on coupling layers, the features along one path interact with that of the other path, and carry all information fed from previous blocks through next blocks. The InvDSNet can also be interpreted from the perspective of multi-task learning where one task for extracting the latent image layer and the other for extracting the snowflake layer. Such two tasks are highly correlated to each other with a strong constraint, *i.e.*, the composition of these two layers should replicate the input image. Such a constraint between two tasks is achieved by the invertibility of InvDSNet.

B. Coupling-in-Coupling Block

As shown in Fig. 5(a), a CIC EB takes two feature tensors as input, one for the latent clean image and the other for the snowflake layer. As a snowflake layer usually has much lower

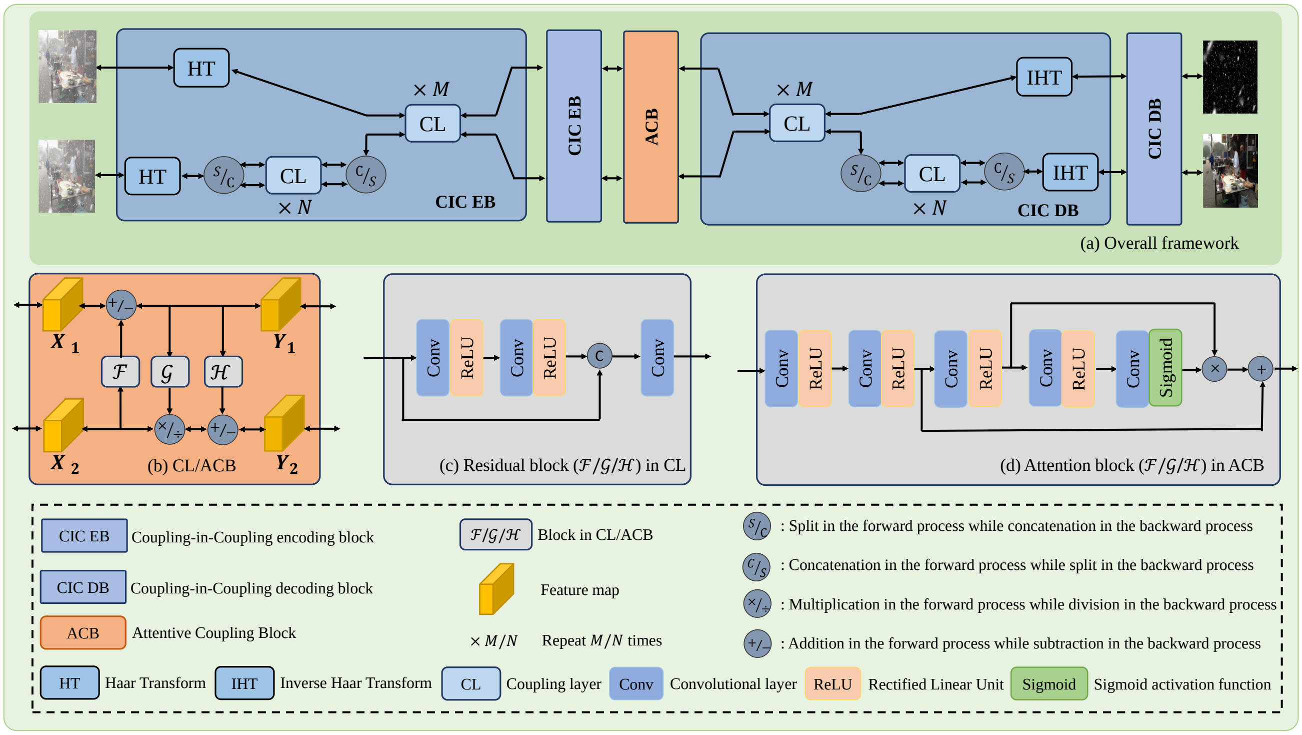
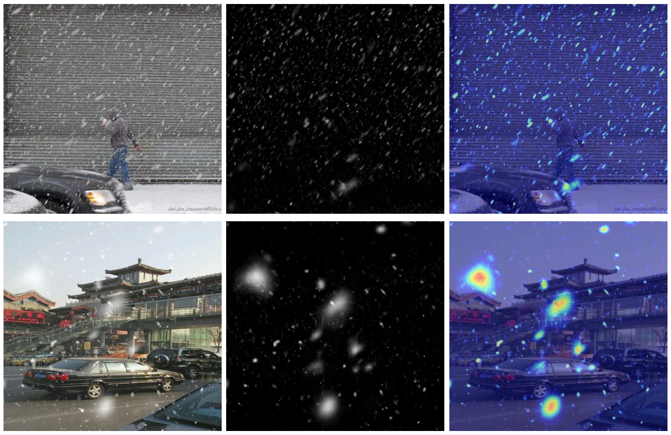

HT
× M
CL
HT
S/C
CL
C/S
× N
CIC EB
CIC EB
ACB
IHT
× M
CL
S/C
CL
C/S
IHT
× N
CIC DB
CIC DB
(a) Overall framework
+/−
X 1
Y 1
F
G
H
X 2
×/÷
+/−
Y 2
(b) CL/ACB
Conv
ReLU
Conv
ReLU
C
Conv
(c) Residual block (F/G/H) in CL
Conv
ReLU
Conv
ReLU
Conv
ReLU
Conv
ReLU
Conv
Sigmoid
×
+
(d) Attention block (F/G/H) in ACB
CIC EB Coupling-in-Coupling encoding block
CIC DB Coupling-in-Coupling decoding block
ACB Attentive Coupling Block
F/G/H Block in CL/ACB
Feature map
× M/N Repeat M/N times
S/C : Split in the forward process while concatenation in the backward process
C/S : Concatenation in the forward process while split in the backward process
×/÷ : Multiplication in the forward process while division in the backward process
+/− : Addition in the forward process while subtraction in the backward process
HT Haar Transform
IHT Inverse Haar Transform
CL Coupling layer
Conv Convolutional layer
ReLU Rectified Linear Unit
Sigmoid Sigmoid activation function

Fig. 5. Outline of proposed InvDSNet for single image desnowing.

complexity than a clean image in terms of image content, each CIC EB processes its two inputs using two paths of different lengths respectively. The shorter path, which is for the snowflake layer, contains a HT layer for downsizing and successive M coupling layers for feature processing. The longer path, which is for the latent image, contains a HT layer and successive $N + M$ coupling layers. Here the M successive coupling layers are shared between the two paths for progressive feature fusion and refinement. The additional N coupling layers in the path for latent images aim at improving the effectiveness of image feature extraction for the separation process. Note that as each of the successive M or N coupling layers uses the same split ratio in InvDSNet, the pair of the concatenation operation in the previous coupling layer and the split operation in the current coupling layer equals to an identity mapping, which are thus omitted in Fig. 5(b). The split ratio is set to $1/3$ for the successive N coupling layers and $1/1$ for the successive M coupling layers. Regarding the functions $\mathcal{F}, \mathcal{G}, \mathcal{H}$ in each coupling layer, we use the residual block [50] shown in Fig. 5(c) for facilitating training and enhancing feature flow.

C. Attentive Coupling Block

Attention mechanism [38], [51], [52] is introduced to improve the efficiency of snowflake layer separation in the latent feature space, which is implemented by inserting an ACB in the middle layer of InvDSNet. An ACB is a coupling layer for ensuring the invertibility of InvDSNet, but with different definitions of $\mathcal{F}, \mathcal{G}, \mathcal{H}$ from the ones used in CIC blocks; see Fig. 5(b)(c)(d). More specifically, the $\mathcal{F}, \mathcal{G}, \mathcal{H}$ in ACB share the same structure, which consists of four consecutive Conv+ReLU layers and a Conv+Sigmoid layer. The third Conv+ReLU layer outputs a feature tensor denoted by $\boldsymbol{F}$, which is passed to the subsequent layers to form an attention map $\boldsymbol{A}$. The generation of $\boldsymbol{A}$ will be supervised during training; see the next subsection. After $\boldsymbol{A}$ is obtained, the attention is done by outputting an attentive feature tensor $\tilde{\boldsymbol{F}} = \boldsymbol{F} \odot \boldsymbol{A}$. There is also a skip connection to the last layer for better information fusion and propagation. See Fig. 6 for the visualization of attention maps generated in the ACB, where the snowflakes with various transparencies, sizes and shapes can be captured by the generated attention maps. As



(a) Snowy image (b) Attention map $\boldsymbol{A}_{\mathcal{F}}$ (c) Heat map

Fig. 6. Attention map generated by ACB in InvDSNet. The heat map is plotted by highlighting the regions of the snowy image according to the attention map.

snowflakes are nearly white in each channel, for the efficiency on memory usage and computing, the ACB only predicts a single-channel attention map and uses it for all channels.

D. Training Loss

Let $(\boldsymbol{I}_{\text{s}}, \boldsymbol{I}_{\text{c}}, \boldsymbol{S})$ denote a triplet of a snowy image, its latent image and its snowflake layer in training data. Recall that an INN has both the forward and reverse modes. The training scheme of InvDSNet utilizes its invertibility to measure the errors in both the forward separation process and the reverse reconstruction process. Such a scheme allows us exploit the supervision from not only $\boldsymbol{I}_{\text{c}}$, but also $\boldsymbol{S}$ and $\boldsymbol{I}_{\text{s}}$, for reducing possible overfitting. Accordingly, the total loss function for training is defined in a two-way manner which includes a forward loss $\mathcal{L}_{\text{forward}}$ and a reverse loss $\mathcal{L}_{\text{reverse}}$:

$$\mathcal{L}_{\text{total}} := \mathcal{L}_{\text{forward}} + \mathcal{L}_{\text{reverse}}. \tag{11}$$

1) Forward loss: The forward loss function is defined by

$$\mathcal{L}_{\text{forward}} := \mathcal{L}_{\text{sep}} + \lambda_{\text{f}}\mathcal{L}_{\text{att-f}}, \tag{12}$$

where the separation loss $\mathcal{L}_{\text{sep}}$ measures the prediction accuracy on the latent image and snowflake layer from an input snowy image, the forward attention loss $\mathcal{L}_{\text{att-f}}$ measures the quality of the generated snow attention maps in the forward process, and the weight $\lambda_{\text{f}} \in \mathbb{R}^+$ balances the two terms. Let $\boldsymbol{I}'_{\text{c}}, \boldsymbol{S}'$ denote the estimates of the latent image and the snowflake layer in the forward process respectively, predicted from $\boldsymbol{I}_{\text{s}}$ using InvDSNet. Then we define

$$\mathcal{L}_{\text{sep}} = \|\boldsymbol{I}'_{\text{c}} - \boldsymbol{I}_{\text{c}}\|_2 + \beta\|\boldsymbol{S}' - \boldsymbol{S}\|_2, \tag{13}$$

with a weight $\beta \in \mathbb{R}^+$. As for the forward attention loss, recall that there are three attention maps denoted by $\boldsymbol{A}_{\mathcal{F}}, \boldsymbol{A}_{\mathcal{G}}, \boldsymbol{A}_{\mathcal{H}}$ generated in the ACB, and we define

$$\mathcal{L}_{\text{att-f}} = \|\boldsymbol{A}_{\mathcal{F}} - \boldsymbol{S}\|_2 + \|\boldsymbol{A}_{\mathcal{G}} - \boldsymbol{S}\|_2 + \|\boldsymbol{A}_{\mathcal{H}} - \boldsymbol{S}\|_2. \tag{14}$$

2) Reverse loss: The reverse loss function is defined by

$$\mathcal{L}_{\text{reverse}} := \mathcal{L}_{\text{recon}} + \lambda_{\text{r}}\mathcal{L}_{\text{att-r}}, \tag{15}$$

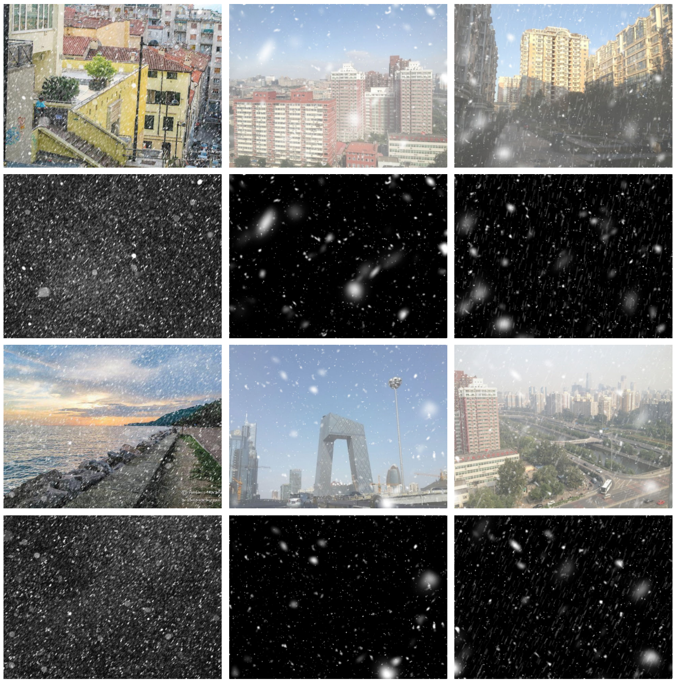
with a reconstruction loss $\mathcal{L}_{\text{recon}}$, a reverse attention loss $\mathcal{L}_{\text{att-r}}$, and a weight $\lambda_{\text{r}} \in \mathbb{R}^+$ set to the same as λ_{f} in practice. Let $\boldsymbol{I}'_{\text{s}}, \boldsymbol{I}''_{\text{s}}$ denote the snowy images reconstructed from two paths respectively, using the reverse mode of InvDSNet fed by the ground-truths $\boldsymbol{I}_{\text{c}}$ and $\boldsymbol{S}$. Note that $\boldsymbol{I}'_{\text{s}}, \boldsymbol{I}''_{\text{s}}$ correspond to the two duplicated inputs for InvDSNet in the forward process, whereas they can be different in the reverse process. The reconstruction loss $\mathcal{L}_{\text{recon}}$ in (15) measures the consistency between the reconstructed snowy images and the original input ones, which is defined as

$$\mathcal{L}_{\text{recon}} = \|\boldsymbol{I}'_{\text{s}} - \boldsymbol{I}_{\text{s}}\|_2 + \gamma\|\boldsymbol{I}''_{\text{s}} - \boldsymbol{I}_{\text{s}}\|_2. \tag{16}$$

The reverse attention loss shares a similar form with the forward one, which is defined based on the attention maps $\bar{\boldsymbol{A}}_{\mathcal{F}}, \bar{\boldsymbol{A}}_{\mathcal{G}}, \bar{\boldsymbol{A}}_{\mathcal{H}}$ generated in the reverse process:

$$\mathcal{L}_{\text{att-r}} = \|\bar{\boldsymbol{A}}_{\mathcal{F}} - \boldsymbol{S}\|_2 + \|\bar{\boldsymbol{A}}_{\mathcal{G}} - \boldsymbol{S}\|_2 + \|\bar{\boldsymbol{A}}_{\mathcal{H}} - \boldsymbol{S}\|_2. \tag{17}$$

It is worth mentioning the reverse process of InvDSNet indeed corresponds to a formation model for images with snowflakes, which may be utilized by the reverse loss for better training.



(a) Snow100K (b) SRRS (c) CSD

Fig. 7. Sample images and corresponding snowflake maps from three datasets.

V. Experiments

The performance of InvDSNet is evaluated on three benchmark datasets which include both synthetic and real snowy images, and it is compared to that of some existing methods of different types.

A. Experimental Settings

1) Datasets: The benchmark datasets for performance evaluation include the Snow100K dataset [4], the dataset of Snow Removal in Realistic Scenario (SRRS) [6], and the Comprehensive Snow Dataset (CSD) [12]. Each dataset contains a large number of training samples, *i.e.*, the triplets consisting of a snowy image, its latent image, and its snowflake layer (map). See Fig. 7 for some sample images of each dataset. The details of these datasets are summarized as follows.

- Snow100K [4]: It is a large dataset with 100000 triplets. The snowflake layers are generated by PhotoShop with variations in density, shape, size, transparency, and moving trajectory. One half of the triplets are used for training, and the other half is used for test. The dataset also provides some real snowy images downloaded via Flicker API for visual evaluation.
- SRRS [6]: This dataset contains 15000 triplets. The snowy images are synthesized with veiling effects, and the snowflakes in synthesis are varied in transparency, size, and moving trajectory. Following the standard protocol provided by the dataset, we use 13000 triplets for training and the rest for test.
- CSD [12]: There are 10000 triplets in this dataset. The synthesized snowflake layers exhibit large variations in the transparency, size and shape of snowflakes, *e.g.*, they may contain streak-like snowflakes. In addition, the

TABLE I
QUANTITATIVE COMPARISON OF DIFFERENT METHODS ON THREE BENCHMARK DATASETS.

Method	Snow100K		Small/Medium/Large	SRRS		CSD		#Params(M)	Time(s)
	PSNR(dB)	SSIM	Snow100K PSNR(dB)	PSNR(dB)	SSIM	PSNR(dB)	SSIM		
MGF [8]	22.41	0.77	24.32/22.99/19.95	15.78	0.74	13.98	0.67	-	-
DesnowNet [4]	30.11	0.93	32.33/30.86/27.16	20.38	0.84	20.13	0.81	15.60	1.38
JSTASR [6]	28.59	0.86	31.40/29.11/25.32	25.82	0.89	27.96	0.88	65.00	0.87
DesnowGAN [13]	31.11	**0.95**	33.43/31.87/28.06	-	-	27.09	0.88	-	-
HDCW-Net [12]	24.10	0.80	24.84/24.75/22.75	27.78	0.92	29.06	0.91	6.99	0.14
ShapeAttention [34]	29.94	0.89	30.93/29.98/28.92	26.56	0.90	27.85	0.88	7.01	0.23
MAWR [16]	-	-	-	-	-	31.35	0.95	28.71	0.39
InvDN [42]	27.99	0.81	28.83/28.44/26.74	26.49	0.88	27.46	0.86	7.26	0.12
InvDSNet [Ours]	**32.41**	0.93	**34.39/33.17/29.69**	**29.25**	**0.95**	**31.85**	**0.96**	**6.94**	**0.06**

synthesis also takes the veiling effects and blurring effects into account. Following the experimental setup of [12], we use 8000 triplets for training and 2000 for test.

2) Methods for comparison: Eight related methods are used for performance comparison, including

- a traditional single image desnowing method: MGF [8];
- four deep learning-based single image desnowing methods: DesnowNet [4], JSTASR [6], DesnowGAN [13] and HDCWNet [12];
- one recent deep learning method for recovering images taken in general bad weather conditions: MAWR [16];
- one deep learning-based method for single image raindrop removal: ShapeAttention [34].
- one efficient INN recently originally proposed for single image denoising: InvDN [42].

The DNN models of the last three methods are retrained on the same training data of image desnowing as ours, with hyper-parameters tuned up. The results of all the compared methods are directly quoted from their corresponding papers whenever available, or obtained by running the codes with recommended parameter settings released in the public domain, using the same experimental data as ours. If neither the results nor the codes are available, we leave the results blank in the tables.

3) Implementation details of InvDSNet: Though all experiments, we set $M = 2$ and $N = 8$ for InvDSNet. In training, all images are randomly cropped into 128×128 patches. The hyper-parameters λ, β and γ of the training loss are set to be 0.05, 0.1 and 0.1 respectively. The model weights are initialized by Xavier. The Adam optimizer is used with batch size 8. The initial learning rate is set to $3e^{-5}$ for the first 100 epochs and $1e^{-5}$ for the last 200 epochs. The InvDSNet is implemented in PyTorch and run on a single NVIDIA GTX 1080 Ti GPU. The code will be made public upon the paper's acceptance.

B. Quantitative Results and Analysis

The quantitative results of the three benchmark datasets are listed in Table I. The effectiveness of desnowing is measured by Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) index. Note that during its construction, the Snow100K dataset is also divided into three subsets according to the size of snowflakes, which are denoted by Snow100K-Small/Medium/Large respectively. Thus, we report the results on Snow100K in terms of both the full test set and the three subsets. It can be seen that InvDSNet is the best performer across all the three datasets in terms of PSNR, with notifiable improvement over the second-best performers. That is, InvDSNet not only outperforms the methods designed for other or general weathers, *e.g.*, ShapeAttention and MAWR, but also performs noticeably better than those designed for single image desnowing, *e.g.*, DesnowNet, JSTASR and HDCW-Net. The SSIM results of InvDSNet are also the highest on CSD and SRRS, while the second highest on Snow100K.

We also compare the complexity of different DNN models in terms of the number of parameters and the running time in processing a 480×640 image on a single RTX 1080Ti GPU. It can be seen that InvDSNet has the smallest number of model parameters among all the compared methods. This not only implies that InvDSNet consumes less memory than other compared models during test, but also indicates that the performance gain of InvNet is not from enlarging the model, but from the architecture design. In addition, our InvNet also enjoys the fastest running speed. Particularly, its running time is less than one half of that of its best competitor, HDCW-Net. All above results have shown the advantages of InvDSNet in both desnowing performance and computational complexity.

To further analyze the performance of InvDSNet in terms of image degradation caused by light snow and heavy snow respectively, we redivide the Snow100K-Small dataset according to the density of snowflakes, as we found that it contains a sufficient number of images with light snow. The density of snowflakes is estimated by the ratio of the total area of all snowflake regions over the total number of image pixels. The snowflake regions are obtained by thresholding the ground-truth snowflake map provided by the dataset by the Ostu method [53]. An image is viewed as containing light snow if its estimated snowflake density is less than 30%; otherwise it is viewed as image with heavy snow. As a result, we obtain 5287 images with light snow and 11324 images with heavy snow. We evaluate the performance of InvDSNet on these two subsets respectively, without model retraining, and the results are listed in Table II. The methods with released codes for producing the results are also included for comparison. It can be seen that our model performs the best on both light-snow images and heavy-snow images.

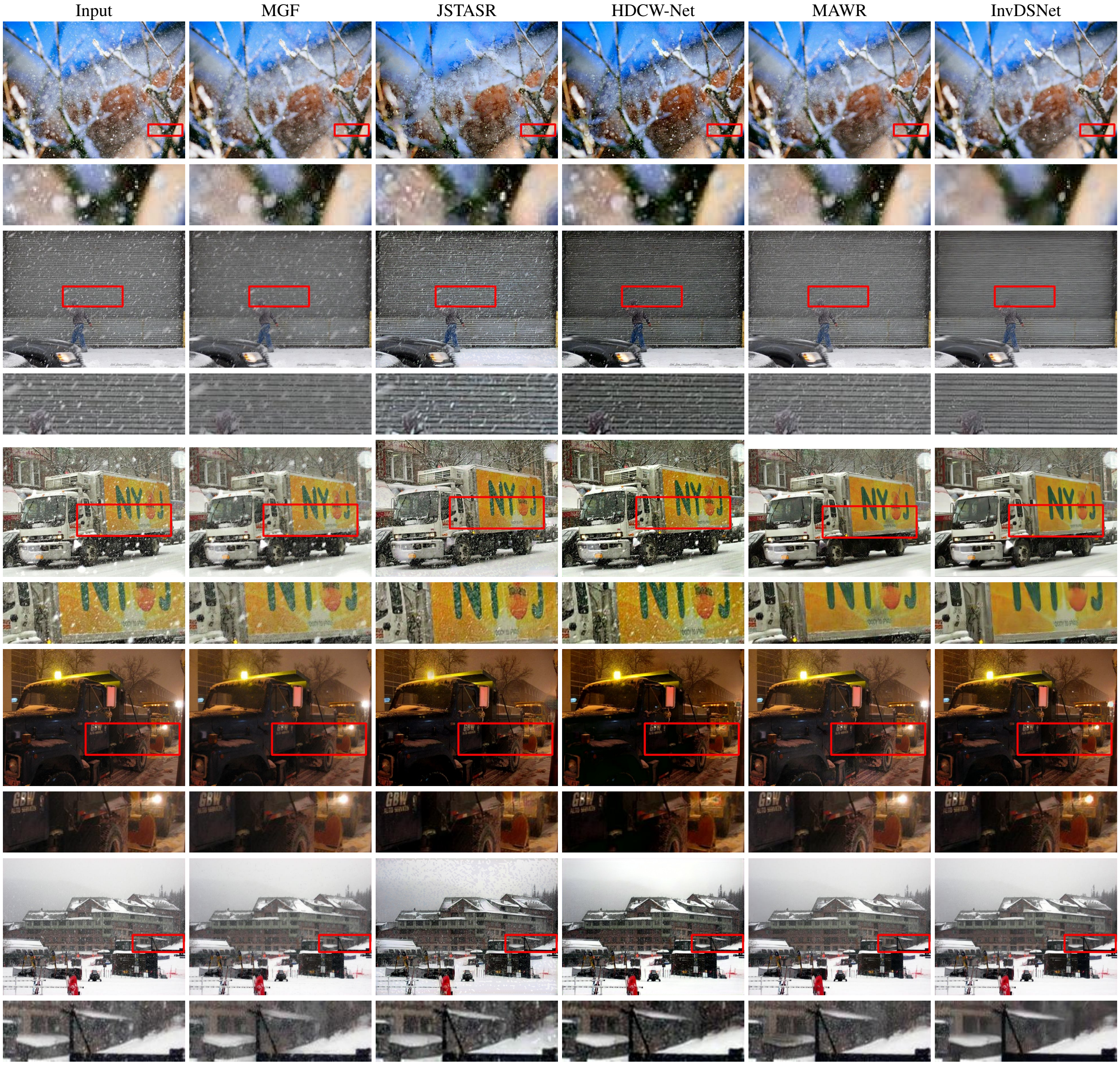


Fig. 8. Visual comparison of desnowed images from selected methods on real-world snowy images from the Snow100K dataset.

TABLE II
RESULTS IN TERMS OF LIGHT SNOW AND HEAVY SNOW ON SNOW100K-SMALL DATASET.

Method	Light snow		Heavy snow	
	PNSR(dB)	SSIM	PNSR(dB)	SSIM
MGF [8]	24.71	0.82	23.96	0.80
HDCW-Net [12]	25.65	0.84	24.46	0.80
InvDN [42]	30.39	0.86	28.22	0.81
InvDSNet [Ours]	**38.73**	**0.98**	**32.27**	**0.92**

C. Qualitative Comparison

The qualitative evaluation is done by visually comparing the desnowed images of different methods in Fig. 8. We select some real-world images from Snow100K for the evaluation. It can be seen that InvDSNet also outperforms other compared methods in terms of visual quality. For instance, in the zoomed-in region of the first sample, other methods cannot remove large snowflakes well, while InvDSNet can do much better. In the second and last samples, other methods cannot well distinguish small snowflakes from bar-like fine textures, leaving a number of snowflakes in the regions with such textures, or even having partial textures smoothed out (*e.g.* MGF). In comparison, our InvDSNet produced a much cleaner yet clear image. In the third sample, all the methods except InvDSNet and MAWR cannot remove the dense snowflakes. Note that MAWR might treat other regions with white color as big snowflakes and thus damaged those regions; see *e.g.* the

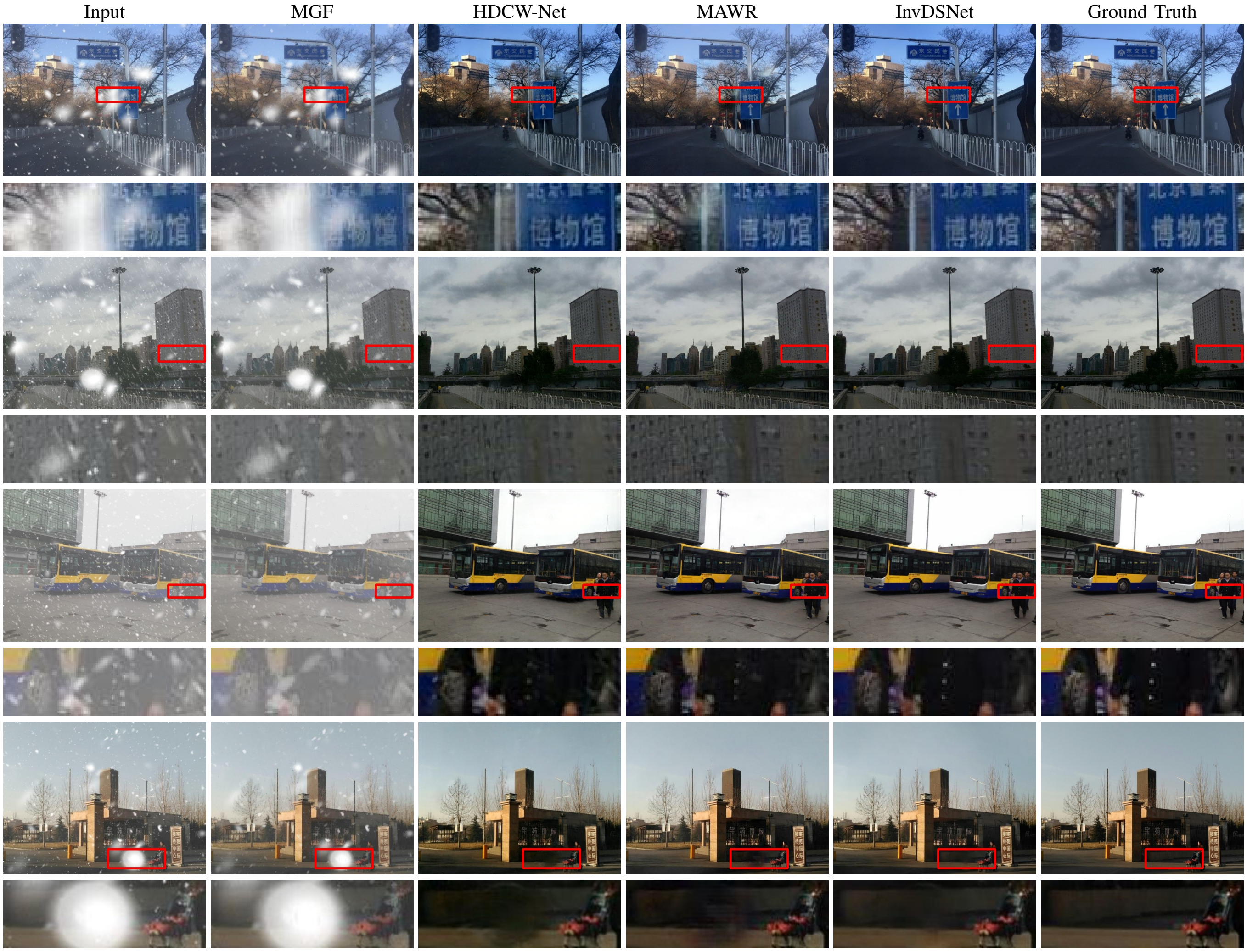


Fig. 9. Visual comparison of desnowed images from selected methods on synthetic snowy images.

bottom of the zoomed-in region. In comparison, InvDSNet not only removed the dense snowflakes but also produced clearer text on the truck. In the forth sample, JSTASR and HDCW-Net darkened the image too much and even removed the halo of the lamp, while MAWR could not remove the snowflakes in the dark. In comparison, InvDSNet not only removed the snowflakes but also preserved the lightness (*e.g.* the halo).

In Fig. 9, we visualize the results from different methods on some synthetic images from the three benchmark datasets, where ground-truth latent images are included for better comparison. Again, we can see that InvDSNet is good at both removing snowflakes and preserving image structures, with better visual quality achieved than other compared methods. Such an improvement on visual quality is consistent with that on quantitative metric. To conclude, InvDSNet is better than other compared methods at removing snowflakes of different shapes, sizes, densities of snowflakes, while preserving image structures and background color well.

D. Ablation Studies

To analyze the contribution of each key component in the proposed InvDSNet, we conduct ablation studies by forming the following baseline models.

- “Plain”: An INN with standard coupling layers is constructed as a baseline, which does not contain any of our proposed blocks. The number of coupling layers is set to 25, which is set to make the resulting model size and network depth as close to our NN as possible. The training is based on a simple forward separation loss.
- “w/o ACB (ACB→CL)”: The ACB is replaced by two coupling layers, which has a similar model size to the original one for a fair comparison.
- “Single path”: Each CIC block is changed to a sequence of $M+N$ coupling layers. It leads to a single-path INN of same size, which maps a snowy image to a clean one.
- “Single path & w/o ACB”: Based on “Single path”, the ACB is further replaced by two coupling layers.
- “w/o HT”: The HT layers in the middle two CIC blocks are removed from InvDSNet.
- “CIC: CL→RB”: The N additional coupling layers in each CIC block are replaced by three residual blocks of 46 channels. The number of residual blocks and the channel number of each residual blocks are set to make the resulting model size and network depth as close to

TABLE III
RESULTS OF ABLATION STUDIES ON CSD.

Model Setting	PSNR(dB)	SSIM
Plain Model	28.56	0.932
w/o ACB (ACB→CL)	31.39	0.948
Single Path	31.12	0.948
Single Path & w/o ACB	29.77	0.939
w/o HT	31.21	0.935
CIC: CL→RB	28.87	0.908
w/o $\mathcal{L}_{\text{reverse}}$	31.43	0.952
w/o $\mathcal{L}_{\text{att}}$	31.34	0.948
w/o $\mathcal{L}_{\text{reverse}}$&$\mathcal{L}_{\text{att}}$	31.02	0.938
Original Model	31.85	0.954

TABLE IV
RESULTS BY SETTING DIFFERENT N AND M ON CSD.

Setting		PSNR(dB)	SSIM	#Parameters(M)
$M = 2$	$N = 6$	31.38	0.952	5.61
	$N = 10$	31.91	0.954	8.26
$N = 8$	$M = 1$	31.03	0.948	5.88
	$M = 3$	31.97	0.955	8.01
$M = 2, N = 8$ (Original)		31.85	0.954	6.94

TABLE V
RESULTS OF STUDY ON ADDITIONAL LOSS FUNCTIONS. ↑ (↓): HIGHER (LOWER) VALUE IMPLIES BETTER PERFORMANCE. ↑ (↓): HIGHER VALUE DENOTES BETTER (WORSE) PERFORMANCE.

Method	PSNR(dB)↑	SSIM↑	LPIPS↓	DISTS↓
InvDSNet Original	**31.85**	0.954	0.036	0.043
InvDSNet w/ SSIM	31.55	**0.963**	0.030	0.037
InvDSNet w/ LPIPS	31.76	0.958	**0.024**	0.034
InvDSNet w/ DISTS	31.80	0.960	0.025	**0.026**
InvDN [42]	27.46	0.861	0.184	0.125
MAWR [16]	31.35	0.941	0.062	0.058

the original ones as possible.

- "w/o $\mathcal{L}_{\text{reverse}}$": The model is trained only using $\mathcal{L}_{\text{forward}}$.
- "w/o $\mathcal{L}_{\text{att}}$": The model is trained without using $\mathcal{L}_{\text{att-f}}$ and $\mathcal{L}_{\text{att-r}}$ in the loss function.
- "w/o $\mathcal{L}_{\text{reverse}}$&$\mathcal{L}_{\text{att}}$": Both $\mathcal{L}_{\text{reverse}}$ and $\mathcal{L}_{\text{att}}$ are removed from the training loss, *i.e.*, the model is trained with the forward separation loss only.

The results of the ablation studies are listed in Table III, where InvDSNet outperforms all the baselines noticeably. (i) The plain model show a significant performance decrease (around 3dB PSNR) over the original InvDSNet, which indicates the effectiveness of our proposed blocks. (ii) The effectiveness of the ACB can be verified by comparing the results of "w/o ACB" and "Original" as well as by comparing the results of "Single Path & w/o ACB" and "Single Path". Particularly, the latter case has a PSNR decrease of 1.35dB when disabling the ACB. (iii) Comparing the results of "Single Path" with "Original", we can see that the proposed dual-path invertible architecture of InvDSNet with CIC structure does benefit image desnowing, which brings 0.73dB PSNR gain. The HT layers in intermediate layers also play an important role which improves the PSNR result with 0.64dB, as they exploit multi-scale representation during progressive feature refinement along the network depth dimension. (iv) There is a PSNR drop of 2.98dB when replacing the additional N coupling layers in CIC blocks with resicual blocks. This indicates the CIC design can better extract features for the separation task. (v) Regarding the reverse loss $\mathcal{L}_{\text{reverse}}$ and the attention loss pair ($\mathcal{L}_{\text{att-f}}, \mathcal{L}_{\text{att-r}}$), both of them have a noticeable contribution to the performance, which are 0.42dB and 0.51dB respectively. This is probably because that such loss functions reduce the overfitting of InvDSNet. In addition, $\mathcal{L}_{\text{reverse}}$ and ($\mathcal{L}_{\text{att-f}}, \mathcal{L}_{\text{att-r}}$) play different roles as a noticeable PSNR drop is observed when removing both of them.

To analyze the performance impact of the values of M and N, *i.e.*, the number of coupling layers in the CIC blocks, we conduct an ablation study by fixing M or N to its original value while varying the value of the other. Concretely, we fix $M = 2$ and set $N = 6, 8, 10$ respectively, and then fix $N = 8$ and set $M = 1, 2, 3$ respectively. The results of the corresponding models as well as their model size (in terms of number of parameters) are listed in Table IV. It can be seen that as M or N increases, the performance of InvDSNet increases, but the increment is small when M and N are sufficiently large. The setting of $M = 2, N = 8$ gives a good balance between the performance and model complexity.

E. Study on Additional Loss Functions

As discussed in [54], [55], existing classic quantitative metrics for measuring image quality are not always consistent with the judgment from human perception. Indeed, many quantitative metrics (*e.g.* [54]–[57]) have been developed for evaluating visual image quality with higher consistency to human perception. The study of this paper focuses on the design of new DNN architectures and loss functions for recovering images, where the loss functions depend on the choice of image quality metrics. A simple ℓ_2-norm-related loss is implemented for measuring the prediction error during training and the quality of the results. Nevertheless, the forward and backward loss functions we propose do not call special properties of ℓ_2-norm. They can be easily adapted to other quality metrics for measuring prediction errors.

In this experiment, we evaluate the performance impact brought by adapting the training loss functions to different image quality metrics including SSIM, LPIPS [56] and DISTS [57]. In addition, we also would like to see how the NNs optimized for one quality metric behaves when their outputs are measured using another quality metric. Thus, we also use LPIPS and DISTS for evaluation. The results are listed in Table V. Two baselines are included for comparison, which also perform worse than InvDSNet in the new metrics. It can also be seen that although the PSNR drops slightly when adding SSIM/LPIPS/DISTS into the loss functions, the values of all these three metrics are improved simultaneously over the original ones. Especially, adding the DISTS loss achieves improvements on the SSIM, LPIPS and DISTS metrics with minimal PSNR drop cost. Such results indicate that the performance of the proposed DNN architecture and loss functions do not rely on any specific metric. Adopting more advanced image quality metrics in the loss functions can be helpful for achieving better quality even measured with different metrics, which can also result in the performance gain consistent with human perception.

VI. Conclusion

In this paper, we proposes InvDSNet, a dual-path attentive INN for single image desnowing. Benefiting from the invertibility of coupling layers, the two paths in InvDSNet interacts with each other for feature refinement and disentanglement, while keeping all available details for reconstructing the latent image. Together with the coupling-in-coupling layers and attentive coupling layers, the InvDSNet can effectively remove snowflakes while preserving image details. In the extensive experiments, the InvDSNet noticeably outperforms existing methods, quantitatively and qualitatively. In addition, the InvDSNet also has lower model complexity and higher computational efficiency compared to existing DNN models.

Due to its invertibility, the InvDSNet in its reverse mode can be viewed as an image formation process for snowflake-corrupted images. Since the formation process of snowflake-corrupted images is not simple, our approach can be viewed as implicitly learning and exploiting such a process (*e.g.* via the reverse loss), instead of explicitly exploiting it. In future, we would like to investigate the extension of InvDSNet to solving other bad-weather image processing problems where the underlying physics of the degradation process are complex.

References

[1] T. U. Kaempfer, M. Hopkins, and D. Perovich, "A three-dimensional microstructure-based photon-tracking model of radiative transfer in snow," *Journal of Geophysical Research: Atmospheres*, vol. 112, no. D24, 2007.

[2] M. Li, X. Cao, Q. Zhao, L. Zhang, and D. Meng, "Online rain/snow removal from surveillance videos," *IEEE Transactions on Image Processing*, vol. 30, pp. 2029–2044, 2021.

[3] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4210–4219.

[4] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, and J.-N. Hwang, "Desnownet: Context-aware deep network for snow removal," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 3064–3073, 2018.

[5] R. Li, R. T. Tan, and L.-F. Cheong, "All in one bad weather removal using architectural search," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3175–3185.

[6] W.-T. Chen, H.-Y. Fang, J.-J. Ding, C.-C. Tsai, and S.-Y. Kuo, "Jstasr: Joint size and transparency-aware snow removal algorithm based on modified partial convolution and veiling effect removal," in *Proceedings of European Conference on Computer Vision*. Springer, 2020, pp. 754–770.

[7] S.-C. Pei, Y.-T. Tsai, and C.-Y. Lee, "Removing rain and snow in a single image using saturation and visibility features," in *IEEE International Conference on Multimedia and Expo Workshops*. IEEE, 2014, pp. 1–6.

[8] X. Zheng, Y. Liao, W. Guo, X. Fu, and X. Ding, "Single-image-based rain and snow removal using multi-guided filter," in *Proceedings of International Conference on Neural Information Processing*. Springer, 2013, pp. 258–265.

[9] X. Ding, L. Chen, X. Zheng, Y. Huang, and D. Zeng, "Single image rain and snow removal via guided l0 smoothing filter," *Multimedia Tools and Applications*, vol. 75, no. 5, pp. 2697–2712, 2016.

[10] 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*, vol. 26, no. 8, pp. 3936–3950, 2017.

[11] S.-C. Huang, D.-W. Jaw, B.-H. Chen, and S.-Y. Kuo, "Single image snow removal using sparse representation and particle swarm optimizer," *ACM Transactions on Intelligent Systems and Technology*, vol. 11, no. 2, pp. 1–15, 2020.

[12] W.-T. Chen, H.-Y. Fang, C.-L. Hsieh, C.-C. Tsai, I. Chen, J.-J. Ding, S.-Y. Kuo *et al.*, "All snow removed: Single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss," in *Proceedings of IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4196–4205.

[13] D.-W. Jaw, S.-C. Huang, and S.-Y. Kuo, "Desnowgan: An efficient single image snow removal framework using cross-resolution lateral connection and gans," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1342–1350, 2020.

[14] P. Li, M. Yun, J. Tian, Y. Tang, G. Wang, and C. Wu, "Stacked dense networks for single-image snow removal," *Neurocomputing*, vol. 367, pp. 152–163, 2019.

[15] Z. Li, J. Zhang, Z. Fang, B. Huang, X. Jiang, Y. Gao, and J.-N. Hwang, "Single image snow removal via composition generative adversarial networks," *IEEE Access*, vol. 7, pp. 25 016–25 025, 2019.

[16] W.-T. Chen, Z.-K. Huang, C.-C. Tsai, H.-H. Yang, J.-J. Ding, and S.-Y. Kuo, "Learning multiple adverse weather removal via two-stage knowledge learning and multi-contrastive regularization: Toward a unified model," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 653–17 662.

[17] L. Dinh, J. Sohl-Dickstein, and S. Bengio, "Density estimation using real nvp," *arXiv preprint arXiv:1605.08803*, 2016.

[18] J. Xu, W. Zhao, P. Liu, and X. Tang, "Removing rain and snow in a single image using guided filter," in *Proceedings of IEEE International Conference on Computer Science and Automation Engineering*, vol. 2. IEEE, 2012, pp. 304–307.

[19] ——, "An improved guidance image based method to remove rain and snow in a single image," *Computer and Information Science*, vol. 5, no. 3, p. 49, 2012.

[20] D. Rajderkar and P. Mohod, "Removing snow from an image via image decomposition," in *Proceedings of IEEE International Conference on Emerging Trends in Computing, Communication and Nanotechnology*. IEEE, 2013, pp. 576–579.

[21] A. Akbari, M. Trocan, S. Sanei, and B. Granado, "Joint sparse learning with nonlocal and local image priors for image error concealment," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 8, pp. 2559–2574, 2019.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4700–4708.

[23] B. Cheng, J. Li, Y. Chen, S. Zhang, and T. Zeng, "Snow mask guided adaptive residual network for image snow removal," *arXiv preprint arXiv:2207.04754*, 2022.

[24] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, 2010.

[25] Y. Yan, W. Ren, Y. Guo, R. Wang, and X. Cao, "Image deblurring via extreme channels prior," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4003–4011.

[26] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *International Journal of Computer Vision*, vol. 86, no. 2-3, p. 256, 2010.

[27] J. Bossu, N. Hautière, 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*, vol. 93, no. 3, pp. 348–367, 2011.

[28] J.-H. Kim, J.-Y. Sim, and C.-S. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2658–2670, 2015.

[29] J. Tian, Z. Han, W. Ren, X. Chen, and Y. Tang, "Snowflake removal for videos via global and local low-rank decomposition," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2659–2669, 2018.

[30] B. Yang, Z. Jia, J. Yang, and N. K. Kasabov, "Video snow removal based on self-adaptation snow detection and patch-based gaussian mixture model," *IEEE Access*, vol. 8, pp. 160 188–160 201, 2020.

[31] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of IEEE/CVF International Conference on Computer Vision*, 2019, pp. 6202–6211.

[32] D. Engin, A. Genç, and H. Kemal Ekenel, "Cycle-dehaze: Enhanced cyclegan for single image dehazing," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 825–833.

[33] R. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu, "Attentive generative adversarial network for raindrop removal from a single image," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2482–2491.

[34] Y. Quan, S. Deng, Y. Chen, and H. Ji, "Deep learning for seeing through window with raindrops," in *Proceedings of IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2463–2471.

[35] J. Peng, Y. Xu, T. Chen, and Y. Huang, "Single-image raindrop removal using concurrent channel-spatial attention and long-short skip connections," *Pattern Recognition Letters*, vol. 131, pp. 121–127, 2020.

[36] W. Luo, J. Lai, and X. Xie, "Weakly supervised learning for raindrop removal on a single image," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 5, pp. 1673–1683, 2020.

[37] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 11, pp. 3943–3956, 2019.

[38] K. Jiang, Z. Wang, P. Yi, C. Chen, Z. Han, T. Lu, B. Huang, and J. Jiang, "Decomposition makes better rain removal: An improved attention-guided deraining network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3981–3995, 2020.

[39] L. Cai, Y. Fu, T. Zhu, Y. Xiang, Y. Zhang, and H. Zeng, "Joint depth and density guided single image de-raining," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 7, pp. 4108–4121, 2021.

[40] M. Xiao, S. Zheng, C. Liu, Y. Wang, D. He, G. Ke, J. Bian, Z. Lin, and T.-Y. Liu, "Invertible image rescaling," in *Proceedings of European Conference on Computer Vision*. Springer, 2020, pp. 126–144.

[41] Z. Guan, J. Jing, X. Deng, M. Xu, L. Jiang, Z. Zhang, and Y. Li, "Deepmih: Deep invertible network for multiple image hiding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[42] Y. Liu, Z. Qin, S. Anwar, P. Ji, D. Kim, S. Caldwell, and T. Gedeon, "Invertible denoising network: A light solution for real noise removal," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 13 365–13 374.

[43] J. Liang, A. Lugmayr, K. Zhang, M. Danelljan, L. Van Gool, and R. Timofte, "Hierarchical conditional flow: A unified framework for image super-resolution and image rescaling," in *Proceedings of IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4076–4085.

[44] J. Whang, E. Lindgren, and A. Dimakis, "Composing normalizing flows for inverse problems," in *Proceedings of International Conference on Machine Learning*. PMLR, 2021, pp. 11 158–11 169.

[45] J. Li, K. Qin, R. Xu, and H. Ji, "Deep scale-aware image smoothing," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2022, pp. 2105–2109.

[46] R. Zhao, T. Liu, J. Xiao, D. P. Lun, and K.-M. Lam, "Invertible image decolorization," *IEEE Transactions on Image Processing*, vol. 30, pp. 6081–6095, 2021.

[47] J. Zhang, H. Wang, X. Wu, and W. Zuo, "Invertible network for unpaired low-light image enhancement," *arXiv preprint arXiv:2112.13107*, 2021.

[48] Y. Xing, Z. Qian, and Q. Chen, "Invertible image signal processing," in *Proceedings of IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6287–6296.

[49] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[51] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[52] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "Cbam: Convolutional block attention module," in *Proceedings of European Conference on Computer Vision*, 2018, pp. 3–19.

[53] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[54] W. Zhang, K. Ma, G. Zhai, and X. Yang, "Task-specific normalization for continual learning of blind image quality models," *arXiv preprint arXiv:2107.13429*, 2021.

[55] W. Zhang, D. Li, C. Ma, G. Zhai, X. Yang, and K. Ma, "Continual learning for blind image quality assessment," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[56] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 586–595.

[57] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

Yuhui Quan received the Ph.D. degree in Computer Science from South China University of Technology in 2013. He worked as the postdoctoral research fellow in Mathematics at National University of Singapore from 2013 to 2016. He is currently the associate professor in Computer Science at South China University of Technology. His research interests include image restoration, unsupervised learning, and sparse representation.

Xiaoheng Tan is currently a M.Sc candidate of Computer Science at South China University of Technology. He is working on machine learning and image processing.

Yan Huang received the B.Sc. degree in Intelligence Science and Technology from Hunan University in 2013, and the Ph.D. degree in Computer Science and Technology from South China University of Technology (SCUT) in 2018. She worked as the postdoctoral research fellow at SCUT from 2018 to 2020. She is currently the associate professor at School of Computer Science and Engineering in SCUT. Her research interests include computer vision and deep learning.

Yong Xu received the B.S., M.S., and Ph.D. degrees in mathematics from Nanjing University, Nanjing, China, in 1993, 1996, and 1999, respectively. He was a Post-Doctoral Research Fellow of computer science with South China University of Technology from 1999 to 2001. He is currently a professor in Computer Science at South China University of Technology. His current research interests include computer vision and image processing.

Hui Ji received the B.Sc. degree in Mathematics from Nanjing University in China, the M.Sc. degree in Mathematics from National University of Singapore and the Ph.D. degree in Computer Science from the University of Maryland, College Park. In 2006, he joined National University of Singapore as an assistant professor in Mathematics. Currently, he is an associate professor in mathematics at National University of Singapore. His research interests include image processing and machine learning.