SmartAssign: Learning A Smart Knowledge Assignment Strategy for Deraining and Desnowing

Yinglong Wang[†] Chao Ma[§] Jianzhuang Liu[†]
[†]Huawei Noah's Ark Lab
[§]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
ylwanguestc@gmail.com, chaoma@sjtu.edu.cn, liu.jianzhuang@huawei.com

Abstract

Existing methods mainly handle single weather types. However, the connections of different weather conditions at deep representation level are usually ignored. These connections, if used properly, can generate complementary representations for each other to make up insufficient training data, obtaining positive performance gains and better generalization. In this paper, we focus on the very correlated rain and snow to explore their connections at deep representation level. Because sub-optimal connections may cause negative effect, another issue is that if rain and snow are handled in a multi-task learning way, how to find an optimal connection strategy to simultaneously improve deraining and desnowing performance. To build desired connection, we propose a smart knowledge assignment strategy, called SmartAssign, to optimally assign the knowledge learned from both tasks to a specific one. In order to further enhance the accuracy of knowledge assignment, we propose a novel knowledge contrast mechanism, so that the knowledge assigned to different tasks preserves better uniqueness. The inherited inductive biases usually limit the modelling ability of CNNs, we introduce a novel transformer block to constitute the backbone of our network to effectively combine long-range context dependency and local image details. Extensive experiments on seven benchmark datasets verify that proposed SmartAssign explores effective connection between rain and snow, and improves the performances of both deraining and desnowing apparently. The implementation code will be available at https://gitee.com/mindspore/models/tree/master/research/cv/SmartAssign.

1. Introduction

Bad weather types, such as haze, rain, and snow inevitably degrade the visual quality of images, meanwhile decrease the performances of other downstream computer vision tasks, e.g., autonomous driving [57]. Existing methods mainly focus on single weather types, e.g., deraining [14, 19, 26, 45, 46, 48–50, 56, 60], dehazing [5, 10, 31, 41, 54], and desnowing [6, 7, 32, 49]. However, these methods usually ignore the connections among these weather types, which, if used properly, may simultaneously improve the performance of multiple image recovery tasks.

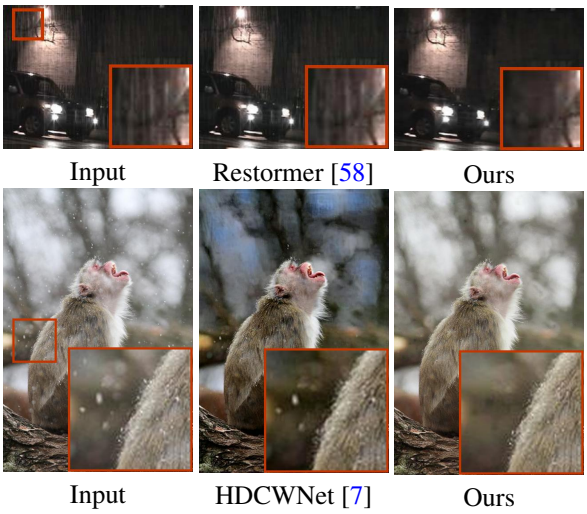

Figure 1. Given challenging rainy (with blurry rain streaks) and snowy (with high bright snowflakes) images , the proposed method effectively removes the artifacts of rain and snow simultaneously, achieving better results than the state-of-the-art approaches. This is attributed to the unique knowledge which captures accurate features of rain/snow as well as the common knowledge boosting the generalization of our model to real data.

Some methods attempt to explore the connections among different weather types by handling them with an unified architecture and one set of pre-trained weights, e.g., [8, 25, 27]. But they neglect the difference of multiple weather types, the uniqueness belonging to single weather types may harm the performance of other weather recovery tasks. Therefore, the performances of such unified networks

are usually lower than the ones for single weather types [8].

In this paper, we focus on the very similar rain and snow and develop a novel Multi-Task Learning (MTL) strategy to explore their connections at deep representation level, meanwhile avoiding the uniqueness of one weather type from damaging the performance of the other weather recovery task. Specifically, our goal is to accurately find the representations (i.e., connections) which can be shared by deraining and desnowing to simultaneously enhance both their performance, and meanwhile determine the exclusive representations (i.e., uniqueness) for one task to specially promote its own performance as well as avoid such uniqueness from damaging the other task. To facilitate the description of our method, we define the deep representation of a single network channel as a *knowledge atom*. All the knowledge atoms constitute the knowledge learned by networks.

Similar to conventional MTL method, we also use a backbone encoder $\mathcal{E}(\cdot)$ to learn the whole image-recovery knowledge simultaneously from rainy and snowy images, and two task-targeted decoders $\mathcal{D}_{dr}(\cdot)$ and $\mathcal{D}_{ds}(\cdot)$ are followed in parallel to remove rain and snow, separately. Conventional MTL takes the whole knowledge as the input for the subsequent decoders. Though such mechanism makes the best of the connections between both tasks, the influences of the uniqueness of single tasks are neglected, i.e., the uniqueness of rain may harm the performance of desnowing, and vice versa. Instead, we propose a novel Gated Knowledge Filtering Module (GKFM) to select optimal knowledge atoms for both tasks via a highly smart strategy, so that the connections between both tasks are sufficiently explored and the uniqueness of single tasks is properly used. To coordinate with GKFM, we design a Task-targeted Knowledge FeedForward mechanism (TKFF) to let every knowledge atom flow to its related tasks. Through our GKFM and TKFF, we realize a smart knowledge assignment, in which both tasks adaptively explore their connections and uniqueness via gradient backward-propagation. Hence, we term our MTL mechanism as *SmartAssign*.

In order to further enhance the accuracy of knowledge assignment, i.e., toward optimally exploring the connections and uniqueness of deraining and desnowing, we introduce a novel knowledge contrast, making the same kind of knowledge atoms more closer, and the ones belonging to different kinds more discriminative under a similarity metric. In this process, all the knowledge atoms are transformed into a new low-dimension feature space by a Dimension Reduction Module (DRM) to avoid model collapse when operating on the original high-dimension knowledge atoms.

Currently, CNNs are still the mainstream choice for image recovery. However, the inherited inductive biases limit their modelling capacity for long-range context dependency. Though they can also obtain a large receptive field by stacking a deep architecture, such indirect modelling is indeed inferior to that of a transformer, which models both short and long range dependency directly via self-attention. In this paper, we adopt transformer blocks to constitute our backbone encoder $\mathcal{E}(\cdot)$. Usually, transformer needs sufficient training pairs to ensure good performance. Hence, our transformer block introduces a gated CNN branch to complement limited training data via the inductive biases. Moreover, the locality of CNN helps to recover degraded image details and the gated operation is used to reduce redundant features caused by the combination of CNN and transformer. Figure 1 gives two examples of deraining and desnowing. By contrast, our method obtains better image recovery quality on both tasks than SOTA methods.

Our contributions are summarized in the following:

- We propose a novel knowledge assignment strategy, i.e., SmartAssign, to excavate the connections and uniqueness of rain and snow, so that their connections are used to enhance the performance of both tasks and the uniqueness is applied to boost corresponding task and avoided from damaging the other task.
- We propose a novel knowledge contrast mechanism to further boost the accuracy of knowledge assignment, in which a dimension reduction module (DRM) is introduced to stabilize the training of our model.
- We propose a novel transformer block to make the best use of the superiority of self-attention and convolution, in which gated operations are introduced to alleviate the feature redundancy.

2. Related Work

Deraining. Early deraining methods usually utilize dictionary learning [35] to factorize a rainy image, and the rain-removed result is obtained by reconstructing the non-rain dictionary atoms [4, 9, 17, 21, 30, 33, 47, 49]. In contrast to conventional methods, deep learning has shown better deraining performance and running efficiency. Some deep learning based methods learn a direct mapping from rainy images to rain-free ones, e.g., SPANet [46], PIGWM [62], TUM [8], DGUNet [37], AirNet [25] and KiT [24].

Because the patterns of rain streaks are simpler than these of rain-free background, some other works use a network to learn easier rainy residual, which is subtracted from rainy image to obtain the deraining result. These methods include DDN [14], JORDER [55, 56], DID-MDN [60], RESCAN [29], MSPFN [19], DRD-Net [11], MPRNet [59], RCDNet [45], MAXIM [43], TransWeather [44], Uformer [53], etc. Though residual based models can deal with rain streaks well, they cannot remove haze-like effect caused by the accumulation of tiny raindrops, which usually leads to low image contrast. Some attempts focus on developing a physical model to formulate the formation of rainy

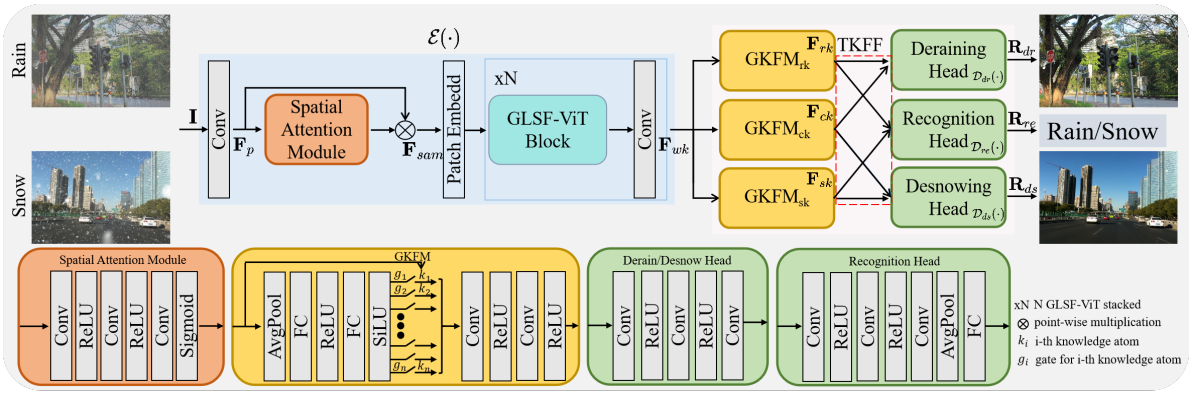

Figure 2. Overview of our SmartAssign framework. SmartAssign first learns $\mathbf{F}_{wk}$ together from $\mathcal{T}_{dr}$, $\mathcal{T}_{ds}$ and $\mathcal{T}_{re}$. Then GKFM and TKFF are used to explore the connections $\mathbf{F}_{ck}$ and uniqueness $\mathbf{F}_{rk}$/$\mathbf{F}_{sk}$ of $\mathcal{T}_{dr}$ and $\mathcal{T}_{ds}$ to optimally assign knowledge atoms to related tasks.

images, and a network is trained under the driving of the model to handle haze-like effect. These methods include PYM+GAN [26] and ASV-joint [51].

Desnowing. Early works rely on guided filters [16] to remove snow in single images [12, 61]. Similar to deraining, an over-complete dictionary can also be used to enhance snowy images [49]. Liu et al. present DesnowNet to remove opaque snow particles, in which multiple scales are utilized to model the diversity of snow [32]. In [6], a snow model is proposed by allowing veiling effect and a network called JSTASR is designed to classify and further remove snow. Wavelet transform and contradictive channel features are proposed in [7] to remove snow in a hierarchical way. In [8], two-stage knowledge distillation and multi-contrastive regularization are developed to handle multiple weather conditions, including snow.

Multi-task learning. Multi-task learning methods are usually classified into two categories according to the parameter sharing strategies [39]. The first category is hard-parameter sharing, in which a backbone encoder is shared by all tasks and multiple task-targeted decoders branch out [2, 13, 18, 20, 23, 27, 38]. The second category is soft-parameter sharing, in which every task has its independent network branch. Different branches share parameters via a defined mechanism, e.g., Cross-stitch [36], Sluice [40], and NDDR [15]. AdaShare is another different parameter sharing method [42]. It uses a single network to execute multi-task training, in which different tasks can selectively choose layers to constitute their own execution paths.

3. Proposed Method

Rain and snow are two different weather types. Their differences mainly lie in: 1) different transparency, leading to higher pixel intensities of snowflakes than rain streaks; 2) different sizes, falling speed and directions, separately forming rain streaks and snowflakes in images. Despite of differences, they still have some commonalities: 1) Due to strong reflection to light, they both possess higher intensities than neighboring regions [49]; 2) veiling effect could appear in both of them [6, 51]; 3) their clear backgrounds share the same distribution. Besides such apparent differentiae and commonalities in RGB space, in this paper, we focus on exploring their connections and uniqueness in deep feature space, and design an effective MTL algorithm to enhance the performances of both deraining and desnowing.

Assume deraining and desnowing are denoted as $\mathcal{T}_{dr}$ and $\mathcal{T}_{ds}$, we first learn the whole knowledge $\mathbf{F}_{wk}$ from both rain and snow images in a MTL manner. In $\mathbf{F}_{wk}$, some knowledge atoms $\mathbf{F}_{ck}$ can be shared by both tasks, i.e., their connections, some $\mathbf{F}_{rk}$/$\mathbf{F}_{sk}$ can only be exclusively used by $\mathcal{T}_{dr}$/$\mathcal{T}_{ds}$, i.e., their uniqueness. Our core idea is to find an optimal strategy to assign $\mathbf{F}_{wk}$ to $\mathcal{T}_{dr}$ and $\mathcal{T}_{ds}$ (SmartAssign), so that their performances can be enhanced simultaneously. In order to further improve the accuracy of knowledge assignment, we introduce another auxiliary task $\mathcal{T}_{re}$ to recognize rainy and snowy scenes to make knowledge atoms in $\mathbf{F}_{wk}$ more discriminative. Moreover, $\mathcal{T}_{re}$ also helps network extract high-level semantic knowledge of rain and snow, facilitating their removing from images.

3.1. SmartAssign

Figure 2 illustrates an overview of SmartAssign. We first learn $\mathbf{F}_{wk}$ via a backbone encoder $\mathcal{E}(\cdot)$ as follows:

$$\mathbf{F}_{wk} = \mathcal{E}(\mathbf{I}), \tag{1}$$

where $\mathbf{I}$ is rainy/snowy image. During training, rainy and snowy datasets are combined together to constitute our training dataset. In order to determine the shared and exclusive knowledge atoms, we adopt three gated knowledge

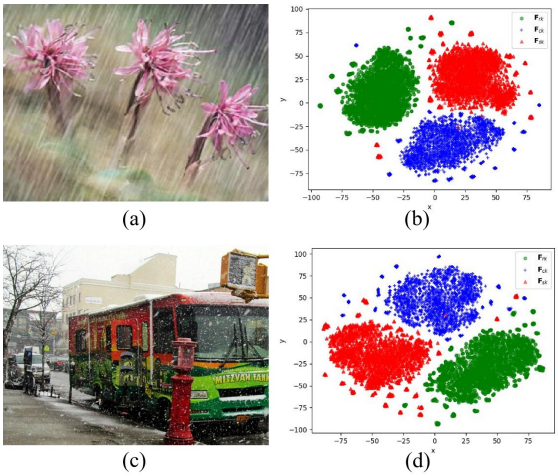

Figure 3. The t-SNE distributions of $\mathbf{F}_{rk}$, $\mathbf{F}_{ck}$ and $\mathbf{F}_{sk}$. (a) A rainy image. (b) Their t-SNE distributions for (a). (c) A snowy image. (d) Their t-SNE distributions for (c). The results illustrate that $\mathbf{F}_{rk}$, $\mathbf{F}_{ck}$ and $\mathbf{F}_{sk}$ possess different distributions for both rainy and snowy images. Hence, they respectively contain representative image contents.

filtering module (GKFM) to filer $\mathbf{F}_{wk}$ as follows:

$$\begin{aligned}\mathbf{F}_{rk} &= \mathrm{GKFM}_{rk}(\mathbf{F}_{wk}),\\ \mathbf{F}_{ck} &= \mathrm{GKFM}_{ck}(\mathbf{F}_{wk}),\\ \mathbf{F}_{sk} &= \mathrm{GKFM}_{sk}(\mathbf{F}_{wk}),\end{aligned} \tag{2}$$

where $\mathbf{F}_{rk}$ is the unique knowledge for $\mathcal{T}_{dr}$, $\mathbf{F}_{sk}$ is for $\mathcal{T}_{ds}$, and $\mathbf{F}_{ck}$ is the shared knowledge for both of them. Figure 3 shows their t-SNE [34] distributions. We observe that they represent different distributions, and thus stand for different meanings. The latter experiments will further illustrate their effectiveness on $\mathcal{T}_{dr}$ and $\mathcal{T}_{ds}$. After obtaining the connections and uniqueness, we adopt a novel and effective task-targeted knowledge feedforward (TKFF) mechanism to assign $\mathbf{F}_{rk}$, $\mathbf{F}_{ck}$ and $\mathbf{F}_{sk}$ to related tasks. Note that our TKFF is not a network architecture, but a novel task-guided knowledge assignment strategy. Different from conventional MTL taking the whole knowledge $\mathbf{F}_{wk}$ as the input for the following decoders, our TKFF utilizes $\mathbf{F}_{wk}$ more elaborately and accurately, letting the common knowledge $\mathbf{F}_{ck}$ enhance the performance of both tasks, and the unique knowledge $\mathbf{F}_{rk}$/$\mathbf{F}_{sk}$ only boosts corresponding tasks.

Figure 2 visually shows how our TKFF assigns knowledge to different tasks. $\mathbf{F}_{rk}$ flows to the deraining head $\mathcal{D}_{dr}(\cdot)$, and $\mathbf{F}_{sk}$ is assigned to the desnowing head $\mathcal{D}_{ds}(\cdot)$. The shared $\mathbf{F}_{ck}$ flows to both $\mathcal{D}_{dr}(\cdot)$ and $\mathcal{D}_{ds}(\cdot)$ to enhance both their performance. Because the connection $\mathbf{F}_{ck}$ is helpless for the auxiliary $\mathcal{T}_{re}$, the recognition head $\mathcal{D}_{re}(\cdot)$ only takes the unique knowledge $\mathbf{F}_{rk}$ and $\mathbf{F}_{sk}$ as inputs. Such process is formulated as follows:

$$\begin{aligned}\mathbf{R}_{dr} &= \mathcal{D}_{dr}(\mathbf{F}_{rk}, \mathbf{F}_{ck}),\\ \mathbf{R}_{re} &= \mathcal{D}_{re}(\mathbf{F}_{rk}, \mathbf{F}_{sk}),\\ \mathbf{R}_{ds} &= \mathcal{D}_{ds}(\mathbf{F}_{sk}, \mathbf{F}_{ck}),\end{aligned} \tag{3}$$

where $\mathbf{R}_{dr}$, $\mathbf{R}_{re}$ and $\mathbf{R}_{ds}$ are the results of deraining $\mathcal{T}_{dr}$, recognition $\mathcal{T}_{re}$ and desnowing $\mathcal{T}_{ds}$, respectively.

We would like to point out that our SmartAssign enables different tasks to adaptively control the closing/opening of the gates in GKFM for all knowledge atoms via the gradient backpropagation, so that an optimal knowledge assignment strategy will be learned to obtain the desired connections and uniqueness of different tasks. That is the reason that our method is termed as SmartAssign.

3.2. Network Structure

As shown in Figure 2, our network consists of three main parts: the shared backbone encoder $\mathcal{E}(\cdot)$, GKFM and the decoders $\mathcal{D}_{dr}(\cdot)$, $\mathcal{D}_{re}(\cdot)$ and $\mathcal{D}_{ds}(\cdot)$ for the three tasks. We give their detailed architecture in the following.

Encoder. We first use a convolutional layer to project a RGB image $\mathbf{I}$ into feature space:

$$\mathbf{F}_p = \mathrm{Conv}(\mathbf{I}). \tag{4}$$

Because rain/snow usually possesses salient characteristics in the shallow feature space, we adopt a **S**patial **A**ttention **M**odule (SAM) to highlight such saliencies, facilitating the discrimination of rain/snow from background:

$$\mathbf{F}_{sam} = \phi(\mathrm{Conv}_3(\mathbf{F}_p)) \odot \mathbf{F}_p, \tag{5}$$

where $\phi(\cdot)$ is the Sigmoid activation function, $\odot$ is pointwise multiplication, the subscript number 3 means 3 convolutional layers. The locality bias usually restricts the modelling capacity of CNN architectures for long-range context dependencies. Instead of only using CNN architecture, we introduce a transformer structure to learn the whole features $\mathbf{F}_{wk}$, in which a novel **G**ated **L**ong and **S**hort range feature **F**usion **Vi**sion **T**ransformer (GLSF-ViT) block is designed to combine the merits of local convolutional and global self-attention mechanism. $\mathbf{F}_{wk}$ can be obtained as follows:

$$\mathbf{F}_{wk} = \text{GLSF-ViT}_N(\mathbf{F}_{sam}), \tag{6}$$

where N means there are N GLSF-ViT block.

GLSF-ViT. As shown in Figure 4, our GLSF-ViT is comprised of two branches. In the upper branch, a **M**ulti-head **S**elf-**A**ttention module (MSA) and a MLP module are concatenated in cascade to model long-range dependency of a patch-embedded feature $\mathbf{F}_{pe}$, which is formulated as:

$$\begin{aligned}\mathbf{F}_{msa} &= \mathrm{MSA}(\mathrm{LN}(\mathbf{F}_{pe})) + \mathbf{F}_{pe},\\ \mathbf{F}_{mlp} &= \mathrm{MLP}(\mathrm{LN}(\mathbf{F}_{msa})) + \mathbf{F}_{msa},\end{aligned} \tag{7}$$

where $\mathrm{LN}(\cdot)$ is layer normalization for stable training and fast convergence. Self-attention is the core operation in MSA. If Queries ($\mathbf{Q}$), Keys ($\mathbf{K}$) and Values ($\mathbf{V}$) have been derived from $\mathbf{F}_{pe}$, the self-attention is calculated as:

$$\mathrm{Attn}(\mathbf{Q},\mathbf{K},\mathbf{V}) = \mathrm{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}})\mathbf{V}, \tag{8}$$

where d is embedding dimension. In the lower branch, a CNN block is used to extract more image details, in which SiLU is activated as the gate for each feature channel to adaptively select useful feature maps, meanwhile avoiding feature redundancy by rejecting uesless ones, the details are:

$$\begin{aligned} \mathbf{F}_{cv} &= \mathrm{Conv}(\mathrm{LN}(\mathbf{F}_{pe})), \\ \mathbf{F}_{local} &= \mathbf{F}_{cv} \odot \mathrm{Gate}(\mathrm{FC}_2(\mathrm{Pool}(\mathbf{F}_{cv}))), \end{aligned} \tag{9}$$

where $\mathrm{Gate}(\cdot)$ is the gate operation, $\mathrm{FC}(\cdot)$ is fully-connected layer, $\mathrm{Pool}(\cdot)$ is average pooling. The output of our GLSF-ViT is:

$$\mathbf{F}_{glsf} = \mathbf{F}_{mlp} + \mathbf{F}_{local}. \tag{10}$$

GKFM. The modules $\mathrm{GKFM}_{rk}(\cdot)$, $\mathrm{GKFM}_{ck}(\cdot)$ and $\mathrm{GKFM}_{sk}(\cdot)$ taking $\mathbf{F}_{wk}$ as input possess the same architecture. The core design is a knowledge filtering mechanism with an adaptive gating operation, in which the gating operation is implemented with SiLU activation layer. Besides, two convolutional layers are followed to refine the filtered knowledge. The details are formulated as follows:

$$\mathbf{F}_{gkfm} = \mathrm{Conv}_2(\mathbf{F}_{wk} \odot \mathrm{Gate}(\mathrm{FC}_2(\mathrm{Pool}(\mathbf{F}_{wk})))), \tag{11}$$

Decoders. The structures for decoders $\mathcal{D}_{dr}(\cdot)$, $\mathcal{D}_{re}(\cdot)$ and $\mathcal{D}_{ds}(\cdot)$ are as follows:

$$\begin{aligned} \mathbf{R}_{dr} &= \mathrm{Conv}_3(\mathbf{F}_{rk}, \mathbf{F}_{ck}), \\ \mathbf{R}_{re} &= \mathrm{FC}(\mathrm{Pool}(\mathrm{Conv}_3(\mathbf{F}_{rk}, \mathbf{F}_{sk}))), \\ \mathbf{R}_{ds} &= \mathrm{Conv}_3(\mathbf{F}_{sk}, \mathbf{F}_{ck}). \end{aligned} \tag{12}$$

3.3. Loss Functions

Image recovery loss. $\mathcal{T}_{dr}$/$\mathcal{T}_{ds}$ aim to recover the degraded image $\mathbf{I}$ by rain/snow. They possess the same image recovery losses. In our work, we use Charbonnier loss [3] in the RGB image domain, the gradient loss and sparsity loss in the gradient domain to constrain the training of $\mathcal{T}_{dr}$/$\mathcal{T}_{ds}$. Assume $\mathbf{R}$ is the recovery result of $\mathbf{I}$, and $\mathbf{G}$ is corresponding ground truth, these losses are formulated as follows:

$$\mathcal{L}_{char} = \sqrt{\|\mathbf{R}-\mathbf{G}\|^2 + \epsilon^2}, \tag{13}$$

where ϵ is set as 10^{-3}. Because image details are more salient in the gradient domain, we add gradient loss as:

$$\mathcal{L}_{grad} = \| \bigtriangledown \mathbf{R} - \bigtriangledown \mathbf{G}\|^2, \tag{14}$$

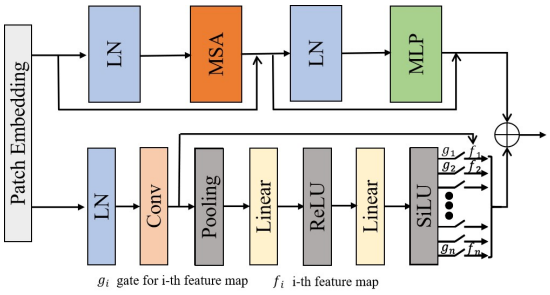

Figure 4. Architecture of our GLSF-ViT. In the upper branch, self-attention is adopted to capture long range dependency. In the lower branch, a CNN module is introduced to recover more image details. The gating operation is to avoid information redundancy.

where $\bigtriangledown$ denotes the combination of the horizontal and vertical gradient operators. Sparsity is an intrinsic property of images, and has been verified to be effective in image restoration [50]. We thus use the unsupervised quasi-sparsity loss to keep intact object textures:

$$\mathcal{L}_{sparsity} = \sum_{m,n} |\omega_{m,n} * \mathbf{R}| + |\omega_{m,n} * (\mathbf{I}-\mathbf{R})|, \tag{15}$$

where $\omega_{m,n}$ is the nth 3×3 filter centered at the mth pixel, and $*$ is the convolution operation.

Recognition loss. The loss for the recognition task $\mathcal{T}_{re}$ is:

$$\mathcal{L}_{recog} = \|\mathbf{R}_{re} - \mathbf{L}\|^2, \tag{16}$$

where $\mathbf{L}$ is the label indicating rainy or snowy scenes, and $\mathbf{R}_{re}$ is the prediction result for $\mathcal{T}_{re}$.

Contrastive loss. The knowledge $\mathbf{F}_{rk}$, $\mathbf{F}_{ck}$, and $\mathbf{F}_{sk}$ contain representative contents and play different roles, they should be exclusive from each other. We thus design a novel knowledge contrast mechanism to reach this goal. They are first transformed into three low-dimension $C\times F$ feature matrices $\mathbf{M}_{rk}$, $\mathbf{M}_{ck}$ and $\mathbf{M}_{sk}$ via a dimension reduction module (DRM), respectively, in which each row is a feature vector (the details are in supplement). The contrastive loss is then computed:

$$\begin{aligned} \mathbf{Z} = \frac{1}{3}(&\mathbf{M}_{rk}\mathbf{M}_{rk}^\top/(\mathbf{M}_{rk}\mathbf{M}_{ck}^\top + \mathbf{M}_{rk}\mathbf{M}_{sk}^\top) \\ &+ \mathbf{M}_{ck}\mathbf{M}_{ck}^\top/(\mathbf{M}_{ck}\mathbf{M}_{rk}^\top + \mathbf{M}_{ck}\mathbf{M}_{sk}^\top) \\ &+ \mathbf{M}_{sk}\mathbf{M}_{sk}^\top/(\mathbf{M}_{sk}\mathbf{M}_{rk}^\top + \mathbf{M}_{sk}\mathbf{M}_{ck}^\top)), \end{aligned} \tag{17}$$

$$\mathcal{L}_{contra} = \frac{1}{C^2}\sum_{i=1}^{C}\sum_{j=1}^{C}\mathbf{Z}(i,j), \tag{18}$$

where / denotes element-wise division.

At last, our overall loss function is:

$$\mathcal{L} = \mathcal{L}_{char} + \mathcal{L}_{grad} + \mathcal{L}_{recog} + \lambda_1\mathcal{L}_{sparsity} + \lambda_2\mathcal{L}_{contra}, \tag{19}$$

where λ_1 and λ_2 are balance parameters.

Datasets	MAXIM [43] PSNR/SSIM	DGUNet [37] PSNR/SSIM	MPRNet [59] PSNR/SSIM	ASV-joint [51] PSNR/SSIM	Restormer [58] PSNR/SSIM	TUM [8] PSNR/SSIM	Ours PSNR/SSIM
Rain-streak	33.01/0.924	32.47/0.919	32.11/0.917	31.08/0.905	32.91/0.923	30.72/0.892	**33.16/0.931**
Rain1200	30.78/0.897	31.54/0.896	32.91/0.916	31.77/0.893	31.48/0.901	29.57/0.858	**33.07/0.927**
Real-rain	33.53/0.958	35.83/0.948	36.04/0.946	35.67/0.927	35.34/0.946	22.75/0.855	**36.55/0.962**

Table 1. PSNR and SSIM comparison with SOTA methods. The best and second best numbers are in bold and underlined, respectively.

Input | MAXIM [43] | DGUNet [37] | MPRNet [59] | ASV-joint [51] | Restormer [58] | TUM [8] | Ours | GT

Figure 5. Qualitative comparisons with SOTA methods on a synthetic rainy images. (Zooming in for better view.)

Methods	Snow100K-L PSNR/SSIM	SRRS PSNR/SSIM	CSD PSNR/SSIM
JSTASR [6]	20.16/0.657	25.82/0.892	27.52/0.873
HDCW [7]	20.57/0.676	27.78/0.923	29.06/0.914
TUM [8]	25.66/0.851	28.03/0.926	30.10/0.933
Ours	**29.45/0.923**	**30.53/0.931**	**32.50/0.957**

Table 2. PSNR and SSIM comparison with SOTA methods on three snowy testing datasets. The best and second best numbers are in bold and underlined, respectively.

4. Experiments

4.1. Datasets and Implementation Details

Datasets. For deraining, we adopt "Rain-streak" [28], "Rain1200" [60], and "Real-Rain" [46] datasets. For desnowing, we use "Snow100K" [32], "SRRS" [6] and "CSD" [7] datasets. "Rain-streak"[1] contains 20800 pairs, we randomly select 2000 samples for testing, and the remaining is for training.

To make fair comparisons, we let all models be trained on the same datasets. During deraining comparisons, only "CSD" is used to match different rainy datasets to train our model. During desnowing comparison, only "rain1200" is adopted to match different snowy datasets. Under such strategy, three different performances can be obtained on "rain1200" when matching different snowy datasets. However, we observe that these performances are very close with each other, i.e., $\pm$0.07dB for PSNR and $\pm$0.01 for SSIM. Similar phenomenon happens on "CSD" when matching different rainy datasets, $\pm$0.1dB for PSNR and $\pm$0.05 for SSIM. Hence, in this paper, the reported performances on these two datasets are simultaneously produced by the model trained by using these two datasets.

[1]Download from https://github.com/lsy17096535/Single-Image-Deraining

Training details. We implement our model using the MindSpore Lite tool [1]. During training, all the patches are randomly cropped from the original image pairs with a fixed size of 256×256, the batch size is 4. Adam [22] is selected as the optimizer. The initial learning rate is 0.0004, and decreases by multiplying 0.5 when the loss stops reducing. λ_1 and λ_2 are set to 0.01, 0.001, respectively.

To evaluate our method, recent SOTA deraining methods DGUNet [37], MAXIM [43], Restormer [58], TUM [8], MPRNet [59], ASV-joint [51] are selected to make comparisons. Desnowing works are relatively fewer and the codes of some works are not available so far. We select the most recent JSTASR [6], HDCWNet [7] and TUM [8] to make comparisons. PSNR and SSIM [52] are selected as the objective metrics to assess different works quantitatively.

4.2. Comparison with State-of-the-Arts

Quantitative evaluation on synthetic data. Table 1 reports PSNR and SSIM values of different deraining methods. We observe that MAXIM, DGUNet, MPRNet and Restormer comprehensively obtain good quantitative values. Especially for MAXIM and MPRNet, they obtain more second best metrics. Restormer is relatively stable, i.e., its performances are consistently high on all datasets, showing good generalization. By contrast, our method obtains the best results, which is attributed to exploring the connections and uniqueness of rain and snow. Table 2 shows the performance of different methods on desnowing datasets. Our method shows highly superior performance on snow removal, obtaining 4.0 dB higher PSNR on the challenging Snow100K-L dataset than the second best TUM.

Qualitative evaluation on synthetic data. Figure 5 shows the deraining results of different methods on a synthetic images. We observe that MAXIM and our method

Figure 6. Qualitative comparisons with SOTA methods on a synthetic snow image. (Zooming in for better view.)

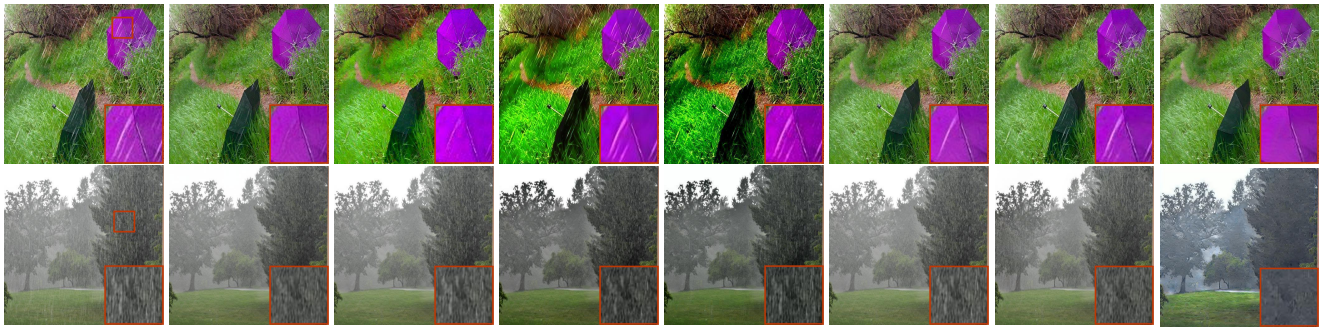

Figure 7. Qualitative comparisons with SOTA methods on two real rainy images. (Zooming in for better view.)

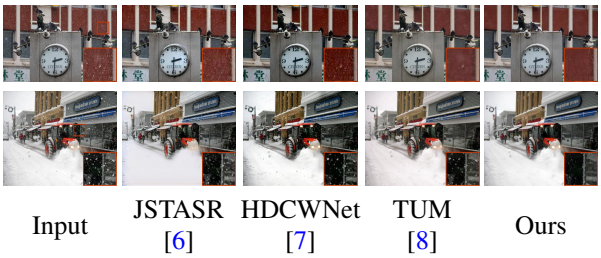

Figure 8. Qualitative comparisons with SOTA methods on two real-world snow images.

obtain the better performance in removing fine rainy traces. Other methods only remove apparent rain streaks, the fine rainy traces still remain. In Figure 6, we show the desnowing results of different methods on a synthetic snowy images. We observe that there are still apparent snowflakes existing in the results of JSTASR and HDCWNet. TUM obtains good visual quality. By comparison, ours removes more snowflakes and recovers clearer image details.

Qualitative evaluation on real data. Figure 7 shows the deraining results on real-world rainy images. MAXIM, DGUNet, Restormer and TUM produce good visual quality on the first image, removing majority of rain streaks and recovering more image details. However, SOTA methods cannot handle the second challenging one well. By contrast, our method removes more rain streaks and recovers better visual quality. Figure 8 visually shows desnowing results. Our method removes most snowflakes, especially for some fine-grained snow spots. We explore the connections of rain and snow, and adopt a novel knowledge filtering mechanism to avoid uniqueness of one task from damaging the other one. All these designs work cooperatively and generate superior performances on both $\mathcal{T}_{dr}$ and $\mathcal{T}_{ds}$.

4.3. Ablation Studies

Effectiveness of key designs. There are many key designs in our SmartAssign. Due to space limitation, we selectively study the following aspects by ablating experiments and only report their performances on the representative "Rain1200" and "CSD" datasets. 1) $\mathcal{V}_1$: ablating knowledge contrast; 2) $\mathcal{V}_2$: ablating GKFM and TKFF, i.e., the conventional MTL mechanism; 3) $\mathcal{V}_3$: ablating gated CNN module in GLSF-ViT; 4) $\mathcal{V}_4$: single task training, $\mathcal{V}_4^r$ for $\mathcal{T}_{dr}$, $\mathcal{V}_4^s$ for $\mathcal{T}_{ds}$; 5) $\mathcal{V}_5$: ablating auxilary task $\mathcal{T}_{re}$; 6) $\mathcal{V}_6$: our SmartAssign. The quantitative and visual results are given in Table 3 and Figure 9. We observe that the performance decreases after respectively ablating these designs. $\mathcal{V}_2$ obtains lowest values, illustrating the effectiveness of exploring connections and uniqueness. Our SmartAssign obtains the best results via combining different designs.

Effectiveness of unique and common knowledge. To study the effect of $\mathbf{F}_{rk}$, $\mathbf{F}_{ck}$ and $\mathbf{F}_{sk}$, we freeze the pre-trained parameters before TKFF, and 1) only use $\mathbf{F}_{ck}$ to train $\mathcal{D}_{dr}(\cdot)$ and $\mathcal{D}_{ds}(\cdot)$, termed as $\mathcal{K}_1$; 2) only use $\mathbf{F}_{rk}$ and $\mathbf{F}_{sk}$ to train $\mathcal{D}_{dr}(\cdot)$ and $\mathcal{D}_{ds}(\cdot)$, respectively, termed as $\mathcal{K}_2$. Table 4 shows that the performance decreases after ablating unique/common knowledge, respectively. But $\mathcal{K}_2$ is better than $\mathcal{K}_1$ consistently on rainy/snowy dataset, illustrating that unique knowledge plays more important role. Figure 10

Figure 9. Ablating components leads to performance decrease. Especially, GKFM and TKFF ($\mathcal{V}_2$), and the gated CNN module in GLSF-ViT ($\mathcal{V}_3$) play key roles in our method.

Datasets	$\mathcal{V}_1$ PSNR/SSIM	$\mathcal{V}_2$ PSNR/SSIM	$\mathcal{V}_3$ PSNR/SSIM	$\mathcal{V}_4^r/\mathcal{V}_4^s$ PSNR/SSIM	$\mathcal{V}_5$ PSNR/SSIM	$\mathcal{V}_6$ PSNR/SSIM
Rain1200	32.34/0.906	32.11/0.894	32.46/0.916	32.19/0.919	33.01/0.921	**33.07/0.927**
CSD	31.73/0.941	31.93/0.933	32.03/0.946	32.20/0.944	32.41/0.953	**32.50/0.957**

Table 3. PSNR and SSIM comparison of different variants of our model. Note that $\mathcal{V}_4^r/\mathcal{V}_4^s$ only handle rain/snow weather. Their decreased performance illustrates that they play positive roles as a part of our SmartAssign.

Datasets	$\mathcal{K}_1$ PSNR/SSIM	$\mathcal{K}_2$ PSNR/SSIM	Ours PSNR/SSIM
Rain1200	30.11/0.904	31.43/0.913	**33.07/0.927**
CSD	29.86/0.915	31.65/0.932	**32.50/0.957**

Table 4. The quantitative values consistently decrease after ablating the unique knowledge $\mathbf{F}_{rk}$/$\mathbf{F}_{sk}$ and the shared knowledge $\mathbf{F}_{ck}$, illustrating that they all functions positively during deraining and desnowing.

Figure 10. Ablating $\mathbf{F}_{rk}$, $\mathbf{F}_{ck}$ and $\mathbf{F}_{sk}$ decreases performance, showing their positive contributions to our model.

shows similar results, $\mathcal{K}_2$ removes more rain/snow than $\mathcal{K}_1$.

Limitations. Although SmartAssign achieves better performance than SOTA methods, there are still some open problems which are not completely solved by our model. The limitation of our SmartAssign mainly lies in desnowing task. Some fine snow traces may remain in the final results for some images with complex snowy patterns. An example is the second image in Figure 8. This can be understood as the light snowy traces with blurry patterns are difficult to be simulated by synthetic training pairs, which is a limitation of all existing methods including SmartAssign. In the future, we will focus on this limitation and attempt to solve it by introducing some effective priors.

5. Conclusion

In this paper, we proposed a novel multi-task learning strategy, SmartAssign, to develop the connections and uniqueness of rainy and snowy images, in which a novel knowledge assignment approach is proposed to adaptively determine the shared and exclusively-used knowledge atoms for deraining and desnowing. To better discriminate knowledge atoms, we proposed a novel knowledge contrast mechanism to enhance the similarity of knowledge atoms belonging to the same kind and highlight the discrimination of knowledge atoms belong to different kinds. Moreover, an effective transformer block was proposed to combine the long- and short-range modelling capacities of self-attention and convolution, respectively. Extensive experiments on six benchmark datasets have been conducted, which illustrate that our proposed SmartAssign is highly effective to both deraining and desnowing.

Acknowledgements

We gratefully acknowledge the support of MindSpore [1], CANN (Compute Architecture for Neural Networks) and Ascend AI Processor used for this research.

References

[1] Mindspore. https://www.mindspore.cn/.

[2] Hakan Bilen and Andrea Vedaldi. Integrated perception with recurrent multi-task neural networks. In *NeurIPS*, 2016.

[3] Pierre Charbonnier, Laure Blanc-Feraud, Gilles Aubert, and Michel Barlaud. Two deterministic half-quadratic regularization algorithms for computed imaging. In *ICIP*, 1994.

[4] Duanyu Chen, Chien-cheng Chen, and Liwei Kang. Visual depth guided color image rain streaks removal using sparse coding. *IEEE TCSVT*, 24(8):1430–1455, 2014.

[5] Weiting Chen, Jian-Jiun Ding, and Sy-Yen Kuo. Pms-net: robust haze removal based on patch map for single images. In *CVPR*, 2019.

[6] Weiting Chen, Hao-Yu Fang, Jian-Jium Ding, Cheng-Che Tsai, and Sy-Yen Kuo. Jstasr: joint size and transparency-aware snow removal algorithm based on modified partial convolution and veiling effect removal. In *ECCV*, 2020.

[7] Weiting Chen, Hao-Yu Fang, Cheng-Lin Hsieh, Cheng-Che Tsai, I-Hsiang Chen, Jian-Jium Ding, and Sy-Yen Kuo. All snow removed: single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss. In *ICCV*, 2021.

[8] Wei-Ting Chen, Zhi-Kai Huang, Cheng-Che Tsai, Hao-Hsiang Yang, Jian-Jiun Ding, and Sy-Yen Kuo. Learning multiple adverse weather removal via two-stage knowledge learning and multi-contrastive regularization: toward a unified model. In *CVPR*, 2022.

[9] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *ICCV*, 2013.

[10] Sourya Das and Saikat Dutta. Fast deep multi-patch hierarchical network for nonhomogeneous image dehazing. In *CVPRW*, 2020.

[11] Sen Deng, Mingqiang Wei, Jun Wang, Yidan Fang, Luming Liang, Haoran Xie, Fu Lee Wang, and Meng Wang. Detail-recovery image deraining via context aggregation networks. In *CVPR*, 2020.

[12] Xinghao Ding, Liqin Chen, Xianhui Zheng, Yue Huang, and Delu Zeng. Single image rain and snow removal via guided l0 smoothing filter. *ACM MTA*, 75(5):2697–2712, 2016.

[13] Nikita Dvornik, Konstantin Shmelkov, Julien Mairal, and Cordelia Schmid. Blitznet: a real-time deep network for scene understanding. In *ICCV*, 2017.

[14] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *CVPR*, 2017.

[15] Yuan Gao, Jiayi Ma, Mingbo Zhao, Wei Liu, and Alan L Yuille. Nddr-cnn: Layerwise feature fusing in multi-task cnns by neural discriminative dimensionality reduction. In *CVPR*, 2019.

[16] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. In *ECCV*, 2010.

[17] De-An Huang, Li-Wei Kang, Yu-Chiang Frank Wang, and Chia-Wen Lin. Self-learning based image decomposition with applications to single image denoising. *IEEE TMM*, 16(1):83–93, 2014.

[18] Junshi Huang, Rogerio S Feris, Qiang Chen, and Shuicheng Yan. Cross-domain image retrieval with a dual attribute-aware ranking network. In *ICCV*, 2015.

[19] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *CVPR*, 2020.

[20] B Jou and S. F. Chang. Deep cross residual learning for multi-task visual recognition. In *ACM MM*, 2016.

[21] Liwei Kang, Chia-wen Lin, and Yu-hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE TIP*, 21(4):1742–1755, 2012.

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[23] Iasonas Kokkinos. Ubernet: training a universal convolutional neural network for low-, mid-, and high-level vision using diverse datasets and limited memory. In *CVPR*, 2017.

[24] Hunsang Lee, Hyesong Choi, Kwanghoon Sohn, and Dongbo Min. Knn local attention for image restoration. In *CVPR*, 2022.

[25] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *CVPR*, 2022.

[26] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *CVPR*, 2019.

[27] Ruoteng Li, Robby T. Tan, and Loong-Fah Cheong. All in one bad weather removal using architectural search. In *CVPR*, 2020.

[28] Siyuan Li, Iago Breno Araujo, Wenqi Ren, Zhangyang Wang, Eric K. Tokuda, Roberto Hirata Junior, Roberto Cesar-Junior, Jiawan Zhang, Xiaojie Guo, and Xiaochun Cao. Single image deraining: a comprehensive benchmark analysis. In *CVPR*, 2019.

[29] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*, 2018.

[30] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *CVPR*, 2016.

[31] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Griddehazenet: attention-based multi-scale network for image dehazing. In *ICCV*, 2019.

[32] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: context-aware deep network for snow removal. *IEEE TIP*, 27(6):3064–3073, 2018.

[33] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, 2015.

[34] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *JMLR*, 9(11):2579–2605, 2008.

[35] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *JMLR*, pages 19–60, 2010.

[36] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert. Cross-stitch networks for multi-task learning. In *CVPR*, 2016.

[37] Chong Mou, Qian Wang, and Jian Zhang. Deep generalized unfolding networks for image restoration. In *CVPR*, 2022.

[38] Rajeev Ranjan, Vishal M. Patel, and Rama Chellappa. Hyperface: a deep multi-task learning framework for face detection, landmark localization, pose estimation, and gender recognition. *IEEE TPAMI*, 41(1):121–135, 2019.

[39] Sebastian Ruder. An overview of multi-task learning in deep neural networks, 2017.

[40] Sebastian Ruder, Joachim Bingel, Isabelle Augenstein, and Sogaard. Latent multi-task architecture learning. In *AAAI*, 2019.

[41] Yuanjie Shao, Lerenhan Li, Wenqi Ren, Changxin Gao, and Nong Sang. Domain adaptation for image dehazing. In *CVPR*, 2020.

[42] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: learning what to share for efficient deep multi-task learning. In *NeurIPS*, 2020.

[43] Zhengzhong Tu, Hossein Talebi, Han Zhang, Feng Yang, Peyman Milanfar, Alan Bovik, and Yinxiao Li. Maxim: Multi-axis mlp for image processing. In *CVPR*, 2022.

[44] Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M. Patel. Transweather: Transformer-based restoration of images degraded by adverse weather conditions. In *CVPR*, 2022.

[45] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *CVPR*, 2020.

[46] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*, 2019.

[47] Yinglong Wang, Chen Chen, Shuyuan Zhu, and Bing Zeng. A framework of single-image deraining method based on analysis of rain characteristics. In *ICIP*, 2016.

[48] Yinglong Wang, Dong Gong, Jie Yang, Qinfeng Shi, Anton van den Hengel, Dehua Xie, and Bing Zeng. Deep single image deraining via modeling haze-like effect. *IEEE TMM*, 23:2481–2492, 2020.

[49] Yinglong Wang, Shuaicheng Liu, Chen Chen, and Bing Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE TIP*, 26(8):3936–3950, 2017.

[50] Yinglong Wang, Chao Ma, and Bing Zeng. Multi-decoding deraining network and quasi-sparsity based training. In *CVPR*, 2021.

[51] Yinglong Wang, Yibing Song, Chao Ma, and Bing Zeng. Rethinking image deraining via rain streaks and vapors. In *ECCV*, 2020.

[52] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[53] Zhendong Wang, Xiaodong Cun, Jianming Bao, Wengang Zhou, Jianzhuang Liu, and Houqiang Li. Uformer: a general u-shaped transformer for image restoration. In *CVPR*, 2022.

[54] Haiyan Wu, Yanyun Qu, Shaohui Liu, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Contrastive learning for compact single image dehazing. In *CVPR*, 2021.

[55] 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 *CVPR*, 2017.

[56] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Shuicheng Yan, and Zongming Guo. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE TPAMI*, 42(6):1377–1393, 2019.

[57] Wenhan Yang, Ye Yuan, Wenqi Ren, and Jiaying Liu et al. Advancing image understanding in poor visibility environments: a collective benchmark study. *IEEE TIP*, 29:5737–5752, 2020.

[58] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: efficient transformer for high-resolution image restoration. In *CVPR*, 2022.

[59] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *CVPR*, 2021.

[60] He Zhang and Vishal M. Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, 2018.

[61] Xianhui Zheng, Yinghao Liao, Wei Guo, Xueyang Fu, and Xinghao Ding. Single-image-based rain and snow removal using multi-guided filter. In *ICONIP*, 2013.

[62] Man Zhou, Jie Xiao, Yifan Chang, Xueyang Fu, Aiping Liu, Jinshan Pan, and Zheng-Jun Zha. Image de-raining via continual learning. In *CVPR*, 2021.