TransWeather: Transformer-based Restoration of Images Degraded by Adverse Weather Conditions

Jeya Maria Jose Valanarasu, Rajeev Yasarla, and Vishal M. Patel
Johns Hopkins University
{jvalana1,ryasarl1,vpatel36} @jhu.edu

Abstract

Removing adverse weather conditions like rain, fog, and snow from images is an important problem in many applications. Most methods proposed in the literature have been designed to deal with just removing one type of degradation. Recently, a CNN-based method using neural architecture search (All-in-One) was proposed to remove all the weather conditions at once. However, it has a large number of parameters as it uses multiple encoders to cater to each weather removal task and still has scope for improvement in its performance. In this work, we focus on developing an efficient solution for the all adverse weather removal problem. To this end, we propose TransWeather, a transformer-based end-to-end model with just a single encoder and a decoder that can restore an image degraded by any weather condition. Specifically, we utilize a novel transformer encoder using intra-patch transformer blocks to enhance attention inside the patches to effectively remove smaller weather degradations. We also introduce a transformer decoder with learnable weather type embeddings to adjust to the weather degradation at hand. TransWeather achieves significant improvements across multiple test datasets over both All-in-One network as well as methods fine-tuned for specific tasks. TransWeather is also validated on real world test images and found to be more effective than previous methods. Implementation code can be found in the supplementary document. Code is available at https://github.com/jeya-maria-jose/TransWeather.

1. Introduction

Weather conditions like rain, fog, and snow reduce the visibility and corrupt the information captured by an image. This drastically affects the performance of many computer vision algorithms like detection, segmentation and depth estimation [3, 5, 41, 52, 59] which are important parts of autonomous navigation and surveillance systems [28, 34–36]. Hence, it is essential to remove adverse weather effects from images in order to make these vision systems more reliable. Also, a clean image without any weather degradation is desired in photography. Early methods for weather removal involve modelling priors for weather conditions using empirical observations [13, 45, 46]. These priors have to be modelled separately for each weather condition and a common prior modelled for all weather conditions is not effective. Recently, Convolutional Neural Networks (CNNs) based solutions have been explored extensively for deraining [11, 37, 53, 56, 61, 63, 69, 70, 76], dehazing [8, 19, 41, 57, 69, 71, 72], desnowing [29, 44, 73] and raindrop removal [37, 40, 66]. Transformer-based methods

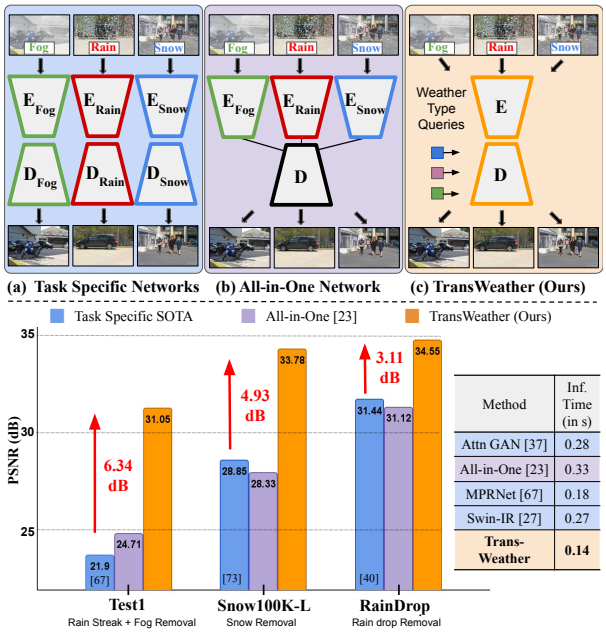

Figure 1. *Top row:* **Weather Removal Frameworks -** (a) Separate networks designed for each type of weather removal like rain, fog and snow. (b) All-in-One Network [23] proposes a framework with separate encoders for each task but a generic decoder. (c) Our proposed method, Transweather, has a single encoder and a decoder and learns weather type queries to solve all adverse weather removal efficiently. *Bottom row:* **Performance Comparison -** A *single model instance* of TransWeather achieves significant performance boost when compared to both All-in-One framework and state-of-the-art networks designed and trained individually for specific tasks while also being faster during inference.

have also been explored for weather removal tasks achieving better performance than CNNs [38, 48, 74]. Most of these methods just focus on one task at hand or fine-tune the model separately for each task. Although they achieve excellent performance, these are not generic solutions for all adverse weather removal problems as the networks have to trained separately for each task. This makes it difficult to adopt them for real-time systems as there have to be multiple models making it computationally complex. Also, the system would have to decide and switch between a series of weather removal algorithms (Figure 1 (a)) making the pipeline more complicated.

Recently, Li *et al*. [23] proposed an All-in-One bad weather removal network which was the first work to propose an algorithm that takes in an image degraded by any weather condition as input and predicts the clean image. All-in-One network was tested across 3 datasets of rain, fog, and snow removal and achieved better or comparable performance than the previous methods which were tuned individually on separate datasets. All-in-One network is CNN-based and uses multiple encoders. In particular, it uses separate encoders for the different weather degradation at hand and uses neural architecture search to find the best network to address the problem (Figure 1 (b)). This network is still computationally complex as there are multiple encoders. To the best of our knowledge, no other methods apart from All-in-One network [23] have been proposed for a generic adverse weather removal in the literature. Although recent methods like MPR-Net [67], U-former [55], Swin-IR [27] have been proposed as generic restoration networks validated on multiple datasets, they are still fine-tuned on the individual datasets and do not use a single model for all the weather removal tasks.

In this work, we propose a single encoder-single decoder transformer network, called TransWeather, to tackle all adverse weather removal problems at once. Instead of using multiple encoders, we introduce weather type queries in the transformer decoder to learn the task (Figure 1 (c)). Here, the multi-head self attention mechanisms take in weather type queries as input and match it with keys and values taken from features extracted from the transformer encoder. These weather type embeddings are learned along with the network to understand and adjust to the weather degradation type present in the image. The decoded features and the hierarchical features obtained form the encoder are fused and projected to the image space using a convolutional block. Thus, TransWeather just has one encoder and one decoder to learn the weather type as well as produce the clean image. Transformers are good at extracting rich global information when compared to CNNs [9]. However, we argue that when the patches are large like in ViT [9], we fail to attend much to the information within the patch. Weather degradations like rain streak, rain drop and snow are usually small in size and so multiple artifacts can occur within a single patch.

To this end, we propose a novel transformer encoder with intra-patch transformer (Intra-PT) blocks. Intra-PT works on sub-patches created from the original patches and excavates features and details of smaller patches. Intra-PT thus focuses on attention inside the main patches to remove weather degradations effectively. We use efficient self-attention mechanisms to calculate the attention between sub-patches to keep the computational complexity low. From our experiments, we find that introducing Intra-PT blocks enhances the performance of transformer and helps it adapt better to weather removal tasks. We train our network on a similar configuration as All-in-One and obtain superior performance across multiple test datasets for rain removal, snow removal, fog removal and even a combination of these weather degradations. We also outperform the methods designed specifically for these individual tasks which are finetuned on those datasets. We also show that TransWeather is fast during inference. Finally, we also test TransWeather on real-world weather degraded images, achieving excellent performance compared to the previous methods. TransWeather can act as an efficient backbone in the future for generic weather removal frameworks.

The key contributions of this work are as follows:

- We propose TransWeather - an efficient solution for all adverse weather removal problem with just a single encoder and a single decoder using transformers. We propose using weather type queries to efficiently handle the All-in-One problem.
- We propose a novel transformer encoder using intra-patch transformer (Intra-PT) blocks to cater to fine detail feature extraction for low-level vision tasks like weather removal.
- We achieve state-of-the-art performance on multiple datasets. We also validate the effectiveness of the proposed method on real-world images.

2. Related Works

Adverse weather removal problems like deraining [16, 21, 25, 51, 61, 65, 76], dehazing [1, 2, 10, 22, 42, 69], desnowing [29, 44, 44, 73] and rain drop removal [37, 39, 40, 66] have been extensively explored in the literature.

Rain Streak Removal: Yang et al. [61] used a recurrent network to decompose rain layers to different layers of various streak types to remove the rain. Zhang et al. [70] proposed using a conditional GAN for image deraining. Yasarla et al. [64] explored using Gaussian processes to perform transfer learning from synthetic rain data to real-world rain data. Quan et al. [39] used a complementary cascaded network to remove rain streaks and raindrops in a unified framework. A more detailed survey of various rain removal methods can be found in [62].

Fog Removal: Li et al. [18] proposed a CNN network considering both atmospheric light and transmission map to

perform dehazing. Ren et al. [43] proposed pre-processing a hazy image to generate multiple inputs thus introducing color distortions to perform dehazing. Zhang and Patel [68] proposed a pyramid CNN network for image dehazing. Zhang et al. [72] proposed a hierarchial density aware network for image dehazing.

Rain drop Removal: You et al. [66] proposed using temporal information to perform video-based raindrop removal. Qian et al. [37] used an attention GAN to remove raindrop and also introduced a new dataset. Quan et al. [40] used a dual attention mechanism to remove effects of raindrops.

Snow Removal: Desnow-Net [29] was one of the first CNN-based methods proposed to remove snow from an image. Li et al. [20] proposed a stacked dense network for snow removal. Chen et al. [6] proposed JSTASR in which a size and transparency aware method was proposed to remove snow. Recently, DDMSNet [72] proposed a deep dense multiscale network using semantic and geometric priors for snow removal.

All-in-One Weather Removal: All-in-One Network [23] was proposed to handle multiple weather degradations using a single network. All-in-One uses a generator with multiple task-specific encoders and a common decoder. It uses a discriminator to classify the degradation type and only backpropagates the loss to specific encoders. It also uses neural architecture search to optimize the feature extraction from the encoder.

Transformers in low-level vision: Since the introduction of Vision Transformer (ViT) [9] for visual recognition, transformers have been widely adopted for various computer vision tasks [12, 31, 49, 60, 75]. Especially for low-level vision, Image processing transformer [4] shows how pretraining a transformer on large-scale datasets can help in obtaining a better performance for low-level applications. U-former [55] proposed a U-Net based transformer architecture for restoration problems. Swin-IR [27] adopted Swin Transformer [30] for image restoration. Zhao et al. [74] proposed a local-global transformer specifically for image dehazing. A multi-branch network [48] for deraining was also proposed based on swin transformer. In ETDNet [38], an efficient transformer block to extract features in a coarse to fine way for image deraining was proposed. 粗粒度

Unlike the above methods, we propose a transformer-based single-encoder single-decoder network to solve all adverse weather removal tasks using a single model instance. Our Transformer encoder is also modified to cater to low-level tasks with the introduction of intra-patch transformer block. Our transformer decoder is trained with weather type queries to learn the task and uses that information to restore the clean image.

3. Proposed Method - TransWeather

In the literature, different weather phenomenons have been modelled differently with regards to the underlying physics involved. Rain drop [37] is modelled as

$$\mathbf{I} = (1 - \mathbf{M}) \odot \mathbf{B} + \mathbf{R}, \tag{1}$$

where $\mathbf{I}$ is the degraded image, $\mathbf{M}$ is the mask, $\mathbf{B}$ is the background and $\mathbf{R}$ is the raindrop residual. 指带雾|雨等的图片 Heavy rain with rain streaks and fog effect [21] is modelled as

$$\mathbf{I} = \mathbf{T} \odot (\mathbf{B} + \sum_{i}^{n} \mathbf{R}_i) + (1 - \mathbf{T}) \odot \mathbf{A}, \tag{2}$$

where $\mathbf{T}$ is the transmission map produced by scattering effect, and $\mathbf{A}$ is the atmospheric light in the scene. According to [29], snow is generally modeled as

$$\mathbf{I} = \mathbf{M} \odot \mathbf{S} + \mathbf{M} \odot (1 - \mathbf{z}), \tag{3}$$

where $\mathbf{z}$ is a mask indicating snow and $\mathbf{S}$ corresponds to snow flakes. All-in-One method [23] generalizes the adverse weather removal problem as

$$\mathbf{B} = \mathbf{D}(\mathbf{E}_p(\mathbf{I}_p)), \tag{4}$$

where $\mathbf{E}$ corresponds to the encoder and $\mathbf{D}$ corresponds to the decoder. p represents the weather type present in the image. Note that for each weather type a different encoder is used. In this work, we follow a similar formulation of all adverse weather removal as

$$\mathbf{B} = \mathbf{T}(\mathbf{I}_p), \tag{5}$$

where $\mathbf{T}$ corresponds to TransWeather which consists of a weather agnostic 未知 encoder and decoder network unlike All-in-One Network. The weather type queries are learnt along with the parameters of $\mathbf{T}()$ thus making the problem setup more generic. We motivate this setup because a problem as generic as weather removal cannot be addressed by merely seeking for perfection on solving individual tasks. This formulation not only makes the process computationally efficient, but also helps in using complimentary information between the tasks to further improve the performance. Furthermore, it is also grounded with regards to how human vision works as our visual cortex 视觉皮层 can perform multiple tasks without any difficulty. This view is widely agreed in neurobiology as the visual cortex does not have different modules for different perception tasks [24, 32].

3.1. Network Architecture

Given a degraded image $\mathbf{I}$ of size $H \times W \times 3$, we first divide it into patches. We then feed forward the patches to a transformer encoder containing transformer blocks at different stages. Across each stage, the resolution is reduced to make sure the transformer learns both coarse and fine information. We then use a transformer decoder block that uses the encoded features as keys and values while using learnable weather type query embeddings as queries. The extracted features are then passed through a convolutional projection block to get the clean image of dimensions $H \times W \times 3$. An overview of the network architecture of

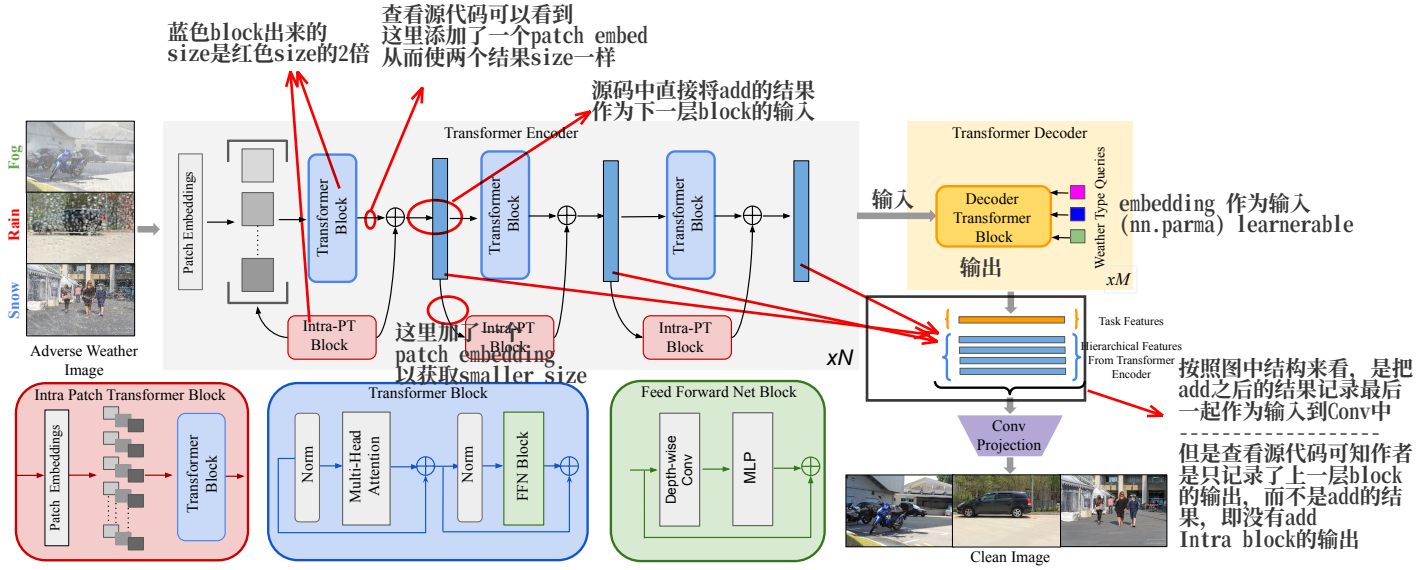

Figure 2. **Overview of the proposed TransWeather network.** A degraded image is forwarded to transformer encoder to extract hierarchical features. The encoder has intra-patch transformer blocks to extract features from smaller sub-patches created from the main patch. The transformer decoder has learnable weather type queries to obtain the task feature. Then, the hierarchical features from encoder as well as the task feature from decoder are forwarded to a convolutional projection block to obtain the clean image.

TransWeather can be found in Figure 2. In the following sections, we describe these components in detail.

3.1.1 Transformer Encoder

We generate a hierarchical feature representation of the input image by extracting multi-level features in the transformer encoder. The features are extracted at different stages in the encoder thus facilitating extraction of both high-level and low-level features. Across each stage, we perform overlapped patch merging [59]. Using this we combine overlapping feature patches to get features of the same size as that of non-overlapped patches before passing the features to the next stage结合重叠特征块以获得与非重叠块
相同大小的特征，然后将特征传递到下一阶段

Transformer Block: In each transformer block, we use multi-head self-attention layers and feed forward networks to calculate the self-attention features. The computation can be summarized as:

$$\mathbf{T}_i(\mathbf{I}_i) = FFN(MSA(\mathbf{I}_i) + \mathbf{I}_i), \tag{6}$$

where $\mathbf{T}()$ represents the transformer block, $FFN()$ represents the feed forward network block, $MSA()$ represents multi head self-attention, $\mathbf{I}$ is the input and i represents the stage in the encoder. Similar to the original self-attention network, the heads of queries ($\mathbf{Q}$), keys ($\mathbf{K}$) and values ($\mathbf{V}$) have same dimensions and are calculated as:

$$\text{Attn}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V}, \tag{7}$$

where d represents the dimensionality. Note that we have multiple attention heads in each transformer block and that number is a hyper-parameter which we vary across each stage in the transformer encoder. More details regarding the hyper-parameter settings can be found in the supplementary document. We reduce the complexity of the original self-attention from $O(N^2)$ to $O(\frac{N^2}{R})$ by introducing a reduction ratio R [54]. We reshape the keys into a dimension from a dimension of (N, C) to a dimension of $(\frac{N}{R}, C.R)$. We then use a linear layer to get the second dimension back to C from $C.R$. Hence, the keys get a dimension of $\frac{N}{R} \times C$ thus reducing the complexity while calculating the self attention. The self-attention features are then passed to a FFN block. The FFN block used here has a slight variation from ViT as we introduce using depth-wise convolution to MLP inspired from [26, 58, 59]. Using depth-wise convolution here helps bring locality information and provide positional information for transformers as shown in [59]. The computation in the FFN block can be summarized as follows:

$$FFN_i(\mathbf{X}_i) = MLP(GELU(DWC(MLP(\mathbf{X}_i)))) + \mathbf{X}_i,$$

where $\mathbf{X}$ refers to self-attention features, DWC is depth-wise convolution [7], $GELU$ is Gaussian error linear units [14], MLP is multi-layer perceptron, i indicates the stage.

Intra-Patch Transformer Block: The intra-patch transformer blocks are present in between each stage in the transformer encoder. These blocks take in the sub-patches created from the original patches as the input. These sub-patches are fixed in dimensions half of height and width of the original patch. Intra-PT also utilizes a similar transformer block as explained above. We use a high R value to make the computation very efficient in the Intra-PT block. Intra-PT block helps in extracting fine details helpful in removing smaller degradation as we operate on smaller patches. Note that the Intra-PT block creates patches at the feature level except at the first stage where it is done at the image level. The output self-attention features from the Intra-PT block are added to the self-attention features from the main block across the same stage. The formulation of feed forward process in our transformer encoder can be summarized as follows:

$$\mathbf{Y}_i = MT_i(\mathbf{X}_i) + IntraPT_i(P(\mathbf{X}_i)) \tag{8}$$

where $\mathbf{I}$ is input to the transformer across each stage, $\mathbf{Y}$ is the output across each stage, $MT()$ is the main transformer

block, $IntraPT$ is the intra-patch transformer block, $P()$ corresponds to the process of creating sub-patches from the input patches and i denotes the stage.

3.1.2 Transformer Decoder

In the original transformer decoder [50], an autoregressive decoder is used to predict the output sequence one element at a time. Detection transformer (DETR) [3] uses object queries to decode the box coordinates and class labels to produce the final predictions. Inspired from them, we define weather type queries to decode the task, predict a task feature vector and use it to restore the clean image. These weather type queries are learnable embeddings which are learnt along with the other parameters of our network. These queries attend to the feature outputs from the transformer encoder. The transformer decoder here operates at a single stage but has multiple blocks. We illustrate the transformer decoder block in Figure 3. These transformer blocks are similar to encoder-decoder transformer blocks [50]. Unlike self-attention transformer block where **Q**, **K** and **V** are taken from the same input, here **Q** is weather type learnable embedding while **K** and **V** are the features taken from the last stage of the transformer encoder. The output decoded features represent the task feature vector and are fused with the features extracted across the transformer encoder at each stage. All of these features are forwarded to the convolutional tail to reconstruct the clean image.

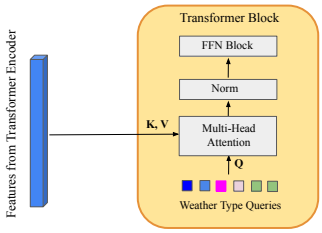

Figure 3. **Configuration of the transformer block in the decoder.** The queries here are learnable embeddings representing the weather type while the keys and values are features taken from the last stage of the transformer encoder.

3.1.3 Convolutional Projection Block

The set of hierarchical transformer encoder features and task features from the transformer decoder are passed through a set of 4 convolutional layers to output the clean image. We also use an upsampling layer before every convolutional layer to get back to the original image size. We also have skip connections across each stage in the convolutional tail from the transformer encoder. We also use a *tanh* activation function in the final layer.

3.2. Loss

Our network is trained in an end-to-end fashion using a smooth L1-loss between the prediction ($\hat{\mathbf{I}}$) and the ground truth ($\mathbf{G}$) defined as follows:

$$\mathcal{L}_{smoothL_1} = \begin{cases} 0.5\mathbf{E}^2 & \text{if } |\mathbf{E}| < 1 \\ |\mathbf{E}| - 0.5 & \text{otherwise}, \end{cases} \tag{9}$$

where $\mathbf{E} = \hat{\mathbf{I}} - \mathbf{G}$. We also add a perceptual loss that measures the discrepancy between the features of prediction and the ground truth. We extract these features using a VGG16 network [47] pretrained on ImageNet. We extract features from the 3^{rd}, 8^{th} and 15^{th} layers of VGG16 to calculate the perceptual loss. The perceptual loss is formulated as follows

$$\mathcal{L}_{perceptual} = \mathcal{L}_{MSE}(VGG_{3,8,15}(\hat{\mathbf{I}}), VGG_{3,8,15}(\mathbf{G})).$$

The total loss can be summarized as follows

$$\mathcal{L}_{total} = \mathcal{L}_{smoothL_1} + \lambda \mathcal{L}_{perceptual}, \tag{10}$$

where λ is a weight that controls the contribution of $\mathcal{L}_{perceptual}$ and L1-loss on the overall loss.

4. Experiments

We conduct extensive experiments to show the effectiveness of our proposed method. In what follows, we explain the datasets, implementation details, experimental settings, results and comparison with state-of-the-art methods.

4.1. Datasets

We train our network on a combination of images degraded from a variety of adverse weather conditions similar to All-in-One Network [23]. We follow the same training set distribution used in All-in-One for fair comparison. The training data consists of 9,000 images sampled from Snow100K [29], 1,069 images from Raindrop [37] and 9,000 images of Outdoor-Rain [21]. Snow100K has synthetic images degraded by snow, raindrop has real raindrop images and Outdoor-Rain has synthetic images degraded by both fog and rain streaks. We term this combination of training data as "All-Weather" for better representation.

We test our methods on both synthetic and real-world datasets. We use the Test1 dataset [21, 23], the RainDrop test dataset [37] and the Snow100k-L test set [29] for testing our method. In addition, we also evaluate on real-world images degraded by rain streaks and rain drops.

4.2. Implementation Details

We implement our method using Pytorch framework [33] and train it using an NVIDIA RTX 8000 GPU. We use an Adam optimizer [17] and a learning rate of 0.0002. We use a learning rate scheduler that anneals the learning rate by 2 after 100 and 150 epochs. The network is trained for a total of 200 epochs with a batch size of 32. Other hyperparameters regarding the TransWeather architecture can be found in the supplementary document.

4.3. Comparison with state-of-the-art methods

First, we compare our method with state-of-the-art methods which are designed specifically for each task: rain drop

Type	Method	Venue	PSNR (↑)	SSIM (↑)
Task Specific	DetailsNet + Dehaze (DHF) [11]	CVPR 2017	13.36	0.5830
	DetailsNet + Dehaze (DRF) [11]	*CVPR 2017*	15.68	0.6400
	RESCAN + Dehaze (DHF) [25]	*ECCV 2018*	14.72	0.5870
	RESCAN + Dehaze (DHF) [25]	*ECCV 2018*	15.91	0.6150
	pix2pix [15]	*CVPR 2017*	19.09	0.7100
	HRGAN [21]	*CVPR 2019*	21.56	0.8550
	Swin-IR [27]	*CVPR 2021*	23.23	0.8685
	MPRNet [67]	*CVPR 2021*	21.90	0.8456
Multi Task	All-in-One [23]	*CVPR 2020*	**24.71**	**0.8980**
	TransWeather	-	**31.05**	**0.9509**

Table 1. **Quantitative Comparison on the Test1 (rain+fog) dataset based on PSNR and SSIM.** DHF represents De-Hazing First and DRF represents De-Raining First. **Red** and **Blue** corresponds to first and second best results. ↑ means higher the better.

Type	Method	Venue	PSNR (↑)	SSIM (↑)
Task Specific	DetailsNet [11]	*CVPR 2017*	19.18	0.7495
	DesnowNet [29]	TIP 2018	27.17	0.8983
	JSTASR [6]	ECCV 2020	25.32	0.8076
	Swin-IR [27]	CVPR 2021	28.18	0.8800
	DDMSNET [73]	TIP 2021	**28.85**	0.8772
Multi Task	All-in-One [23]	CVPR 2020	28.33	**0.8820**
	TransWeather	-	**33.78**	**0.9287**

Table 2. **Quantitative Comparison on the SnowTest100k-L test dataset based on PSNR and SSIM. Red** and **Blue** corresponds to first and second best results. ↑ means higher the better.

Type	Method	Venue	PSNR (↑)	SSIM (↑)
Task Specific	Pix2pix [15]	*CVPR 2017*	28.02	0.8547
	Attn. GAN [37]	*CVPR 2018*	30.55	0.9023
	Quan et al. [40]	*ICCV 2019*	**31.44**	0.9263
	Swin-IR [27]	*CVPR 2021*	30.82	0.9035
	CCN [39]	*CVPR 2021*	31.34	**0.9500**
Multi Task	All-in-One [23]	*CVPR 2020*	31.12	0.9268
	TransWeather	-	**34.55**	**0.9502**

Table 3. **Quantitative comparison on the RainDrop test dataset based on PSNR and SSIM. Red** and **Blue** corresponds to first and second best results. ↑ means higher the better.

removal, snow removal and rain+haze removal. For rain drop removal, we compare the performance with state-of-the-art methods like Attention GAN [37], Quan et al. [40], and complementary cascaded network (CCN) [39]. For snow removal, we compare with Desnow-Net [29], JSTASR [6] and Deep Dense Multi-Scale Network (DDMSNet) [73]. For rain+fog removal, we compare with HRGAN [21], Details-Net [11], Recurrent squeeze-and-excitation context aggregation Net (RESCAN) [25], and Multi-Stage Progressive Restoration Network (MPRNet) [67]. We also compare with a recent transformer network Swin-IR [27] for all datasets. Note that all these methods are single-task handling networks which are fine-tuned for specific datasets.

We also compare the performance of our method with All-in-One network [23] which is trained to perform all the above tasks with a *single model instance*. Our method TransWeather is also trained to perform all these tasks using a *single model instance*.

4.3.1 Referenced Quality Metrics

We use PSNR and SSIM to evaluate the performance of different models. We tabulate the quantitative results in terms of PSNR and SSIM in Tables 1, 2, and 3 while evaluating on the Test1 (fog+rain removal), Snow100K-L (snow removal) and RainDrop (rain drop removal) test datasets, respectively. As Test1 has both fog and rain, we sequentially apply deraining and dehazing methods for fair comparison on this dataset. For example, while using Details-Net and RESCAN for deraining, we apply Multi-scale boosted dehazing network (MSBDN) [8] for dehazing. Note that from our experiments we found MSBDN to be the best performing network for dehazing. We compare the performance while applying deraining first, then dehazing and also vice-versa. We train Swin-IR and MPRNet directly on "Outdoor-Rain" (training split of Test1) and test it on Test1 for fair comparison. Similarly, Swin-IR was trained on Snow100K dataset, RainDrop and tested on SnowTest100k-L, RainDrop test datasets respectively. It can be noted that some recent methods like CCN and DDMSNet when fine-tuned on the individual datasets outperform All-in-One. TransWeather outperforms All-in-One as well as all the task-specific methods by a significant margin as we cater to low-level weather details as well as use weather queries to efficiently handle the All-in-One problem.

4.3.2 Visual Quality Comparison

Synthetic Images We illustrate the predictions from synthetic test datasets like Test1 and Snow100k-L in Figures 4 and 5. It can be seen that Transweather achieves visually pleasing results compared to the previous methods. It works very well in removing both fog and rain streaks as can be seen in Figure 4 while other methods including All-in-One fail to remove at least one of the degradations. It can be seen from Figure 5 that our method removes even the snow particles which are very small in structure while All-in-One has hard time removing them.

Real-World Images We illustrate the predictions from real test datasets like RainDrop and Real-World images in Figures 6 and 7. It can be seen in both the figures that Tran-

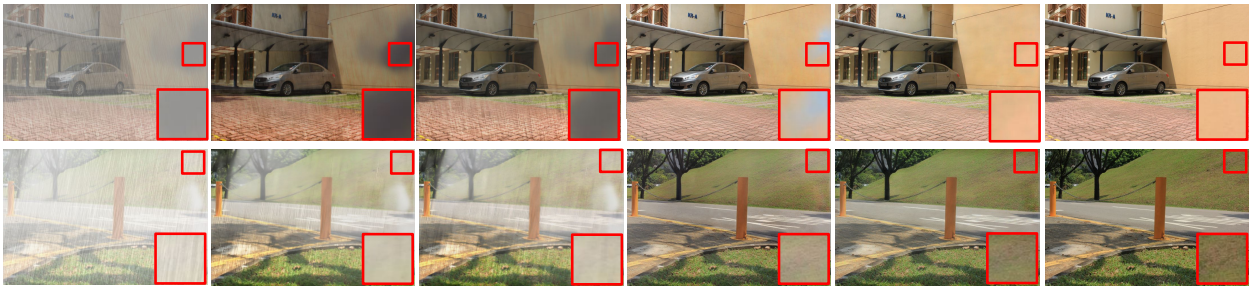

Figure 4. **Sample qualitative results on the Test1 dataset.** Red box corresponds to the zoomed-in patch for better comparison.

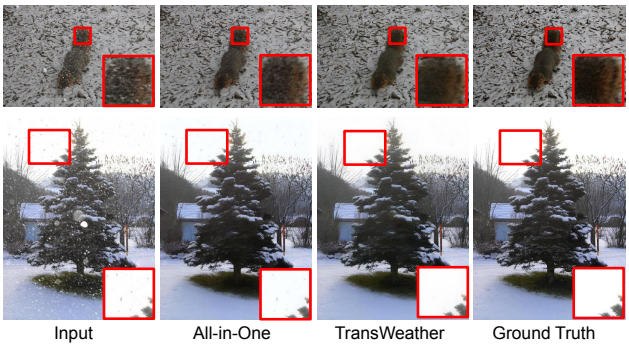

Figure 5. **Sample qualitative results on the Snow100k-L dataset.** Red box corresponds to the zoomed-in patches.

sweather removes even the finest rain streaks or drops when compared to the previous methods.

5. Discussions

Ablation Study: We conduct an ablation study to understand the contributions of individual components proposed in the TransWeather architecture. We start with a base transformer encoder architecture and a conv tail. We call this configuration Transformer Base. We then convert the transformer encoder to hierarchical transformer (HE) encoder to extract both high-level and low-level features where we perform patch merging between each stage in the transformer encoder. We then add the intra patch transformer block (Intra-PT) in the encoder. Then we add learnable weather type queries and a transformer decoder block to learn the task embeddings. This configuration corresponds to the TransWeather architecture. All of these models are trained on All-Weather and tested on the Raindrop test dataset. The results of ablation study can be found in Table 4. It can be observed that each individual contribution of this work helps in improving the performance.

Method	PSNR (↑)	SSIM (↑)
Transformer Base	30.12	0.8512
+ HE	31.62	0.8671
+ HE + Intra-PT	32.37	0.9463
+ HE + Intra-PT +Weather Queries	34.55	0.9502

Table 4. **Ablation Study on the RainDrop test dataset.** HE denotes converting to hierarchical transformer encoder and Intra-PT represents intra-patch transformer blocks.

What do the weather queries learn? The weather queries are embeddings which learn what type of degradation is present in the image. These queries help in predicting the corresponding task vector which is helpful to inject the task information to get a better prediction. To show this, we visualize the attention maps of eight random queries (out of 512) for three images corresponding to different weather degradations in Figure 8. It is interesting to observe that queries *Q1*, *Q3*, and *Q6* activate highly for foggy image. They attend throughout the image to all the places afflicted by the fog. Queries *Q2*, *Q4* and *Q8* are observed to activate highly for rainy images and the attention maps are sparse corresponding to the rain details. Similarly, queries *Q5* and *Q7* activate to snow images more when compared to images with rain and fog. This shows that different queries activate for different weather degredations helping TransWeather learn the underlying weather condition and give better predictions. It can also be noted that when an image is degraded by multiple weather conditions, multiple task type queries activate to encode specific tasks. This can be observed from the middle row of Figure 8 where queries that attend to both fog and rain activate as the image is degraded by both of these conditions.

Inference Time: In Figure 1 (bottom row), we compare the inference speed in terms of seconds. The time reported in the table corresponds to the time taken by each model feed forward an image of dimensions 256×256 during the inference stage. We note that our method is faster (with just 0.14 seconds per image) during inference when compared to the previous weather removal methods. TransWeather has 31 M parameters which are less than that of All-in-One Network which has 44 M parameters.

Differences from All-in-One: As the All-in-One network [23] is the first method to look into using a single model instance for all weather removal problems, we present clear differences of our method from All-in-One. First, All-in-One is a CNN-based method while TransWeather uses a transformer backbone built specifically for low-level vision tasks with an extra focus on operating on smaller patches. All-in-One uses multiple encoders while TransWeather utilizes a single encoder. All-in-One uses adversarial training and neural architecture search while TransWeather just uses a combination of L1 and perceptual loss making the train-

Figure 6. **Sample qualitative results on the RainDrop dataset.** Red box corresponds to the zoomed-in patches for better comparison.

Figure 7. **Sample qualitative results on the Real-World images.** Red box corresponds to the zoomed-in patch for better comparison. Note that these are real-world images with no availability of ground truth.

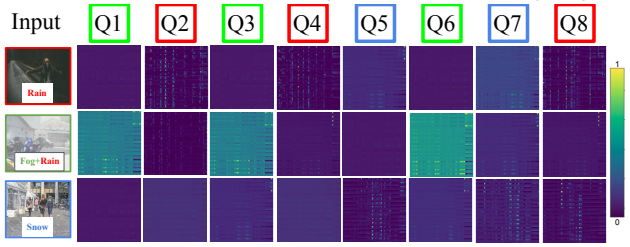

Figure 8. **Attention maps with respect to different queries.** Rows correspond to input image with different weather degradations and the columns correspond to attention maps with different queries (*Q1* to *Q8*). **Red**, **Green**, and **Blue** boxes correspond to queries that activate most to *Rain*, *Fog* and *Snow* respectively. Best viewed when zoomed in and in color.

ing more stable. TransWeather also has a faster inference speed, lesser number of parameters while obtaining better quantitative and qualitative performance.

Limitations: Although TransWeather achieves better performance than previous methods, there are still some open problems that TransWeather does not solve. TransWeather does not perform well in some real world images afflicted by high intensity rains. This can be understood as sometimes real-rain is very different in terms of streak size and intensity and are difficult to model. Moreover, if the intensity of rain is high, it creates a splattering effect when it hits the surface of objects or people in the scene. Removing this splattering effect is still a limitation by all methods including TransWeather. A sample limitation is illustrated in Figure 9.

Figure 9. **Limitations of our method:** High intensity rain and the splattering effect of rain is not removed by any method.

6. Conclusion

In this work, we proposed TransWeather - an efficient transformer-based solution for the all adverse weather removal problem. We focus on building a *single model instance* which can remove any weather degradation present in the image. We build a single encoder-decoder network for restoration while using learnable weather type queries in the decoder to learn the type of weather degradation and use that information for the weather removal process. We also propose a novel transformer encoder architecture base which work on sub-patches thus aiding transformers to remove small weather degradations more efficiently. We extensively experiment on multiple synthetic and real-world datasets where we push the current state-of-the-art by a significant amount using a *single model instance* while also obtaining a faster inference speed. We also obtain better visual results when tested on real-world adverse weather images.

Acknowledgement: This work was supported by an ARO grant W911NF-21-1-0135.

References

[1] Dana Berman, Shai Avidan, et al. Non-local image dehazing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1674–1682, 2016. 2

[2] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016. 2

[3] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 1, 5

[4] Hanting Chen, Yunhe Wang, Tianyu Guo, Chang Xu, Yiping Deng, Zhenhua Liu, Siwei Ma, Chunjing Xu, Chao Xu, and Wen Gao. Pre-trained image processing transformer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12299–12310, 2021. 3

[5] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *Proceedings of the European conference on computer vision (ECCV)*, pages 801–818, 2018.

[6] Wei-Ting Chen, Hao-Yu Fang, Jian-Jiun 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 *European Conference on Computer Vision*, pages 754–770. Springer, 2020. 3, 6

[7] François Chollet. Xception: Deep learning with depthwise separable convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1251–1258, 2017. 4

[8] Hang Dong, Jinshan Pan, Lei Xiang, Zhe Hu, Xinyi Zhang, Fei Wang, and Ming-Hsuan Yang. Multi-scale boosted dehazing network with dense feature fusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2157–2167, 2020. 1, 6

[9] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2, 3

[10] Raanan Fattal. Dehazing using color-lines. *ACM transactions on graphics (TOG)*, 34(1):1–14, 2014. 2

[11] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3855–3863, 2017. 1, 6

[12] Meng-Hao Guo, Jun-Xiong Cai, Zheng-Ning Liu, Tai-Jiang Mu, Ralph R Martin, and Shi-Min Hu. Pct: Point cloud transformer. *Computational Visual Media*, 7(2):187–199, 2021. 3

[13] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010. 1

[14] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 4

[15] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017. 6

[16] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE transactions on image processing*, 21(4):1742–1755, 2011. 2

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 5

[18] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, pages 4770–4778, 2017. 2

[19] Pengyue Li, Jiandong Tian, Yandong Tang, Guolin Wang, and Chengdong Wu. Deep retinex network for single image dehazing. *IEEE Transactions on Image Processing*, 30:1100–1115, 2020. 1

[20] Pengyue Li, Mengshen Yun, Jiandong Tian, Yandong Tang, Guolin Wang, and Chengdong Wu. Stacked dense networks for single-image snow removal. *Neurocomputing*, 367:152–163, 2019. 3

[21] Ruoteng Li, Loong-Fah Cheong, and Robby T Tan. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1633–1642, 2019. 2, 3, 5, 6

[22] Runde Li, Jinshan Pan, Zechao Li, and Jinhui Tang. Single image dehazing via conditional generative adversarial network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8202–8211, 2018. 2

[23] Ruoteng Li, Robby T Tan, and Loong-Fah Cheong. All in one bad weather removal using architectural search. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3175–3185, 2020. 1, 2, 3, 5, 6, 7

[24] Wu Li, Valentin Piëch, and Charles D Gilbert. Perceptual learning and top-down influences in primary visual cortex. *Nature neuroscience*, 7(6):651–657, 2004. 3

[25] Xia Li, Jianlong Wu, Zhouchen Lin, Hong Liu, and Hongbin Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018. 2, 6

[26] Yawei Li, Kai Zhang, Jiezhang Cao, Radu Timofte, and Luc Van Gool. Localvit: Bringing locality to vision transformers. *arXiv preprint arXiv:2104.05707*, 2021. 4

[27] Jingyun Liang, Jiezhang Cao, Guolei Sun, Kai Zhang, Luc Van Gool, and Radu Timofte. Swinir: Image restoration using swin transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1833–1844, 2021. 2, 3, 6

[28] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 1

[29] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE Transactions on Image Processing*, 27(6):3064–3073, 2018. 1, 2, 3, 5, 6

[30] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 3

[31] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021. 3

[32] Justin NJ McManus, Wu Li, and Charles D Gilbert. Adaptive shape processing in primary visual cortex. *Proceedings of the National Academy of Sciences*, 108(24):9739–9746, 2011. 3

[33] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. 5

[34] Asanka G Perera, Yee Wei Law, and Javaan Chahl. Uav-gesture: A dataset for uav control and gesture recognition. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018. 1

[35] Aditya Prakash, Kashyap Chitta, and Andreas Geiger. Multi-modal fusion transformer for end-to-end autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7077–7087, 2021. 1

[36] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 1

[37] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018. 1, 2, 3, 5, 6

[38] Qin Qin, Jingke Yan, Qin Wang, Xin Wang, Minyao Li, and Yuqing Wang. Etdnet: An efficient transformer deraining model. *IEEE Access*, 9:119881–119893, 2021. 2, 3

[39] Ruijie Quan, Xin Yu, Yuanzhi Liang, and Yi Yang. Removing raindrops and rain streaks in one go. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9147–9156, 2021. 2, 6

[40] Yuhui Quan, Shijie Deng, Yixin Chen, and Hui Ji. Deep learning for seeing through window with raindrops. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2463–2471, 2019. 1, 2, 3, 6

[41] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015. 1

[42] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*, pages 154–169. Springer, 2016. 2

[43] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2018. 3

[44] Weihong Ren, Jiandong Tian, Zhi Han, Antoni Chan, and Yandong Tang. Video desnowing and deraining based on matrix decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4210–4219, 2017. 1, 2

[45] Stefan Roth and Michael J Black. Fields of experts: A framework for learning image priors. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 860–867. IEEE, 2005. 1

[46] Leonid I Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992. 1

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 5

[48] Fuxiang Tan, YuTing Kong, Yingying Fan, Feng Liu, Daxin Zhou, Long Chen, Liang Gao, Yurong Qian, et al. Sdnet: mutil-branch for single image deraining using swin. *arXiv preprint arXiv:2105.15077*, 2021. 2, 3

[49] Jeya Maria Jose Valanarasu, Poojan Oza, Ilker Hacihaliloglu, and Vishal M. Patel. Medical transformer: Gated axial-attention for medical image segmentation. In *Medical Image Computing and Computer Assisted Intervention – MICCAI 2021*, pages 36–46, Cham, 2021. Springer International Publishing. 3

[50] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 5

[51] Hong Wang, Qi Xie, Qian Zhao, and Deyu Meng. A model-driven deep neural network for single image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3103–3112, 2020. 2

[52] Lijun Wang, Jianming Zhang, Oliver Wang, Zhe Lin, and Huchuan Lu. Sdc-depth: Semantic divide-and-conquer network for monocular depth estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 541–550, 2020. 1

[53] 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 *Proceed-*

ings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 12270–12279, 2019. 1

[54] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. *arXiv preprint arXiv:2102.12122*, 2021. 4

[55] Zhendong Wang, Xiaodong Cun, Jianmin Bao, and Jianzhuang Liu. Uformer: A general u-shaped transformer for image restoration. *arXiv preprint arXiv:2106.03106*, 2021. 2, 3

[56] Wei Wei, Deyu Meng, Qian Zhao, Zongben Xu, and Ying Wu. Semi-supervised transfer learning for image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3877–3886, 2019. 1

[57] Haiyan Wu, Yanyun Qu, Shaohui Lin, Jian Zhou, Ruizhi Qiao, Zhizhong Zhang, Yuan Xie, and Lizhuang Ma. Contrastive learning for compact single image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10551–10560, 2021. 1

[58] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 4

[59] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint arXiv:2105.15203*, 2021. 1, 4

[60] Sen Yang, Zhibin Quan, Mu Nie, and Wankou Yang. Transpose: Keypoint localization via transformer. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 11802–11812, 2021. 3

[61] Wenhan Yang, Robby T Tan, Jiashi Feng, Zongming Guo, Shuicheng Yan, and Jiaying Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE transactions on pattern analysis and machine intelligence*, 42(6):1377–1393, 2019. 1, 2

[62] Wenhan Yang, Robby T. Tan, Shiqi Wang, Yuming Fang, and Jiaying Liu. Single image deraining: From model-based to data-driven and beyond, 2019. 2

[63] Rajeev Yasarla and Vishal M Patel. Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8405–8414, 2019. 1

[64] Rajeev Yasarla, Vishwanath A Sindagi, and Vishal M Patel. Syn2real transfer learning for image deraining using gaussian processes. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2726–2736, 2020. 2

[65] Rajeev Yasarla, Jeya Maria Jose Valanarasu, and Vishal M Patel. Exploring overcomplete representations for single image deraining using cnns. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):229–239, 2020. 2

[66] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Adherent raindrop modeling, detectionand removal in video. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1721–1733, 2015. 1, 2, 3

[67] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Multi-stage progressive image restoration. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14821–14831, 2021. 2, 6

[68] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2018. 3

[69] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 695–704, 2018. 1, 2

[70] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology*, 30(11):3943–3956, 2019. 1, 2

[71] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Joint transmission map estimation and dehazing using deep networks. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(7):1975–1986, 2019. 1

[72] Jingang Zhang, Wenqi Ren, Shengdong Zhang, He Zhang, Yunfeng Nie, Zhe Xue, and Xiaochun Cao. Hierarchical density-aware dehazing network. *IEEE Transactions on Cybernetics*, 2021. 1, 3

[73] Kaihao Zhang, Rongqing Li, Yanjiang Yu, Wenhan Luo, and Changsheng Li. Deep dense multi-scale network for snow removal using semantic and depth priors. *IEEE Transactions on Image Processing*, 30:7419–7431, 2021. 1, 2, 6

[74] Dong Zhao, Jia Li, Hongyu Li, and Long Xu. Hybrid local-global transformer for image dehazing. *arXiv preprint arXiv:2109.07100*, 2021. 2, 3

[75] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6881–6890, 2021. 3

[76] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *Proceedings of the IEEE international conference on computer vision*, pages 2526–2534, 2017. 1, 2