Structure Representation Network and Uncertainty Feedback Learning for Dense Non-Uniform Fog Removal

Yeying Jin[1][0000−0001−7818−9534], Wending Yan[1,3][0000−0001−5993−8405], Wenhan Yang[2][0000−0002−1692−0069], and Robby T. Tan[1,3][0000−0001−7532−6919]

[1] National University of Singapore,
[2] Nanyang Technological University,
[3] Yale-NUS College

`jinyeying@u.nus.edu, e0267911@u.nus.edu, wenhan.yang@ntu.edu.sg,`
`robby.tan@{nus,yale-nus}.edu.sg`

Abstract. Few existing image defogging or dehazing methods consider dense and non-uniform particle distributions, which usually happen in smoke, dust and fog. Dealing with these dense and/or non-uniform distributions can be intractable, since fog's attenuation and airlight (or veiling effect) significantly weaken the background scene information in the input image. To address this problem, we introduce a structure-representation network with uncertainty feedback learning. Specifically, we extract the feature representations from a pre-trained Vision Transformer (DINO-ViT) module to recover the background information. To guide our network to focus on non-uniform fog areas, and then remove the fog accordingly, we introduce the uncertainty feedback learning, which produces uncertainty maps, that have higher uncertainty in denser fog regions, and can be regarded as an attention map that represents fog's density and uneven distribution. Based on the uncertainty map, our feedback network refines our defogged output iteratively. Moreover, to handle the intractability of estimating the atmospheric light colors, we exploit the grayscale version of our input image, since it is less affected by varying light colors that are possibly present in the input image. The experimental results demonstrate the effectiveness of our method both quantitatively and qualitatively compared to the state-of-the-art methods in handling dense and non-uniform fog or smoke.
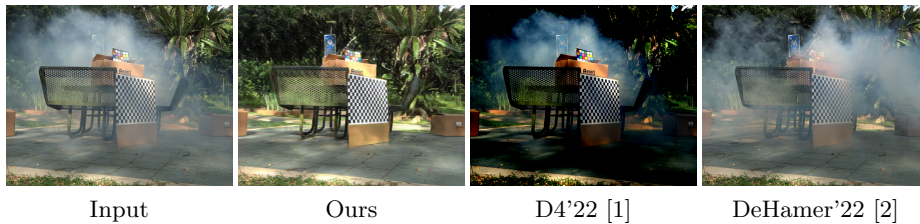
1 Introduction

Atmospheric particles, such as fog, haze, dust and smoke particles, can degrade the visibility of a scene significantly as shown in Fig. 1. These particles can be modeled as [3]:

$$\mathbf{I}(\mathbf{x}) = \mathbf{J}(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))\,\mathbf{A}, \qquad (1)$$

[†] Our data and code is available at: `https://github.com/jinyeying/FogRemoval`

Input   Ours   D4'22 [1]   DeHamer'22 [2]

Fig. 1. Visual comparisons of different methods: the state-of-the-art CNN-based method [1] and transformer-based method [2] in dense and/or non-uniform fog.

where $\mathbf{I}$ is an observed RGB color vector, $\mathbf{x}$ is the pixel location. $\mathbf{J}$ is the scene radiance. $\mathbf{A}$ is the atmospheric light, and t is the transmission. The first term is called direct attenuation, and the second term is called airlight. Transmission t can be modeled as $t(\mathbf{x}) = \exp(\beta(\mathbf{x})d(\mathbf{x}))$, where β is the particle attenuation factor that depends on the density of the particle distribution and the size of particles; while d is the depth of the scene with respect to the camera. Most existing methods assume the uniformity of the particle distributions, which means they assume β to be independent from $\mathbf{x}$. Note that, in this paper, we deal with fog, haze, atmospheric dust and smoke that can be dense and/or non-uniform. However, for clarity, we write fog to represent them.

Many methods have been proposed to deal with fog degradation. Existing fully supervised CNN-based methods [4,5,6,7,8,9,10] require clean ground truths, which are intractable to obtain particularly for non-uniform fog. Synthetic images, unfortunately, cannot help that much for dense and/or non-uniform fog. Synthesizing non-uniform fog is difficult and computationally expensive, and dense synthetic fog has significant gaps with real dense fog. Semi-supervised methods [11,12,13,14] adopt the domain adaptation. However, the huge domain gap between synthetic and real dense and/or non-uniform fog images is not easy to align. Unsupervised methods [15,16,17,18,1] make use of statistical similarity between unpaired training data, and are still less effective compared with semi-supervised or supervised methods. Importantly, unsupervised methods can generate hallucinations, particularly in dense fog areas. Recently, ViT-based dehazing methods [2,19] have been proposed; however, memory and computation complexity slow down the convergence [20], causing unreliable performance on real-world high-resolution non-uniform fog images.

In this paper, our goal is to remove fog, particularly dense or non-uniform fog, or a combination of the two (dense and non-uniform). Unlike non-dense uniform fog, where the human visual perception can still discern the background scenes, dense and/or non-uniform fog significantly weakens the information of the background scenes (see Fig. 1). To achieve our goal, first, we exploit the representation extracted from DINO-ViT [21], a self-supervised pre-trained model in order to recover background structures. DINO-ViT captures visual representations from data, e.g., scene structure representations, based on self-similarity prior [22]. Second, since the recovery of the $\mathbf{A}$ is challenging [23], to avoid the direct recovery of $\mathbf{A}$, we introduce a grayscale feature multiplier to learn fog

degradation in an end-to-end manner. Grayscale images are less affected by multi-colored light sources (skylight, sunlight, or cars' headlights, etc.) as well as the colors of the particle scattered lights (whitish for fog, yellowish or reddish for haze or atmospheric dust or smoke). We can multiply the grayscale features and our feature multiplier (derived from the model in Eq. (1)), to ensure our features are unaffected by the airlight and thus are more reliable when applied to our multiplier consistency loss.

Third, we propose an uncertainty-based feedback learning that allows our network to pay more attention to regions that are still affected by fog based on our uncertainty predictions, iteratively. Since the network usually has high uncertainty on the dense fog regions (because background information is washed out by the fog and the input image contains less background information in those regions), we can use an uncertainty map as an attention cue to guide the network to differentiate dense fog region from the rest of input image. In one iteration, if our output still contains fog in some regions, the uncertainty map will indicate those regions, and in the next iteration, our method will focus on these regions to further defog them.

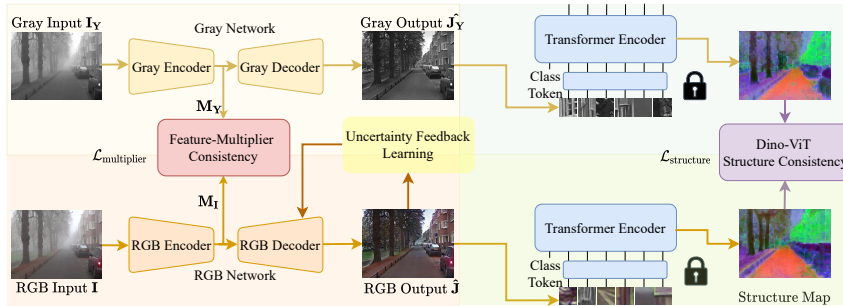To sum up, our main contributions and novelties are as follows:

 − To the best of our knowledge, our method is the first single-image defogging network that performs robustly in dense non-uniform fog, by combining structure representations from ViT and features from CNN as feature regularization. Thus, the background information under fog can be preserved and extracted.
 − We propose the grayscale feature multiplier that acts as feature enhancement and guides our network to learn to extract clear background information.
 − We introduce the uncertainty feedback learning in our defogging network, which can refine the defogging results iteratively by focusing on areas that still suffer from fog.

Experimental results show that our method is effective in removing dense and/or non-uniform fog images, outperforming the state-of-the-art methods both quantitatively and qualitatively.

2   Related Works

Non-learning methods introduced priors from the atmosphere scattering model. Tan [24] estimates the airlight to increase contrast, Fattal [25] estimates transmission, which is statistical uncorrelated to surface shading, He et al. [26] introduce the dark channel prior, Berman et al. [27] propose a haze-line constraint, and Meng et al. [28] estimate transmission using its minimum boundary.

CNN-based methods allow faster results [29,30,31,32]. DehazeNet [4] and MSCNN [33] use CNN, DCPDN [6] trains densely connected pyramid network to estimate transmission map. AODNet [5], GFN [34], [35] applies CGAN, EPDN [7] applies pix2pix, they end-to-end output clear images. Griddehazenet [8] designs attention-based [36] grid network, MSBDN [9] designs boosted multi-scale decoder, FFA-Net [37] proposes feature fusion attention network, AECR-Net [10]
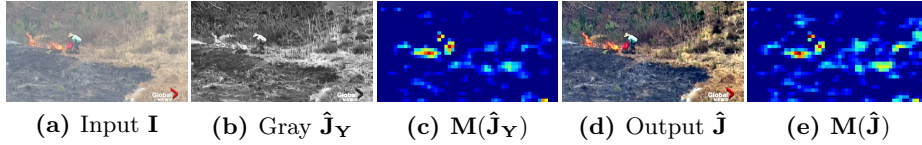
Fig. 2. The pipeline of our network, which consists of (i) grayscale feature multiplier (top left), (ii) structure representation network (right), and (iii) uncertainty feedback learning (middle). The grayscale feature multiplier ($\mathbf{M_Y}$) provides features (red) from CNN, and guides the RGB network to enhance features. The structure representation network provides structure representations (purple) from fixed and pre-trained DINO-ViT, to recover background information.

applies contrastive learning. Few fully-supervised methods [38,39,40] are proposed to deal with Dense-Haze [41]. All these methods employ fully supervised learning and hence require ground truths to train their networks. However, obtaining a large number of real dense or non-uniform fog images and their corresponding ground truths is intractable. Semi-supervised methods [11,12,13,14] have been introduced, unfortunately, they still suffer from gaps between synthetic and real fog images. Unsupervised methods [15,16,17,18,1] are mainly CycleGAN-based. However, the generated images can easily render artefacts (structures that are not originally in the input image) when unpaired training data is used. Though all these methods perform well on normal fog dataset, they are CNN-based, and tend to perform poorly on dense and non-uniform fog [2] since CNN fails to model long-range pixel dependencies [42].

Recently, ViT-based dehazing [2,19] has made progress. DehazeFormer [19] is trained on synthetic fog images (RESIDE outdoor dataset [43]), which are not realistic and cause unreliable performance on real-world fog images. DeHamer [2] combines CNN and Transformer for image dehazing; however, memory and computation complexity slow down the convergence [20], causing inefficient performance on real-world high resolution fog images. In contrast, our method exploits features from both ViT and CNN.

3   Proposed Method

Fig. 2 shows the pipeline of our architecture, which consists of three parts: (i) grayscale feature multiplier, (ii) structure representation network, and (iii) uncertainty feedback learning. Since grayscale images are less affected by multi-colored light sources and colorful particle scattered lights, we develop a grayscale network to guide our RGB network. Hence, in our pipeline, we have two parallel

(a) Input $\mathbf{I}$ (b) Gray $\hat{\mathbf{J}}_{\mathbf{Y}}$ (c) $\mathbf{M}(\hat{\mathbf{J}}_{\mathbf{Y}})$ (d) Output $\hat{\mathbf{J}}$ (e) $\mathbf{M}(\hat{\mathbf{J}})$

Fig. 3. Visualization of the features extracted from the grayscale feature multiplier. (a) Input fog image $\mathbf{I}$, (b) Grayscale output image $\hat{\mathbf{J}}_{\mathbf{Y}}$, (c) Sample feature map for $\hat{\mathbf{J}}_{\mathbf{Y}}$, (d) Output fog-free image $\hat{\mathbf{J}}$, and (e) Sample feature map for $\hat{\mathbf{J}}$. We can observe that features in (c) for the grayscale fog images are less affected by fog, and can effectively guide the features in (e) owing to our multiplier consistency loss.

subnetworks: one for processing the grayscale input image, and the other one for processing the RGB input image.
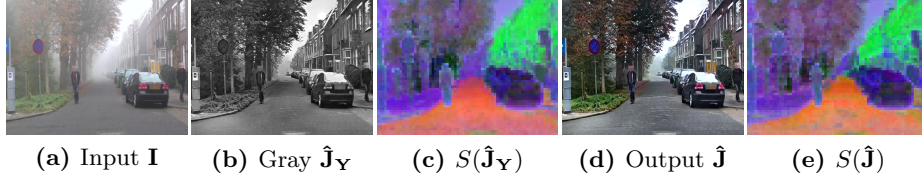
3.1   Grayscale Feature Multiplier

Feature Multiplier Dense and/or non-uniform fog suffers from low contrast and degraded features. To extract clear background features, we design a subnetwork to predict the amount by which these features should be enhanced. Considering the fog model in Eq. (1) and to avoid the challenges of predicting atmosphere light $\mathbf{A}$ [23], we turn the relationship between fog $\mathbf{I}$ and clear images $\mathbf{J}$ into a multiplier relationship: $\mathbf{J}(\mathbf{x}) = \mathbf{I}(\mathbf{x})\mathbf{M}(\mathbf{x})$, which is called $\mathbf{M}$ feature multiplier [44], where $\mathbf{M}(\mathbf{x}) = \frac{\mathbf{I}(\mathbf{x}) + t(\mathbf{x})\mathbf{A} - \mathbf{A}}{\mathbf{I}(\mathbf{x})t(\mathbf{x})}$.

The feature multiplier $\mathbf{M}$ depends on atmospheric light $\mathbf{A}$ and transmission $t(\mathbf{x})$, which are both unknown. Moreover, $\mathbf{A}$ is an RGB color vector; implying that in order to estimate $\mathbf{M}$, there are four unknowns in total for each pixel: 3 for the RGB values of $\mathbf{A}$ and 1 for t. These unknowns influence the accuracy of the network in learning the correct value of $\mathbf{M}$. To overcome the difficulty, we propose to employ a grayscale feature multiplier, where all variables in the grayscale feature multiplier become scalar variables. Consequently, the number of unknowns the network needs to learn is reduced to only two variables for each pixel: $t(\mathbf{x})$ and $\mathbf{A}$. Note that, to avoid the direct recovery of $\mathbf{A}$, our network implicitly includes $\mathbf{A}$ in the feature multiplier.

Grayscale-Feature Multiplier We feed the grayscale image, $\mathbf{I}_{\mathbf{Y}}$, to our grayscale encoder, which estimates the grayscale feature multiplier $\mathbf{M}_{\mathbf{Y}}$. We multiply grayscale features and $\mathbf{M}_{\mathbf{Y}}$ before feeding them to our grayscale decoder. We train the grayscale network independently from the RGB network, using both synthetic and unpaired real images. Once the grayscale network is trained, we freeze it, and employ it as the guidance for training the RGB network.

As for the RGB network, the RGB encoder takes the RGB image as input, $\mathbf{I}$, and estimates the color feature multiplier $\mathbf{M}_{\mathbf{I}}$. Having estimated $\mathbf{M}_{\mathbf{I}}$, we multiply it with the RGB features and feed the multiplied features to our RGB decoder. As shown in Fig. 2, we constrain the learning process of our RGB network by

(a) Input $\mathbf{I}$ (b) Gray $\hat{\mathbf{J}}_\mathbf{Y}$ (c) $S(\hat{\mathbf{J}}_\mathbf{Y})$ (d) Output $\hat{\mathbf{J}}$ (e) $S(\hat{\mathbf{J}})$

Fig. 4. Visualization of structure representations. (a) Input fog image $\mathbf{I}$, (b) Grayscale output image $\hat{\mathbf{J}}_\mathbf{Y}$, (c) DINO-ViT keys for $\hat{\mathbf{J}}_\mathbf{Y}$, (d) Output fog-free image $\hat{\mathbf{J}}$, and (e) DINO-ViT keys for $\hat{\mathbf{J}}$. We can observe that DINO-ViT representations in (c) capture structure scene/object parts (*e.g.* cars, trees, buildings), and are less affected by fog.

imposing a consistency loss between the grayscale feature multiplier, $\mathbf{M}_\mathbf{Y}$, and the RGB feature multiplier $\mathbf{M}_\mathbf{I}$. We call this loss a multiplier consistency loss.

Multiplier Consistency Loss To constrain the RGB feature multiplier $\mathbf{M}_\mathbf{I}$, we utilize the grayscale feature multiplier $\mathbf{M}_\mathbf{Y}$ as guidance. Based on the Gray World assumption [45], we define the multiplier consistency loss as:
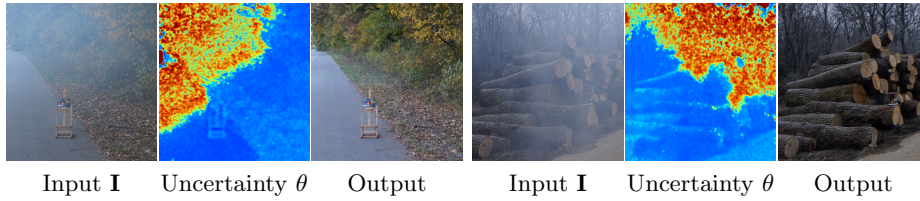
$$\mathcal{L}_{\text{multiplier}} = \|\mathbf{M}_\mathbf{I} - \mathbf{M}_\mathbf{Y}\|_2 \,, \tag{2}$$

where $\mathbf{M}_\mathbf{I}$ and $\mathbf{M}_\mathbf{Y}$ are the feature multipliers of the RGB and grayscale images. To construct this loss, first, we train our grayscale network independently from our RGB network. By training the grayscale network on both synthetic and real images, $\mathbf{M}_\mathbf{Y}$ is optimized. Once the training is completed, we freeze the grayscale network. Subsequently, we train our RGB network.

In this training stage, the network losses are the same as those in the grayscale network, except all the images used to calculate the losses are now RGB images. Unlike the training process of the grayscale network, however, we need to apply the multiplier consistency loss $\mathcal{L}_{\text{multiplier}}$ to train the RGB network. Note that, the reason we use the loss to enforce $\mathbf{M}_\mathbf{I}$ and $\mathbf{M}_\mathbf{Y}$ to be close, and do not use $\mathbf{M}_\mathbf{Y}$ as the feature multiplier for the RGB network (i.e., $\mathbf{M}_\mathbf{I} = \mathbf{M}_\mathbf{Y}$) is because we intend to train the RGB convolution layers; so that, in the testing stage, we do not need the grayscale network.

3.2   Structure Representation Network

A few methods [22,46,47,48] have exploited self-similarity-based feature descriptors to obtain structure representations. Unlike these methods, to reveal the clear background structures, we use deep spatial features obtained from DINO-ViT [49], which has been proven to learn meaningful visual representations [50]. Moreover, these powerful representations are shared across different object classes. Specifically, we use keys' self-similarity in the attention model, at the deepest transformer layer. In Fig. 4, we show the Principal Component Analysis (PCA) visualization of the keys' self-similarity and demonstrate the three top components as RGB at layer 11 of DINO-ViT. As one can observe, the

Input **I**   Uncertainty θ Output   Input **I**   Uncertainty θ Output

Fig. 5. Uncertainty maps of O-HAZE [51] dataset. The (b) uncertainty map indicates the fog intensity.

structure representations capture the clear background parts, which helps the network significantly preserve the background structures.

Dino-ViT Structure Consistency Loss Our Dino-ViT structure consistency loss encourages the deep-structure representations of the RGB output to be similar to the grayscale features, since the grayscale features are robust to fog:

$$\mathcal{L}_{\text{structure}} = \left\| S(\hat{\mathbf{J}}) - S(\hat{\mathbf{J}}_{\mathbf{Y}}) \right\|_F, \tag{3}$$

where S is the self-similarity descriptor, defined by the difference in the self-similarity of the keys extracted from the attention module, with $n \times n$ dimension, where n is the number of patches. $\|\cdot\|_F$ is the Frobenius norm. The self-similarity descriptor is defined as:
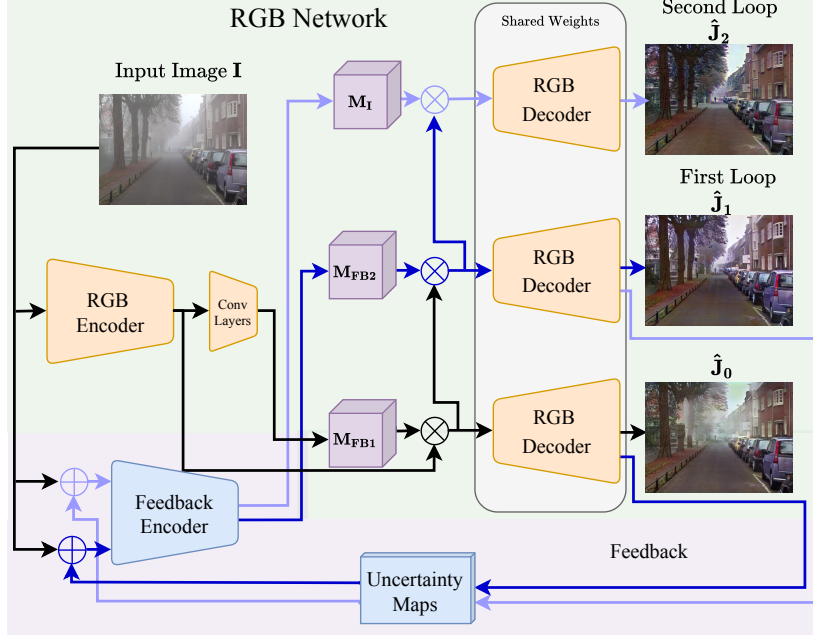
$$S(\hat{\mathbf{J}})_{ij} = \text{cos-sim}(k_i(\hat{\mathbf{J}}), k_j(\hat{\mathbf{J}})) = 1 - \frac{k_i(\hat{\mathbf{J}}) \cdot k_j(\hat{\mathbf{J}})}{\left\| k_i(\hat{\mathbf{J}}) \right\| \cdot \left\| k_j(\hat{\mathbf{J}}) \right\|}, \tag{4}$$

where $\text{cos-sim}(\cdot)$ is the cosine similarity between keys, k_i are the spatial keys.

3.3   Uncertainty Feedback Learning

Uncertainty Map The main challenge of dealing with dense non-uniform fog distributions is how to differentiate the dense fog regions from the light fog regions. To address this problem, we exploit an uncertainty map as an attention map to guide the network to differentiate dense fog regions from the rest of the input image. Since the network produces higher uncertainty for the denser fog regions. Each value in the uncertainty map represents the confidence of the defogging operation at the corresponding pixel (i.e. the variance). The higher the value, the more uncertain the network's prediction for that pixel.

To generate an uncertainty map together with the defogged result, we add a multi-task decoder to our network. Note that the defogged result and the uncertainty map are decoded from the same features, since there is only one encoder. We assume that the defogged output $\hat{\mathbf{J}}$ follows a Laplace distribution,

Fig. 6. Architecture of the uncertainty feedback learning. This network refines the performance of the RGB network.

where the mean of this distribution is the clear ground truth $\mathbf{J}^{gt}$ [52,53]. Under this assumption, we can define a likelihood function as follows:

$$p(\mathbf{J}^{gt}|\mathbf{I}) = \frac{1}{2\theta}\exp(-\frac{\left\|\hat{\mathbf{J}} - \mathbf{J}^{gt}\right\|_1}{\theta}), \tag{5}$$

where θ is the variance of the Laplace distribution. In our implementation, we define this variance as the uncertainty of the defogged output $\hat{\mathbf{J}}$. Therefore, Eq. (5) includes both outputs generated by our multi-task network. Taking the logarithm of both sides of Eq. (5) and maximizing it, we can obtain: $\arg\max_\theta \ln p(\mathbf{J}^{gt}|\mathbf{I}) = -\frac{\left\|\hat{\mathbf{J}} - \mathbf{J}^{gt}\right\|_1}{\theta} - \ln\theta$.

For the first term in this likelihood $-\frac{\left\|\hat{\mathbf{J}} - \mathbf{J}^{gt}\right\|_1}{\theta}$, we simply convert the negative sign to positive and put it into the loss function. The second term $-\ln\theta$, we convert it to $\ln(\theta + 1)$ to avoid negative infinity when θ is zero. Hence, the uncertainty loss we will minimize is expressed as follows:

$$\mathcal{L}_{\text{unc}} = \frac{\left\|\hat{\mathbf{J}} - \mathbf{J}^{\mathbf{gt}}\right\|_1}{\theta} + \ln(\theta + 1). \tag{6}$$

Uncertainty Feedback Learning Unfortunately, the results of our baseline network might still suffer from the remaining fog. There are two possible rea-

sons. First, the effectiveness of our multiplier consistency loss depends on the grayscale network's performance. While we can see in our ablation studies that this grayscale guidance improves the defogging performance, the grayscale network cannot completely remove fog all the time. Second, our discriminative loss cannot fully suppress fog for any input image, since we do not have paired training ground truths for real images.

To address this problem, we introduce uncertainty feedback learning, which the architecture is shown in Fig. 6. We provide our network extra attention to the different densities of fog based on our network-generated uncertainty maps. Specifically, we introduce uncertainty feedback learning to make our network focus on areas where fog is still visible and to defog these areas iteratively. In one iteration, if our output still contains fog in some regions, then the uncertainty map will indicate those regions, and in the next iteration, our method will focus on these regions to further defog them.

As shown in Fig. 6, we feedforward the uncertainty map together with the input image into the feedback encoder, producing a new feedback feature multiplier $\mathbf{M_{FB}}$. We multiply this multiplier with the RGB features, and feed the multiplication result to the RGB decoder, generating the enhanced output, $\hat{\mathbf{J}}_1$. To train our RGB network and the feedback network, we use real images (that do not have ground truths) and apply only the discriminative loss. We compute the loss of this output $\hat{\mathbf{J}}_i$ (where i is the index of the iterations) with the same loss functions as the initial output image $\hat{\mathbf{J}}$, and backpropagate the errors. We iterate this process a few times to obtain the final output. The number of iterations is constrained by the GPU memory. From our experiments, we found that the uncertainty map tends to be unchanged after two or three iterations.

3.4   Overall Losses

In training our network, we use both synthetic images with ground truths and real images without ground truths. For both the grayscale and RGB networks, we feedforward a set of synthetic images into the network, which outputs the predicted clear synthetic images. For the same batch, we feedforward a set of real images into the network, producing the predicted real images. Having obtained the predicted clear images of both the synthetic and real images, we then train the discriminators in the grayscale channel of the grayscale and RGB networks. Training discriminators requires a set of reference images, which must be real and clear (without fog). Our reference images include the ground truth images of the synthetic fog images (paired), and other real images that with no correlation to our input images (unpaired).

We multiply each loss function with its respective weight, and sum them together to obtain our overall loss function:

$$\mathcal{L} = \lambda_m \mathcal{L}_{\text{multiplier}} + \lambda_s \mathcal{L}_{\text{structure}} + \lambda_u \mathcal{L}_{\text{unc}} + \mathcal{L}_{\text{MSE}} + \lambda_d \mathcal{L}_{\text{dis}}, \qquad (7)$$

where λ are the weights for the respective losses. $\lambda_m = 1$, $\lambda_u = 1$, their values are obtained empirically. $\lambda_s = 0.1$, $\lambda_d = 0.005$, their values are followed default

Table 1. Quantitative results on Dense-HAZE, NH-HAZE, O-HAZE and self-collected smoke datasets.

Method	Dense-HAZE [41]		NH-HAZE [56]		O-HAZE [51]		SMOKE	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
DCP [26]	10.06	0.39	10.57	0.52	16.78	0.65	11.26	0.26
DehazeNet [4]	13.84	0.43	16.62	0.52	17.57	0.77	-	-
AODNet [5]	13.14	0.41	15.40	0.57	15.03	0.54	-	-
GDN [8]	-	-	-	-	23.51	**0.83**	15.19	0.53
MSBDN [9]	15.37	0.49	19.23	0.71	24.36	0.75	13.19	0.34
FFA-Net [37]	14.39	0.45	19.87	0.69	22.12	0.77	-	-
AECR-Net [10]	15.80	0.47	19.88	**0.72**	-	-	-	-
DeHamer'22 [2]	16.62	**0.56**	20.66	0.68	17.02	0.43	13.31	0.28
Ours	**16.67**	0.50	**20.99**	0.61	**24.61**	0.75	**18.83**	**0.62**

setting. $\mathcal{L}_{\mathrm{MSE}}$ is the Mean Squared Error (MSE) loss (applied only to synthetic images), $\mathcal{L}_{\mathrm{dis}}$ is the discriminative loss.

4   Experimental Results

Implementation We use two sets of data to train our networks: real fog images and reference clear images, synthetic fog images and their ground truth. For the real fog images, we train on self-collected and Internet fog images. For the clear reference images, we collect clear images from Google Street View and Internet. For synthetic training images, we render fog images (Eq. 1) from clear images, taken from Cityscapes [55], which provides 2,975 pairs of RGB images and their disparity maps. Subsequently, we fine-tune the model on different datasets. For self-collected smoke images, we fine-tune the model on the 110 self-collected smoke images and clean pairs, and 100 unpaired Internet clean references. We also collect 12 other pairs of fog data for evaluation. Our data is publicly available.
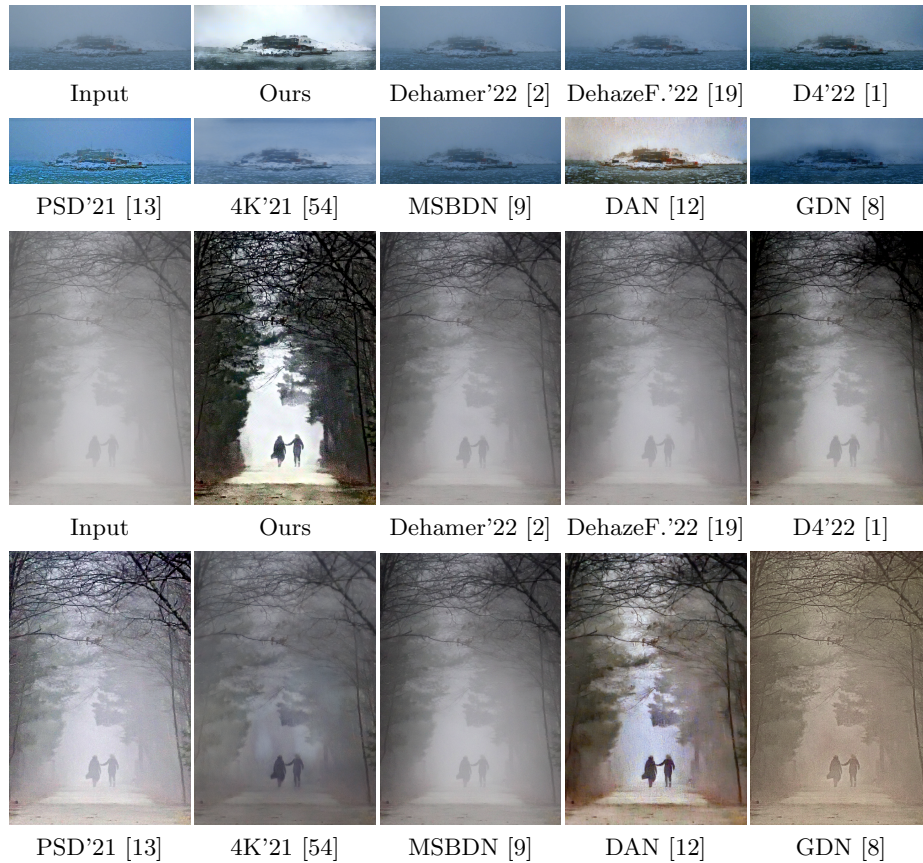
Datasets We collected real smoke data by ourselves. We used a fog machine to generate fog, where we fixed the camera pose to record fog images and their paired ground truth. Ancuti et al. [51] propose the O-HAZE dataset consisting of 45 pairs of hazy/clear scenes using a smoke generator to simulate the atmospheric scattering effect in the real world. Using the same smoke generator equipment, Ancuti et al. [41] also propose the Dense-HAZE and NH-HAZE [56,57], which both consist of 55 pairs of hazy/clear scenes, 45 training, 5 validation and 5 test images. The scenes in the Dense-HAZE and NH-HAZE datasets are similar to the O-Haze dataset, but the smoke density is much higher and more non-homogeneous.

Baselines We evaluate our method against the non-learning method Non-Local Image Dehazing (NLD) [27], state-of-the-art transformer-based dehazing meth-

Fig. 7. Qualitative evaluation results on real fog machine and O-Haze [51] images.

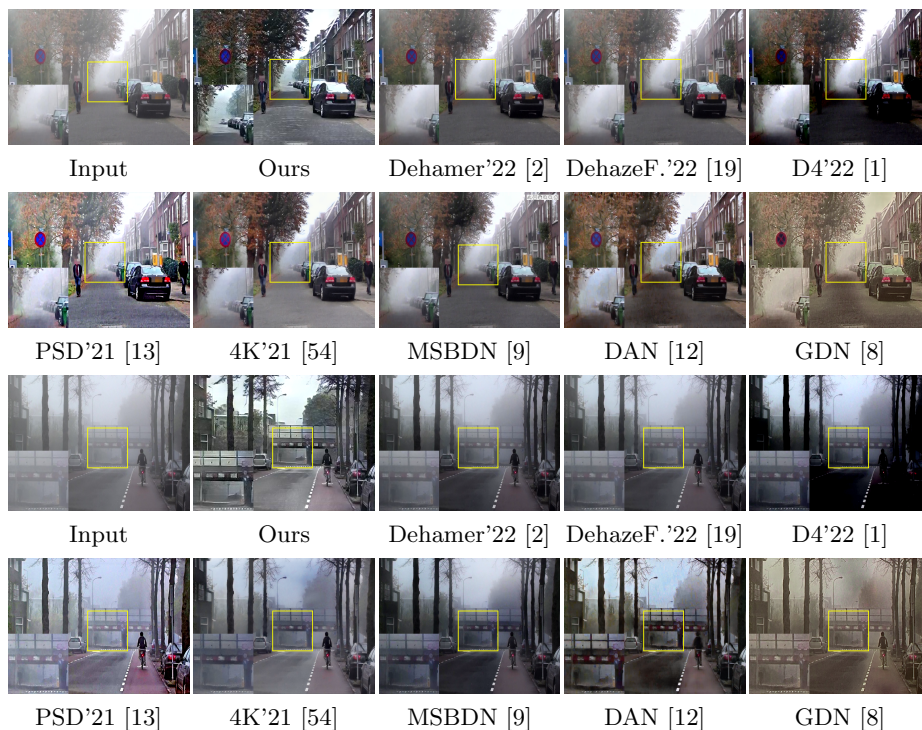ods [2,19], CNN-based methods: GridDehazeNet (GDN) [8], Domain Adapta-

Fig. 8. Comparison results on commonly used test foggy images. (a) Input images. (b) Our results. (c)∼(g) Results of the state-of-the-art methods.

tion Network (DAN) [12], Multi-Scale Boosted Dehazing Network (MSBDN) [9], 4KDehazing (CVPR21) [54], PSD (CVPR21) [13], D4 (CVPR22) [1], etc.

Qualitative Comparisons Comparisons on the self-collected fog and O-HAZE dataset are shown in Fig. 7. The baseline methods do not perform well on the images. Some results are too dark, and some still have fog left. Also, since the generated fog is not uniform in the Dense-Haze and NH-Haze datasets, some fog still remains. The deep learning baselines are not able to defog such dense fog adequately.

Figs. 8 to 9 show the input dense non-uniform fog images, our defogging results, and the results of the state-of-the-art methods. Due to the uniform and/or severe fog density, the input images are degraded by multiple factors like blur, contrast, sharpness, and color distortion. As shown in the figures, our method outperforms the state-of-the-art methods on real dense fog images.
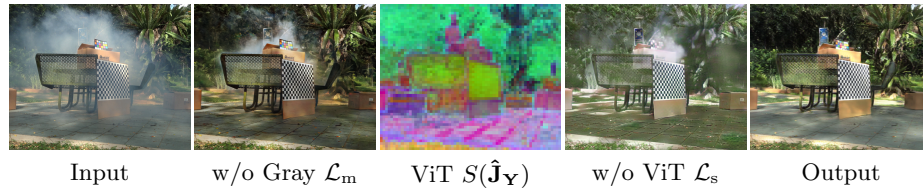
| Input | Ours | Dehamer'22 [2] | DehazeF.'22 [19] | D4'22 [1] |

| PSD'21 [13] | 4K'21 [54] | MSBDN [9] | DAN [12] | GDN [8] |

| Input | Ours | Dehamer'22 [2] | DehazeF.'22 [19] | D4'22 [1] |

| PSD'21 [13] | 4K'21 [54] | MSBDN [9] | DAN [12] | GDN [8] |

Fig. 9. Comparison results on real dense fog images. (a) Input images. (b) Our results. (c)∼(g) Results of the state-of-the-art methods. Our results show better visibility.
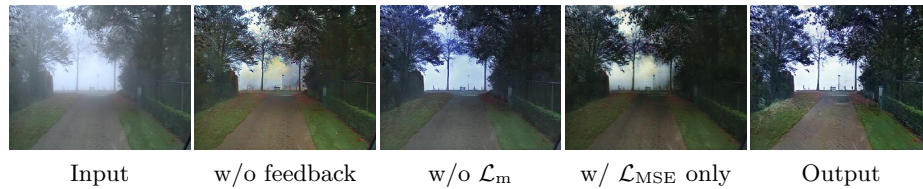
Quantitative Comparisons We also conduct quantitative evaluations on O-HAZE, NH-Haze and Dense-Haze, which are shown in Table 1. We measure the restoration performance using the Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity (SSIM); higher is better. Our method achieves the best PSNR, SSIM performance.

4.1   Ablation Studies

We conduct ablation studies to analyze the characteristics of the proposed algorithm. We first evaluate the grayscale feature multiplier, if the grayscale network is removed, the RGB network will have no guidance from grayscale and the results are shown in Fig. 10b. To show the effectiveness of using ViT, we remove the structure consistency loss, the results are shown in Fig. 10d. We then remove the uncertainty feedback network from our model. After training with the same semi-supervised training strategy and the same loss functions, the results are shown in Fig. 11b. We can observe that the results are not as effective as those using the feedback network. In Fig. 11c, we replace our multiplier generator with the normal generator. Therefore, we can observe more fake content, such as the fake leaves on the tree. Finally, Fig. 11d shows the results of using the MSE loss

| Input | w/o Gray $\mathcal{L}_m$ | ViT $S(\hat{\mathbf{J}}_{\mathbf{Y}})$ | w/o ViT $\mathcal{L}_s$ | Output |

Fig. 10. Ablation studies on (b) without using our grayscale multiplier consistency loss $\mathcal{L}_{\text{multiplier}}$, and (d) without using our Dino-ViT structure consistency loss $\mathcal{L}_{\text{structure}}$. (e) is our final output. (c) shows DINO-ViT capture scene structure, helping the network to recover the background information.

| Input | w/o feedback | w/o $\mathcal{L}_m$ | w/ $\mathcal{L}_{\text{MSE}}$ only | Output |

Fig. 11. Ablation studies on (b) without uncertainty feedback network; (c) without multiplier consistency loss; (d) defogging results from our model with the MSE loss only. (e) is our final output.

only. The typical results of fully supervised deep learning methods trained on synthetic images are unsatisfactory. Some fog still remains, details are lost, and some regions are considerably dark.

5   Conclusion

We have proposed a learning-based defogging method that targets dense and/or non-uniform fog. Our method combines the structure representations from ViT and the features from CNN as feature regularization that can guide our network to recover background information. Our pipeline consists of a grayscale network and an RGB network. We introduced the grayscale feature multiplier, which is designed to enhance features. Aside from the new structure loss and multiplier consistency loss, we also introduced uncertainty feedback learning that refines the performance of the RGB generator network. Experimental results show that our method works for dense and/or non-uniform fog, and outperforms the state-of-the-art methods.

Acknowledgment

References

1. Yang, Y., Wang, C., Liu, R., Zhang, L., Guo, X., Tao, D.: Self-augmented unpaired image dehazing via density and depth decomposition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 2037–2046
2. Guo, C.L., Yan, Q., Anwar, S., Cong, R., Ren, W., Li, C.: Image dehazing transformer with transmission-aware 3d position embedding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 5812–5820
3. Koschmieder, H.: Theorie der horizontalen Sichtweite. Number V. 2 in Beiträge zur Physik der freien Atmosphäre. Keim & Nemnich (1924)
4. Cai, B., Xu, X., Jia, K., Qing, C., Tao, D.: Dehazenet: An end-to-end system for single image haze removal. IEEE Transactions on Image Processing **25** (2016) 5187–5198
5. Li, B., Peng, X., Wang, Z., Xu, J., Feng, D.: Aod-net: All-in-one dehazing network. In: Proceedings of the IEEE international conference on computer vision. (2017) 4770–4778
6. Zhang, H., Patel, V.M.: Densely connected pyramid dehazing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 3194–3203
7. Qu, Y., Chen, Y., Huang, J., Xie, Y.: Enhanced pix2pix dehazing network. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 8160–8168
8. Liu, X., Ma, Y., Shi, Z., Chen, J.: Griddehazenet: Attention-based multi-scale network for image dehazing. In: Proceedings of the IEEE/CVF international conference on computer vision. (2019) 7314–7323
9. Dong, H., Pan, J., Xiang, L., Hu, Z., Zhang, X., Wang, F., Yang, M.H.: Multi-scale boosted dehazing network with dense feature fusion. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 2157–2167
10. Wu, H., Qu, Y., Lin, S., Zhou, J., Qiao, R., Zhang, Z., Xie, Y., Ma, L.: Contrastive learning for compact single image dehazing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 10551–10560
11. Li, L., Dong, Y., Ren, W., Pan, J., Gao, C., Sang, N., Yang, M.H.: Semi-supervised image dehazing. IEEE Transactions on Image Processing **29** (2019) 2766–2779
12. Shao, Y., Li, L., Ren, W., Gao, C., Sang, N.: Domain adaptation for image dehazing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2020) 2808–2817
13. Chen, Z., Wang, Y., Yang, Y., Liu, D.: Psd: Principled synthetic-to-real dehazing guided by physical priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 7180–7189
14. Li, Y., Chang, Y., Gao, Y., Yu, C., Yan, L.: Physically disentangled intra-and inter-domain adaptation for varicolored haze removal. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 5841–5850
15. Huang, L.Y., Yin, J.L., Chen, B.H., Ye, S.Z.: Towards unsupervised single image dehazing with deep learning. In: 2019 IEEE International Conference on Image Processing (ICIP), IEEE (2019) 2741–2745
16. Golts, A., Freedman, D., Elad, M.: Unsupervised single image dehazing using dark channel prior loss. IEEE Transactions on Image Processing **29** (2019) 2692–2701

17. Li, B., Gou, Y., Gu, S., Liu, J.Z., Zhou, J.T., Peng, X.: You only look yourself: Unsupervised and untrained single image dehazing neural network. International Journal of Computer Vision **129** (2021) 1754–1767
18. Zhao, S., Zhang, L., Shen, Y., Zhou, Y.: Refinednet: A weakly supervised refinement framework for single image dehazing. IEEE Transactions on Image Processing **30** (2021) 3391–3404
19. Song, Y., He, Z., Qian, H., Du, X.: Vision transformers for single image dehazing. arXiv preprint arXiv:2204.03883 (2022)
20. Zhu, X., Su, W., Lu, L., Li, B., Wang, X., Dai, J.: Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159 (2020)
21. Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A.: Emerging properties in self-supervised vision transformers. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 9650–9660
22. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2007) 1–8
23. Sulami, M., Geltzer, I., Fattal, R., Werman, M.: Automatic recovery of the atmospheric light in hazy images. In: IEEE International Conference on Computational Photography (ICCP). (2014)
24. Tan, R.T.: Visibility in bad weather from a single image. In: 2008 IEEE conference on computer vision and pattern recognition, IEEE (2008) 1–8
25. Fattal, R.: Single image dehazing. ACM transactions on graphics (TOG) **27** (2008) 1–9
26. He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE transactions on pattern analysis and machine intelligence **33** (2010) 2341–2353
27. Berman, D., Treibitz, T., Avidan, S.: Single image dehazing using haze-lines. IEEE transactions on pattern analysis and machine intelligence **42** (2018) 720–734
28. Meng, G., Wang, Y., Duan, J., Xiang, S., Pan, C.: Efficient image dehazing with boundary constraint and contextual regularization. In: Proceedings of the IEEE international conference on computer vision. (2013) 617–624
29. Li, Y., You, S., Brown, M.S., Tan, R.T.: Haze visibility enhancement: A survey and quantitative benchmarking. Computer Vision and Image Understanding **165** (2017) 1–16
30. Ye, T., Jiang, M., Zhang, Y., Chen, L., Chen, E., Chen, P., Lu, Z.: Perceiving and modeling density is all you need for image dehazing. arXiv preprint arXiv:2111.09733 (2021)
31. Lin, B., Zhang, S., Bao, F.: Gait recognition with multiple-temporal-scale 3d convolutional neural network. In: ACM MM. (2020)
32. Lin, B., Zhang, S., Yu, X.: Gait recognition via effective global-local feature representation and local temporal aggregation. In: ICCV. (2021)
33. Ren, W., Liu, S., Zhang, H., Pan, J., Cao, X., Yang, M.H.: Single image dehazing via multi-scale convolutional neural networks. In: European conference on computer vision, Springer (2016) 154–169
34. Ren, W., Ma, L., Zhang, J., Pan, J., Cao, X., Liu, W., Yang, M.H.: Gated fusion network for single image dehazing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2018) 3253–3261
35. Li, R., Pan, J., Li, Z., Tang, J.: Single image dehazing via conditional generative adversarial network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. (2018) 8202–8211

36. Jin, Y., Sharma, A., Tan, R.T.: Dc-shadownet: Single-image hard and soft shadow removal using unsupervised domain-classifier guided network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. (2021) 5027–5036

37. Qin, X., Wang, Z., Bai, Y., Xie, X., Jia, H.: Ffa-net: Feature fusion attention network for single image dehazing. In: Proceedings of the AAAI Conference on Artificial Intelligence. Volume 34. (2020) 11908–11915

38. Dudhane, A., Singh Aulakh, H., Murala, S.: Ri-gan: An end-to-end network for single image haze removal. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2019) 0–0

39. Bianco, S., Celona, L., Piccoli, F., Schettini, R.: High-resolution single image dehazing using encoder-decoder architecture. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2019) 0–0

40. Morales, P., Klinghoffer, T., Jae Lee, S.: Feature forwarding for efficient single image dehazing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. (2019) 0–0

41. Ancuti, C.O., Ancuti, C., Sbert, M., Timofte, R.: Dense-haze: A benchmark for image dehazing with dense-haze and haze-free images. In: 2019 IEEE international conference on image processing (ICIP), IEEE (2019) 1014–1018

42. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020)

43. Li, B., Ren, W., Fu, D., Tao, D., Feng, D., Zeng, W., Wang, Z.: Benchmarking single-image dehazing and beyond. IEEE Transactions on Image Processing **28** (2018) 492–505

44. Li, R., Tan, R.T., Cheong, L.F., Aviles-Rivero, A.I., Fan, Q., Schonlieb, C.B.: Rainflow: Optical flow under rain streaks and rain veiling effect. In: Proceedings of the IEEE International Conference on Computer Vision. (2019) 7304–7313

45. Buchsbaum, G.: A spatial processor model for object colour perception. Journal of the Franklin institute **310** (1980) 1–26

46. Zheng, C., Cham, T.J., Cai, J.: The spatially-correlative loss for various image translation tasks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2021) 16407–16417

47. Kolkin, N., Salavon, J., Shakhnarovich, G.: Style transfer by relaxed optimal transport and self-similarity. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2019) 10051–10060

48. Jin, Y., Yang, W., Tan, R.T.: Unsupervised night image enhancement: When layer decomposition meets light-effects suppression. arXiv preprint arXiv:2207.10564 (2022)

49. Tumanyan, N., Bar-Tal, O., Bagon, S., Dekel, T.: Splicing vit features for semantic appearance transfer. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. (2022) 10748–10757

50. Amir, S., Gandelsman, Y., Bagon, S., Dekel, T.: Deep vit features as dense visual descriptors. arXiv preprint arXiv:2112.05814 (2021)

51. Ancuti, C.O., Ancuti, C., Timofte, R., De Vleeschouwer, C.: O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In: Proceedings of the IEEE conference on computer vision and pattern recognition workshops. (2018) 754–762

52. Kendall, A., Gal, Y.: What uncertainties do we need in bayesian deep learning for computer vision? Advances in neural information processing systems **30** (2017)

53. Ning, Q., Dong, W., Li, X., Wu, J., Shi, G.: Uncertainty-driven loss for single image super-resolution. Advances in Neural Information Processing Systems **34** (2021) 16398–16409
54. Zheng, Z., Ren, W., Cao, X., Hu, X., Wang, T., Song, F., Jia, X.: Ultra-high-definition image dehazing via multi-guided bilateral learning. In: 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), IEEE (2021) 16180–16189
55. Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B.: The cityscapes dataset for semantic urban scene understanding. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (2016) 3213–3223
56. Ancuti, C.O., Ancuti, C., Timofte, R.: Nh-haze: An image dehazing benchmark with non-homogeneous hazy and haze-free images. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. (2020) 444–445
57. Ancuti, C.O., Ancuti, C., Vasluianu, F.A., Timofte, R.: Ntire 2020 challenge on nonhomogeneous dehazing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. (2020) 490–491