# Nighttime Defogging Using High-Low Frequency Decomposition and Grayscale-Color Networks

Wending Yan<sup>1[0000-0001-5993-8405]</sup>, Robby T. Tan<sup>1,2[0000-0001-7532-6919]</sup>, and Dengxin Dai<sup>3[0000-0001-5440-9678]</sup>

<sup>1</sup>National University of Singapore, <sup>2</sup>Yale-NUS College, <sup>3</sup>ETH Zurich {eleyanw, robby.tan}@nus.edu.sg, dai@vision.ee.ethz.ch

Abstract. We address the problem of nighttime defogging from a single image by introducing a framework consisting of two modules: gravscale and color modules. Given an RGB foggy nighttime image, our grayscale module takes the grayscale version of the image as input, and decomposes it into high and low frequency layers. The high frequency layers contain the scene texture information, which is less affected by fog. While the low frequency layers contain the scene layout/structure information including fog and glow. Our grayscale module then enhances the visibility of the textures in the high frequency layers, and removes the presence of glow and fog in the low frequency layers. Having processed the high/low frequency information, it fuses the two layers to obtain a grayscale defogged image. Our second module, the color module, takes the original RGB image, and process it similarly to what the grayscale module does. However, to obtain fog-free high and low frequency information, the module is guided by the grayscale module. The reason of doing this is because grayscale images are less affected by multiple colors of atmospheric light, which are commonly present in nighttime scenes. Moreover, having the grayscale module allows us to have consistency losses between the outputs of the two modules, which is critical to our framework, since we do not have paired ground-truths for our real data. Our extensive experiments on real foggy nighttime images show the effectiveness of our method.

### 1 Introduction

Fog in nighttime can significantly degrades visibility. Few methods have been proposed to address this degradation problem, e.g.: [27, 17, 26]. Unfortunately, due to the complexity of illumination colors, the presence of multiple light sources and the strong glow, these methods still perform suboptimally as shown in Fig. 1. To our knowledge, there is no deep-learning based nighttime defogging method that has been proposed so far. The major reason is because collecting pairs of foggy nighttime images and their corresponding clear nighttime images, which are necessary for training a network, is significantly intractable. Rendering synthetic foggy nighttime images is a possible solution, however to render them



Li et al. [17]

Zhang et al. [26]

Fig. 1: Top left: Input image. Top right: Our result. Bottom left: Li et al.'s result [17]. Bottom right: Zhang et al.'s result [26]. These methods are the state of the art in nighttime defogging. Zoom-in for better visualization.

physically realistic for many different nighttime scenes and atmospheric conditions is challenging, since various information is required (such as depths, light source locations, particle distributions, how lights with different colors and intensities interact, etc).

Unlike foggy daytime images, foggy nighttime images suffer from low light, strong noise, multiple light sources, non-uniform distribution of atmospheric light intensity, multiple colors of lights, strong glow, etc. Because of these problems, directly applying existing daytime defogging methods to foggy nighttime images will not be effective. One of the reasons is that most of the methods (e.g. [24, 7, 12, 19, 3, 22]) assume a single atmospheric light (thus, a uniformly ambient light color), which is inapplicable to most of foggy nighttime images, since there are usually many man-made light sources with different colors in a single image.

In this paper, we introduce a framework that consists of grayscale and color modules. Given an RGB input image, the grayscale module takes the grayscale version of the input RGB image as input. The grayscale module decomposes the image into high and low frequency layers. These layers are then processed by two separate networks: a network for the high frequency layers that enhances the textures, and a network for the low frequency layers that removes glow and fog. Subsequently, fusing the processed high and low frequency information will produce a defogged grayscale image. Our second module, the color module, takes the original RGB image as input. The operations in the color module are similar to those in the grayscale module. Unlike the grayscale module, however, we train the module based on the outputs of the grayscale module. The reason of doing this is because we found defogging a grayscale image is more effective than defogging the RGB version of the image alone. This is mainly because, the grayscale image is less complicated by multiple colors of atmospheric light (see the detailed discussion in Sect. 4). Moreover, having the grayscale module, we can have consistency losses between the outputs of the two modules, which is critical to our framework, since we do not have paired ground-truths for our real data. To train our networks, we employ a few foggy-nighttime synthetic images, which come with ground-truths, and real foggy-nighttime images that have no paired ground-truths.

As a summary, our contributions are as follows:

- We introduce a nighttime defogging method based on the grayscale and color modules. Our method works for both grayscale and RGB input image.
- We propose the use of the low and high frequency layer decomposition. The high frequency layer contains the scene texture information, which is less affected by fog and ambient light colors; while, the low frequency layer contains the scene structure information affected by fog and glow.
- We introduce new consistency losses between the outputs of the grayscale and color modules, which are useful to strengthen the training process, particularly when we do not have paired ground-truths.

# 2 Related Work

Many methods have been proposed to deal with fog or haze in daytime (e.g., [24, 7, 12, 19, 3, 22, 14, 25, 23, 15, 6]). Tan [24] proposes a method by maximizing contrast. Fattal [7] estimates the transmission map by assuming that it is independent on the surface shading. He et al. [12] propose with the dark channel prior. Meng et al. [19] predict the transmission map with minimum boundaries and contextual regularization. Berman et al. [3] present a haze-line prior. In the era of deep learning, Ren et al. [22] design a multi-scale network for estimating the transmission map. Li et al. [14] propose all-in-one network, which predicted defogged images without estimating the transmission map and atmospheric light. Li et al. [15] apply a conditional GAN and Engin et al. [6] applied a cycle GAN to their method. More detailed discussion can be found in [18]. However, all these existing methods do not have good performance in nighttime fog removal. They have problem in estimating the atmospheric light. Even CNN-based methods have not been shown to be able to handle nighttime complex ambient lights, due to the lack of paired ground-truths.

There are few methods that specifically address nighttime fog removal. Pei and Lee [20] transfer the colors of a foggy nighttime image into the foggy daytime style, and apply the dark channel prior. Zhang et al. [27] propose a method using varying illumination compensation and post-processing. This technique brings more realistic results, but also color artifacts. Li et al. [17] apply glow removal before fog removal. The glow is decomposed from input image by a layer separation method. Then, the deglowed image is further defogged using



Fig. 2: The pipeline of our framework that consists of two main modules: grayscale module (top) and the color module (bottom). For the images, zoom-in for better visualization.

the dark channel prior. Ancuti et al. [1] provide a multi-scale fusion approach to enhance the nighttime fog visibility. They compute the atmospheric light component on an image patch level and derive three components, which will be weighed and fused to obtain the final output. Zhang et al. [26] propose a prior named maximum reflectance prior. They claim that the ambient illumination can be estimated by this prior. Like most of the nighttime defogging methods, the method employs the dark prior channel.

# 3 Proposed Method

Fig. 2 shows the pipeline of our method, which consists of two modules: grayscale and color modules. Given an RGB input image, the grayscale module takes the grayscale version of the input rgb image as input. It then decomposes the input into high and low frequency layers. A network dedicated to process the high frequency layers boosts the texture information of the layers. Another network dedicated to process the low frequency layers removes glow and fog. From the low and high frequency layers, a weight map is computed to suppress the noise of the processed high frequency information. Finally, the grayscale defogged output is obtained by fusing the low/high frequency information. The color module does similar processes, and thus we can create losses that can keep the outputs of the grayscale and color modules to be consistent. Note that, for real training data, we do not have paired ground-truths. The details of the discussion on the two modules are as follows.

#### 3.1 Grayscale Module

There are two main reasons of having the grayscale module in our solution. First, in nighttime, ambient lights with multiple colors are common. This is due to the fact that there are many man-made light sources with various colors. Unlike nighttime colored images, nighttime grayscale images are less complicated by the colors of multiple light (more discussion in Sect. 4). Second, for real foggy nighttime images, we do not have their clean fog-free ground-truths. Hence, by having the grayscale module, we can have additional constraints by enforcing the consistency between the outputs of the grayscale and color modules.

High/Low Frequency Layers and Networks The grayscale module decomposes the input image into a few high and low frequency layers. Any state of the art decomposition techniques can be used (e.g. [11]). In our implementation, we produces 8 high frequency layers, and their 8 corresponding low frequency layers using different decomposition parameters. The reason of doing this is to reduce missing information possibly caused by inappropriate choice of parameters with respect to the input image. The high frequency layers contain the scene textures, and the low frequency layers contain the scene layout/structures. Due to their smoothness, fog and glow will go to the low frequency layers, and consequently the high frequency layers is not much affected by fog and glow (see Fig. 2). We separate the high and low frequency is because they represent different aspects of a foggy scene, and hence they should be processed differently. Otherwise, when removing fog, we may also remove genuine textures of the scene.

After decomposition, we process the high frequency layers to boost the texture information. The main reason of why we want to boost the textures is because in foggy nighttime images, the contrast is low, and thus the textures can be weak. We design an autoencoder network to handle this task, and we call it high-frequency network. The input of this network is the concatenation of the high frequency layers. Like the high frequency part, for the low frequency layers, we create a low-frequency network to process them. Our intention of creating the network is to suppress the presence of fog and glow. We call the outputs of the high/low frequency networks high/low frequency maps, respectively.

**Fusing High/Low Frequency Maps** Having obtained the high/low frequency maps, we can directly fuse them. However, since boosting the textures in the high frequency layers also means boosting the noise, prior fusing the two maps, we compute a weight map. The weights in this map represents the noise levels of the

high frequency map. Thus, multiplying the weight map with the high frequency map will reduce noise. Note that, foggy nighttime images suffer significantly from noise; this is because many parts of foggy nighttime images are regions with low light. We adopt an existing technique [16] to compute the weight map, where we first generate a coarse map from the discrete cosine transform (DCT) coefficients of the high frequency map. Larger DCT coefficients represent more texture details. Thus, we set the sum of coefficients as the coarse weight map. However, since the DCT is computed patch by patch, the weight map may looks coarse. As suggested in [16], we utilize the low frequency map to smoothen the weights. This refined weight map has high values on texture regions and low values on uniform regions like sky regions. Fig. 2, the top right image, shows a sample of our weight map.

In the grayscale module, we have a few losses: smoothness, pair and discriminative. The smoothness loss is applied only to the low frequency network; while the grayscale pair and discriminative losses are applied to the final defogged image. Hence, the backpropagated errors from the grayscale pair and discriminative losses will go to both the high and low frequency networks.

- Grayscale Smoothness Loss The loss is defined as:

$$\mathcal{L}_{smooth}^{gray} = \mathbb{E}\big[ (J_{gray}^f - LF_{gray}[J_{gray}^f]) * f_2 \big], \tag{1}$$

where LF[.] is the low frequency network, and  $J_{gray}^{f}$  is the grayscale foggy nightime input (the superscript f stands for fog).  $f_{2}$  is the second order Laplacian filter and \* is the convolution operator. The loss enforces that the removed fog and glow by the network must be smooth spatially. The subtraction is done pixel by pixel. The operator  $\mathbb{E}[S]$ , where S is any matrix, implies adding every element of the matrix, and dividing the total sum with the size of the matrix.

 Grayscale Pair Loss Our synthetic nightime foggy data has paired clean ground-truths. Thus, we can employ the pair loss:

$$\mathcal{L}_{pair}^{gray} = \mathbb{E}\left[ \|M_{gray}[J_{gray}^f] - J_{gray}^{gt}\| \right], \tag{2}$$

where  $J_{gray}^{f}$  and  $J_{gray}^{gt}$  are paired synthetic grayscale fog image and clean ground-truth image, respectively.  $M_{gray}[.]$  is our grayscale module.

- Grayscale Discriminative Loss We apply a discriminative loss only to real images that do not have paired ground-truths. We define our discriminative loss as:

$$\mathcal{L}_{GAN}^{gray} = -\log(\sigma[D_{gray}[M_{gray}[J_{gray}^f]]]), \qquad (3)$$

where  $\sigma[.]$  is the sigmoid function, and  $D_{gray}$  is the discriminator [9].

Overall, the total loss functions of the grayscale module is:

$$\mathcal{L}_{total}^{gray} = \mathcal{L}_{pair}^{gray} + \lambda_1^{gray} \mathcal{L}_{GAN}^{gray} + \lambda_2^{gray} \mathcal{L}_{smooth}^{gray}, \tag{4}$$

 $\overline{7}$ 



Fig. 3: The inputs and outputs of our grayscale module. Two left images show a pair of our input and output. Two right images show the same. Zoom-in for better visualization.

where  $\lambda_1^{gray}$  and  $\lambda_2^{gray}$  are the weighting factors. Empirically in our experiments, we set  $\lambda_1^{gray} = 0.1$  and  $\lambda_2^{gray} = 1$ .

Fig. 3 shows some defogging results using the grayscale module. The glow and fog in input images are significantly removed, while the textures of the background are still retained. Note that, fog-free nighttime images without glow and sufficient lighting usually suffer from the low light problem. This is because glow is in fact light scattered by particles in the atmosphere. Hence, after we successfully remove fog and thus glow, it is natural that the image looks dimmer than the input image. We will discuss about this further in the later part of the following section.

#### 3.2 Color Module

In the color module, we take the original RGB input image as input. The operations in this module are similar to those of the grayscale module. However, in this module, we do not apply the high and low frequency decomposition technique like in the grayscale module. Instead, we input the RGB foggy nighttime image to the low frequency network directly, and the grayscale version of the input image to the high frequency network. The high and low frequency networks in this module are then guided by the counterparts in the grayscale module.

**High/Low Frequency Networks** Since the high frequency map is independent from colors, the input of the high frequency network is the grayscale version of the input image. Like its counterpart in the grayscale module, the task of the high frequency network is to boost textures. The high frequency network learns to boost textures largely from the consistency loss that measures the consistency between its output and the outputs of the high frequency network in the

grayscale module. Note that, unlike the high frequency network in the grayscale module that takes high frequency layers as input, in this module, the high frequency module takes the grayscale input image. Hence, we cannot simply use the one in the grayscale module.

Similarly, the low frequency network is also trained using the consistency with respect to the output of the low frequency counterpart in the grayscale module. However, different from the low frequency network in the grayscale module, the low frequency network in the color module produces a colored image. Hence, we convert the colored low frequency map to grayscale before applying the consistency loss. Doing this process, however, allows the color in the low frequency map to be shifted, since for real data, we do not have their paired ground-truths. To tackle this problem, we need another loss that ensures the correctness of the colors of the low frequency map. Here are all the losses for training the high and low frequency networks in the color module. Note that, after all the networks in the grayscale module have been trained, we freeze them.

#### - High Frequency Consistency Loss We define it as:

$$\mathcal{L}_{HF}^{rgb} = \mathbb{E}\left[ \| [HF_{gray}[J_{gray}^{f}]] - HF_{rgb}[J_{gray}^{f}] \| \right], \tag{5}$$

where  $HF_{gray}[.]$  and  $HF_{rgb}[.]$  are the low frequency networks from the grayscale and color modules, respectively.

- Low Frequency Consistency Loss We define it as:

$$\mathcal{L}_{LF}^{rgb} = \mathbb{E} \left[ \| LF_{gray}[J_{gray}^f] - G \left( LF_{rgb}[J_{rgb}^f] \right) \| \right], \tag{6}$$

where  $LF_{gray}[.]$  and  $LF_{rgb}[.]$  are the low frequency networks from the grayscale and color modules, respectively. Function G(.) is to convert an rgb image to its grayscale version.

- Hue Loss This loss is to ensure there is no color shift in the output of the low frequency network. The basic idea is that if we can estimate the color (or chromaticity) of the atmospheric light of the input RGB image, then we can normalize the image such that the atmospheric light colors will be canceled. Once the input image is normalized, it is known that the hue values of the input fog image will be the same as those of the normalized clean fog-free image (e.g. [13]). Hence, we define the loss as:

$$\mathcal{L}_{hue}^{rgb} = \mathbb{E} \left[ \|Hue[CN[LF_{rgb}[J_{rgb}^{f}], A[x_{rgb}^{f}]]] - Hue[CN[J_{rgb}^{f}, A[J_{rgb}^{f}]]\| \right],$$
(7)

where Hue[.] are a function to compute the hue values of an image. CN[.] is the function that normalizes an image based on the atmospheric light chromaticity. A[.] is the function to estimate the atmospheric light chromaticity. We can use any existing technique to estimate the atmospheric light chromaticity for every pixel such as that described in [17, 26]. **Fusing High and Colored Low Frequency Maps** Having obtained the grayscale high and colored low frequency maps, we multiple the grayscale high frequency map with the weight map we obtained in the grayscale module to suppress noise. Subsequently, we fuse the weighted high frequency map and the colored low frequency map. To ensure good quality of outputs, we also impose a few loss functions to train the networks in this color module:

 Color Pair Loss This loss is only for our paired synthetic data. The loss is defined as:

$$\mathcal{L}_{pair}^{rgb} = \mathbb{E}\big[ \|M_{rgb}[J_{rgb}^f] - J_{rgb}^{gt}\|\big],\tag{8}$$

where  $x_{rgb}^{f}$  and  $x_{rgb}^{gt}$  are paired synthetic fog image and clean ground-truth image, respectively.

- Color Discriminative Loss We define the loss as:

$$\mathcal{L}_{GAN}^{rgb} = -\log(\sigma[D_{rgb}[M_{rgb}[J_{rgb}^{f}]]]), \qquad (9)$$

where  $x_{rab}^{J}$  is a real colored foggy nighttime image.

 Image Metric Loss We include an unsupervised loss based on contrast and acutance [2], which are defined as:

$$\mathcal{L}_{IQ}^{rgb} = \mathbb{E} \left[ (1 - C[M_{rgb}[J_{rgb}^{f}]] + C[J_{rgb}^{f}]) + (1 - Ac[M_{rgb}[J_{rab}^{f}]] + Ac[J_{rgb}^{f}]) \right],$$
(10)

where the C[.] and Ac[.] are the functions to obtain the contrast and acutance of an image.

Overall, the total loss functions for color module is:

$$\mathcal{L}_{total}^{rgb} = \mathcal{L}_{pair}^{rgb} + \lambda_1^{rgb} \mathcal{L}_{GAN}^{rgb} + \lambda_2^{rgb} \mathcal{L}_{LF}^{rgb} + \lambda_3^{rgb} \mathcal{L}_{HF}^{rgb} + \lambda_4^{rgb} \mathcal{L}_{hue}^{rgb} + \lambda_5^{rgb} \mathcal{L}_{IQ}^{rgb},$$
(11)

where  $\lambda_i^{rgb}$ ,  $i = \{1...5\}$  are the weighting factors. Empirically in our experiments, we set  $\lambda_1^{rgb} = 0.1$ ,  $\lambda_2^{rgb} = 1$ ,  $\lambda_3^{rgb} = 1$ ,  $\lambda_4^{rgb} = 0.5$  and  $\lambda_5^{rgb} = 1$ .

**Intensity Boosting** Degradation in foggy nighttime images is caused not only by fog but also by other nighttime conditions such as low light and noise. Hence, successfully removing fog does not mean that we can have an output that is bright and noise free, since low light and noise are inherent even in clear nighttime images. In fact, foggy nighttime images look relatively brighter than clear nighttime images mainly due to the presence of glow (= the light scattered into the atmosphere). Hence, when we successfully remove fog and thus glow, implying removing the scattered light, it is natural that the images look darker.

Nevertheless, we can boost the intensity of our output to improve the visibility quality using our current framework. Although we have to note that, the



Fig. 4: After modified the low frequency consistency loss, the image brightness is boosted.

issue of nighttime visibility enhancement is still an open problem (e.g. [10, 4]), and thus effective nighttime/low-light visibility enhancement is actually beyond the scope of our defogging paper. In our intensity boosting, we simply modify the low frequency consistency loss in Eq.(6), namely by brightening up the reference grayscale low-frequency map; since, the low frequency map in the color module influences the tone and brightness of the final output. Our modified loss is expressed as:

#### - Modified Low Frequency Consistency Loss:

$$\mathcal{L}_{LF}^{boost} = \mathbb{E}\left[ \left\| B\left( \left[ LF_{gray}[J_{gray}^{f}] \right] \right) - G\left( LF_{rgb}[J_{rgb}^{f}] \right) \right\| \right], \tag{12}$$

where B(.) is the intensity boosting function, where we use a gamma function. G(S), where S is any matrix, means for each element in S we power it to  $\gamma$ . In other words:  $S_{ij}^{\gamma}$ , where  $\gamma$  in our experiment is 0.5. As a result of this, In Eq. (11), the loss  $\mathcal{L}_{LF}^{rgb}$  is replaced by this loss  $\mathcal{L}_{LF}^{boost}$ .

Fig. 4 shows the results of our intensity boosting.

# 4 Discussion: Multiple Light Colors

In this section, we discuss why using grayscale images (and thus our grayscale module) can be more effective in dealing with multiple-light colors, which are commonly present in the nighttime images.

Most daytime defogging methods (e.g., [24, 8, 12]) assume the atmospheric light is uniform across the input image. , which is a reasonable assumption, since in foggy daytime, the dominant light source is only the skylight. This assumption can thus simplify the problem of defogging, since we do not need to estimate the atmospheric light **A** for every pixel locally, reducing the complexity of the already ill-posed problem. Unfortunately, this assumption cannot be applied to most foggy nighttime images. Since in foggy nighttime, the skylight is usually much dimmer than the man-made light sources; and, there are multiple intensities/colors of these light sources, and their potential combination. This implies, that we need to estimate the atmospheric light locally for every pixel



Fig. 5: Comparison of the high and low frequency maps: The color module without guidance, the grayscale module and the color module with guidance from the grayscale module.

in the input image, which render a more complication of the problem. Existing nighttime defogging methods (e.g. [17, 1, 26]) propose a few techniques to estimate the atmospheric light locally, based on the patches of the input image. The basic assumption is that in a small patch, there is a uniform atmospheric light. While to some extent the proposed techniques work, their accuracy relies on the decided patch size. If the size is too large, the accuracy will drop, since the patch can actually contains multiple light colors. And, if the size is too small, the accuracy will also drop, since the background intensity/color can be uniform and mistakenly estimated as the light intensity/color

In our proposed method, to deal with the problem, we leverage the grayscale version of the RGB input image. Our underlying idea is that grayscale images do not have color information, and thus the problem of multiple colors of the atmospheric light is not present. This reduces the complexity of the problem, and our graysacle module can focus on extracting the background properties, i.e., high and low frequencies, without worrying the correctness of their colors. Fig. 5 show the results of the high and low frequency from an RGB image (processing them directly without the guidance from its grayscale images), from a grayscale version of the image, and from our color module guided by our grayscale module. As can be observed, the grayscale high/low frequency can reveal more about the background properties. Moreover, the one without the guidance shows more artifacts, fogginess and color shifts in some regions.

# 5 Experimental Results

In our experiments, we compare our method with the following baselines: Li et al. [17], Ancuti et al. [1], and Zhang et al. [26], which are the three stateof-art nighttime defogging methods. Besides, we also compare our results with

Table 1: Quantitative results on our synthetic foggy nighttime data.		
	PSNR	SSIM
Input Image	18.987	0.6764
Li et al. [17]	21.024	0.6394
Zhang et al. [26]	20.921	0.6461
Ancuti et al. [1]	20.585	0.6233
Berman et al. [3]	19.085	0.5373
EPDN [21]	22.565	0.7330
Without Teacher Module	26.163	0.8185
Without Decomposition	26.328	0.8219
Without Weights Map	26.892	0.8327
Our Result	26.997	0.8499



Fig. 6: Ablation studies on the high/low frequency decomposition, grayscale/color modules, loss functions and weight map. Zoom-in for better visualization.

the state-of-art daytime defogging methods Berman et al. [3] and EPDN [21]. In our training process, we combine synthetic data with ground-truths and real foggy nighttime data without ground-truths. The synthetic data is generated by a video game engine, GTA5 [5], and our real foggy nighttime data are collected from the Internet.

For the quantitative evaluation, we use 200 pairs of synthetic data, which are generated from GTA5. The quantitative evaluation results are shown in Table 1, where our method shows better performance on both PSNR and SSIM by significant margins compared to all the baseline methods. Fig. 7 shows the qualitative evaluation results on real foggy nighttime images. As one can notice that our method qualitatively provides better defogging results compared with the results of the other baseline methods.



Fig. 7: Qualitative comparisons with the state of the art methods on real images. Zoom-in for better visualization.

#### 5.1 Ablation Studies

To show the effectiveness of our grayscale module in helping the color module, we train our color module without the losses that involve the grayscale module. In other words, we cut off the color module from the grayscale module. The network architecture in this case is similar to the grayscale module, but the input of the low frequency network is the RGB low frequency layers (instead of the grayscale low frequency layers), and the same decomposition technique is applied. The first row second column of Fig. 6 shows the defogging results. As can be seen, the fog is still considerably noticeable.

To show the effectiveness of the high/low frequency decomposition, we remove the two networks in our modules, replace them with one autoencoder network, and apply our losses that are relevant. As shown in the first row third column of Fig. 6, there are some regions still affected by fog, as the network is mistaken it with genuine textures. The ablation studies on the high/low frequency grayscale guidance (i.e. the high/low frequency consistency losses) are shown in the last two columns and first row of Fig. 6.

The second row of Fig. 6 shows that the effectiveness of other losses. The intensity boosting is important, since the image becomes dimmer after we remove the fog and glow. Without the hue loss, the outputs have red or other color shifting. Our hue loss can constrain the color shifting of the RGB outputs. To show the effectiveness of our image metric loss, we remove this loss from our color module. The results are shown in the second row and third column in Fig. 6. After adding this unsupervised image metric loss, the outputs of our color module are constrained to have higher contrast and higher fidelity. The second row fourth column of the figure shows the results when we do not apply the weight map. By zooming-in the images, one can notice some visible noise and artifacts particularly in the dark regions. By applying the weight map, the noise/artifact will be suppressed, as shown in the last column of the figure.

### 6 Conclusion

We have introduced a learning-based nighttime defogging method. To our knowledge, this is the first time, a deep learning-based method is dedicated to handle nighttime defogging problem. To achieve our goal, we design grayscale and color modules, which rely mainly on the high/low frequency layers to enhance textures and at the same time suppress glow, fog and noise. Due to the lack of paired real ground-truths, our training process employs both paired synthetic data and unpaired real data. For this, we introduce new consistency losses between the outputs of the grayscale and color modules. Experimental results and evaluations, both quantitative and qualitative, show the effectiveness of our method.

# Acknowledgment

This work is supported by MOE2019-T2-1-130.

15

## References

- Ancuti, C., Ancuti, C.O., De Vleeschouwer, C., Bovik, A.C.: Night-time dehazing by fusion. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 2256–2260. IEEE (2016)
- Barbosa, W.V., Amaral, H.G., Rocha, T.L., Nascimento, E.R.: Visual-qualitydriven learning for underwater vision enhancement. In: 2018 25th IEEE International Conference on Image Processing (ICIP). pp. 3933–3937. IEEE (2018)
- 3. Berman, D., Avidan, S., et al.: Non-local image dehazing. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1674–1682 (2016)
- Chen, C., Chen, Q., Xu, J., Koltun, V.: Learning to see in the dark. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3291– 3300 (2018)
- 5. Doan, A.D., Jawaid, A.M., Do, T.T., Chin, T.J.: G2D: from GTA to Data. arXiv preprint arXiv:1806.07381 pp. 1–9 (2018)
- Engin, D., Genc, A., Kemal Ekenel, H.: Cycle-dehaze: Enhanced cyclegan for single image dehazing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 825–833 (2018)
- Fattal, R.: Single image dehazing. ACM transactions on graphics (TOG) 27(3), 72 (2008)
- 8. Fattal, R.: Dehazing using color-lines 34(1), 13 (2014)
- Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems. pp. 2672–2680 (2014)
- Guo, X., Li, Y., Ling, H.: Lime: Low-light image enhancement via illumination map estimation. IEEE Transactions on Image Processing 26(2), 982–993 (2016)
- 11. He, K., Sun, J., Tang, X.: Guided image filtering. In: European conference on computer vision. pp. 1–14. Springer (2010)
- He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. IEEE transactions on pattern analysis and machine intelligence **33**(12), 2341–2353 (2011)
- Jiang, Y., Sun, C., Zhao, Y., Yang, L.: Image dehazing using adaptive bi-channel priors on superpixels. Computer Vision and Image Understanding 165, 17–32 (2017)
- 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. pp. 4770–4778 (2017)
- 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. pp. 8202–8211 (2018)
- Li, Y., Guo, F., Tan, R.T., Brown, M.S.: A contrast enhancement framework with jpeg artifacts suppression. In: European conference on computer vision. pp. 174– 188. Springer (2014)
- Li, Y., Tan, R.T., Brown, M.S.: Nighttime haze removal with glow and multiple light colors. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 226–234 (2015)
- Li, Y., You, S., Brown, M.S., Tan, R.T.: Haze visibility enhancement: A survey and quantitative benchmarking. Computer Vision and Image Understanding 165, 1–16 (2017)

- 16 W. Yan, R. Tan et al.
- 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. pp. 617–624 (2013)
- Pei, S.C., Lee, T.Y.: Nighttime haze removal using color transfer pre-processing and dark channel prior. In: 2012 19th IEEE International Conference on Image Processing. pp. 957–960. IEEE (2012)
- Qu, Y., Chen, Y., Huang, J., Xie, Y.: Enhanced pix2pix dehazing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8160–8168 (2019)
- 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. pp. 154–169. Springer (2016)
- 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. pp. 3253–3261 (2018)
- Tan, R.T.: Visibility in bad weather from a single image. In: 2008 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1–8. IEEE (2008)
- Zhang, H., Patel, V.M.: Densely connected pyramid dehazing network. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3194–3203 (2018)
- 26. Zhang, J., Cao, Y., Fang, S., Kang, Y., Wen Chen, C.: Fast haze removal for nighttime image using maximum reflectance prior. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7418–7426 (2017)
- 27. Zhang, J., Cao, Y., Wang, Z.: Nighttime haze removal based on a new imaging model. pp. 4557–4561. IEEE (2014)