
UNCLOUD: Universal Image Defogging Using Image Inpainting Techniques

Kyle Li*

University of Illinois at Urbana-Champaign
Champaign, IL 61820
kyleli2@illinois.edu

Deming Chen

University of Illinois at Urbana-Champaign
Champaign, IL 61820
dchen@illinois.edu

Yuhong Li

University of Illinois at Urbana-Champaign
Champaign, IL 61820
leeyh@illinois.edu

Abhi Kamboj

University of Illinois at Urbana-Champaign
Champaign, IL 61820
akamboj2@illinois.edu

Abstract

In recent years, image defogging has made significant progress. However, many defogging methods either rely on expensive sensors or lack the ability to achieve optimal results under all types of foggy conditions using a singular trained model. In this paper, we propose UNCLOUD, a universal image-based defogging method. UNCLOUD utilizes an unconventional novel approach that treats foggy areas of an image as irregular partial-holes and attempts to recover a non-foggy image by using image inpainting techniques. Specifically, we propose the usage of masked convolution layers, where convolutions are weighted and renormalized in proportion to the estimated intensity of fog at each pixel. Experiments on benchmark images demonstrate that in comparison to other singular image-based defogging methods, UNCLOUD achieves superior performance under all types of foggy weather conditions. In our testing, UNCLOUD achieved an SSIM of 0.47 and PSNR of 13.64 on the DENSE-HAZE dataset, while existing image-based methods only achieve a maximum SSIM and PSNR of 0.44 and 11.43 respectively.

1 Introduction

Fog, haze, and other obscuring weather conditions are common atmospheric phenomena that involve the suspension of particles in the air. Images taken under such conditions are often partially obscured and experience reduced clarity. The objective of image defogging is to estimate a fog-free image from an observed foggy image. Image defogging is highly applicable to vision-based systems. These include surveillance and autonomous driving, which are critical to security and traffic safety. Despite the importance of these systems, they are not always able to perform properly under all weather conditions, and they become particularly ineffective in obscuring foggy conditions.

1.1 Background

The degradation of foggy images perceived by the human eye and vision-based systems can be modeled by the following equation [15, 17, 16]:

$$I(x) = J(x)t(x) + A(1 - t(x)) \quad (1)$$

*<https://www.kylebli.com>



Figure 1: Image-based defogging by UNCLOUD. Top row: foggy image. Bottom row: corresponding image defogged by UNCLOUD.

where $I(x)$ is the observed foggy image, $J(x)$ is the non-foggy image, $t(x)$ is the medium transmission map, A is the global atmospheric light, and x indexes pixels in the images.

The transmission map $t(x)$ represents the portion of light that is not scattered by fog and reaches the camera. $t(x)$ can be expressed as

$$t(x) = e^{-\beta d(x)} \quad (2)$$

where β is the atmospheric scattering coefficient and $d(x)$ is the scene depth.

Using Eq.(1), the task of image defogging, where the objective is to recover non-foggy image J given foggy image I , can be simplified to estimating the two unknowns, A and t . While early methods [9] show that it is entirely possible to algorithmically estimate A and t to recover a good-visibility J using Eq.(1), their results often times show visible deviation from the ground truth. Thus, more recent methods introduce various deep learning methods to estimate A and t , or bypass Eq.(1) entirely and learn the relationship between I , A , t , and J from scratch.

1.2 Objective

While there are existing methods of obtaining a defogged version of an image or video taken on a foggy day [12, 9, 22, 19, 14], not many methods provide a singular image-based model that can adapt to and remove fog from all fog conditions. Fog conditions include heavy fog, non-uniform fog, and no fog, in which case the model should leave the image unaltered. Instead of providing a singular universal model, many state-of-the-art solutions provide various versions of the same model that is trained to excel on specific data sets. The issue with this is that in real-world applications, foggy weather conditions are constantly changing, and it may be difficult for a defogging solution to decide which pre-trained model is most suitable for the observed foggy conditions.

In this paper, we propose a novel universal image-based defogging method: UNCLOUD. The purpose of universal image-based defogging is to be able to equip any vision-based system with a universal defogging solution, and not need to manually switch between defogging models based on the observed weather conditions.

The novelty of our proposed method, UNCLOUD, resides in its image-inpainting-based approach to defogging images using masked convolution layers. Image inpainting is the task of filling in holes in an image (see Fig. 2). UNCLOUD is based on the observation that foggy images are similar to images with holes in them. Specifically, the foggy portions of images, like holes, are parts of the image with missing information. However, unlike holes in an image, fog is a non-binary mask, meaning that foggy portions of an image are not devoid of all information, but instead are partially missing information, since part of the information, which is not scattered by fog, still makes it into the observed image. Thus, unlike traditional image inpainting, the image-inpainting-based approach

utilized by UNCLOUD accommodates the non-binary property of foggy images and adaptively considers the existing information beneath the fog at each pixel.

To achieve this, we propose the usage of a *Masked Convolution Layer*, which takes a foggy image and fog mask as input. A fog mask depicts the fog intensity at each corresponding pixel in a foggy image using a value between 0 and 1. The masked convolution layer then performs a masked convolution operation, which is a convolution that is weighted and renormalized in proportion to the fog mask, and subsequently performs a mask update step. The concept of a masked convolution layer is similar to the partial convolution layer proposed by Liu et al. [13] for image inpainting, however their weighted and renormalized convolution operation is strictly limited to binary masks. Our primary extension of this is the additional ability to accommodate non-binary masks, where the pixels in a foggy image and fog mask are incrementally updated based on the values of neighboring pixels and their corresponding mask. Using a transmission map, t , as an initial fog mask, UNCLOUD employs a series of masked convolution layers to iteratively remove fog from foggy images and produce reliable defogged images that outperform state-of-the-art models.

Similar to how image inpainting methods selectively fill in holes, the main benefit of using an image-inpainting-based approach to defogging is its ability to selectively defog pixels labeled as *foggy* in a given image. Under the assumption that UNCLOUD is able to accurately identify foggy pixels and their respective fog intensities, this image-inpainting-based approach enables UNCLOUD to universally defog images as it neither requires prior knowledge of the fog condition in a given image, nor does it require prior training on a foreign fog condition present in a given image.

In summary, our contributions are as follows:

- We propose UNCLOUD, which uses *masked convolutions* to achieve state of the art image defogging.
- We demonstrate that our image-inpainting-based approach to image defogging enables UNCLOUD to universally defog images and achieve state of the art results regardless of fog condition.
- To the best of our knowledge, we are the first to utilize such an approach to image defogging.

2 Related works

2.1 Dark channel prior

The dark channel prior [9] is defined as the observation that in non-foggy outdoor images, there should be at least one color channel with very low intensity in any non-sky patch of pixels. The dark channel is defined as the minimum of all color channels in a local patch, and is expressed as:

$$I^{dark}(x) = \min_{c \in \{r,g,b\}} \left(\min_{y \in \Omega(x)} I^c(y) \right) \quad (3)$$

where $I^c(y)$ is an RGB color channel of I corresponding to color channel c , and $\Omega(x)$ is a patch of pixels in I centered at index x . $I^{dark}(x)$, known as the dark channel, is statistically observed to have a low near-zero intensity in non-foggy images.

It is observed that the dark channel correlates closely with the transmission map t from Eq.(1) and Eq.(2). The approximation is expressed as $t(x) \propto 1 - I^{dark}(x)$. After further refinement, He et al. suggested the following equation to estimate $t(x)$ [9]:

$$\tilde{t}(x) = 1 - w \min_{c \in \{r,g,b\}} \left(\min_{y \in \Omega(x)} \left(\frac{I^c(y)}{A^c} \right) \right) \quad (4)$$

where $\tilde{t}(x)$ is the estimated transmission map, w ($0 < w \leq 1$) is a constant parameter that allows for a small degree of *naturally-occurring haze* to remain for distant objects, A^c is the estimated atmospheric light corresponding to color channel c , and the remaining variables hold the same meaning as they do in Eq.(3). Note the process for estimating A in foggy image I is as follows: First, pick the top 0.1% brightest pixels in the dark channel I^{dark} . Second, among these pixels, choose the pixel with the highest intensity in I to be atmospheric light A .

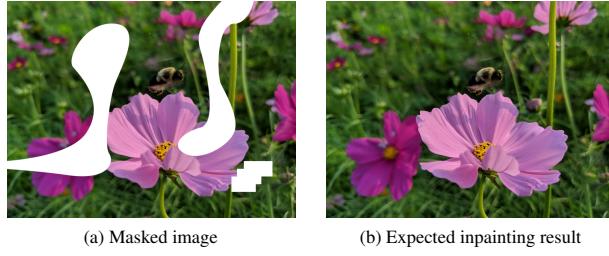


Figure 2: Image with holes and corresponding expected results from image inpainting.

2.2 Image inpainting

Image inpainting is the task of filling in holes in an image. These holes are portions of the image which are missing information and must be recovered by an image inpainting solution.

The state-of-the-art image inpainting method proposed by Liu et al. [13] takes an image with holes in it and a binary mask, which marks the holes in the image, as input. It then uses a series of partial convolution operations and mask updating steps to perform image inpainting. The partial convolution operation and mask updating step are jointly referred to as a Partial Convolution Layer.

The partial convolution operation at each location x is expressed as:

$$I'(x) = \begin{cases} W^T(I_X \odot M) \frac{\text{sum}(1)}{\text{sum}(M)} + b, & \text{if } \text{sum}(M) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

where $I'(x)$ is the new pixel value for pixel $I(x)$, W is the convolutional filter weights and b is the bias for the convolution filter, I_X represents pixel values centered at x for the current convolution window, M is the corresponding binary mask, 1 is the same shape as M but all elements are 1, and \odot is an element-wise multiplication operation.

The mask updating step of the partial convolution layer is expressed as:

$$M'(m) = \begin{cases} 1, & \text{if } \text{sum}(M) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where M is the binary mask corresponding to the convolutional window centered at m , and $M'(m)$ is the new mask value at index m .

The concept behind this method is that most image inpainting methods make heavy use of convolutional layers since they are highly effective in models related to computer vision. However, the convolutional layers in most of these methods equally weight the areas of the image that are holes, which contain useless information, and the areas of the image that are not holes. This causes non-hole portions of the image to be partially corrupted by useless information, and hole portions of the image to consider useless information when being restored by the image inpainting method. Hence, the partial convolution layer aims to fix this by masking out useless information during convolution.

3 Approach

Our proposed method first produces masks for each foggy image before feeding the images and their masks into a model that utilizes masked convolution layers.

3.1 Fog mask synthesis

The goal of fog mask synthesis is to obtain a mask that denotes all the fog pixels and their corresponding fog-intensity in foggy image I . To obtain a fog mask, we must compute a transmission map t , which can be referenced in equation Eq.(1), from a foggy image I . The transmission map t ultimately determines how much atmospheric light, A , obscures each area of the image I due to the fog. This makes the transmission map a suitable fog mask since the purpose of a fog mask is to inform the overall Universal Defogging Model of the intensity of the fog at each pixel.

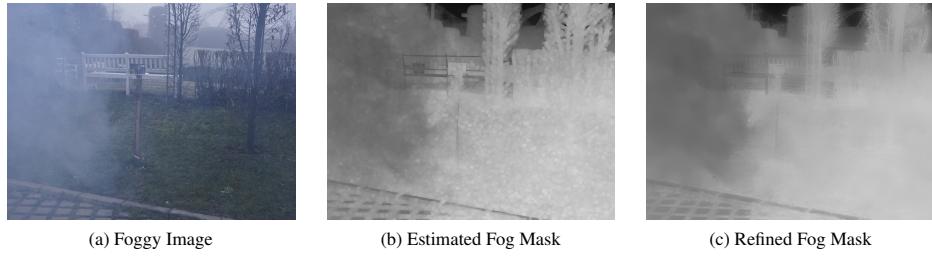


Figure 3: Fog mask synthesis results.

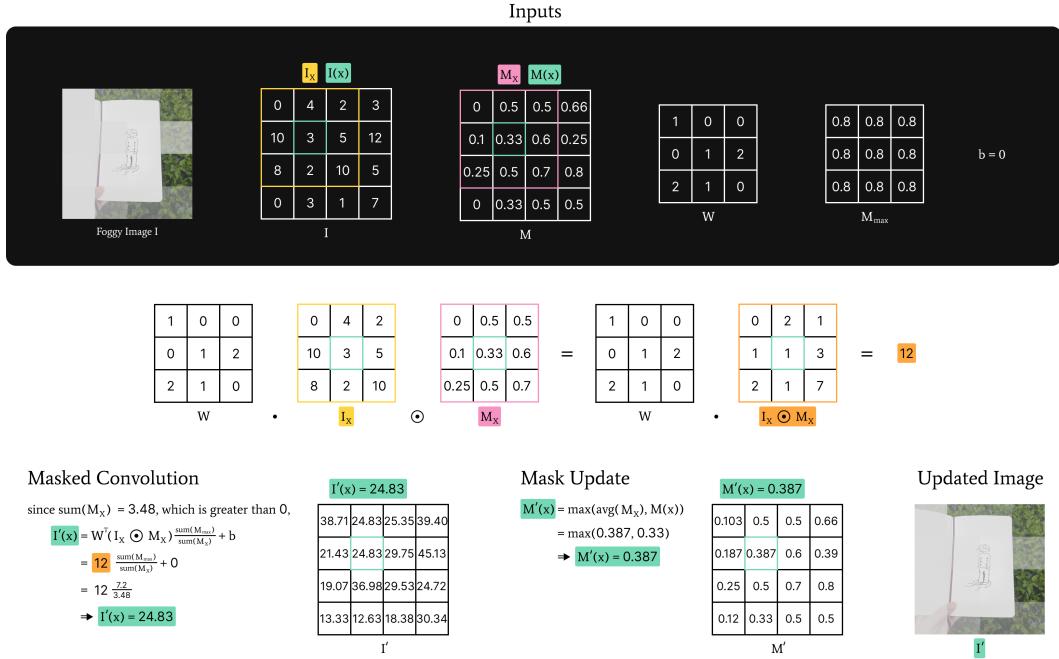


Figure 4: Masked convolution layer example.

The Dark Channel Prior method [9] outlined in Section 2.1 is used to produce the transmission map. First an estimated transmission map, depicted by Fig. 3.b, is computed using Eq.(4). This estimated transmission map is then refined using a soft matting algorithm [11] to produce the refined transmission map or fog mask in Fig. 3.c. In the fog mask, a pixel with a value of 0 indicates that the corresponding pixel in the foggy input image is completely obscured by fog, whereas a pixel from the transmission map with a value of 1 indicates that fog is completely absent from the corresponding pixel in the input foggy image.

3.2 Masked convolution layer

The masked convolution (MConv) layer comprises a masked convolution operation and a mask update step similar to the partial convolutional layer [13] outlined in Section 2.2. It takes a foggy image and a corresponding fog mask as input, and produces an updated image and updated mask as output. For our proposed method, the fog mask synthesis method described in Section 3.1 is used to synthesize the fog mask inputted into the initial MConv layer.

The masked convolution operation at each location x is expressed as:

$$I'(x) = \begin{cases} W^T(I_x \odot M_x) \frac{\text{sum}(M_{max})}{\text{sum}(M_x)} + b, & \text{if } \text{sum}(M_x) > 0 \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

where $I'(x)$ is the new pixel value for pixel $I(x)$, W is the convolutional filter weights, b is the bias for the convolution filter, I_X represents pixel values centered at x for the current convolution window, M_X is the corresponding binary mask, M_{\max} is the same shape as M_X but all elements equal to the max value in M ($\max(M)$), and \odot is an element-wise multiplication operation. Additionally, the scaling factor $\frac{\text{sum}(M_{\max})}{\text{sum}(M_X)}$ normalizes the updated value assigned to $I'(x)$ based on the degree to which the input $I(x)$ is masked by fog.

The mask updating step of the masked convolution layer is expressed as:

$$M'(x) = \max(\text{avg}(M_X), M(x)) \quad (8)$$

where M_X is the mask corresponding to the convolutional window centered at x , $M'(x)$ is the new mask value, and $M(x)$ is the old mask value of mask M at x .

An walk through of each of the operations within the masked convolution layer is visually depicted in Fig. 4. At the top of the figure, inputs to the MConv layer are highlighted by a black box. From left to right the inputs include the 4x4 foggy image itself, hypothetical values I that correspond to the foggy image, a fog mask M that corresponds to the foggy image, convolution weights W , an M_{\max} that corresponds to M , and the bias b which corresponds to the convolution filter. In this figure, we walk through the calculation of one value $I'(x)$ in the updated image, and one value $M'(x)$ in the updated mask. For computing these values, the relevant values in I and M are I_X and M_X , which are highlighted in yellow and pink respectively. The second row of the figure computes $WT(I_X \odot M_X)$. The result is highlighted in orange. This value is used by Eq. 7 in the *Masked Convolution* section of the third row to compute $I'(x)$. In the third row of the figure, $I'(x)$ and $M'(x)$ are computed using Eq. 7 and Eq. 6 respectively. Additionally, an updated version of the inputted foggy image is presented as the resulting output of the MConv layer.

After several iterations of masked convolutions and mask updates, the mask will eventually consist solely of the maximum value from the initial M passed into the model. The reason for only updating the mask to the max value of M in the mask rather than to 1, the maximal value any element of M can possibly have, is that when updating individual elements in the foggy image based on their surrounding pixels and mask, a pixel should only be able to become as un-foggy as the most un-foggy pixel around it. It would not make sense for a pixel, which was defogged using its neighboring pixels as context, to become less foggy than the surrounding pixels. As a result of this behavior, the last MConv layer will receive an input image and a corresponding uniform mask, where each element's value is equivalent to the max value from the original mask passed into the beginning of the model. This last MConv layer learns to uniformly defog the remaining image based on the intensity of the fog denoted by the uniform fog mask passed into the layer.

3.3 Implementation

Dark channel method: The dark channel method outlined in Section 2.1 was implemented using functions from the scikit-image [25] and OpenCV [10] libraries. Regarding the variables w and $\Omega(x)$ in Eq.(4): A value of 0.95 was used for w , and a radius of 15 pixels was used for the size of $\Omega(x)$.

Masked convolution layer: The masked convolution layer is implemented with PyTorch [18]. It takes an input image, as well as a fog mask of the width and height. A fixed convolution layer with no bias and the same kernel size as the masked convolution operation is used to implement the mask update step. This fixed convolution layer's weights are all set to 0 except for the center element, which holds a value of 1.

Network architecture: We use a UNet-like architecture [20] for the UNCLOUD network, similar to the one used in [13]. The UNet is configured the same way as in [13]; however each convolutional layer in the UNet is replaced with an MConv layer.

Loss function: The loss function used by the network is equivalent to that of [13], which is a composition of two per-pixel losses, $\mathcal{L}_{\text{hole}}$ and $\mathcal{L}_{\text{valid}}$, a perceptual loss $\mathcal{L}_{\text{perceptual}}$ proposed by Gatys et al. [8], a style loss term, $\mathcal{L}_{\text{style}_{\text{out}}}$ and $\mathcal{L}_{\text{style}_{\text{comp}}}$, and a total variation loss \mathcal{L}_{tv} . The total loss [13] is expressed as:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{valid}} + 6\mathcal{L}_{\text{hole}} + 0.05\mathcal{L}_{\text{perceptual}} + 120(\mathcal{L}_{\text{style}_{\text{out}}} + \mathcal{L}_{\text{style}_{\text{comp}}}) + 0.1\mathcal{L}_{\text{tv}} \quad (9)$$

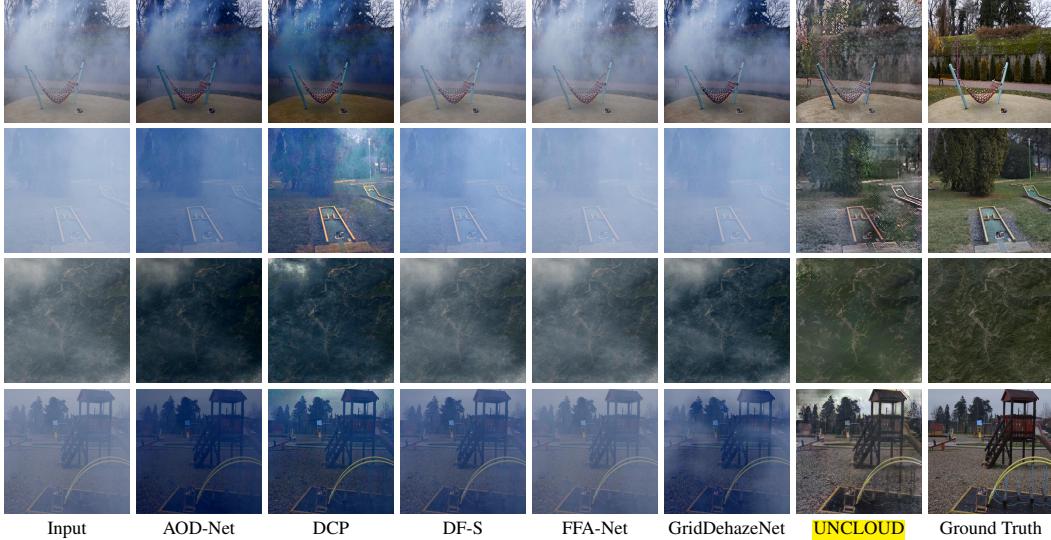


Figure 5: Qualitative comparison of various defogging methods. Top Row: Image from NH-HAZE dataset. Second Row: Image from DENSE-HAZE dataset. Third Row: Image from RS-HAZE dataset. Bottom Row: Image from O-HAZE dataset

4 Experiments

4.1 Datasets

The datasets used to train, and later evaluate the model, are as follows:

- **DENSE-HAZE [3] [4]**: a dataset comprising images of both indoor and outdoor scenes obscured by real haze. The haze in the images is generated by a professional haze machine. The images from this dataset are characterized as heavy uniform fog.
- **Foggy Cityscapes [21]**: a dataset comprising images from the Cityscapes dataset [7] layered with varying degrees of uniform fog. Three degrees of fogginess are generated for each cityscape photo: 600m, 300m, and 150m visibility. Compared to other datasets, the Foggy Cityscapes dataset can be categorized as generally uniform light to medium fog.
- **FRIDA [23]**: a dataset comprising images depicting a computer generated environment with varying levels of fog layered on top. A total of 4 different fog conditions were provided for each scene, ranging from medium to heavy fog. Additionally, foggy photos in the dataset include both uniform and non-uniform fog conditions.
- **FRIDA2 [24]**: an extension of the FRIDA dataset which comprises more images of other environments obscured by the same types of fog presented in FRIDA.
- **I-HAZE [1]**: a dataset comprising images of indoor scenes obscured by real haze generated by a professional haze machine. The fog in the images from this dataset are characterized as generally uniform and of medium intensity.
- **NH-HAZE [5, 6]**: a dataset comprising images of outdoor scenes obscured by real haze generated by a professional haze generator. The fog in the images is characterized as non-homogeneous, leaving parts of images completely obscured while other parts of images are nearly completely obscured.
- **O-HAZE [2]**: a dataset comprising images of outdoor scenes obscured by real haze generated by a professional haze machine. The fog in the images from this dataset are characterized as generally uniform and of medium intensity.
- **RS-HAZE [22]**: a dataset comprising aerial images of various terrains with synthetic fog, resembling clouds, layered over them. This dataset provides 6 different non-uniform fog intensities for each scene, ranging from very light to very heavy fog.

Table 1: Quantitative comparison of various defogging methods

Datasets	Metrics	Models					
		AOD-Net	DCP	DehazeFormer-S	FFA-Net	GridDehazeNet	UNCLOUD
DENSE-HAZE	MSE	5672.38	5246.69	9114.87	9378.37	9040.86	3490.18
	PSNR	10.95	11.43	8.71	8.82	8.85	13.64
	SSIM	0.41	0.44	0.39	0.39	0.39	0.47
Foggy Cityscapes	MSE	1597.33	1374.51	1422.37	2224.10	948.96	391.40
	PSNR	16.41	16.95	18.07	16.38	19.57	23.30
	SSIM	0.70	0.70	0.86	0.83	0.85	0.91
FRIDA	MSE	2984.69	3536.67	3250.27	4272.12	2565.19	988.19
	PSNR	13.70	12.73	13.28	12.16	14.21	18.22
	SSIM	0.76	0.67	0.78	0.76	0.76	0.79
FRIDA2	MSE	4811.90	4862.15	5068.84	6430.35	4760.07	2525.45
	PSNR	11.46	11.45	11.26	10.19	11.50	14.29
	SSIM	0.71	0.58	0.71	0.72	0.66	0.73
I-HAZE	MSE	1928.72	3341.57	1992.92	3467.84	2447.57	1550.85
	PSNR	15.70	13.07	15.60	12.78	14.58	16.25
	SSIM	0.70	0.65	0.74	0.66	0.70	0.72
NH-HAZE	MSE	3452.60	2932.75	4352.65	4438.24	3777.76	1877.34
	PSNR	12.93	13.64	11.99	11.92	12.51	15.65
	SSIM	0.37	0.45	0.41	0.40	0.43	0.60
O-HAZE	MSE	2002.58	1582.42	1012.50	1564.98	1481.03	620.52
	PSNR	15.16	16.63	18.33	16.62	16.47	20.83
	SSIM	0.62	0.69	0.71	0.66	0.70	0.83
RS-HAZE	MSE	2960.33	1129.68	6311.72	6695.18	3689.70	973.20
	PSNR	14.80	18.32	12.80	12.12	14.63	20.52
	SSIM	0.52	0.63	0.58	0.56	0.60	0.71

4.2 Training

When training the model, all eight datasets from Section 4.1 were used. Each dataset was split into a training set and evaluation set. Since some datasets are particularly small, such as the DENSE-HAZE dataset, which contains a total of 33 images, the training sets for all datasets were capped at 250 images in order to ensure that the model spends somewhat equal amounts of time training on each dataset. Thus the model was not trained on more than 250 images from each individual dataset.

The validation set for each dataset comprises between 5 and 50 images depending on the size of the dataset. For each dataset the ratio between the validation set size and the training set size is approximately 1:5.

The first step in the training process is to pre-compute a fog mask for each image using the method outlined in Section 3.1. The model is then trained on the training sets from each respective dataset. The images in the validation sets are not seen by the model during training. When running the model, foggy input images and their corresponding masks are resized to the dimensions of 512x512. The training is conducted on an NVIDIA Titan RTX GPU and a batch size of 8 is used.

4.3 Evaluation

UNCLOUD is qualitatively compared against several methods in Fig. 5 and quantitatively compared against each model in table 1, where the best results for each metric are **bolded**. For each method evaluated in both comparisons, a singular pre-trained model is used. The models used are as follows:

- **AOD-Net [12]**: Only one model provided. Choosing a model was not necessary.
- **DCP [9]**: Only one model provided. Choosing a model was not necessary.
- **DehazeFormer-S (DF-S) [22]**: Outdoor Pretrained Model.
- **FFA-Net [19]**: OTS (Outdoor Training Set) Pretrained Model.
- **GridDehazeNet [14]**: Outdoor Pretrained Model.

In general, when given the option, a model pretrained on outdoor training data was chosen because most datasets used to evaluate each model comprised outdoor images.

Each model is evaluated on the same validation set for each respective dataset as defined in Section 4.2. Specifically, similar to the image preparation used by UNCLOUD, images in each validation set are first transformed to a size of 512x512 before feeding them into each model. The mean squared error (MSE), peak signal to noise ratio (PSNR), and structural similarity (SSIM) are computed for each defogged image using functions from the scikit-image library [25]. Then, for each dataset, the MSE, PSNR, and SSIM metrics are respectively averaged across all images in each respective validation set to produce the metrics for each model in Table 1.

As depicted in Table 1, our method, UNCLOUD, outperforms all models it is compared against in all three metrics, with an exception for a single metric on the I-HAZE dataset. Evaluating the results, we conclude that the novel image inpainting approach to image defogging used by UNCLOUD is both effective and capable of surpassing current state of the art defogging methods.

5 Conclusion

In this paper, we proposed a simple yet effective universal image defogging method called UNCLOUD. UNCLOUD uses masked convolution layers to iteratively remove fog from foggy images. These masked convolution layers utilize non-binary fog masks to perform convolutions that are weighted and renormalized in proportion to the estimated intensity of fog at each pixel in an image. Experiments on benchmark images across eight datasets of varying fog conditions demonstrate that UNCLOUD outperforms existing state of the art image-based defogging methods.

References

- [1] Codruta O. Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. I-haze: a dehazing benchmark with real hazy and haze-free indoor images. In *arXiv:1804.05091v1*, 2018.
- [2] Codruta O. Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. In *IEEE Conference on Computer Vision and Pattern Recognition, NTIRE Workshop*, NTIRE CVPR’18, 2018.
- [3] Codruta O. Ancuti, Cosmin Ancuti, Mateu Sbert, and Radu Timofte. Dense haze: A benchmark for image dehazing with dense-haze and haze-free images. In *IEEE International Conference on Image Processing (ICIP)*, IEEE ICIP 2019, 2019.
- [4] Codruta O Ancuti, Cosmin Ancuti, Radu Timofte, Luc Van Gool, Lei Zhang, and Ming-Hsuan Yang. Ntire 2019 image dehazing challenge report. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE CVPR 2019, 2019.
- [5] Codruta O. Ancuti, Cosmin Ancuti, and Radu Timofte. NH-HAZE: an image dehazing benchmark with non-homogeneous hazy and haze-free images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE CVPR 2020, 2020.
- [6] Codruta O Ancuti, Cosmin Ancuti, Florin-Alexandru Vasluijanu, Radu Timofte, et al. NTIRE 2020 challenge on nonhomogeneous dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE CVPR 2020, 2020.
- [7] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [8] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015. URL <http://arxiv.org/abs/1508.06576>.
- [9] 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, 2011. doi: 10.1109/TPAMI.2010.168.
- [10] Itseez. Open source computer vision library. <https://github.com/itseez/opencv>, 2015.

- [11] Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008. doi: 10.1109/TPAMI.2007.1177.
- [12] Boyi Li, Xiulian Peng, Zhangyang Wang, Ji-Zheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.
- [13] Guilin Liu, Fitzsum A. Reda, Kevin J. Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions, 2018. URL <https://arxiv.org/abs/1804.07723>.
- [14] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Griddehazenet: Attention-based multi-scale network for image dehazing. *CoRR*, abs/1908.03245, 2019. URL <http://arxiv.org/abs/1908.03245>.
- [15] Earl J McCartney. Optics of the atmosphere: scattering by molecules and particles. *New York*, 1976.
- [16] Srinivasa G Narasimhan and Shree K Nayar. Vision and the atmosphere. *IJCV*, 48(3):233–254, 2002.
- [17] Shree K Nayar and Srinivasa G Narasimhan. Vision in bad weather. *ICCV*, 2:820–827, 1999.
- [18] 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. URL <http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf>.
- [19] Xu Qin, Zhilin Wang, Yuanchao Bai, Xiaodong Xie, and Huizhu Jia. Ffa-net: Feature fusion attention network for single image dehazing. *CoRR*, abs/1911.07559, 2019. URL <http://arxiv.org/abs/1911.07559>.
- [20] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL <http://arxiv.org/abs/1505.04597>.
- [21] Christos Sakaridis, Dengxin Dai, and Luc Van Gool. Semantic foggy scene understanding with synthetic data. *International Journal of Computer Vision*, 126(9):973–992, Sep 2018. URL <https://doi.org/10.1007/s11263-018-1072-8>.
- [22] Yuda Song, Zhuqing He, Hui Qian, and Xin Du. Vision transformers for single image dehazing, 2022. URL <https://arxiv.org/abs/2204.03883>.
- [23] J.-P. Tarel, N. Hautière, A. Cord, D. Gruyer, and H. Halmaoui. Improved visibility of road scene images under heterogeneous fog. In *Proceedings of IEEE Intelligent Vehicle Symposium (IV’2010)*, pages 478–485, San Diego, California, USA, 2010. <http://perso.lcpc.fr/tarel.jeanphilippe/publis/iv10.html>.
- [24] J.-P. Tarel, N. Hautière, L. Caraffa, A. Cord, H. Halmaoui, and D. Gruyer. Vision enhancement in homogeneous and heterogeneous fog. *IEEE Intelligent Transportation Systems Magazine*, 4(2):6–20, Summer 2012. <http://perso.lcpc.fr/tarel.jeanphilippe/publis/itsm12.html>.
- [25] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuel Gouillart, Tony Yu, and the scikit-image contributors. scikit-image: image processing in Python. *PeerJ*, 2:e453, 6 2014. ISSN 2167-8359. doi: 10.7717/peerj.453. URL <https://doi.org/10.7717/peerj.453>.