Non-Homogeneous Haze Removal Using Deep Neural Networks

by

Harsh Sharma
202111002

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

Dhirubhai Ambani Institute of Information and Communication Technology

May, 2023

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

Harsh Sharma

Certificate

This is to certify that the thesis work entitled **Non-Homogeneous Haze Removal Using Deep Neural Networks** has been carried out by **Harsh Sharma** for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

M Khare

Prof. Bakul Gohel
Thesis Supervisor

Prof. Manish Khare
Thesis Co-Supervisor

Acknowledgments

First of all I would like to acknowledge my thesis guide, **Prof. Bakul Gohel**, who was there the whole time guiding and supporting me towards the goal. His expertise, skill and knowledge have helped me immensely throughout the journey of the thesis. He was always there to solve any of my problem, be it technical or non-technical and helped me figuring out a way to solve it. Thanks a lot.

I want to thank my thesis co-guide, **Prof. Manish Khare**, for his valuable inputs and suggestions from time to time.

I would also like to thank my former thesis guide, **Prof. Ahlad Kumar**, for introducing me to this problem and igniting an interest for the same. His guidance and adept helped me getting started.

I am grateful to my panel members, **Prof. Manjunath Joshi**, **Prof. Puneet Bhateja**, and **Prof. Shruti Bhilare** for their extremely valuable feedback during my thesis stage I and II presentations.

I am grateful to my friends, **Krunal Botadara**, **Raghav Gorasiya**, **Utkarsh Pandaya**, and **Shrey Garg** who helped me solve problems which i faced during my thesis journey.

Contents

Abstract

Image Dehazing is a famous computer vision application that has been in the research area for the past decade. It involves reducing or removing haze/fog from an image to extract more information and make it more visually appealing overall. In this thesis, we first explain the existing methods to solve the ill-posed problem of image dehazing. We start with the prior-based techniques, which use image processing to dehaze the image after which we shift to learning-based methods, which have been recently developed and are considered state-of-the-art. Following this, we discuss the method proposed by us and it's results as compared to other existing state-of-the-art methods. We have proposed a two stage image dehazing model which utilizes two different deep learning models. The first model is a combination of different convolutional modules like haze detector module, Dark channel prior module, feature extraction module, spatial attention module, feature fusion module and restoration module. The other model is a GAN architecture, pix2pix GAN to be specific with different generator losses. We have obtained PSNR score of 18.11 and SSIM score of 0.6 on NH-HAZE dataset, while we have obtained a PSNR score of 13.79 and a SSIM score of 0.4320 on DenseHaze dataset. Along with these two datasets, we have also tested our model on RESIDE dataset which also gives comparable results.

These days cascading of models is quite popular to make a complex model which can solve the problem with good metrics as compared to a stand-alone model. We have also explored the validity of this statement by comparing the results of a cascaded model with ours. We explore when can a cascade model benefit the result while consuming extra computational power. Along with all these analysis, we have also experimented with different loss functions and observed that different datasets require different loss functions for better performance.

List of Principal Symbols and Acronyms

β	Extinction coefficient
∇	Representing gradient of an image
ω	Representing a patch of an image
ϕ	Representing output feature tensor of a certain convolutional block of VGG16 network for a given input

List of Tables

List of Figures

Chapter 1
Introduction

A clear and haze-free image is essential for computer vision tasks like object detection, image segmentation, image classification, etc. So, if we want to apply some advanced computer vision application to the image, we first need to make it as clear as possible. Simply enough, image dehazing is an image processing/computer vision application which reduces or removes the fog/haze from an input hazy image.

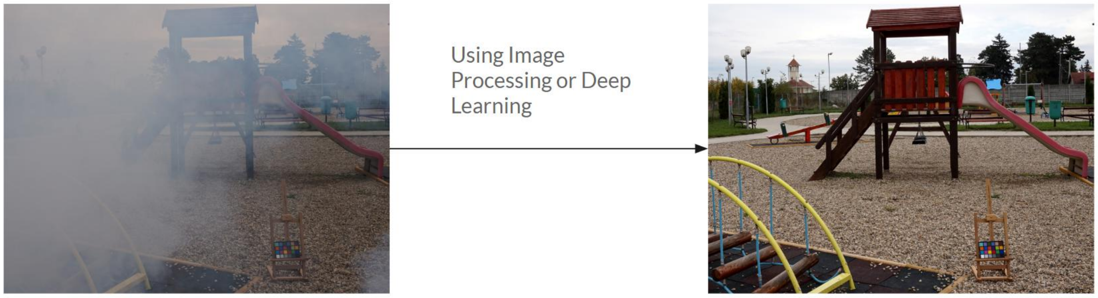


Figure 1.1: Image Dehazing [4]

As shown in Fig. 1.1, we can use traditional image processing techniques or deep learning techniques to remove fog from the image. Also, it's quite clear from Fig. 1.2 that if an object detection algorithm is given this image to detect chairs, it won't be able to detect the chairs which are hidden behind the haze with good accuracy. So, in order for the algorithm to work, we need to pre-process the image by removing the haze.

The irradiance received by the camera from the scene point is attenuated along the line of sight. Furthermore, the incoming light is mixed with the airlight (surrounding light which is reflected into the line of sight by small atmospheric particles). The resulting image loses contrast and color accuracy, as shown in Fig. 1.3. Since the amount of light scattering depends on the distances of the scene points from the camera, the degradation is spatial-variant and usually non-homogeneous.

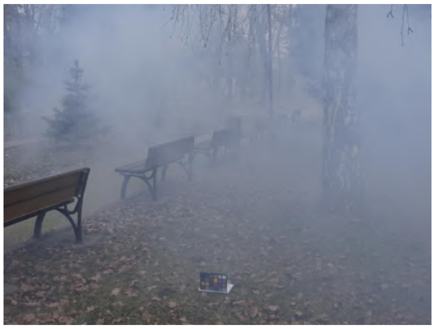


Figure 1.2: Foggy Image[4]

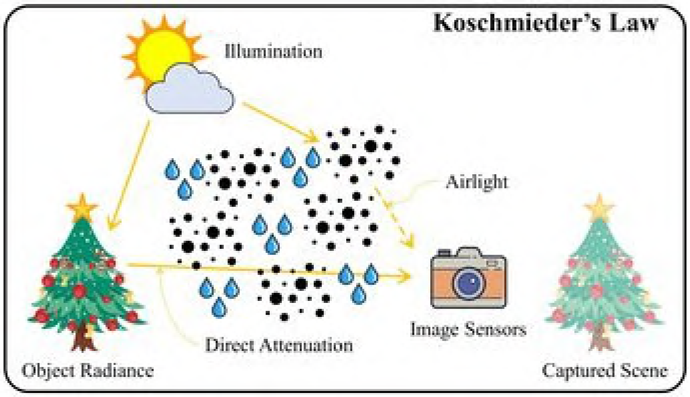


Figure 1.3: Atmospheric scattering model [2]

One of the most crucial tasks is formulating a mathematical model that gives a relationship between hazy and clear image pairs. One such mathematical model is the atmospheric scattering model or the widely known Koschmieder's law. In our scenario, this law can be defined as,

$$I(x) = J(x)t(x) + A(1 - t(x)) \tag{1.1}$$

Where *I(x)* represents the foggy/hazy image, and *J(x)* represents the clear image. x refers to the position of image pixels. A is the global airlight representing the surrounding/ambient light in the atmosphere. *t(x)* is a map that represents the transmission of the intrinsic luminance in the atmosphere. *t(x)* can be further modeled as,

$$t(x) = e^{-\beta d(x)} \tag{1.2}$$

where β is the extinction coefficient, and *d(x)* is the scene depth. Existing dehazing methods can be roughly classified into two categories, i.e., the prior-based and the learning-based. The prior-based methods rely heavily on the atmospheric scattering model to compute the dehazed image. But since there are more than two unknowns in the atmospheric scattering model (*J(x)*, *t(x)*, *A*), it is an ill-posed problem to solve. So the prior-based methods proposes various priors as extra constraints to find the dehazed image. On the other hand, learning-based methods learn to approximate *A* and *t(x)*. Combining *A*, *t(x)*, and *I(x)* using the atmospheric scattering model, they compute the dehazed image. Some learning-based methods directly compute the dehazed image from the hazy image using supervised learning.

1.1 Problem Statement

A homogeneous haze is a haze or fog present in uniform amounts throughout the image. i.e., it is of the same intensity everywhere in the image. While on the other side, non-homogeneous haze is a haze that is present in different amounts at different patches/regions in the image. It is needless to say that it is easier to dehaze a homogeneous hazy image than to dehaze a non-homogeneous hazy image since, in the homogeneous hazy image, we have to find that common haze intensity in the image.

Natural haze is non-homogeneous in nature, so a method which can not only remove haze from the hazy region but also doesn't alter the pixel values of the non-hazy region while maintaining the colour, texture, and luminance of the whole image is desired.

CHAPTER 2
Related Work

This chapter explores most of the current as well as previous methods which were used to dehaze an image. Some of the mentioned methods use advanced image processing, while others use deep neural networks to dehaze an image.

2.1 Dark Channel Prior

Dark channel prior [8] is perhaps one of the most known and earlier methods in image dehazing application. It revolves around the dark channel prior to remove haze from the image. The dark channel prior is based on the observation that most local patches in haze-free non-sky outdoor images contain some pixels which have very low intensity in at least one color channel. So considering the prior, the dark channel for an image *I(x)* is obtained by the below formula:

$$I^{dark}(x) = \min_{c \in \{r,g,b\}} (\min_{y \in \omega(x)} (I^c(y))) \tag{2.1}$$

Where I^c is a color channel of I and $\omega(x)$ is a local patch centered at x. The observation says that except for the sky region, the intensity of I^{dark} is low and tends to be zero if I is a haze-free outdoor image. I^{dark} is the dark channel of I. The transmission map can then be estimated by the below formula:

$$\tilde{t}(x) = 1 - \min_{c}(\min_{y \in \omega(x)} (\frac{I^c(y)}{A^c})) \tag{2.2}$$

The dark channel of the image can be used to estimate atmospheric light. We first pick the top 0.1% brightest pixels in the dark channel. Among these pixels, the pixels with the highest intensity in the input image *I(x)* are selected as the atmospheric light. The final dehaze image can be obtained by rearranging Eq. 1.1 as below,

$$J(x) = \frac{I(x) - A}{max\{t(x), t_0\}} + A \tag{2.3}$$

We restrict the transmission t(x) to a lower bound t0, meaning that a small amount of haze is preserved in dense haze regions. This is done to prevent the denominator from becoming zero. Along with the dehazed image, a high-quality depth map can also be obtained as a by-product of this method. Please refer to appendix A for full DCP proof.

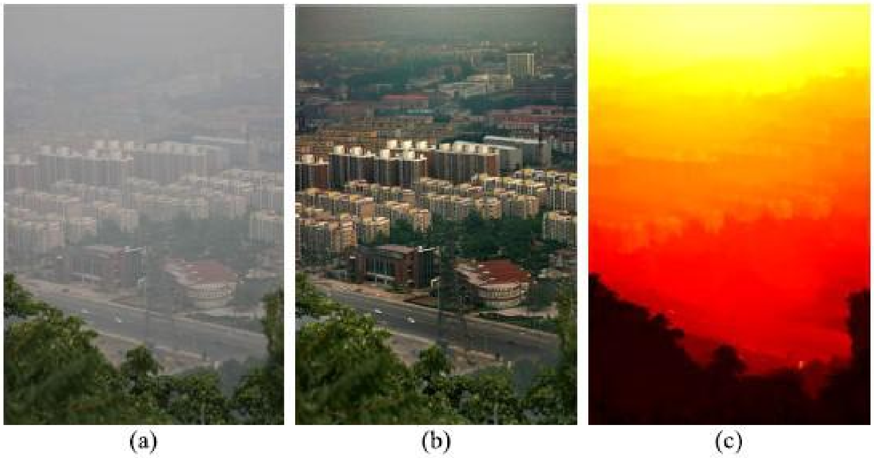


Figure 2.1: (a) Input haze image. (b) Image after haze removal using DCP. (c) Recovered depth map. [8]

2.2 Other Image Processing Techniques

In [11], the authors proposed an adaptive single-image dehazing method. The method first classifies various regions of the hazy input image as less affected, moderately affected, and most affected and subsequently dehazes according to the haze-affected regions. The hazy image, which is separated into three prominent hazy regions, is passed to three dehazing blocks. Each block decomposes the processed hazy image into base and detail layers by choosing different scale factors w.r.t different modules. Then in these modules, the image dehazing and detail enhancement are performed for base and detail layers. Finally, after image dehazing and detail enhancement, the recovered images of three blocks are fused based on the respective regions to obtain the final dehazed output. In [10], the authors proposed a new method which merges bright channel prior along with dark channel prior to boost the performance of dark channel prior and overcome it’s disadvantages.

2.3 Deep Neural Network Techniques

In [5], the authors proposed a deep neural network model which comprises of three sub neural network models. A deep pre-dehazer, feature extractor and an image restoring neural network. The hazy image first goes into the pre-dehazer network which is pre-trained on the dataset. The output of the pre-dehazer is a less hazy image. The output of the pre-dehazer and the input hazy image then goes into the feature extractor network in a parallel fashion. The feature extractor extract the image features which then gets fused by a custom progressive feature fusion technique. At last the fused features are then passed into the image restoring model which produces the final output dehaze image. The model architecture is given in fig. 2.2.

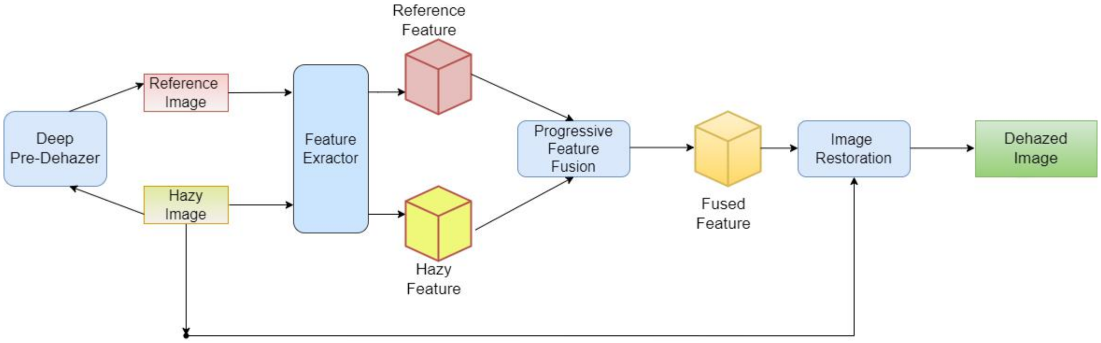


Figure 2.2: Image dehazing using Feature fusion technique

Pre-dehazer and Image restoration modules are modified U-NET Architectures. Feature extraction module is based on convolution layers. Progressive feature fusion module is also based on convolution layers, but the same module is used multiple times in order to progressively fuse the features.

The loss function is summation of three losses. First one is standard L1 loss between the ground truth and the output of the end model as well as the output of deep pre-dehazer. Second loss term is there to preserve the edges of the output dehaze image. This is done by minimizing the L1 distance between the ground truth gradients and the output gradients. Third loss term is to remove possible artifacts from the output image and this is done by minimizing the gradients of the modified transmission map. The authors tested the model on RESIDE and NTRIE2018 dehazing challenge datasets, both of which are homogeneous in nature. They also resized the NTRIE2018 dataset images to 512x512 to reduce computations. Hence the model is not made to handel non-homogeneous images.

In [17], the authors combined dark channel prior method with deep neural networks. In this method, the hazy image is first dehazed using dark channel prior method. After that it is fed into the deep neural network which is U-NET based. This neural network then outputs a refined dehazed image as well as a refined transmission map. A second refined dehaze image is obtained by using the refined transmission map. Both refined dehaze images are than fused to obtain best of both worlds and finally a much better dehaze image is obtained in the output. The U-NET models are trained with the help of a GAN which tries to generate the distribution of haze-free images by distinguishing haze-free images from hazy images. The discriminator is fed unpaired haze-free images half of the time and the output of the U-NET the other half of the time. Since the discriminator is trained to identify which image came from which dataset, the output of the U-NET shifts towards haze-free image and progressively the haze is reduced from the image as the model is trained. The model architecture is given in fig. 2.3.

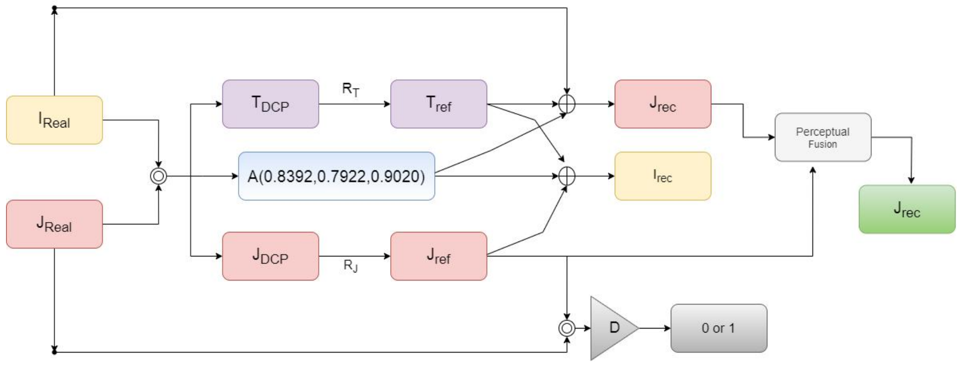


Figure 2.3: Image dehazing using RefineDNet technique

The loss function is a combination of three different losses. First term is the standard GAN loss. Second term is the reconstruction loss which is the L1 difference of hazy image and reconstructed hazy image, while the third loss is the identity loss which helps reduce image artifacts.

The model was trained on RESIDE dataset and tested on SOTS. The model was also trained on D-HAZY dataset. Both of the datasets are homogeneous in nature, hence the model is primarily made to handle homogeneous data. The authors didn't test the model on any non-homogeneous data, which is common occurrence in real-life scenarios.

In [13], the authors proposed a neural network architecture with encoder and decoder pairs for each atmosphere map, transmission map and a reconstructed image. They also included a weight map which learns the best weights to combine the transmission map and atmosphere map to produce a dehazed image.

In [16], the authors proposed an ensemble method for image dehazing. In total there are three models. The first two models are quite same as in [13], while the third model is a combination of encoder-decoder model and a UNET architecture. The final dehaze image is the one which is best obtained from all the three models.

In [14], the authors used one of the state-of-the-art model i.e., VisionTransformer for dehazing purpose. They used UNET architecture as a backbone and implemented vision transformer blocks instead of normal CNN layers. They also modified the normalization layers, activation functions and spatial information aggregation scheme in order to make it compatible for non-homogeneous data. Mostly the authors used homogeneous datasets, but they also used a remote sensing dataset which is non-homogeneous in nature to test their model. The model architecture is given in fig. 2.4.

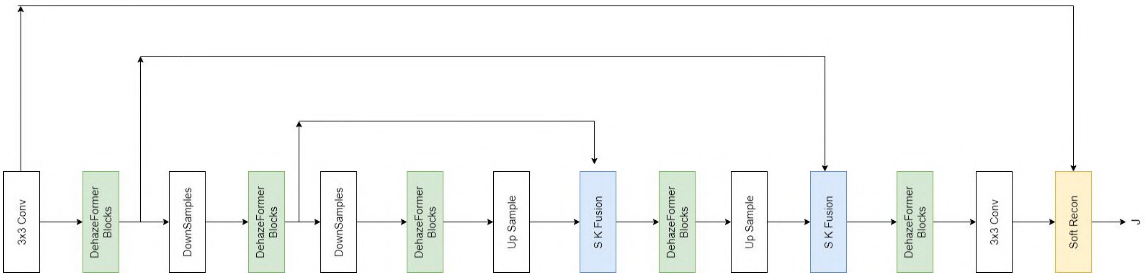


Figure 2.4: Image dehazing using DehazeFormer technique

CHAPTER 3
Proposed Method

After getting through so many methods, be it related with image processing or deep learning, it's clear that image dehazing task is highly complex since it requires the model to predict objects or scenes using nothing but the objects or scenes which are not fully covered with haze and are near the region to dehaze. To tackle this problem, we have built many different models, comprising of many different modules, combined differently. As for a base model, we have used the model which was proposed in [5].

Although the authors in [5] proposed the model for dealing with homogeneous haze, we will be modifying it to deal with non-homogeneous haze as well. Also during the process we will compare our modified author's model with the author's original model. This is done to test one of the author's claims that the use of pre-dehazer in the network boosts the network's performance immensely.

This will also test the idea of cascade deep neural networks which are popular these days. We will find out how much does the first network contribute to the performance of the latter network. We will also find out how does the first and second network perform when they are stand-alone. The full architecture of our model in given in Fig. 3.1.

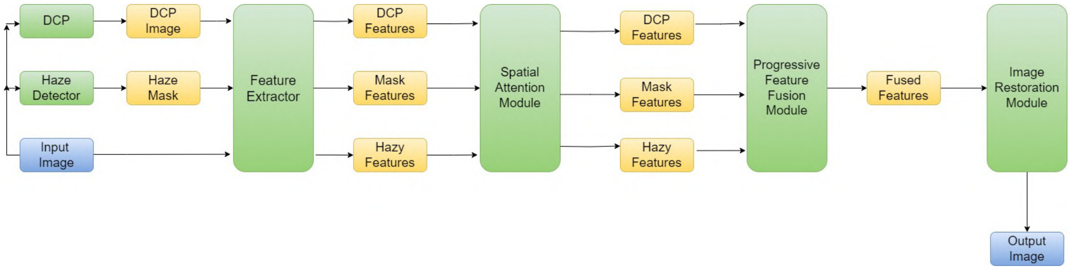


Figure 3.1: Full Architecture of proposed model

Although the full architecture is given in Fig. 3.1, we have experimented with different architectures by removing certain modules from the full architecture. Hence we have obtained many architectures by the combination of whether one module is present or not in the architecture. We then evaluate the model performance on different datasets to obtain the optimum architecture for a particular dataset.

The full architecture consists of many modules like, DCP module, haze detector module, feature extractor module, convolution block attention module, progressive feature fusion module and image restoration module. All of the above modules are explained in detail below.

3.1 DCP

The DCP module is responsible to produce the dark channel prior image of the input image. It utilizes the dark channel prior algorithm which was proposed in [8]. The obtained RGB image is then normalized to have values between -1 and 1.

3.2 Haze Detector

The haze detector module is a standard U-NET architecture which is trained on image-mask pairs for segmentation task. This module is only functional for non-homogeneous haze images since in non-homogeneous haze images we generally have a patch of image where the haze is the most dense. Hence if we can somehow detect which region has more haze and which region has less haze, we can remove the haze more effectively by only working on the hazy region and leaving the haze free region as it is.

This helps in preventing the haze free region pixels from getting mapped to some other pixels which have different colour or intensity. Hence only the hazy region gets affected after the introduction of haze mask. This haze mask is not applicable in homogeneous images, be it dense haze or light haze, since all of the image is filled with haze and there is no particular region of image containing the haze. As for the haze-mask pairs, we manually marked each of the 45 images of the NH-HAZE training dataset for regions having highest haze.

This manual marking of mask is kind of subjective from person to person,

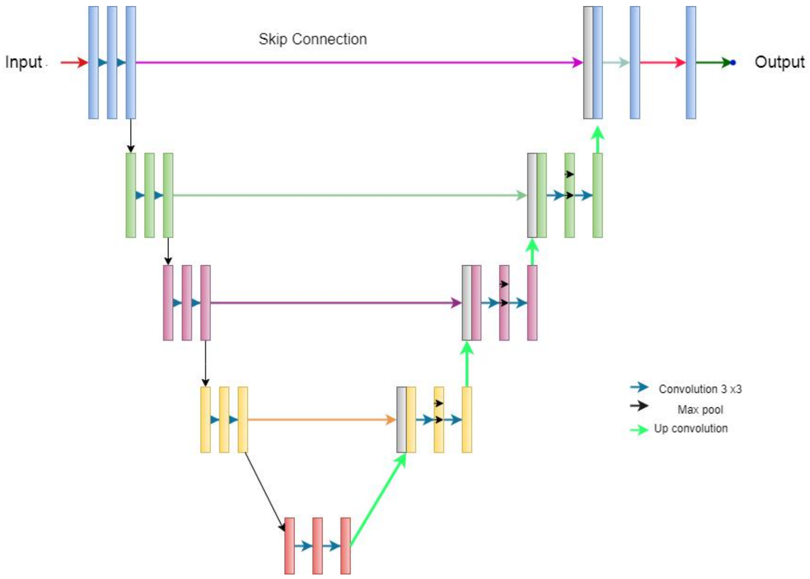


Figure 3.2: Standard U-NET Architecture

but little uncertainty is acceptable since there is no so called boundary between hazy region and non-hazy region, it is a gradient. The masks are binary masks containing a value of 1 if the haze is present and a value of -1 if the haze is not present at that pixel. We know that allowing only two values in our mask, i.e., whether the pixel contains haze or it doesn't is a bit restrictive and we can actually improve the bit resolution by increasing the number of values that the mask can take which can indicate not only if the pixel contains haze but also how much. This is one of our future tasks.

3.3 Attention Module

To further leverage the attention mechanism, we have included an attention module. The module is based on convolution block attention module (CBAM) proposed in [15]. CBAM improves the network's ability to focus on informative features while suppressing irrelevant or noisy signals. The CBAM module consists of two main components: the Channel Attention Module (CAM) and the Spatial Attention Module (SAM).

The Channel Attention Module (CAM) operates on the channel dimension of feature maps. It employs a global average pooling operation to aggregate global information from each channel and then uses a two-layer fully connected network to model the interdependencies among channels. The output of the fully connected network is used to compute channel-wise attention weights. These weights are then multiplied element-wise with the input feature maps to highlight important channels and suppress less relevant ones.

The Spatial Attention Module (SAM), on the other hand, focuses on capturing spatial dependencies within feature maps. It utilizes both the max pooling and average pooling operations to capture the maximum and average activations along spatial dimensions. These pooled representations are then fed into a convolutional layer followed by a sigmoid activation function. The resulting attention map is then multiplied element-wise with the input feature maps to emphasize relevant spatial locations while suppressing others. The brief architecture of CBAM is given in Fig. 3.3

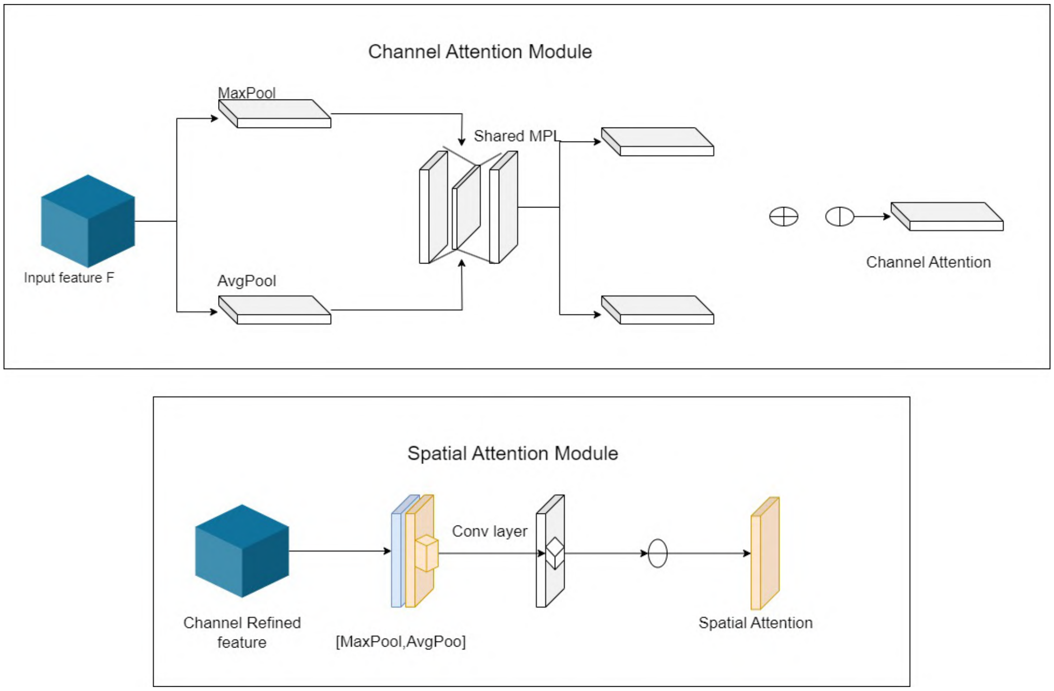


Figure 3.3: Convolution Block Attention Module Architecture

3.4 Feature Extractor Module

The feature extractor module consists of a convolution layer with relu activation and a residual block containing two convolution layers with relu activation. The primary task of this module is to convert the 3 channel input into a 32 channel feature tensor.

This feature extractor module is applied in parallel to all the three inputs, (if full architecture is used) i.e., hazy input image, output of DCP module and haze detector mask. These extracted features are then further refined by passing through the convolution block attention module in parallel which we had discussed earlier.

3.5 Feature Fusion Module

The feature fusion module takes the features from CBAM module and fuses them using convolutional layers. We can have multiple stages of fusion which is a hyper-parameter to be chosen by the user. In our case we have taken it as 2.

3.6 Image Restoration Module

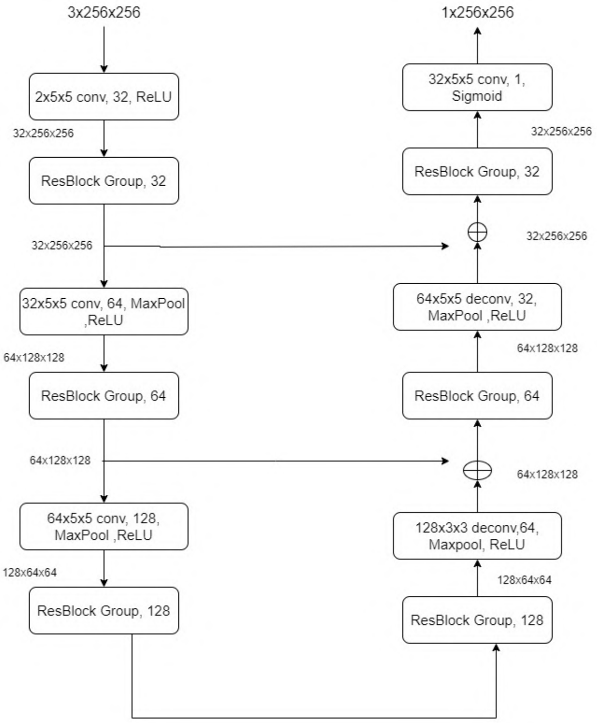


Figure 3.4: Image Restoration Architecture

It is an encoder-decoder architecture where each encoder and decoder contains a convolutional layer or a convolution transpose layer, and a ResBlock Group. Also, the output of each encoder is added to the input of the corresponding decoder as a skip connection. Each convolutional layer and convolution transpose layer, it is followed by a ReLU activation function except the last convolutional layer which is followed by a Sigmoid function to output the transmission map.

Each ResBlock Group consists of three residual blocks (ResBlock), and each ResBlock contains two convolutional layers with a ReLU function in the middle, and its inputs are added to the outputs as the residual connections.

3.7 Loss Functions

We have used the same loss functions which were used in [5]. Only difference is that we didn't use the smooth loss which was used in [5]. This is because we found that our models worked better without the inclusion of smooth loss. Other than that, we have used the below losses:

The Standard L1 loss, which is given as,

$$\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{3.1}$$

The Gradient loss, which is given as,

$$\sum_{i=1}^{n} |\nabla y_i - \nabla \hat{y}_i| \tag{3.2}$$

where y_i is the ground truth clear image, $\hat{y}_i$ is the model predicted image, and ∇ is the gradient operation on an image.

Along with combining these two losses and training the model, we also explored the possibilities of improving the model by changing the loss function entirely. To scout this, we trained all of the models on SSIM loss. This is a loss which emphasises the model to output an image which has better SSIM score. We found that this loss is beneficial and better as compared to the previous L1 + Gradient loss because it not only enhances the SSIM score of the model, but also enhances the PSNR score of the model. So in a way our finding is that the SSIM loss does not penalize the PSNR score to elevate the SSIM score.

The SSIM loss, which is given as,

$$\sum_{i=1}^{n} |1 - SSIM(y_i, \hat{y}_i)| \tag{3.3}$$

Where SSIM(y_i, $\hat{y}_i$) computes the SSIM value between y_i and $\hat{y}_i$.

3.8 Pix2pix GAN

Although most of the models derived from the architecture shown in the Fig 3.1 quantitatively beats the model proposed in [5], it still has an issue which many of the deep learning models face for dehazing. The problem is, although the haze is highly removed from the image, but the output image gets a little blue shifted. i.e., the output is not able to maintain it's colour information for both the hazy regions and for the regions with less haze.

To prevent this from happening and to obtain an image which not only contains less haze but also has preserved it's original colour information, we trained a pix2pix GAN to the output images from the first model. Again, we have experimented with many combinations of inputs with pix2pix GAN to obtain the best input which can produce the highest PSNR and SSIM scores. The architectural concept of Pix2pix GAN is shown in Fig. 3.5

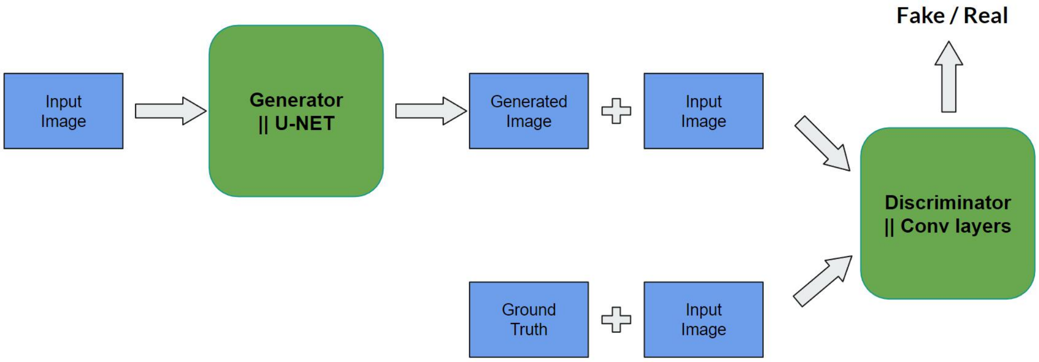


Figure 3.5: Pix2pix GAN Architecture

The pix2pix GAN consists of a generator and a discriminator. The generator is a U-NET inspired architecture which takes in hazy input images and tries to output a clear image. There are three losses dedicated to train the generator, the L1 loss, the generator adversarial loss and the perceptual loss.

The discriminator is a CNN architecture but the catch here is instead of having an output layer of one neuron to predict whether the image came from generated distribution or from clear distribution, the discriminator produces an image of a certain dimensions based on the resolution of input image. Each value in the output of discriminator image corresponds to whether that particular patch came from generated distribution or clear distribution. Hence the discriminator in pix2pix GAN discriminated patch-wise instead of a whole image. There is only one loss to train the discriminator which is the discriminator adversarial loss. The discriminator takes a concatenated input of two images, either hazy image with clear image or hazy image with generated image.

3.9 Loss Functions for Pix2pix GAN

We have used primarily used three loss functions to train the GAN network. These have been listed below with their mathematical equations:

The generator adversarial loss is given as,

$$\sum_{i=1}^{n} -\log(D(G(x_i))) \tag{3.4}$$

The discriminator adversarial loss is given as,

$$\sum_{i=1}^{n} -\log(D(x_i)) - \log(1 - D(G(x_i))) \tag{3.5}$$

The perceptual loss is given as,

$$\sum_{i=1}^{n} (\phi(y_i) - \phi(\hat{y}_i))^2 \tag{3.6}$$

where x_i is input hazy image, y_i is the ground truth clear image, $\hat{y}_i$ is the model predicted image, and ϕ is the output feature tensor of a certain convolutional block of VGG16 network.

Along with these two equations, we have also incorporated the L1 loss in the generator network, given in Eq. 3.1. This helps the network to preserve the colour and intensity when combined with the adversarial loss. It is to be noted that just like in our first model, we tried to replace the L1 loss and the perceptual loss with SSIM loss in hope for better SSIM and PSNR score, but it seems like SSIM score is too unstable with GAN and hence our generator kept on diverging and we were not able to generate a good quality haze free image using only SSIM loss and Adversarial loss.

3.10 Proposed Architecture

The proposed architecture is a cascaded deep neural network of two architectures shown in Fig. 3.1 (CNN network) and in Fig. 3.5 (GAN network). i.e., first the model in Fig. 3.1 (CNN network) is trained on the training set and then it is evaluated on both training and testing set. Now in order to train the model in Fig. 3.5 (GAN network), we use the outputs that we generated from the architecture in Fig. 3.1 (CNN network) for training images and train the model. Finally, after the model in Fig. 3.5 (GAN network) is trained, we feed the outputs that we generated from the architecture in Fig. 3.1 (CNN network) for testing images and obtain the dehazed image for testing data.

So it's a two step process for any new image to dehaze. First it goes into the model in Fig. 3.1 (CNN network) and then the result goes into the model in Fig. 3.5 (GAN network) which is our final result of dehazing.

We have proposed this cascaded architecture because both of the models serve different purposes, and hence gives us the best of both worlds. The CNN model helps in dehazing while the GAN network restores the colour and luminosity information back into the image.

CHAPTER 4

Datasets

We have used recent datasets in the dehazing space to evaluate our model performance. We have picked one dataset which is having non-homogeneous haze images, one dataset which contains dense homogeneous haze images with variable haze intensity, and lastly one dataset which contains homogeneous haze images with uniform light haze intensity. All the datasets and their meta-data is explained in the below sections.

4.1 NH-HAZE 2020

NH-HAZE 2020 [4] was released during IEEE CVPR NTRIE workshop, 2020. NTRIE stands for "New Trends in Image Restoration and Enhancement", which is an annual workshop related to image restoration and enhancement methods. The dataset contains high-resolution non-homogeneous haze-clear image pairs. The meta-data of the dataset is given below:

- Place where images were shot: Outdoor
- Haze construction: Using industrial haze machine
- Image resolution: 1200x1600 pixels
- Number of pair of haze-clear images: 55
- Official train-validation-test split for evaluation: Images numbered 1 to 45 are for training, images numbered 46 to 50 are for validation, and images numbered 51 to 55 are for testing

Some of the images from NH-HAZE dataset is given below in Fig. 4.1, Fig. 4.2, and Fig. 4.3.

4.2 DenseHaze 2019

DenseHaze 2019 [3] was released during IEEE International Conference on Image Processing, 2019. The dataset contains high-resolution variable intensity haze homogeneous haze-clear image pairs. The meta-data of the dataset is given below:

- Place where images were shot: Outdoor + indoor
- Haze construction: Using industrial haze machine
- Image resolution: 1200x1600 pixels
- Number of pair of haze-clear images: 55
- Official train-validation-test split for evaluation: Images numbered 1 to 45 are for training, images numbered 46 to 50 are for validation, and images numbered 51 to 55 are for testing

Some of the images from DenseHaze dataset is given below in Fig. 4.4, Fig. 4.5, and Fig. 4.6.

4.3 RESIDE INDOOR Test

RESIDE [12] stands for "REalistic Single Image Dehazing" which was released in IEEE Transactions on Image Processing, 2019. The dataset contains standard-resolution uniform intensity homogeneous haze-clear image pairs. The RESIDE dataset consists of two sub-divisions, RESIDE Outdoor containing outdoor images and RESIDE Indoor containing indoor images. Further, both Outdoor and Indoor datasets contains training and testing images.

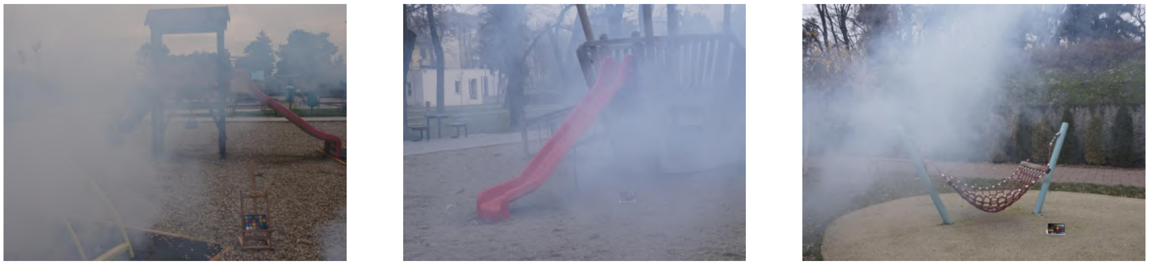
RESIDE Indoor dataset contains 13990 training images and 500 testing images. It is to be worth noting that each image is repeated 10 times in both training and testing directories. That is, each image is mapped to ten images with slightly different haze intensity in each image. RESIDE Outdoor dataset contains 313950 training images and 500 testing images. SOTS(Synthetic objective testing set) is the combination of testing images of both RESIDE Indoor and RESIDE Outdoor.

Since number of training examples in both RESIDE Indoor and Outdoor is huge, and we need to compute DCP of each image in our model, we used RESIDE Indoor Test (Same as SOTS Indoor) as a whole dataset and made a 405:45:50

train:validation:test split to evaluate our model. The meta-data of the dataset which we used is given below:

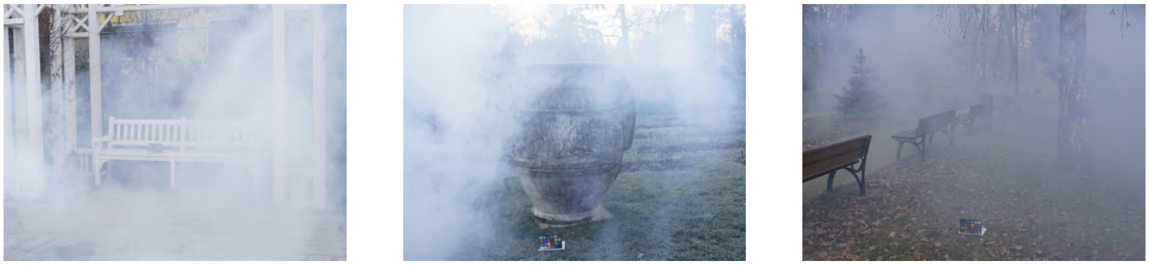
- Place where images were shot: Indoor
- Haze construction: Using software
- Image resolution: 460x620 pixels
- Number of pair of haze-clear images: 500
- Train-test split for evaluation: We randomly split the dataset in the ratio 405:45:50 train:validation:test.

Some of the images from RESIDE Indoor Test dataset is given below in Fig. 4.7, and Fig. 4.8.



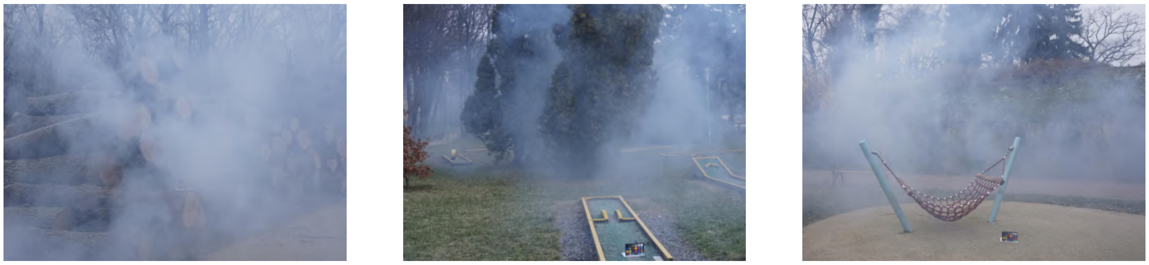
(a) Image number 7 (b) Image number 17 (c) Image number 21

Figure 4.1: Some of the training images in NH-HAZE dataset



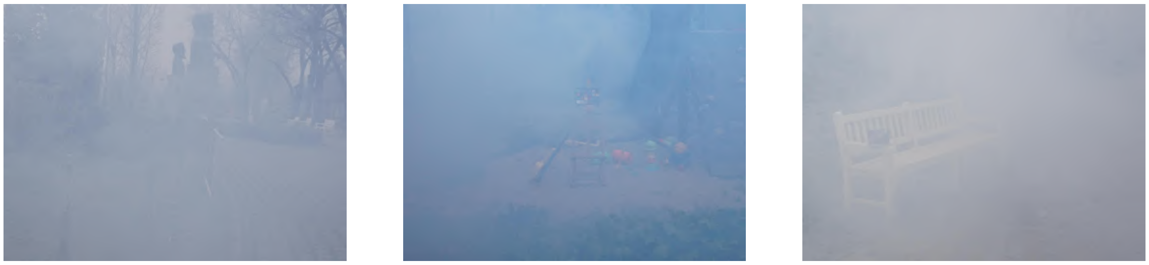
(a) Image number 46 (b) Image number 47 (c) Image number 48

Figure 4.2: Some of the validation images in NH-HAZE dataset



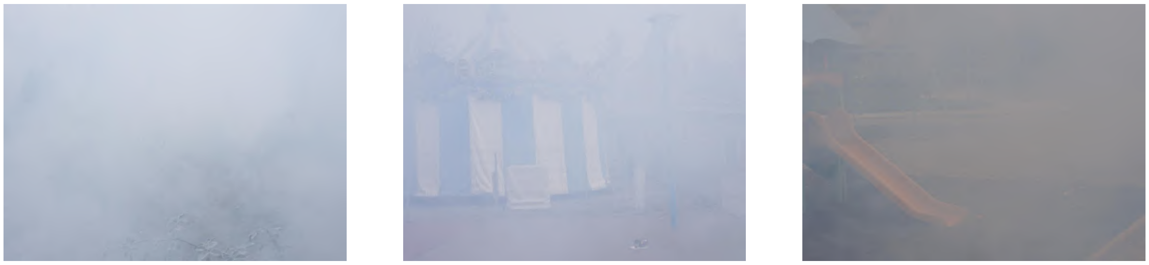
(a) Image number 51 (b) Image number 52 (c) Image number 53

Figure 4.3: Some of the testing images in NH-HAZE dataset



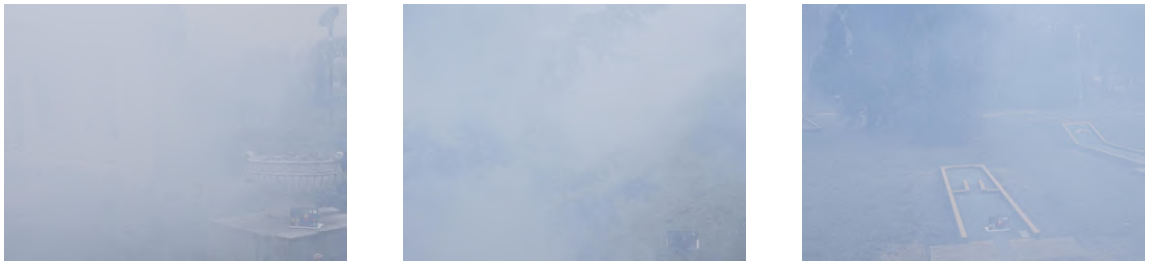
(a) Image number 7 (b) Image number 17 (c) Image number 21

Figure 4.4: Some of the training images in DenseHaze dataset



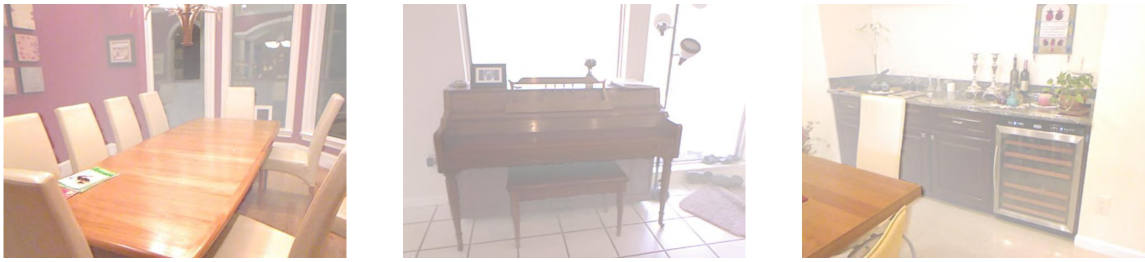
(a) Image number 46 (b) Image number 47 (c) Image number 48

Figure 4.5: Some of the validation images in DenseHaze dataset



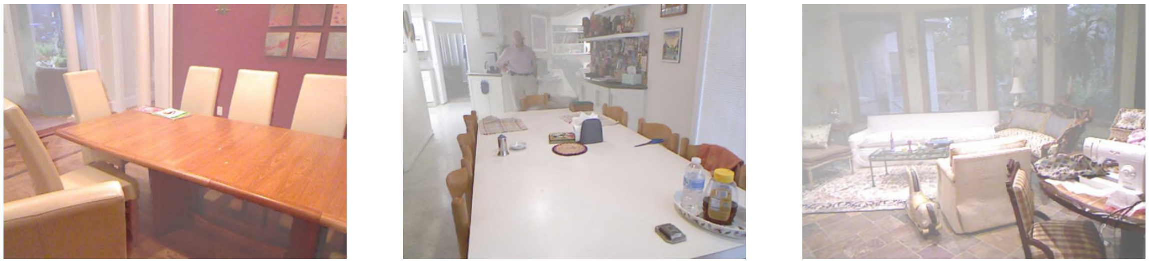
(a) Image number 51 (b) Image number 52 (c) Image number 53

Figure 4.6: Some of the testing images in DenseHaze dataset



(a) Image number 10 (b) Image number 120 (c) Image number 360

Figure 4.7: Some of the training images in SOTS Indoor dataset



(a) Image number 20 (b) Image number 170 (c) Image number 260

Figure 4.8: Some of the testing images in SOTS Indoor dataset

Chapter 5

Experiments

Here we have first discussed the acronyms which we have used to represent the experiments after which we have given all the hyper-parameter settings for each dataset.

Table 5.1: Acronym explaination for experiments

Short Forms	Full Form with explaination
PDH	Pre-Dehazer network which was used by authors in [5]
AU	PFF network which was proposed by authors in [5]
SA	CBAM network
MAS	Haze Detector network
DCP	Dark Channel Prior method

If for example in the experiment table it's mentioned "AU", then it means that the model used is the same as proposed by authors in [5]. If for example it's "AU+SA", then it means that it's model proposed by authors in [5] with integrated CBAM network. Similarly "AU-PDH" means it's the network proposed in [5] without the pre-dehazer network which was used in [5]. Hence it's clear that our proposed model which is in Fig. 3.1, is given by the acronym "AU-PDH+DCP+SA+MAS".

5.1 Hyper-parameters for NH-HAZE and DenseHaze

- Training image size: Randomly cropped 600x800 pixel images for CNN network and randomly cropped 1024x1024 pixel images for GAN models
- Testing image size: Full resolution 1200x1600 pixel images
- Learning rate: 1e-4 for CNN models and 2e-4 for GAN models
- batch size: 2 for CNN models and 3 for GAN models

- Number of epochs: 100 for CNN models and 1000 for GAN models
- CNN model loss coefficients:
  - L1 loss coefficient: 1
  - Gradient loss coefficient: 1
- GAN model loss coefficients:
  - L1 loss coefficient: 100
  - perceptual loss coefficient: 10
  - adversarial loss coefficient: 1

5.2 Hyper-parameters for RESIDE Indoor Test

- Training image size: 460x620 pixel images
- Testing image size: 460x620 pixel images
- Learning rate: 1e-4 for CNN models and 2e-4 for GAN models
- batch size: 2 for CNN models and 12 for GAN models
- Number of epochs: 25 for CNN models and 1000 for GAN models

Chapter 6

Results

The results obtained from all of the different experiments that we conducted is available in this chapter. In order to quantitatively evaluate output image, we have used two metrices, first one is PSNR (Peak Signal to Noise Ratio) and the other one is SSIM (Structure Similarity Index Metric).

6.1 On NH-HAZE Dataset

6.1.1 CNN Network (First Network)

Table 6.1: Results of different first stage models for NH-HAZE dataset

	L1 + Gradient Loss		**SSIM Loss**	
Experiments:	*PSNR*	*SSIM*	*PSNR*	*SSIM*
Hazy Image	11.31	0.418	11.31	0.418
Only DCP (no NN)	12.81	0.438	12.81	0.438
PDH	15.52	0.6039	15.64	0.62
AU	15.26	0.5919	16.01	0.624
AU-PDH	15.59	0.6039	15.79	0.622
AU+SA	15.36	0.604	15.94	0.62
AU-PDH+SA	15.47	0.6020	15.70	0.618
AU-PDH+DCP	15.53	0.6060	16.20	0.624
AU-PDH+MAS	15.11	0.5879	15.70	0.614
AU-PDH+DCP+SA	15.34	0.592	15.93	0.62
AU-PDH+SA+MAS	15.36	0.592	15.48	0.6060
AU-PDH+DCP+MAS	15.80	0.6	16.22	0.62
AU-PDH+DCP+SA+MAS	15.63	0.59	15.89	0.62

"Hazy Image" in the Table 6.1 contains the PSNR and SSIM score of unprocessed raw hazy image inputs as compared with ground truths. We can see from Table 6.1 that for L1 + Gradient Loss, best PSNR score is obtained by "AU-PDH+DCP+MAS" model, while best SSIM score is obtained by "AU-PDH+DCP" model.

It is also apparent that both "PDH" and "AU-PDH" has better PSNR and SSIM scores as compared to "AU", which means we can obtain better results without cascading two networks as done in "AU" and also save on computation time. This proves that the pre-dehazer used in [5] actually works better or at par with the complete end-2-end model proposed by the authors.

For SSIM Loss, the best PSNR score is again obtained by "AU-PDH+DCP+MAS", while the best SSIM score is obtained by "AU" and "AU-PDH+DCP" with "AU-PDH+DCP+MAS" and other similar models just slightly behind. This proves that the addition of DCP, haze detector and attention is actually better as compare to the pre-dehazer used in [5]. Also it can be noticed that whatever be the model, both PSNR and SSIM has improved when SSIM loss is used. Hence selection of suitable loss also impacts the performance metrics.

6.1.2 Pix2pix GAN Network (Second Network)

Table 6.2: Results of different second stage models for NH-HAZE dataset

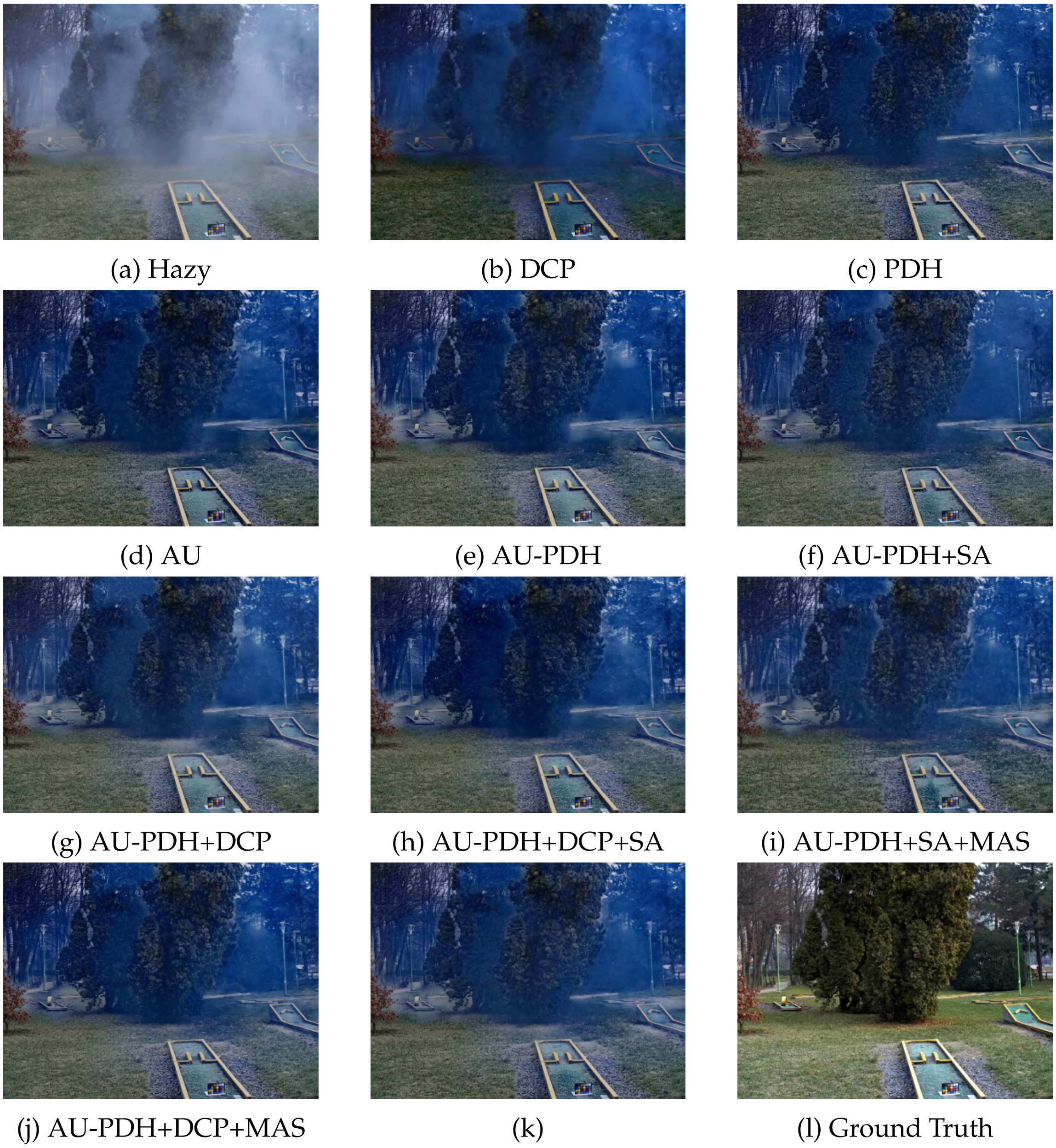
	L1 + VGG + GAN Loss	
Experiments:	*PSNR*	*SSIM*
DCP ->pix2pix	17.82	0.6100
PDH ->pix2pix	17.86	0.6039
AU ->pix2pix	17.72	0.6
AU-PDH+DCP ->pix2pix	18.19	0.6040
AU-PDH+DCP+SA ->pix2pix	17.97	0.6040
AU-PDH+DCP+SA+MAS ->pix2pix	18.11	0.5980
CONCAT(DCP, PDH) ->pix2pix	17.91	0.618
CONCAT(DCP, AU) ->pix2pix	18.45	0.608
CONCAT(DCP, AU-PDH+DCP) ->pix2pix	18.0	0.596
CONCAT(DCP, AU-PDH+DCP+SA) ->pix2pix	18.28	0.6
CONCAT(DCP, AU-PDH+DCP+SA+MAS) ->pix2pix	18.11	0.6

Here "->" means that the output obtained from the model present on left of "->" is given as input to pix2pix GAN for training. For example, "PDH -> pix2pix" means that after training pre-dehazer model, whatever the outputs were obtained were given as an input to train the pix2pix GAN. Similarly for others. Also, "CONCAT" denotes concatenation. For example, "CONCAT(DCP, AU) -> pix2pix" means after training "AU" model whatever the outputs were obtained were first concatinated with their corresponding DCP outputs and then were given as an input to train the pix2pix GAN.

In this case, the best SSIM is obtained by "CONCAT(DCP, AU) -> pix2pix" with "CONCAT(DCP, AU-PDH+DCP+SA) ->pix2pix" just slightly behind, while the best PSNR is obtained by "CONCAT(DCP, PDH) ->pix2pix". Here also it is obvious that "PDH" alone beats "AU" in both PSNR and SSIM scores. Hence there is no need of additional computation and model cascading as proposed in [5].

Also, note that the SSIM scores remain the same even after the GAN model, but the PSNR scores have improved a lot. This is related with the GAN making sure that the colours and intensity of pixels don't deviate much from the clear image.

Qualitative results of various models with L1+Gradient loss is shown in Fig. 6.1 and Fig. 6.2. Qualitative results of various models with ssim loss is shown in Fig. 6.3 and Fig. 6.4. Qualitative results of various models with pix2pix as a second cascade network is shown in Fig. 6.5 and Fig. 6.6.



(a) Hazy

(b) DCP

(c) PDH

(d) AU

(e) AU-PDH

(f) AU-PDH+SA

(g) AU-PDH+DCP

(h) AU-PDH+DCP+SA

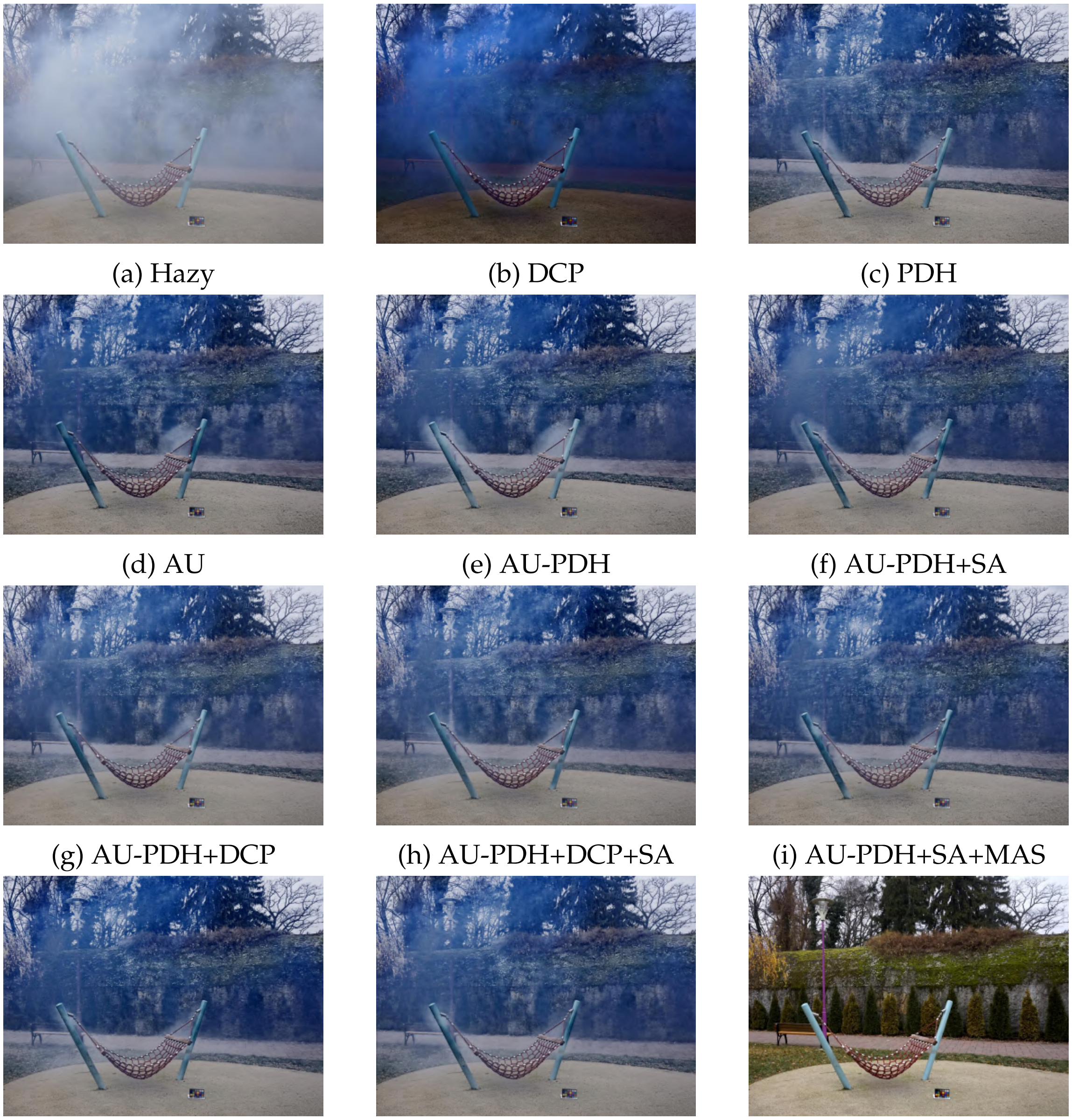
(i) AU-PDH+SA+MAS

(j) AU-PDH+DCP+MAS

(k)

(l) Ground Truth

Figure 6.1: Visual results of various models on image number 52 of NH-HAZE dataset for L1 + Gradient loss. (k): AU-PDH+DCP+SA+MAS.



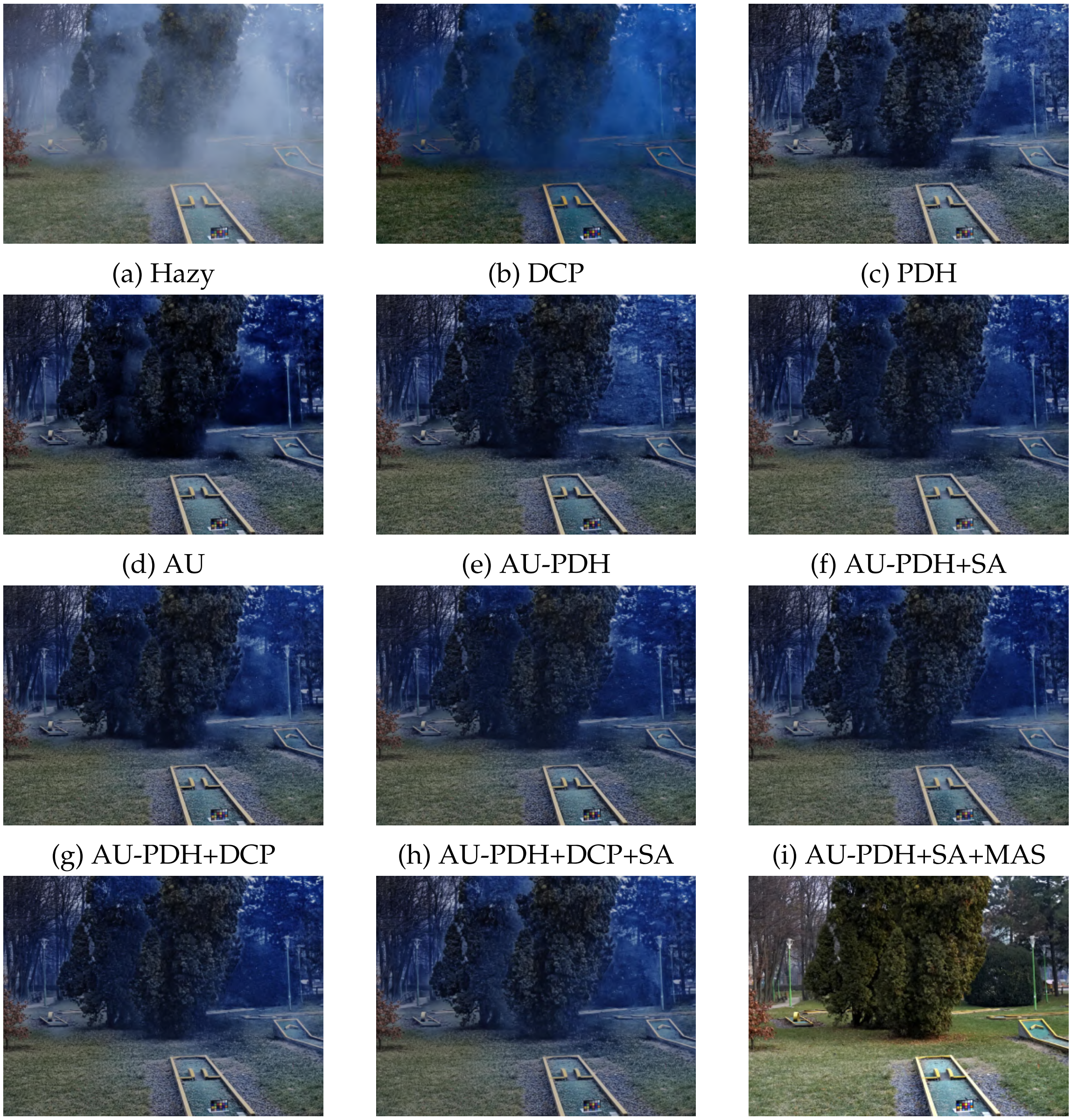
(a) Hazy (b) DCP (c) PDH

(d) AU (e) AU-PDH (f) AU-PDH+SA

(g) AU-PDH+DCP (h) AU-PDH+DCP+SA (i) AU-PDH+SA+MAS

(j) AU-PDH+DCP+MAS (k) (l) Ground Truth

Figure 6.2: Visual results of various models on image number 53 of NH-HAZE dataset for L1 + Gradient loss. (k): AU-PDH+DCP+SA+MAS.



(a) Hazy (b) DCP (c) PDH

(d) AU (e) AU-PDH (f) AU-PDH+SA

(g) AU-PDH+DCP (h) AU-PDH+DCP+SA (i) AU-PDH+SA+MAS

(j) AU-PDH+DCP+MAS (k) (l) Ground Truth

Figure 6.3: Visual results of various models on image number 52 of NH-HAZE dataset for SSIM loss. (k): AU-PDH+DCP+SA+MAS.

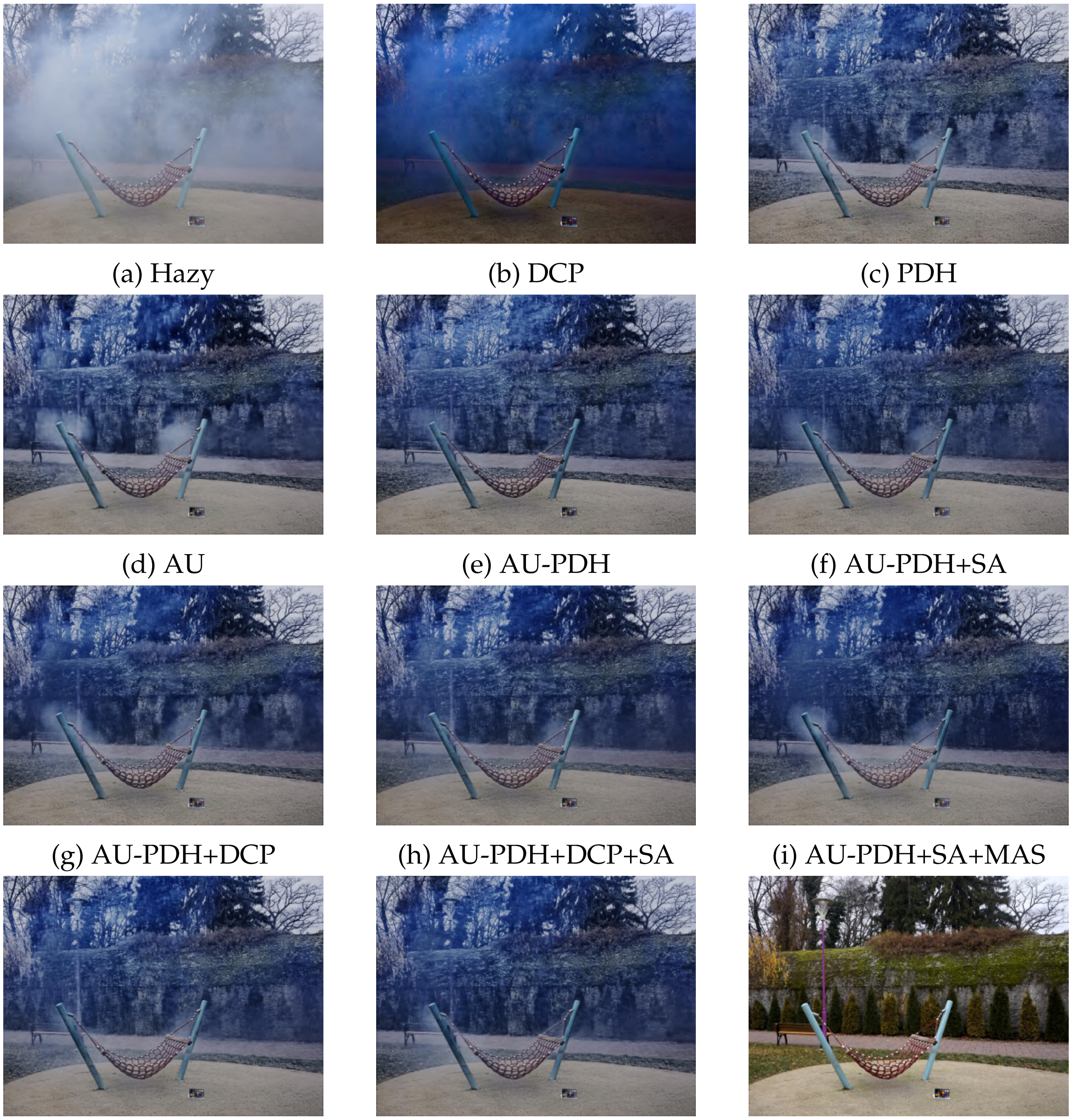
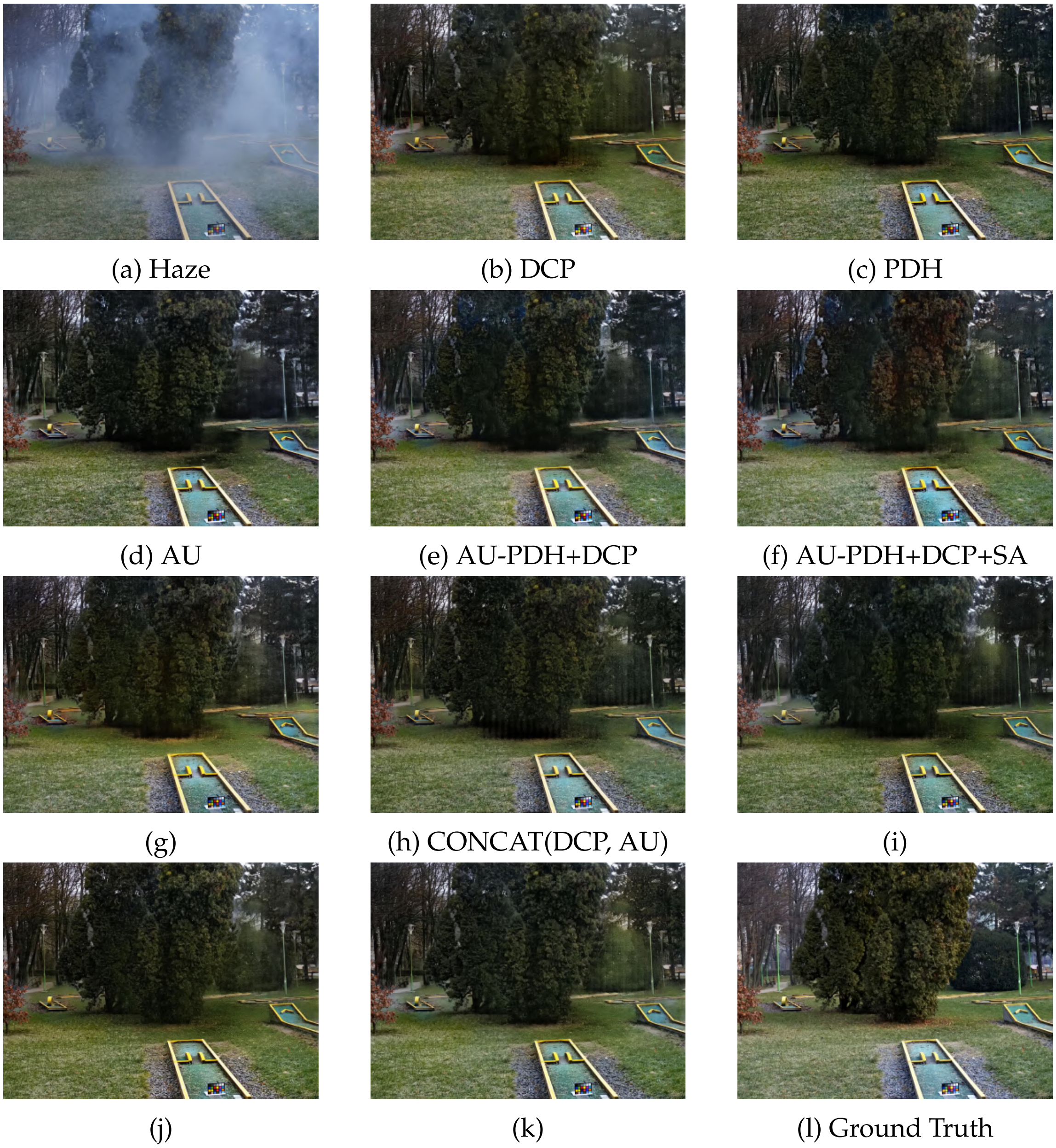

(a) Hazy
(b) DCP
(c) PDH
(d) AU
(e) AU-PDH
(f) AU-PDH+SA
(g) AU-PDH+DCP
(h) AU-PDH+DCP+SA
(i) AU-PDH+SA+MAS
(j) AU-PDH+DCP+MAS
(k)
(l) Ground Truth

Figure 6.4: Visual results of various models on image number 53 of NH-HAZE dataset for SSIM loss. (k): AU-PDH+DCP+SA+MAS.



(a) Haze

(b) DCP

(c) PDH

(d) AU

(e) AU-PDH+DCP

(f) AU-PDH+DCP+SA

(g)

(h) CONCAT(DCP, AU)

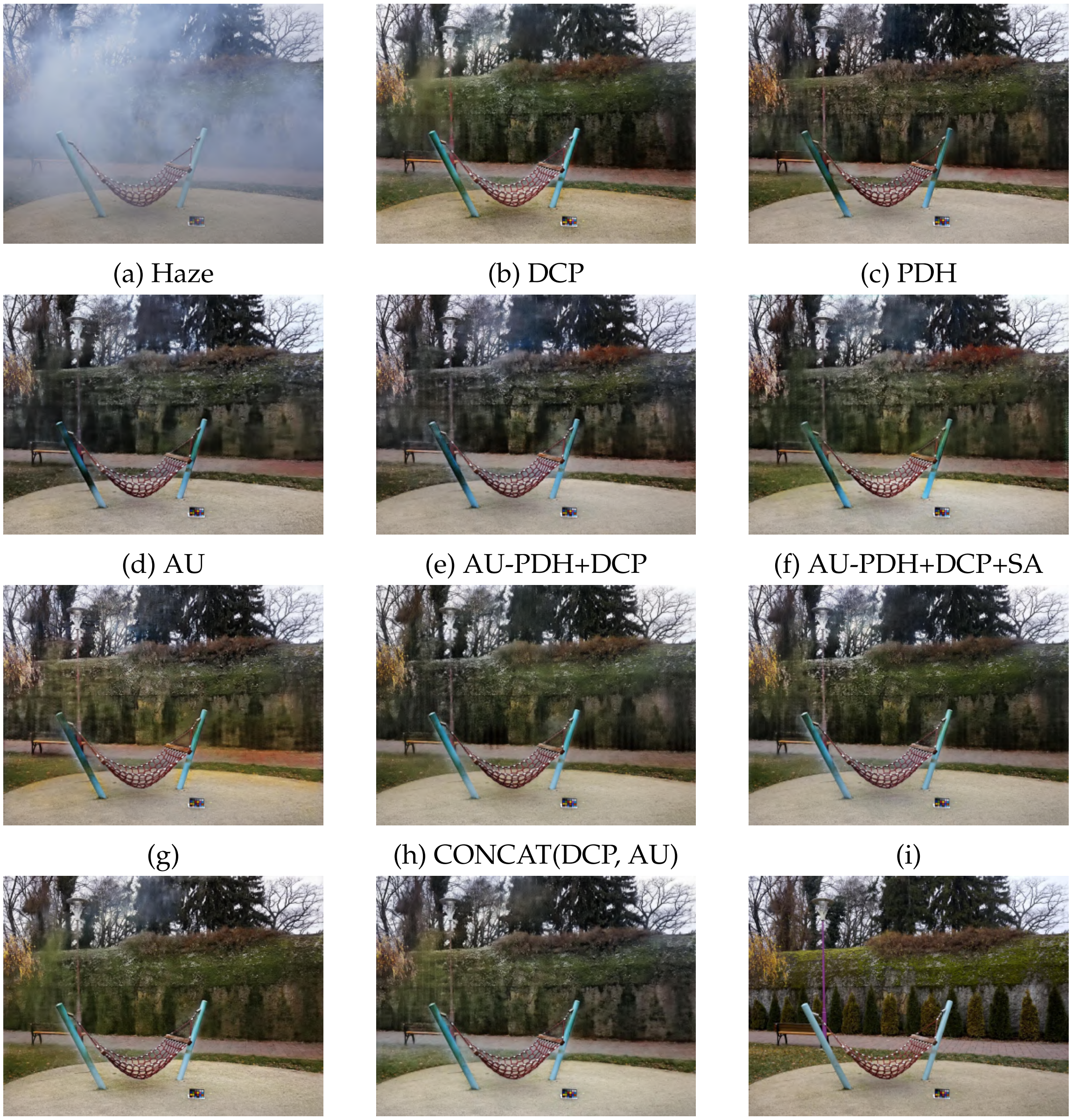
(i)

(j)

(k)

(l) Ground Truth

Figure 6.5: Visual results of various models with pix2pix as a second cascade network on image number 52 of NH-HAZE dataset. (g): AU-PDH+DCP+SA+MAS, (i): CONCAT(DCP, AU-PDH+DCP), (j): CONCAT(DCP, AU-PDH+DCP+SA), (k): CONCAT(DCP, AU-PDH+DCP+SA+MAS).



(a) Haze

(b) DCP

(c) PDH

(d) AU

(e) AU-PDH+DCP

(f) AU-PDH+DCP+SA

(g)

(h) CONCAT(DCP, AU)

(i)

(j)

(k)

(l) Ground Truth

Figure 6.6: Visual results of various models with pix2pix as a second cascade network on image number 53 of NH-HAZE dataset. (g): AU-PDH+DCP+SA+MAS, (i): CONCAT(DCP, AU-PDH+DCP), (j): CONCAT(DCP, AU-PDH+DCP+SA), (k): CONCAT(DCP, AU-PDH+DCP+SA+MAS).

6.2 On DenseHaze Dataset

6.2.1 CNN Network (First Network)

Table 6.3: Results of different first stage models for DenseHaze dataset

	L1 + Gradient Loss		**SSIM Loss**	
Experiments:	*PSNR*	*SSIM*	*PSNR*	*SSIM*
Hazy Image	8.49	0.4439	8.49	0.4439
Only DCP (no NN)	10.99	0.4360	10.99	0.4360
PDH	12.50	0.506	12.56	0.508
AU	12.24	0.532	12.22	0.518
AU-PDH	12.67	0.534	12.51	0.506
AU+SA	12.73	0.53	12.61	0.516
AU-PDH+SA	12.12	0.496	12.19	0.512
AU-PDH+DCP	13.33	0.5459	12.47	0.5359
AU-PDH+DCP+SA	13.27	0.522	12.38	0.53

"Hazy Image" in the Table 6.3 contains the PSNR and SSIM score of unprocessed raw hazy image inputs as compared with ground truths. We can see from Table 6.3 that for L1 + Gradient Loss, best PSNR score is obtained by "AU-PDH+DCP" model, while best SSIM score is also obtained by "AU-PDH+DCP" model.

It's also worth noting that both "PDH" and "AU-PDH" have PSNR greater than "AU", while "AU-PDH" has SSIM greater than "AU".

For SSIM loss, best PSNR is obtained by "AU+SA" and best SSIM is obtained by "AU-PDH+DCP". Note that there is no model containing haze detector module because there is no point of creating a haze mask for DenseHaze dataset since haze is present everywhere.

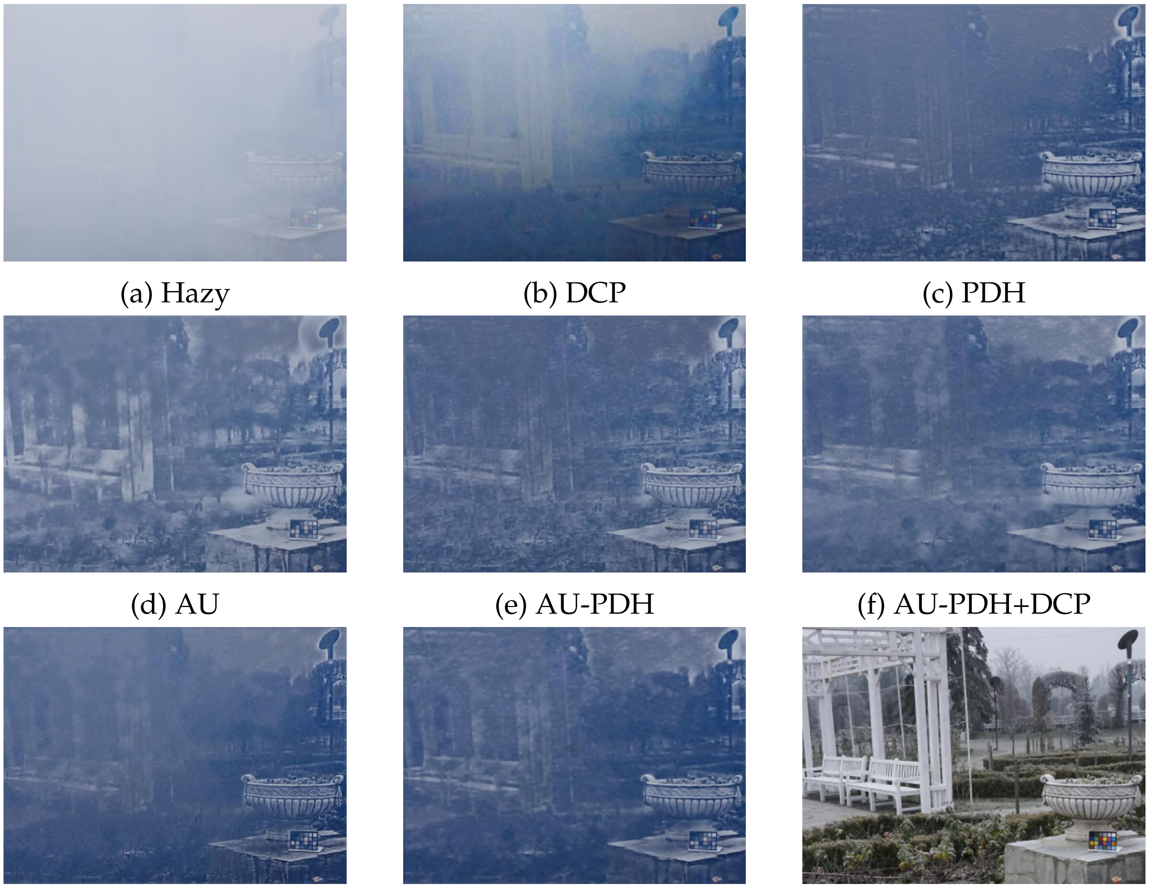
6.2.2 Pix2pix GAN Network (Second Network)

From Table 6.4 it's clear that pix2pix GAN has helped in improving the overall PSNR scores of all the models, but it's interesting that the SSIM scores have been reduced. The best PSNR is obtained from our proposed model, i.e., "AU-PDH+DCP+SA ->pix2pix". Also "CONCAT(DCP, AU-PDH+DCP+SA) ->pix2pix" has the second best SSIM score.

Table 6.4: Results of different second stage models for DenseHaze dataset

	L1 + VGG + GAN Loss	
Experiments:	*PSNR*	*SSIM*
DCP ->pix2pix	13.77	0.406
PDH ->pix2pix	14.29	0.43
AU ->pix2pix	14.15	0.36
AU-PDH+DCP+SA ->pix2pix	14.68	0.364
CONCAT(DCP, PDH) ->pix2pix	13.61	0.446
CONCAT(DCP, AU) ->pix2pix	13.10	0.394
CONCAT(DCP, AU-PDH+DCP+SA) ->pix2pix	13.79	0.4320

Qualitative results of various models with L1+Gradient loss is shown in Fig. 6.7 and Fig. 6.8. Qualitative results of various models with ssim loss is shown in Fig. 6.9 and Fig. 6.10. Qualitative results of various models with pix2pix as a second cascade network is shown in Fig. 6.11 and Fig. 6.12.

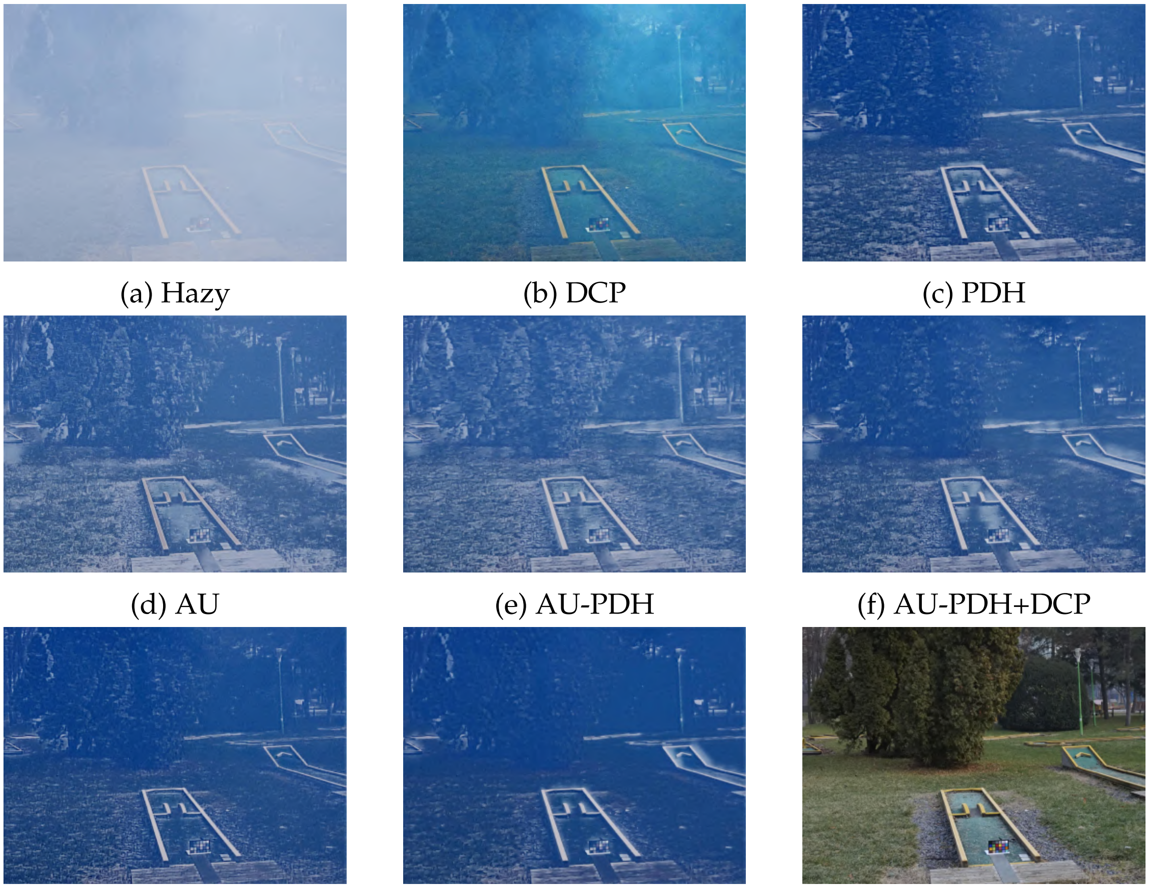


(a) Hazy (b) DCP (c) PDH

(d) AU (e) AU-PDH (f) AU-PDH+DCP

(g) AU-PDH+SA (h) AU-PDH+DCP+SA (i) Ground Truth

Figure 6.7: Visual results of various models on image number 51 of DenseHaze dataset for L1 + Gradient loss.


(a) Hazy (b) DCP (c) PDH

(d) AU (e) AU-PDH (f) AU-PDH+DCP

(g) AU-PDH+SA (h) AU-PDH+DCP+SA (i) Ground Truth

Figure 6.8: Visual results of various models on image number 53 of DenseHaze dataset for L1 + Gradient loss.

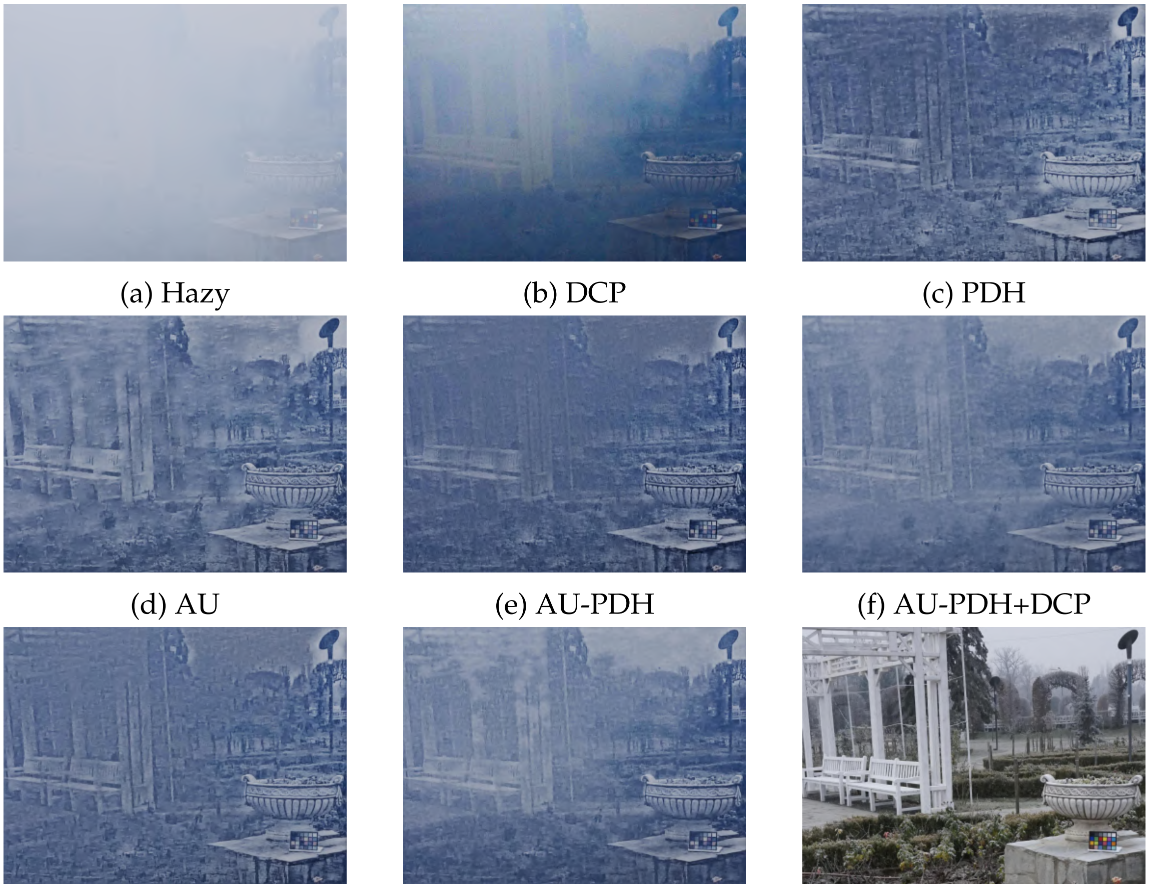
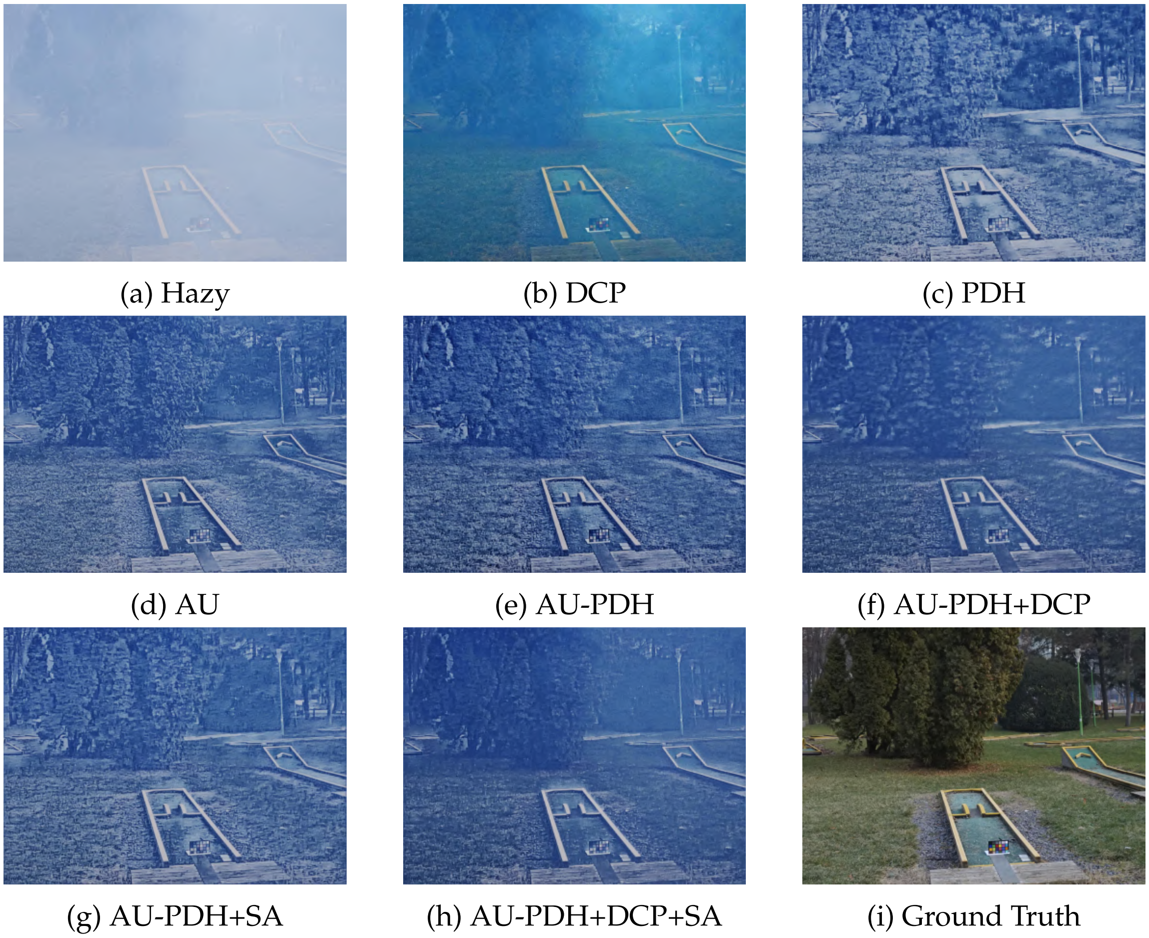

(a) Hazy
(b) DCP
(c) PDH
(d) AU
(e) AU-PDH
(f) AU-PDH+DCP
(g) AU-PDH+SA
(h) AU-PDH+DCP+SA
(i) Ground Truth

Figure 6.9: Visual results of various models on image number 51 of DenseHaze dataset for SSIM loss.



(a) Hazy

(b) DCP

(c) PDH

(d) AU

(e) AU-PDH

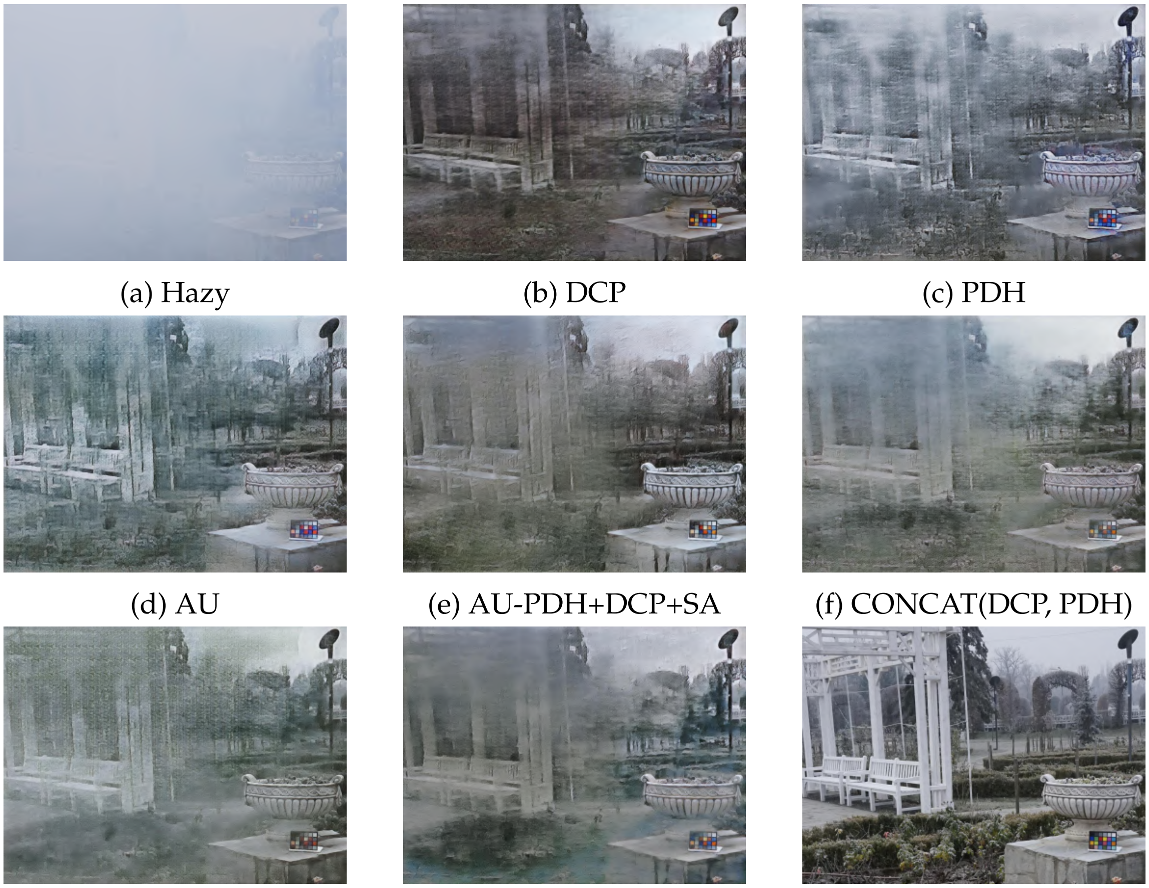
(f) AU-PDH+DCP

(g) AU-PDH+SA

(h) AU-PDH+DCP+SA

(i) Ground Truth

Figure 6.10: Visual results of various models on image number 53 of DenseHaze dataset for SSIM loss.

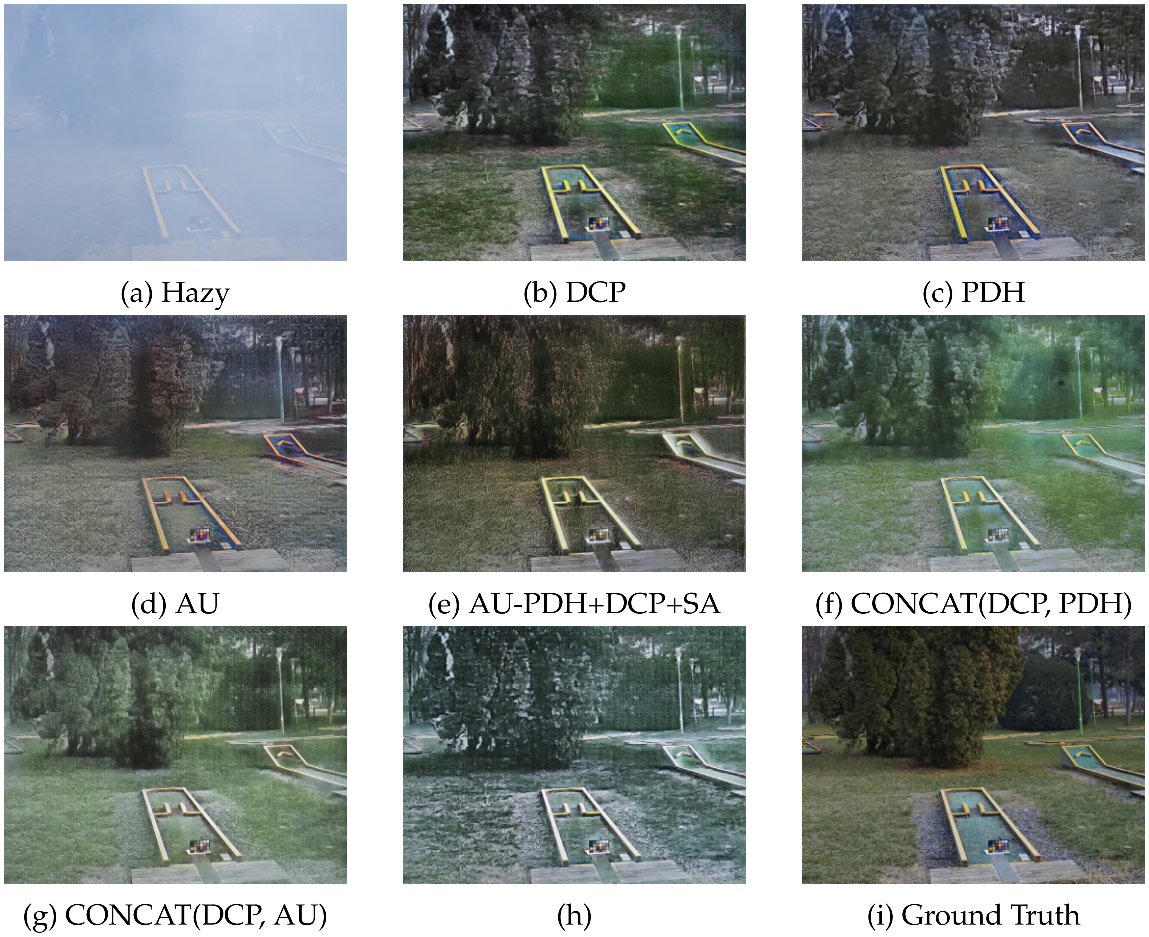


(a) Hazy (b) DCP (c) PDH

(d) AU (e) AU-PDH+DCP+SA (f) CONCAT(DCP, PDH)

(g) CONCAT(DCP, AU) (h) (i) Ground Truth

Figure 6.11: Visual results of various models with pix2pix as a second cascade network on image number 51 of DenseHaze dataset. (h): CONCAT(DCP, AU-PDH+DCP+SA).



(a) Hazy

(b) DCP

(c) PDH

(d) AU

(e) AU-PDH+DCP+SA

(f) CONCAT(DCP, PDH)

(g) CONCAT(DCP, AU)

(h)

(i) Ground Truth

Figure 6.12: Visual results of various models with pix2pix as a second cascade network on image number 53 of DenseHaze dataset. (h): CONCAT(DCP, AU-PDH+DCP+SA).

6.3 On RESIDE Indoor Test (SOTS Indoor) Dataset

6.3.1 CNN Network (First Network)

Table 6.5: Results of different first stage models for SOTS Indoor dataset

	L1 + Gradient Loss		**SSIM Loss**	
Experiments:	*PSNR*	*SSIM*	*PSNR*	*SSIM*
Hazy Image	12.50	0.7184	12.50	0.7184
Only DCP (no NN)	19.80	0.8586	19.80	0.8586
PDH	28.88	0.9521	26.05	0.9514
AU	35.99	0.9749	27.52	0.9633
AU-PDH+SA	32.24	0.9595	26.26	0.9494
AU-PDH+DCP	32.96	0.9627	26.94	0.9558
AU-PDH+DCP+SA	31.70	0.9587	27.03	0.9520

"Hazy Image" in the Table 6.5 contains the PSNR and SSIM score of unprocessed raw hazy image inputs as compared with ground truths. From Table 6.5 it's clear that both the best SSIM and PSNR is obtained by "AU" model which was proposed by the authors in [5] and was designed specifically for RESIDE dataset, hence it's performing really good. Also we can see that our proposed model, "AU-PDH+DCP" performs nearly same as compared to "AU" in both SSIM and PSNR. We can also see that all the SSIM and PSNR scores for models with SSIM loss is less as compared to L1+Gradient loss. This might be because RESIDE dataset is not that complex, hence a complex loss function like SSIM loss having structure, luminance and contrast components performs poorly.

6.3.2 Pix2pix GAN Network (Second Network)

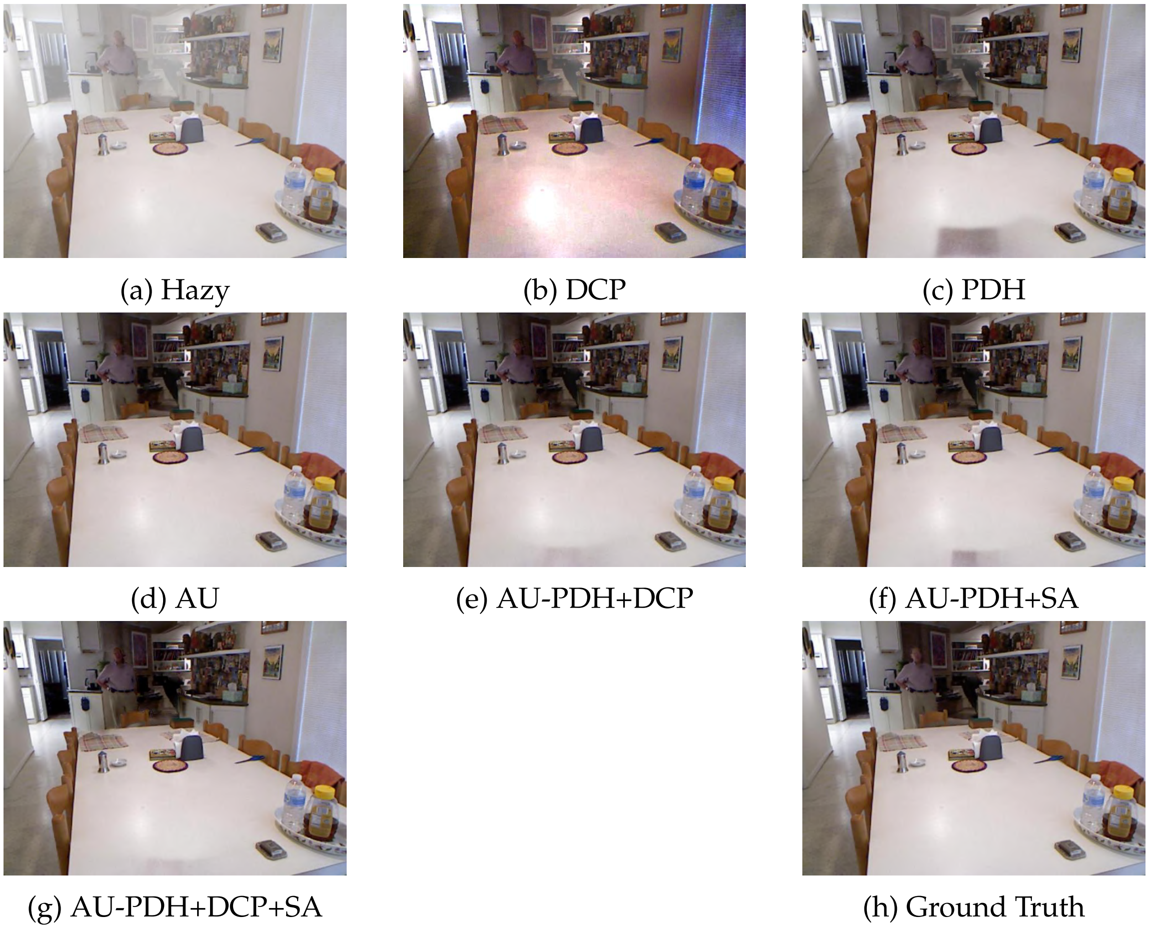
As shown in Table 6.6, both PSNR and SSIM scores of all the models have improved significantly after images are trained on pix2pix GAN. It's clear that without DCP concatination, Our proposed model, "AU-PDH+DCP -> pix2pix" performed better as compared to "AU" model in terms of both SSIM and PSNR scores. While with DCP concatination, "AU" performed better.

Qualitative results of various models with L1+Gradient loss is shown in Fig. 6.13 and Fig. 6.14. Qualitative results of various models with ssim loss is shown

Table 6.6: Results of different second stage models for SOTS Indoor dataset

	L1 + VGG + GAN Loss	
Experiments:	*PSNR*	*SSIM*
PDH ->pix2pix	42.76	0.9896
AU ->pix2pix	43.83	0.9906
AU-PDH+DCP ->pix2pix	43.88	0.9908
AU-PDH+DCP+SA ->pix2pix	42.75	0.9892
CONCAT(DCP, AU) ->pix2pix	44.24	0.9906
CONCAT(DCP, AU-PDH+DCP) ->pix2pix	43.19	0.9896
CONCAT(DCP, AU-PDH+DCP+SA) ->pix2pix	43.28	0.9898

in Fig. 6.15 and Fig. 6.16. Qualitative results of various models with pix2pix as a second cascade network is shown in Fig. 6.17 and Fig. 6.18.



(a) Hazy

(b) DCP

(c) PDH

(d) AU

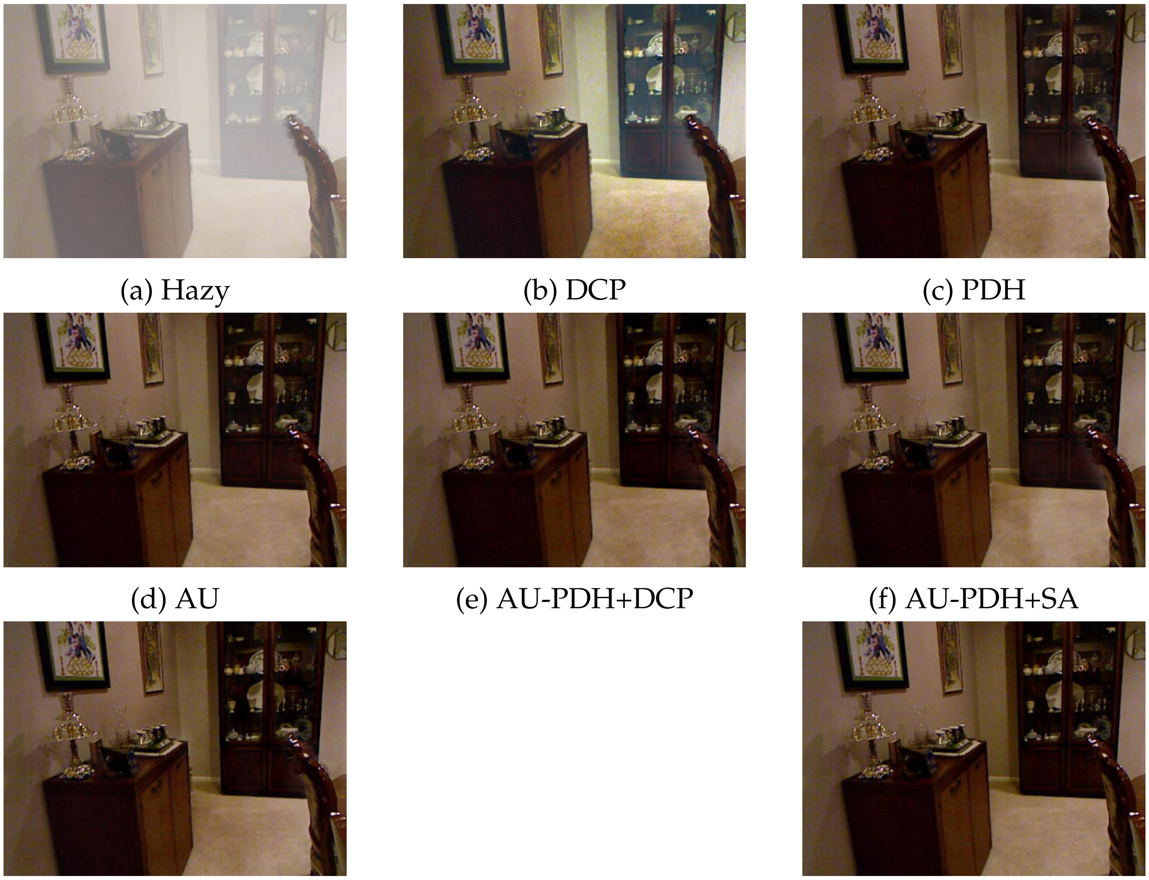
(e) AU-PDH+DCP

(f) AU-PDH+SA

(g) AU-PDH+DCP+SA

(h) Ground Truth

Figure 6.13: Visual results of various models on image number 170 of SOTS Indoor dataset for L1 + Gradient loss.



(a) Hazy

(b) DCP

(c) PDH

(d) AU

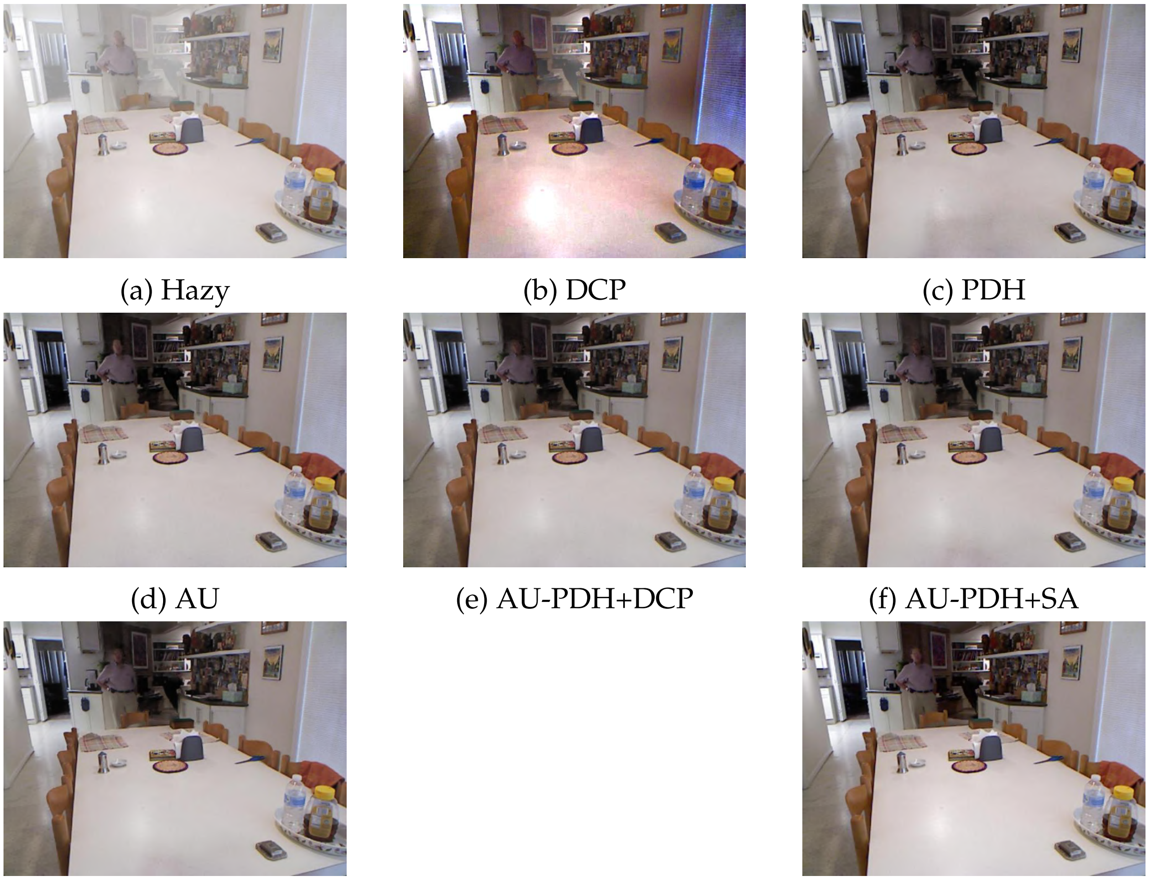
(e) AU-PDH+DCP

(f) AU-PDH+SA

(g) AU-PDH+DCP+SA

(h) Ground Truth

Figure 6.14: Visual results of various models on image number 240 of SOTS Indoor dataset for L1 + Gradient loss.



(a) Hazy

(b) DCP

(c) PDH

(d) AU

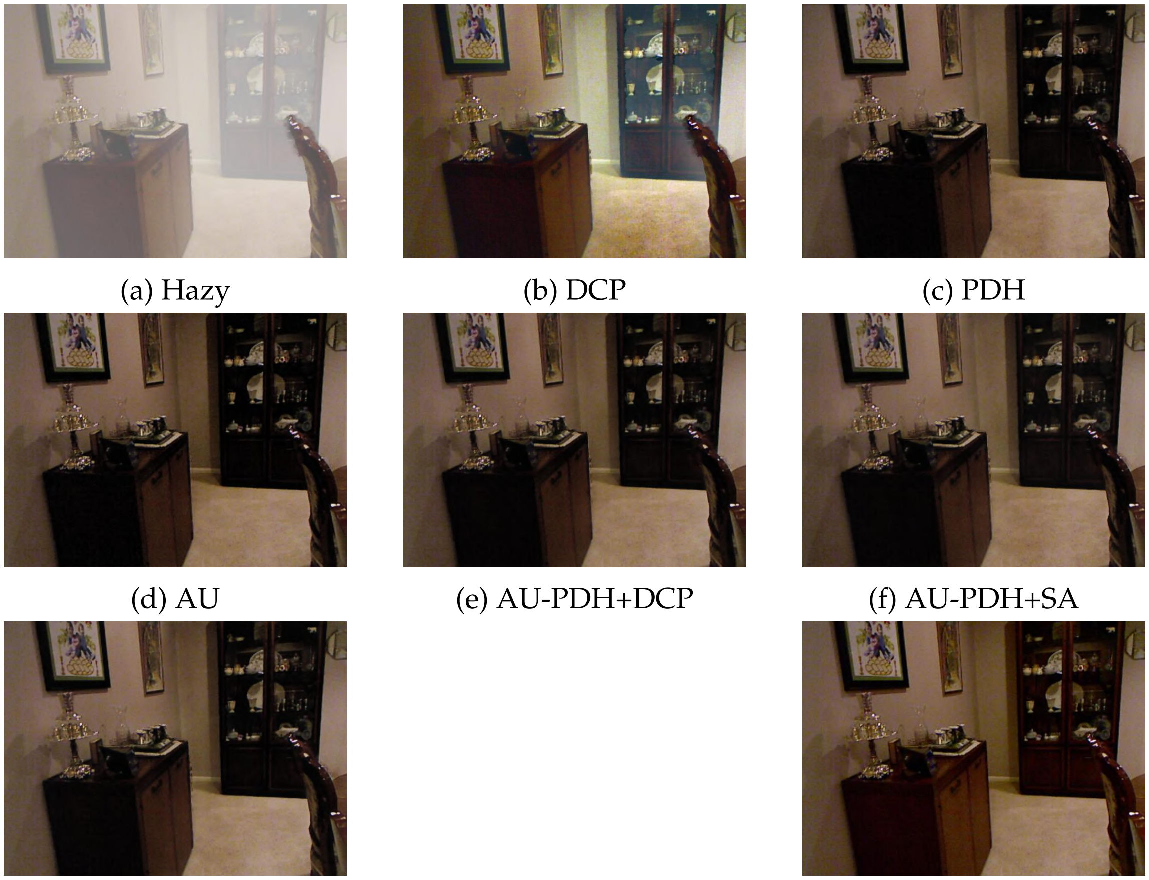
(e) AU-PDH+DCP

(f) AU-PDH+SA

(g) AU-PDH+DCP+SA

(h) Ground Truth

Figure 6.15: Visual results of various models on image number 170 of SOTS Indoor dataset for SSIM loss.



(a) Hazy

(b) DCP

(c) PDH

(d) AU

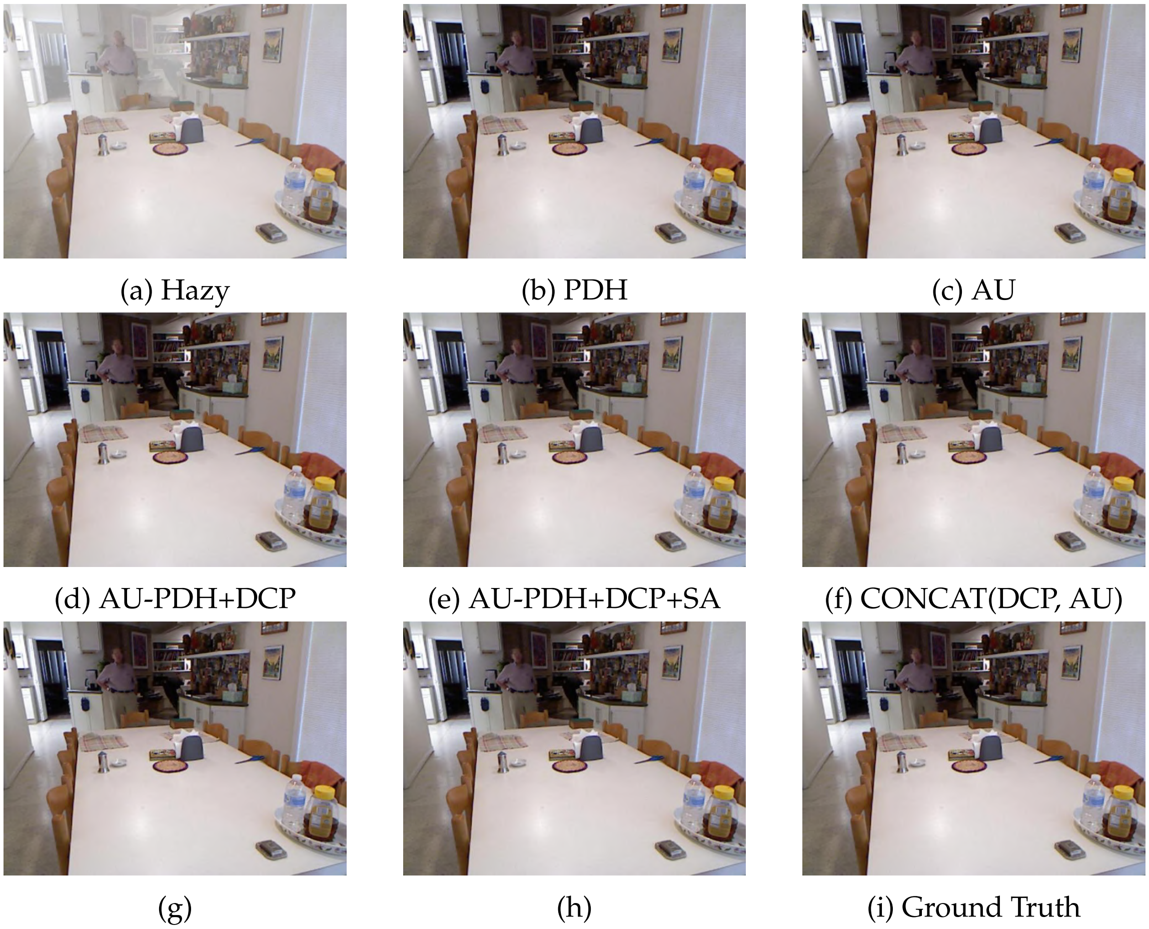
(e) AU-PDH+DCP

(f) AU-PDH+SA

(g) AU-PDH+DCP+SA

(h) Ground Truth

Figure 6.16: Visual results of various models on image number 240 of SOTS Indoor dataset for SSIM loss.



(a) Hazy

(b) PDH

(c) AU

(d) AU-PDH+DCP

(e) AU-PDH+DCP+SA

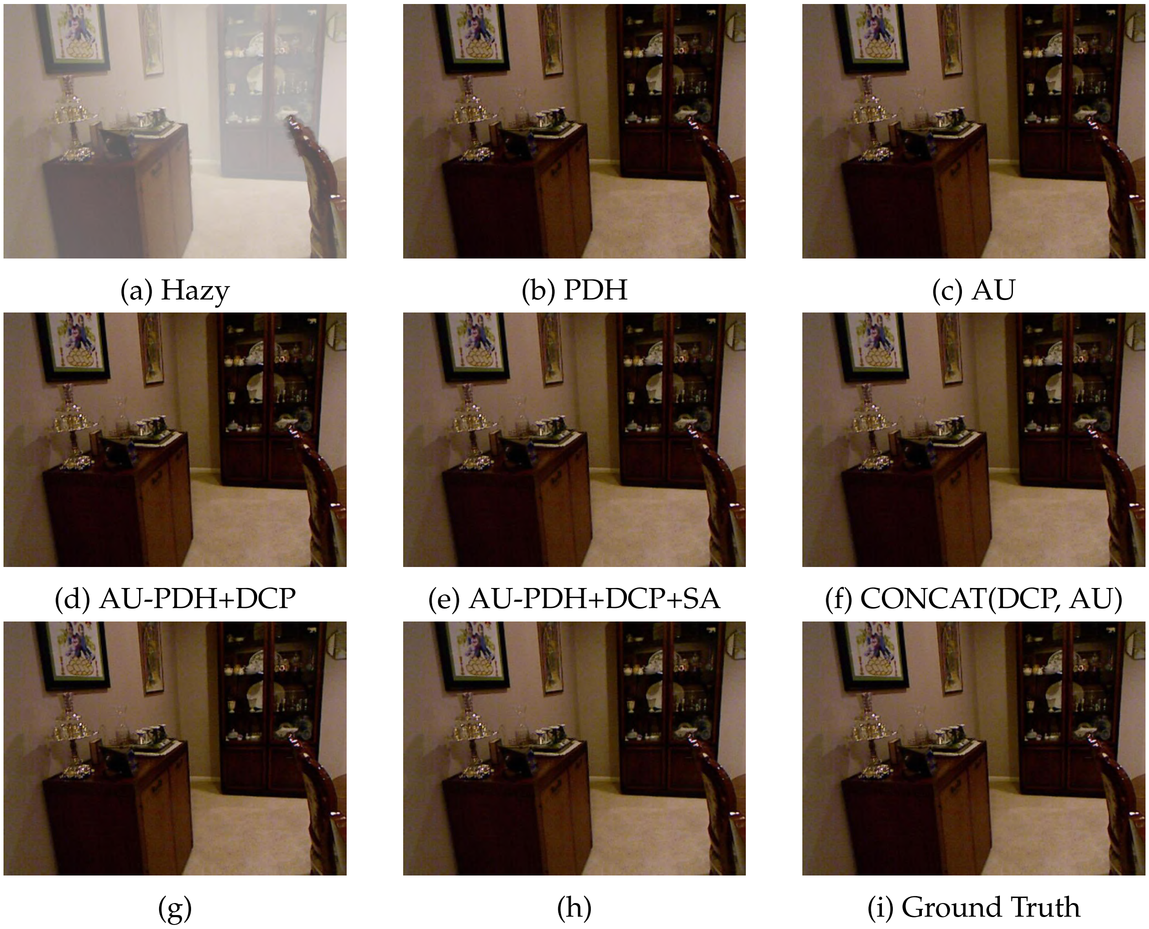
(f) CONCAT(DCP, AU)

(g)

(h)

(i) Ground Truth

Figure 6.17: Visual results of various models with pix2pix as a second cascade network on image number 170 of SOTS Indoor dataset. (g): CONCAT(DCP, AU-PDH+DCP), (h): CONCAT(DCP, AU-PDH+DCP+SA).



(a) Hazy

(b) PDH

(c) AU

(d) AU-PDH+DCP

(e) AU-PDH+DCP+SA

(f) CONCAT(DCP, AU)

(g)

(h)

(i) Ground Truth

Figure 6.18: Visual results of various models with pix2pix as a second cascade network on image number 240 of SOTS Indoor dataset. (g): CONCAT(DCP, AU-PDH+DCP), (h): CONCAT(DCP, AU-PDH+DCP+SA).

6.4 On a real-life hazy image

We have also evaluated our models on a real life image [1] to find how generalizable our models are and how they perform on an image which is of a completely different distribution than of our training data. The image is shown in Fig. 6.19

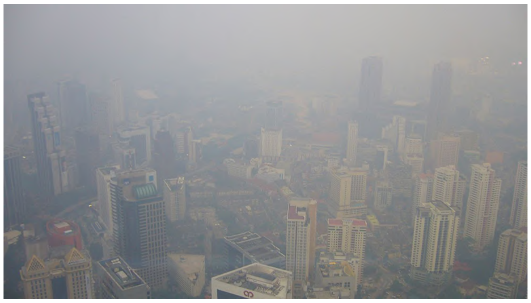


Figure 6.19: Real-life hazy image [1]

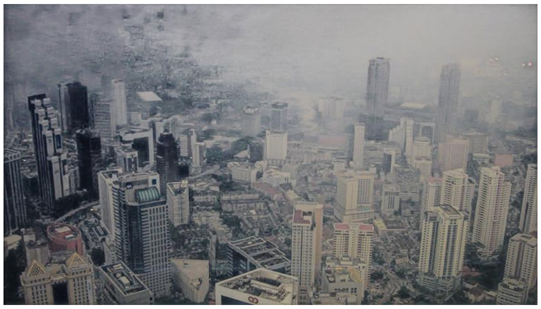


Figure 6.20: Output of AU-PDH+DCP+SA+MAS model trained on NH-HAZE dataset

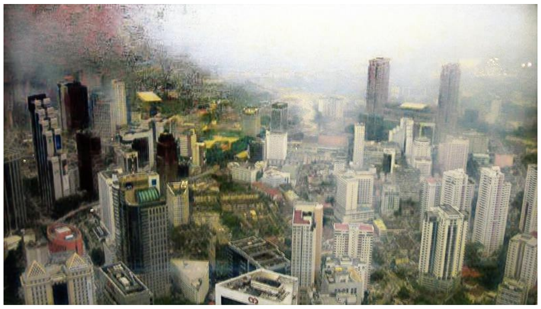


Figure 6.21: Output of CONCAT(DCP, AU-PDH+DCP+SA+MAS) -> pix2pix model trained on NH-HAZE dataset

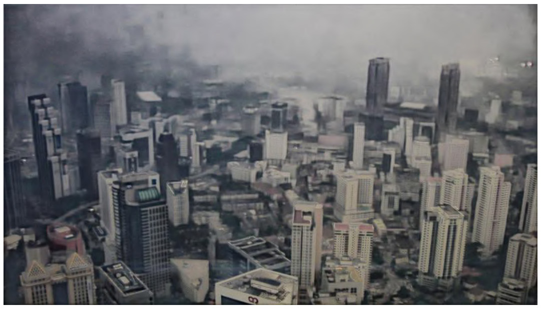


Figure 6.22: Output of AU-PDH+DCP+SA model trained on DenseHaze dataset

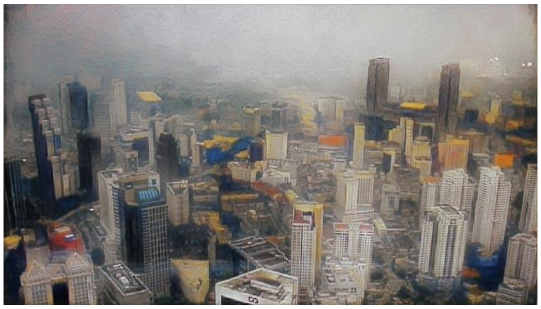


Figure 6.23: Output of CONCAT(DCP, AU-PDH+DCP+SA) -> pix2pix model trained on DenseHaze dataset

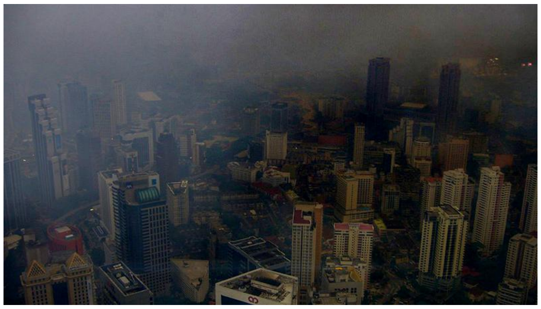


Figure 6.24: Output of AU-PDH+DCP model trained on SOTS Indoor dataset

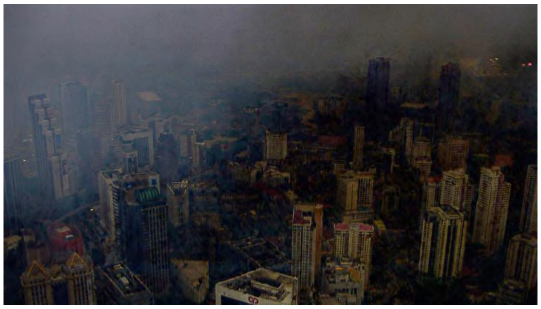


Figure 6.25: Output of AU-PDH+DCP -> pix2pix model trained on SOTS Indoor dataset

CHAPTER 7

Conclusion and Future work

We have proposed a cascaded model for image dehazing which performs really well on non-homogeneous data as well as on homogoneous data. We have also verified that cascaded networks are not always a good choice by comparing progressive feature fusion model and modifying it to work on stand alone mode without cascading. Hence it's really important to make sure that the individual models are getting an improvement overall after cascading. Since that's the only way to make their additional computational cost worth it.

We also saw that loss functions are incredibly important for a problem statement since they affect the result a lot. Changing the loss function can change the outcome by a huge factor, as we saw when we change the loss function from L1+Gradient to SSIM. We also saw that SSIM loss performed better on NH-HAZE dataset while L1+Gradient loss performed better on SOTS Indoor dataset.

Hence, as part of future work, to boost optimizing loss function, we can try to produce a mathematical method which can give us better loss coefficients for better convergence rather than blindly experimenting. we can try incorporating current state-of-the-art models like vision transformers to fit complex data.

We also would like to incorporate some reference less loss functions like the ones proposed in [7]. Reference less models don't require ground truths and hence are more generalizable and operable on wide distribution of images.

Also it's quite apparent that SSIM and PSNR alone don't really give a good view of qualitative metric. Hence we can try different metrics which are more inclined towards human visual perception.

References

[1] https://cdn.climatechangenews.com/files/2015/10/295153252_19d224d3bf_o.jpg.

[2] https://www.researchgate.net/publication/346226974_single-image_visibility_restoration_a_machine_learning_approach_and_its_4k-capable_hardware_accelerator/figures?lo=1.

[3] C. O. Ancuti, C. Ancuti, M. Sbert, and R. 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] C. O. Ancuti, C. Ancuti, and R. 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.

[5] H. Bai, J. Pan, X. Xiang, and J. Tang. Self-guided image dehazing using progressive feature fusion. *IEEE Transactions on Image Processing*, 31:1217–1229, 2022.

[6] C.-F. R. Chen, Q. Fan, and R. Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 357–366, 2021.

[7] C. Guo, C. Li, J. Guo, C. C. Loy, J. Hou, S. Kwong, and R. Cong. Zero-reference deep curve estimation for low-light image enhancement. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 1780–1789, 2020.

[8] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(12):2341–2353, 2011.

[9] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[10] C. Kim. Region adaptive single image dehazing. *Entropy*, 23(11):1438, 2021.

[11] B. P. Kumar, A. Kumar, and R. Pandey. Region-based adaptive single image dehazing, detail enhancement and pre-processing using auto-colour transfer method. *Signal Processing: Image Communication*, 100:116532, 2022.

[12] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2019.

[13] K. Metwaly, X. Li, T. Guo, and V. Monga. Nonlocal channel attention for nonhomogeneous image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 452–453, 2020.

[14] Y. Song, Z. He, H. Qian, and X. Du. Vision transformers for single image dehazing. *IEEE Transactions on Image Processing*, 32:1927–1941, 2023.

[15] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[16] M. Yu, V. Cherukuri, T. Guo, and V. Monga. Ensemble dehazing networks for non-homogeneous haze. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 450–451, 2020.

[17] S. Zhao, L. Zhang, Y. Shen, and Y. Zhou. Refinednet: A weakly supervised refinement framework for single image dehazing. *IEEE Transactions on Image Processing*, 30:3391–3404, 2021.

CHAPTER A
DCP Proof

In this section we have given the proof for Dark Channel Prior method. We know from Eq. 1.1,

$$I(x) = J(x)t(x) + A(1 - t(x)) \tag{A.1}$$

Now let's divide both sides by atmospheric light,

$$\frac{I(x)}{A} = t(x)\frac{J(x)}{A} + (1 - t(x)) \tag{A.2}$$

If we compute patch wise minimum values on each side of the equation we will get,

$$\min_{y\in\omega(x)}(\frac{I^c(x)}{A^c}) = \tilde{t}(x) \min_{y\in\omega(x)}(\frac{J^c(x)}{A^c}) + (1 - \tilde{t}(x)) \tag{A.3}$$

Now if we compute the dark channel on each side of the equation we will get,

$$\min_{y\in\omega(x)}(\min_c \frac{I^c(x)}{A^c}) = \tilde{t}(x) \min_{y\in\omega(x)}(\min_c \frac{J^c(x)}{A^c}) + (1 - \tilde{t}(x)) \tag{A.4}$$

But as per our prior, the dark channel of non hazy image is completely black and hence the dark channel of RHS will become zero. So our equation becomes,

$$\tilde{t}(x) = 1 - \min_c(\min_{y\in\omega(x)}(\frac{I^c(y)}{A^c})) \tag{A.5}$$

Note that the transmission map we obtain is not t(x) but $\tilde{t}(x)$. This is because this transmission map is an approximation of the real transmission map since we used a prior for the computation.