Non-homogeneous Dehazing of Images by Attention Mechanism in Deep Framework

by

Akash Dhedhi
202011053

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

July, 2022

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.

A.T.Dhedhi

Akash Dhedhi

Certificate

This is to certify that the thesis work entitled Non-homogeneous Dehazing of Images by Attention Mechanism in Deep Framework has been carried out by Akash Dhedhi 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.

Prof. Rajib Lochan Das
Thesis Supervisor

Prof. Srimanta Mandal
Thesis Co-Supervisor

Acknowledgments

First and foremost, I am extremely grateful to my supervisors, Prof. Rajib Lochan Das and Prof. Srimanta Mandal, for their invaluable advice, continuous support, and patience during my MTech Thesis. Their immense knowledge and ample experience have encouraged me during my academic research.

I want to thank my peers and friends Tarang Ranpara, Dhyanil Mehta, Shivangi Gajjar, Krishna Savaliya, Manali Bhavsar, and Vatsal Patel who were always there in my ups and downs. Their kind help and support have made my study and life at DA-IICT a wonderful and memorable experience.

Finally, I would like to express my gratitude to my parents and family for their love, care, and invaluable support throughout my life. A very special thanks to my brother Rakesh Dhedhi for encouraging me and pushing me when needed.

Contents

Abstract

The availability of dehazing datasets has enabled various deep learning techniques to perform effectively on hazy images. Most of the developed frameworks focus on removing homogeneous haze. However, homogeneous-centric methods produce sub-optimal results on non-homogeneous haze. The primary reason is that the architectures devised to handle homogeneous haze fail to address the non-uniformity of haze in non-homogeneous case. The secondary reason is the unavailability of enough data for the non-homogeneous scenario. Although many works cite the lack of data as a primary concern for poor performance, we find that even if the homogeneous-centric networks are trained with non-homogeneous data, the produced results are sub-standard. Hence, there is a requirement for a network architecture that can handle non-homogeneous haze in a better way. In this work, we propose to use multiple attention mechanisms in parallel along with pre-trained ConvNeXt blocks. Specifically, we use pixel, channel, and residual channel attention mechanisms. Pixel attention can complement the channel attention in dealing with space-variant haze when connected in parallel. On the other hand, residual channel attention fetches hazy image-related features and caters to better information flow towards the output. Concatenating the attention-based features can yield better results as compared to the existing approaches.

Keywords: Attention mechanism, ConvNeXt, Image Dehazing, Non-homogeneous haze,

List of Tables

List of Figures

CHAPTER 1
Introduction

Atmospheric particles often hinders the visibility of a scene by creating phenomena like haze, fog, etc. This is because that the object reflected light gets attenuated along the line of path. Further, the atmospheric light gets scattered by the atmospheric particles as shown in Fig. 1.1. As a result, a portion of atmospheric light (*a.k.a airlight*) is added with the direct light, received by a camera. Hence, the captured image suffers from lower contrast, color distortion, etc.

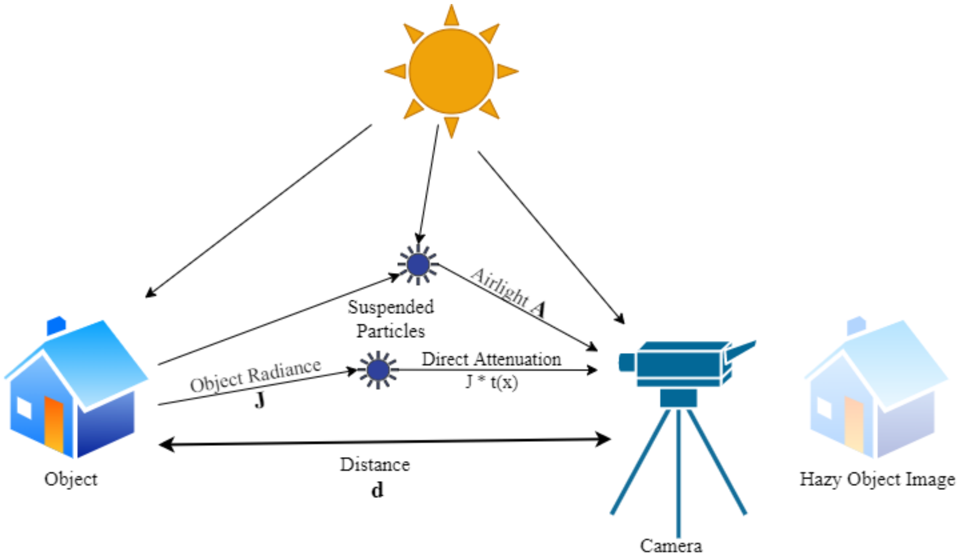
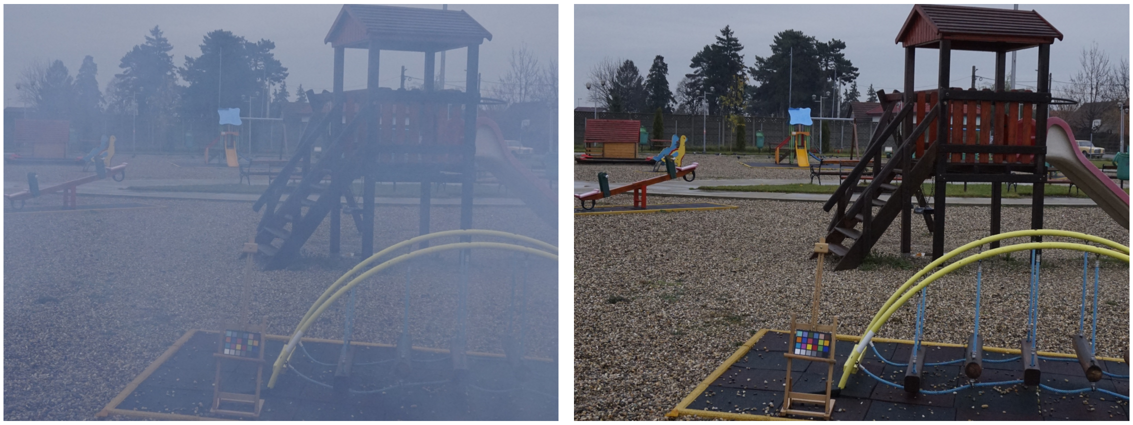


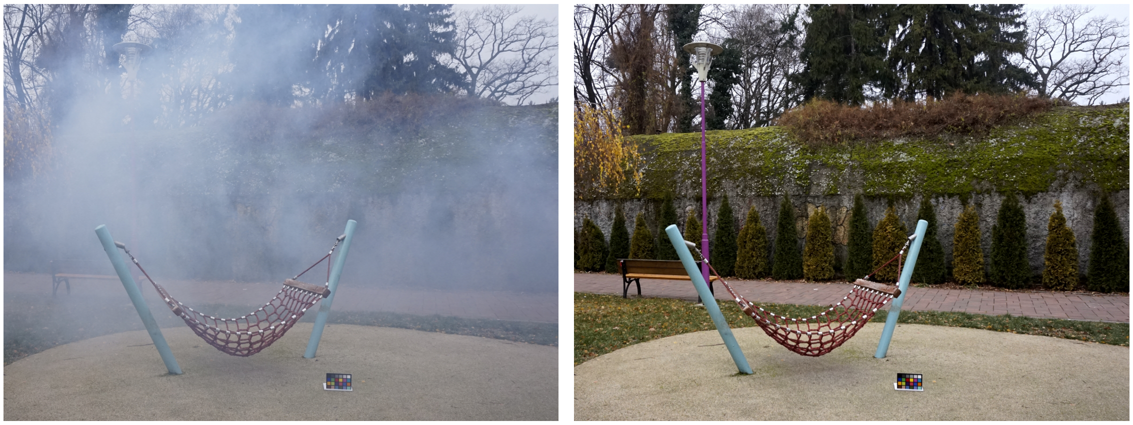
Figure 1.1: Atmospheric Scattering Model(ASM)

This kind of degraded image is not desirable in vision related applications. Hence, the effect of haze or other atmospheric phenomena needs to be reduced. For example, we need to produce an image like Fig.1.3(b) from 1.3(a), which is degraded by haze.



((a)) Hazy Image ((b)) Haze-free Image

Figure 1.2: Example of homogeneous image dehazing



((a)) Hazy Image ((b)) Haze-free Image

Figure 1.3: Example of non-homogeneous image dehazing

Haze can be of two types: i) homogeneous, and ii) non-homogeneous. In homogeneous hazy images, haze is evenly distributed at the same depth level. As illustrated in Fig.1.2(a), the haze is evenly distributed throughout the entire scene. In contrast to homogeneous hazy images, haze presence is unequally distributed in non-homogeneous hazy images, which means that part of the scene is hazy or has a high density compared to other regions. As shown in Fig.1.3(a), the haze is distributed unevenly across the entire scene, with some areas having high-density haze and others having very-low-density haze.

The homogeneous haze type degradation can be approximated by atmospheric scattering model (ASM) [27, 29, 28] as

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

Here, $I(x)$, $J(x)$, and $t(x)$ denote the hazy image captured using the camera, the original scene (haze-free image), and the transmission map, respectively. A is the global atmospheric light, and x indicates the pixel location of the image. In the equation, the first term $J(x)t(x)$ is the direct component, and the second term $A(1 - t(x))$ is known as Airlight. We need to find two unknowns in this method: $t(x)$ and A from $I(x)$. Here, $t(x)$ is directly related to the distance $d(x)$ of the scene point from the camera as $t(x) = e^{-\beta d(x)}$.

Here β is the scattering coefficient. Several dehazing methods follow this model to find out a haze-free image. However, the ASM can not reflect the non-homogeneous nature of haze. As a result, methods developed based on ASM find it difficult to produce an optimal dehazing result on non-homogeneous hazy image.

1.1 Objective

The objective of the thesis are summarized as follows:

- Design an architecture that is suitable for dealing with non-homogeneous haze.
- We require an architecture that can handle limited data challenges and the non-homogeneous nature of haze.
- Even if the dataset is available, one needs to devise an architecture that can address the varying density of non-homogeneous haze images.
- To address the color cast issue, we must include a module in our architecture that can handle it.

1.2 Contribution

We are mainly focused on different attention strategies along with pre-trained deep architecture. The main contributions of the thesis are summarized as follows:

- We propose a deep architecture based on an attention mechanism to dehaze an image that is degraded by a non-homogeneous haze.
- Demonstrate the effectiveness of the pre-trained model for Non-Homogeneous dehazing.

- We investigate and demonstrate that the parallel combination of channel attention and pixel attention is more effective than its serial counterpart for the problem.

- We demonstrate that a pre-trained model for classification can assist in addressing the issue of a lesser amount of data, which is often the case for image dehazing.

- Additionally, we address the issue of the color cast effect on images by employing a multi-colored module in the network.

1.3 Organization of Thesis

The rest of the thesis is structured as follows: Chapter 2 includes Existing techniques for image dehazing, including the classical method and deep learning-based methods. Chapter 3 includes the proposed method containing network architecture and loss function details. In this chapter, we will explained how two branches of our network can use pertained model and attention mechanism to handle non-homogeneous haze. Chapter 3.3 includes the results of our experiments. This chapter compares our results with the existing state-of-the-art methods. Also, we include an ablation study to see the effectiveness of pre-trained models and attention mechanisms. We also include some of the different experiments we did during our thesis work. Chapter 4 includes a chromatic casting image dehazing method containing detailed information about network architecture and how we create the different colored datasets for our training purpose. This chapter discussed how chromatic cast effects work in our network architecture and the conclusion is in chapter 5.

CHAPTER 2

Literature Survey

Image Dehazing is a challenging task because the hazy image does not contain any haze-related information. The distance between camera and object is an important factor to deal with the density of haze in hazy images. Also, hazy images do not carry this distance information. In this chapter, we will go over the various image dehazing methods briefly.

2.1 Classical Methods

According to eq 1.1, we need to find two unknowns for finding haze-free images: atmospheric light and a transmission map that depends on the depth(distance) of the scene point from the camera. Because this distance information is not carried by the image, determining the distance value is tricky. This limitation turned concentration into the classical method[11, 28, 36, 12, 18, 26]. The dehazing task based on a single reference image depends on the characteristics of the haze-free image. Tan[36] noticed that haze-free images have more contrast than hazy images and suggested approaches for boosting image visibility through optimization of local contrast. Dark Channel Prior (DCP) [20] relies on dark pixels in the most haze-free outdoor images that have at least one color channel with very low intensity. DCP fails in the case of sky regions since its color is entirely equivalent to atmospheric light.

2.2 Deep Learning Methods

Instead of relying on hand-crafted features like classical approaches, deep learning methods learn features automatically from training data. Hence, it has become a popular choice for many vision applications, including image dehazing.

2.2.1 Homogeneous Dehazing

Cai et al.[13] have developed DehazeNet as a deep learning method for determining medium transmission map t(x). After computing the transmission map, DehazeNet uses the atmospheric scattering model to find haze-free images. As we discussed above, we required an Atmospheric light value in eq (1.1). The author uses the highest intensity value of hazy images I(x) for atmospheric light. When any white object is present in a scene, then this white object is detected as an atmospheric value which is not valid. In Dehazenet, we still use prior for computing atmospheric light value. There will be some end-to-end mapping between hazy images and haze-free images.

Unlike DehazeNet, AOD-net[23] calculates the atmospheric light and transmission map simultaneously and then uses ASM to produce a haze-free image. Several deep learning approaches calculate haze-free images directly without using ASM by mapping end-to-end between haze-free and hazy images. Qin et al.[30] suggested FFA-net, an attention-based neural network that can directly generate haze-free images without calculating atmospheric light or transmission map. In FFA-net, the input is a Hazy image, which is subsequently processed by an N-group architecture consisting of several skip connections and residual connections. In each group, there is B basic block with an attention mechanism.Finally, they employ the Attention block (Channel attention + pixel attention) and two convolution layers to produce a haze-free image. These approaches perform well on homogeneous hazy images but fail to produce satisfactory results on non-homogeneous datasets.

2.2.2 Non-Homogeneous Dehazing

To handle the non-homogeneous nature of haze, numerous deep learning algorithms have been proposed in NTIRE2020[9] and NTIRE2021[10] challenges.

Existing multi-scale-based approaches require a significant amount of time and memory. To overcome this limitation, Das and Dutta[14] have suggested Deep Multi patch/scale Hierarchical Network (DMSHN and DMPHN) techniques, which rely on hierarchy-based multi-patch and multi-scale neural networks. It also has a shorter runtime than current multi-scale approaches. DMPHM used multi-level architecture, With different patch sizes at each level. They use 1,2 and 4 patches from top to bottom. The top level has only one patch per image. Then, the next level images are divided vertically, so we got two patches per image. At the last level, image is divided horizontally, so we got four patches per image. Instead of using patches, DHSHM uses different levels of scale. The structure of the encoder-decoder is the same as DHPHM. When we go through the upper level to the bottom level, Image size is downsampled by scales 1,2 and 4, respectively. In this encoder-decoder architecture, 15 convolution layers and six residual connections are used in encoder. Encoder and decoder architectures are the same except for two convolutions replaced with De-convolution layers. The author demonstrates that DMPHN outperforms DMSHN because DMPHN integrates local features generated from a finer to a coarser level.

Haze is not entirely removed from hazy images because of non-homogeneous hazy image data scarcity. Some haze is still visible in the images. The pre-trained model is the best solution in deep learning to tackle this issue. Wu et al.[38] have presented KTDN, which uses pre-trained ResNet for feature extraction in the encoder and an attention mechanism in the decoder. This team got the second rank in NTIRE2020[9] challenge, and the method gives significant results compared to other existing methods, although pre-trained models train on a 1k ImageNet dataset and are utilized for classification purposes. As a result, these pre-trained algorithms cannot remove haze from small objects in images successfully.

To take advantage of pre-trained models and work with current data, Yankun et al.[40] introduced two-branch neural networks. This approach extends the KDTN[38] notion by adding another branch that works with current data. This procedure produces excellent results.

2.3 Chapter Summary

In image dehazing, the classical and homogeneous methods work well with a homogeneous dataset. We have lots of data for homogeneous haze like RESIDE[24], D-HAZY[1], DENSE[4, 7], I-HAZE[8], O-HAZE[6], and FRIDA[37]. We can also create data for homogeneous haze by using the Atmospheric scattering model. We need data that contain a depth of scene information like the NYU[35] dataset.

Many deep learning-based methods are introduced which use this homogeneous data. These methods work well with homogeneous haze but fail to address the non-homogeneous haze scenario. We need to face two main challenges when dealing with non-homogeneous dehazing, which are 1) the non-homogeneous nature of haze and 2) limited non-homogeneous data. We can take advantage of a pre-trained model that gives better feature representation and attention mechanisms (channel attention and pixel attention) to deal with these challenges.

CHAPTER 3

Proposed Method

3.1 Network Structure

The network architecture of our method is shown in Fig. 3.1. The main components of our architecture are i) ConvNeXt encoder [25], ii) decoder with attention block, which is a parallel connection of channel attention and pixel attention, iii) Residual Channel Attention Network (RCAN), and iv) feature fusion. The first two components are used in sequence in one stream, whereas the third component is used in parallel to the first two as a second stream. Finally, the features from both the streams are fused to generate a final dehazed result using the last component of our architecture.

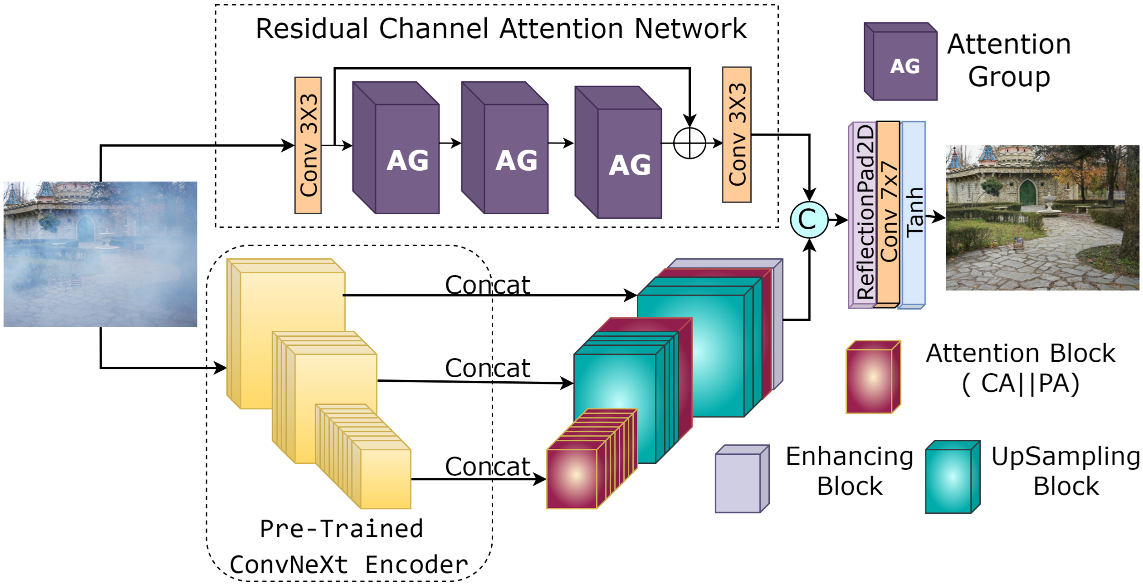


Figure 3.1: The architecture of the proposed model

3.1.1 ConvNeXt Encoder

Datasets consist of hazy and haze-free images have limited number of data. Hence, a deep encoder, trained with very limited data may not be able to provide better representation. Hence, we use pre-trained ConvNeXt model [25] to achieve a suitable and robust feature from the input data. Since, the model is pre-trained with ImageNet dataset to classify images effectively, the pre-trained weights need to be fine-tuned for dehazing purpose using transfer learning method.

The ConvNext architecture [25] is a modified version of ResNet with ConvNet modules, tailored to emulate the vision transformer design. It uses fewer activation fucntions in the form of GELU along with layer normalization. Unlike ResNet, ConvNext uses non-overlapping and depth-wise convolution in an inverted bottleneck structure. This structure supports regularization and FLOPS reduction. For our network, we use only the front part of the ConvNext with 16 times down sampling.

3.1.2 Decoder with Attention Block

The decoder consists of attention blocks, up-sampling blocks and an enhancement block. The attention block contains channel attention and pixel attention in parallel (CA || PA), as can be seen in Fig. 3.2. The channel attention mechanism can be described by

$$A_c = \sigma\left(MLP(Avgpool(F)\right), \tag{3.1}$$

where F is the input feature. It first aggregate spatial information by average pooling operation on F. The resultant features are then fed to an MLP layer. Finally the feature is passed through a sigmoid activation function to produce channel attention A_c, which can be multiplied with F as $F_c = A_c \otimes F$. On the other hand, the pixel attention A_p is produced by 1×1 convolution followed by sigmoid activation on F. The $F_p = A_p \otimes F$ is the output of pixel attention mechanism. The output of CA and PA are multiplied element-wise to produce the final feature map.

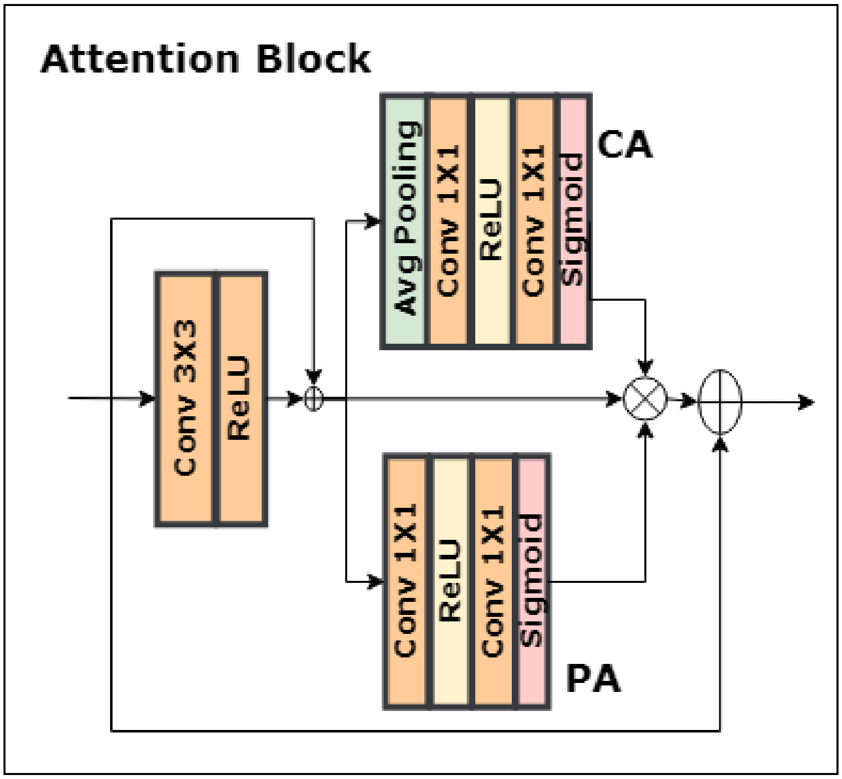


Figure 3.2: Attention Block

The channel attention (CA) brings out channel wise significant information while the pixel attention (PA) conveys pixel level attention. Their parallel connection can complement each other in better manner as compared to their serial interconnection. The reason being that the output of channel attention will be weighted. As a result, the pixel level information is affected. Hence, pixel attention on the output of channel attention will not be suitable as the PA will not be able to explore the unaffected pixel level information. This issue is resolved in our method by connecting CA and PA in parallel. The output of attention mechanism is fed to PixelShuffle layer [34], followed by an enhancement block that fuse multi-scale features.

3.1.3 Residual Channel Attention Block

This stream of network (see Fig. 3.3) assists in complementing the first stream, where the pre-trained ConvNext model is used along with CA || PA. In the first stream, the attention mechanism works on the pre-trained robust feature with fine-tuning to the new data. In the second stream, the residual channel attention blocks (RCAB) [41] play the lead role.

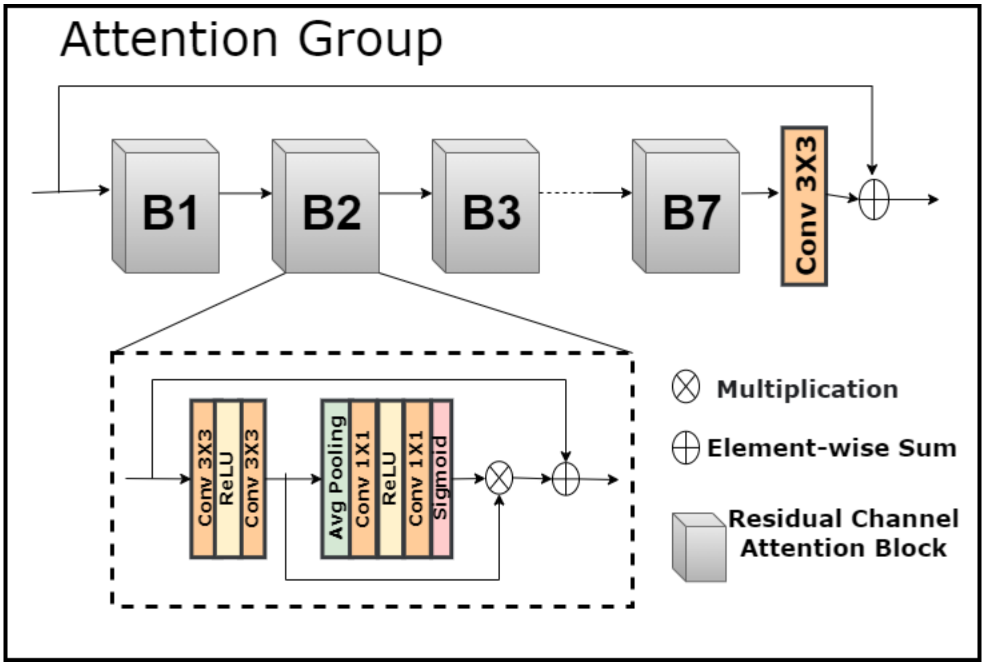


Figure 3.3: Attention Group

In RCAB, the channel attention directly faces the input hazy data to produce a feature with weights highlighting channel-wise important features. Further, short and long skip connections enable a smoother information flow from the input towards output. Additionally, it assists in addressing the over-fitting issue, which is often encountered by deep-learning models due to lesser data.

3.1.4 Feature Fusion

Here we concatenate the features from both the streams and apply one layer of convolution followed by *tanh* activation function to produce the dehazed results.

3.2 Loss Function

The loss function is,

$$L = \gamma_1 L_{l1} + \gamma_2 L_{MS-SSIM} + \gamma_3 L_{perc} + \gamma_4 L_{adv} \qquad (3.2)$$

L comprises of four different components: i) smooth L1 loss (L_{l1}), ii) structural similarity loss ($L_{MS-SSIM}$), iii) perceptual loss (L_{perc}), and iv) adversarial loss (L_{adv}). Hyper-parameters $\gamma_1, \gamma_2, \gamma_3$, and γ_4 are used to balance losses.

3.2.1 Smooth L1 loss

To assure that the predicted images are similar to clean images, we use the smooth L1 loss[19].

$$L_{l1} = \frac{1}{N} \sum_{x=1}^{N} smooth_{L1}(j(x) - f_\theta(i(x))) \tag{3.3}$$

$$smooth_{L1}(p) = \begin{cases} 0.5p^2 & \text{if } |p| < 1. \\ |p| - 0.5 & \text{otherwise.} \end{cases} \tag{3.4}$$

Here $i(x)$ and $j(x)$ are hazy image and ground truth image (vector form), respectively at pixel x. $f_\theta(\cdot)$ represents network function, parameterized by θ. N denotes total number of pixels.

3.2.2 Structure Similarity loss

We use multi-scale structure similarity (MS-SSIM) loss[42] such that the network learns to generate visually pleasing results. Let H and G indicate two common-size windows in the dehazed and clear images, respectively, centered at pixel x. We compute the means and standard deviations of H, G, as μ_H, μ_G, σ_H, and σ_G. The covariance of H and G is given by σ_{HG}. For pixel x, the SSIM is defined as:

$$SSIM(x) = \frac{2\mu_H\mu_G + C_1}{\mu_H^2 + \mu_G^2 + C_1} \cdot \frac{2\sigma_{HG} + C_2}{\sigma_H^2 + \sigma_G^2 + C_2} = l(x) \cdot cs(x) \tag{3.5}$$

where C_1, C_2 are variables to stabilize weak denominator. $l(\cdot)$ is the luminance, and $cs(\cdot)$ denotes contrast and structure measures.

$$L_{MS-SSIM}(x) = 1 - l_M^\alpha(x) \cdot \prod_{k=1}^{M} [cs_k(x)]^{\beta_k} \tag{3.6}$$

M denotes number of scale.

3.2.3 Perceptual loss

We use perceptual loss[22] to give extra supervision in high-level feature space. It is well known that training using perceptual loss helps the model reconstruct useful details more accurately. The perceptual loss function is defined as follows:

$$L_{perc} = \frac{1}{N} \sum_{k} \frac{1}{C_k H_k W_k} ||\phi_k(f_\theta(i)) - \phi_k(j)||_2^2 \tag{3.7}$$

where, $f_\theta(i)$ is dehazed image. j and i are ground truth and hazy images respectively, $\phi_k(\cdot)$ stands for feature map of size $C_k \times H_k \times W_k$.

3.2.4 Adversarial loss

Adversarial loss is said to aid in restoring photo-realistic images. Specifically for small-scaled datasets, the pixel-wised loss function often fails to give adequate supervised data to train a network for extracting photo-realistic features. We include adversarial loss[43] $L_{adv} = \sum_{k=1}^{N} -logD(f_\theta(i))$. Here, D is a discriminator and $D(f_\theta(i))$ is related to the probability of dehazed image $f_\theta(i)$ closer to the ground truth.

3.3 Experiments and Results

We begin by discussing the datasets, training details, and evaluation metrics in this section. Then, we compare our method quantitatively and qualitatively with state-of-the-art dehazing methods. Finally, to further understand the effects of the various modules, we undertake ablation studies.

3.3.1 Dataset for Training

We use two datasets: NTIRE2020[5] and NTIRE2021, for training our model. NTIRE2020 and NTIRE2021 contain 45 and 25, respectively for training. Both datasets contain five pairs of hazy and ground truth images for validation and testing. Ground truths of validation and testing of NTIRE2021 are not available. So we took five pairs of images from training data to test our model. We have 65 images for training, five images for validation and ten images for testing.

3.3.2 Training Details

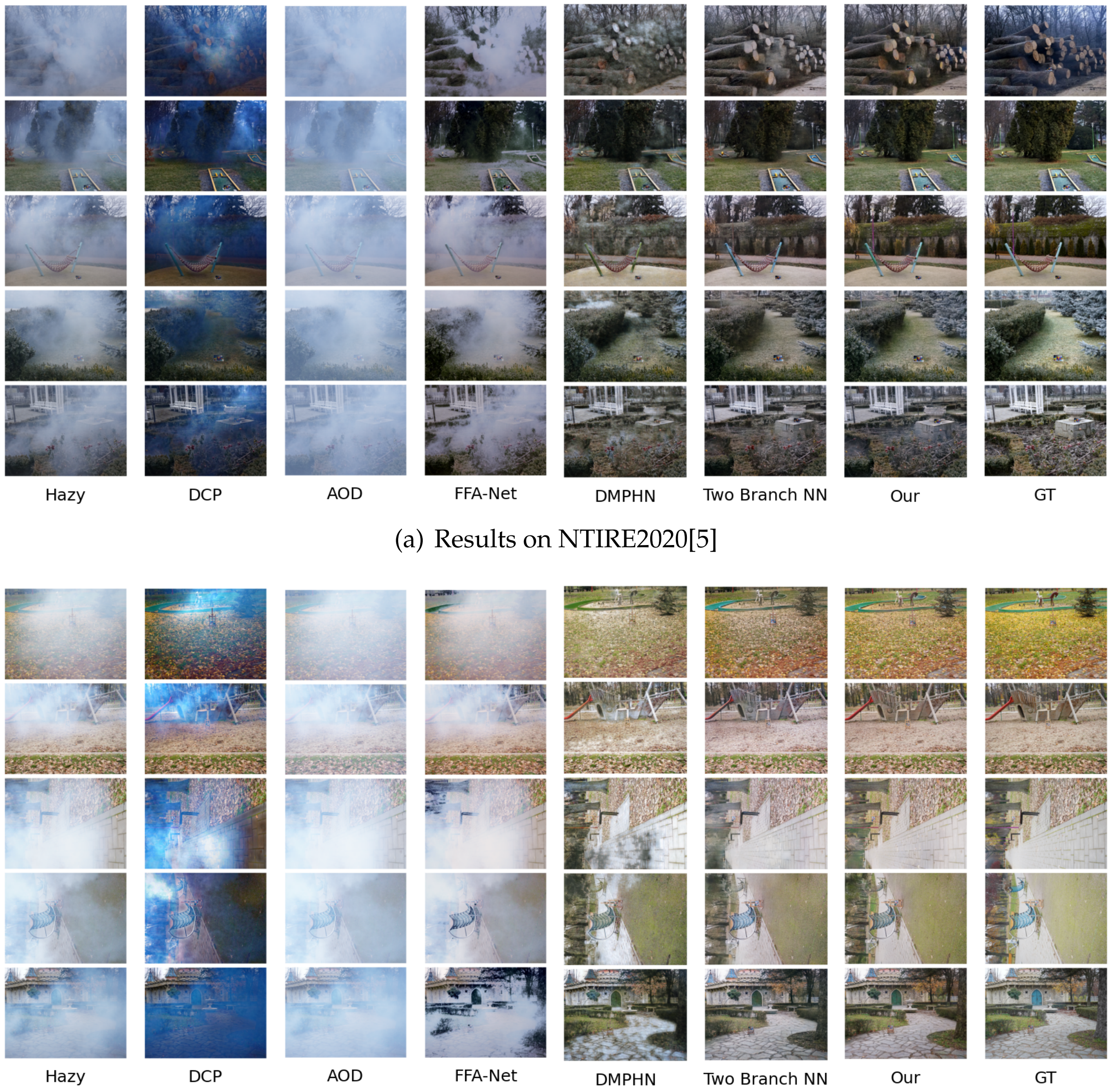
Since there are less number of images available, we flip the training images horizontally as well as vertically. Further, we considered rotating the images by 90, 180, and 270 degrees. We extract overlapping patches of size 256 × 256 from the images for training purpose.

We chose 0.0001 as the initial learning rate. Loss function hyper-parameters $\gamma 1, \gamma 2, \gamma 3$, and $\gamma 4$ are 1.0, 0.5, 0.01, and 0.0005, respectively. The Pytorch library is used to implement our technique. All of the tests are conducted using Nvidia

Tesla P100 GPUs. We use the peak signal-to-noise ratio (PSNR), and the structural similarity index (SSIM) measures for quantitative evaluation.

3.3.3 Comparison with state-of-the-art Methods

Here, we compare our method with state-of-the-art methods: DCP[20], AOD[23], FFA-Net[30], DMPHN[14] and Two Branch NN[40]. To compare our method with state-of-the-art methods, we train all methods with the same dataset of NTIRE2020 and NTIRE2021. The results of all experiments are shown in the Fig.3.4 One can observe that DCP generates dehazed results with color distortion. we can observe that DCP can generate blueish results and is also not able to remove haze. AOD produces undesirable results that we can clearly see in our figure. Unlike AOD, FFAnet removes haze in some images but most images lose their structure and edges. They produce an unacceptably dense color impression. Both methods work fine with homogeneous data but fail in the non-homogeneous dataset. When compared to the previous methods, DMPHN can remove haze from images, however, it cannot entirely remove haze from a hazy image.



(a) Results on NTIRE2020[5]

(b) Results on NTIRE2021

Figure 3.4: Qualitative comparison of NTIRE2020 and NTIRE2021 datasets

Our proposed method and Two branch NN can effectively remove haze from hazy images. When compared to our approach, however, two branch NN has low contrast and Some small regions contain haze. However, if we look closely at the results, we can see that our methods outperform the Two branch NN method (See Fig. 3.5), which follow channel attention and pixel attention in sequence. The improvements in visual results are reflected in the quantitative comparison in terms of PSNR and SSIM values, as can be observed in Table 3.1.

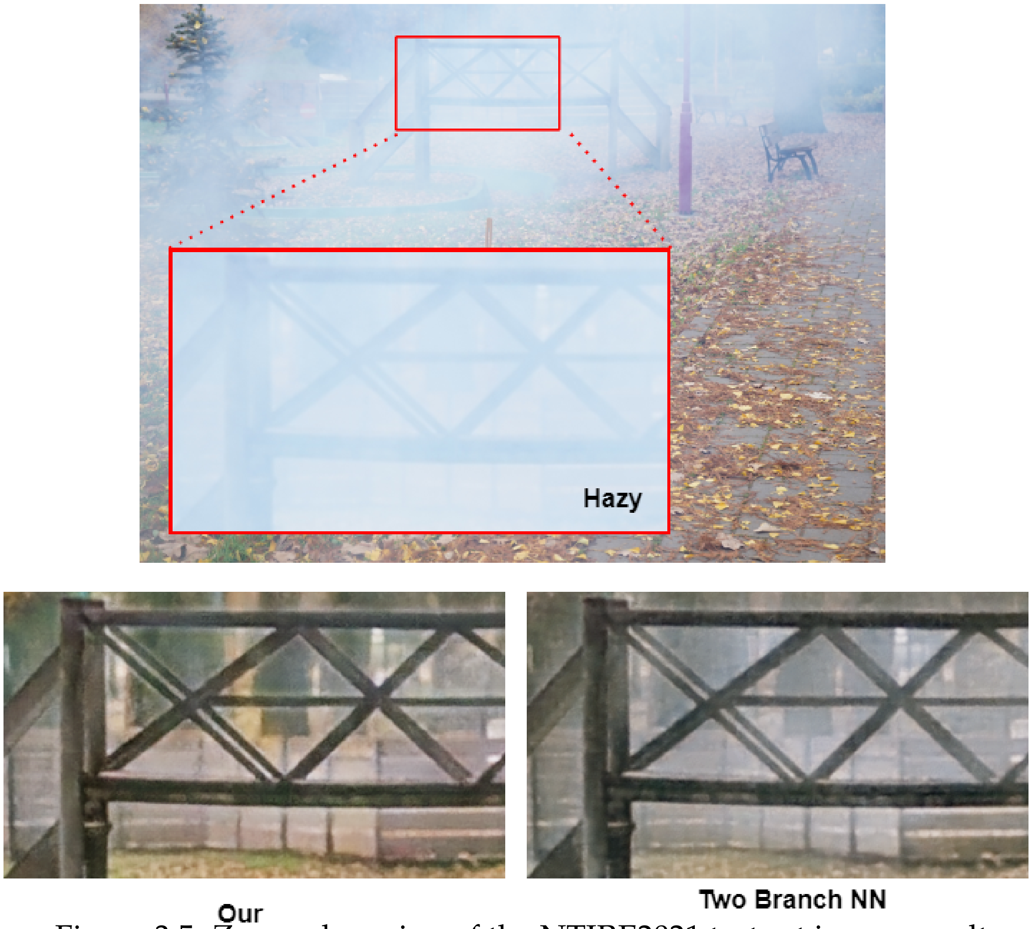


Figure 3.5: Zoomed version of the NTIRE2021 test set image result

Table 3.1: Quantitative comparison

Method	NTIRE2020		NTIRE2021	
	PSNR	*SSIM*	*PSNR*	*SSIM*
DCP[20]	13.26	0.494	11.67	0.71
AOD-net[23]	8.1	0.313	9.66	0.517
FFA[30]	13.74	0.507	12.44	0.624
DMPHN[14]	17.29	0.631	18.16	0.798
Two Branch NN[40]	19.3	0.682	21.09	0.856
Ours	**21.11**	**0.715**	**22.87**	**0.893**

3.3.4 Ablation Study

To analyze the effectiveness of the pre-trained model and parallel connection of channel attention(CA) and pixel attention(PA), we performed ablation study by considering these factors: two pre-trained models (ResNet and ConvNext) and the connection of CA and PA (serial and parallel). We conducted four experiments: i) using ResNet as pre-trained model and serial connection ii) using ResNet as pre-trained model and parallel connection iii) using ConvNext as pre-trained model and serial connection, and iv) using ConvNext as pre-trained model and parallel connection. One can observe in Table 3.2 that for fixed type of connection between CA and PA, ConvNext produces better results. On the other hand, when we fixed the pre-trained model, the CA||PA produces better results as compared to their serial interconnections.

Table 3.2: Ablation study: The best outcomes are highlighted in **bold**.

Method	NTIRE2020		NTIRE2021	
	PSNR	SSIM	PSNR	SSIM
ResNet + CA $\rightarrow PA$	19.3	0.682	21.09	0.856
ResNet + CA $\|\|PA$	20.78	0.702	22.12	0.879
ConvNext + CA $\rightarrow PA$	20.31	0.709	22.09	0.879
ConvNext + CA $\|\|PA$	**21.11**	**0.715**	**22.87**	**0.893**

We conducted one another ablation study for various loss function in order to compare the effectiveness of the various loss function combinations for our model. The results of the ablation study shown in the below Table 3.3.

Table 3.3: Ablation study: various combinations of the loss function

Loss function	**NTIRE2020**		**NTIRE2021**	
	PSNR	***SSIM***	***PSNR***	***SSIM***
L1 + 0.5 * ssim	20.70	0.713	22.75	0.889
L1 + 0.0005 * adv	20.95	0.711	22.31	0.878
L1 + 0.01 * perc	21.02	0.690	22.71	0.864
L1 + 0.5 * ssim + 0.0005 * adv	20.83	0.713	22.40	0.882
L1 + 0.5 * ssim + 0.01 * perc	20.72	0.703	22.77	0.863
L1 + 0.01 * perc + 0.0005 * adv	20.37	0.666	21.94	0.833
L1 + 0.5 * ssim + 0.01 * perc + 0.0005 * adv	**21.11**	**0.715**	**22.87**	**0.893**

Notation in Table : (L1 : Smooth L1 loss , ssim : Structure Similarity loss, perc: Perceptual loss , adv: Adversarial loss)

In the Table 3.3, we can observe that we get better results when we use all four losses than in other experiments. So we consider this loss as our final loss. If we try to change the hyper-parameters values, we have many possibilities. So for that reason, we tried two other experiments (two times increase and two times decrease all parameter values). The results of these two experiments are shown in the below Table 3.4 .

The Table 3.4 demonstrates that we achieved better results when we set our

Table 3.4: Ablation study: modified weight of the four losses

Loss function	**NTIRE2020**		**NTIRE2021**	
	PSNR	***SSIM***	***PSNR***	***SSIM***
L1 + 1 * ssim + 0.02 * perc + 0.0010 * adv	20.46	0.699	22.68	0.864
L1 + 0.25 * ssim + 0.005 * perc + 0.00025 * adv	20.77	0.701	22.86	0.861
L1 + 0.5 * ssim + 0.01 * perc + 0.0005 * adv	**21.11**	**0.715**	**22.87**	**0.893**

loss function's hyper-parameters $\gamma_1, \gamma_2, \gamma_3$, and γ_4 to 1.0, 0.5, 0.01, and 0.0005 respectively.

3.3.5 Real-world images and results

During fumigation on the campus of DA-IICT, we are taking some images from the cafeteria area. Captured images show non-homogeneous nature because smoke is unevenly distributed over the scene. Fig.3.6 shows an example of captured photos. In Fig.3.6, the first column shows captured hazy images, and the second column shows our model output. So, we test this image in our model. We can observe that our algorithm can remove some haze levels but not wholly remove haze from images. One reason behind this is that the image is captured in the presence of sunlight, so sunlight is additionally added. Another reason is that we are training our model on the smoggy data set, and here this real-world image contains smoke. However, smog and smoke have different light scattering characteristics. That's why they fail to handle smoggy images.

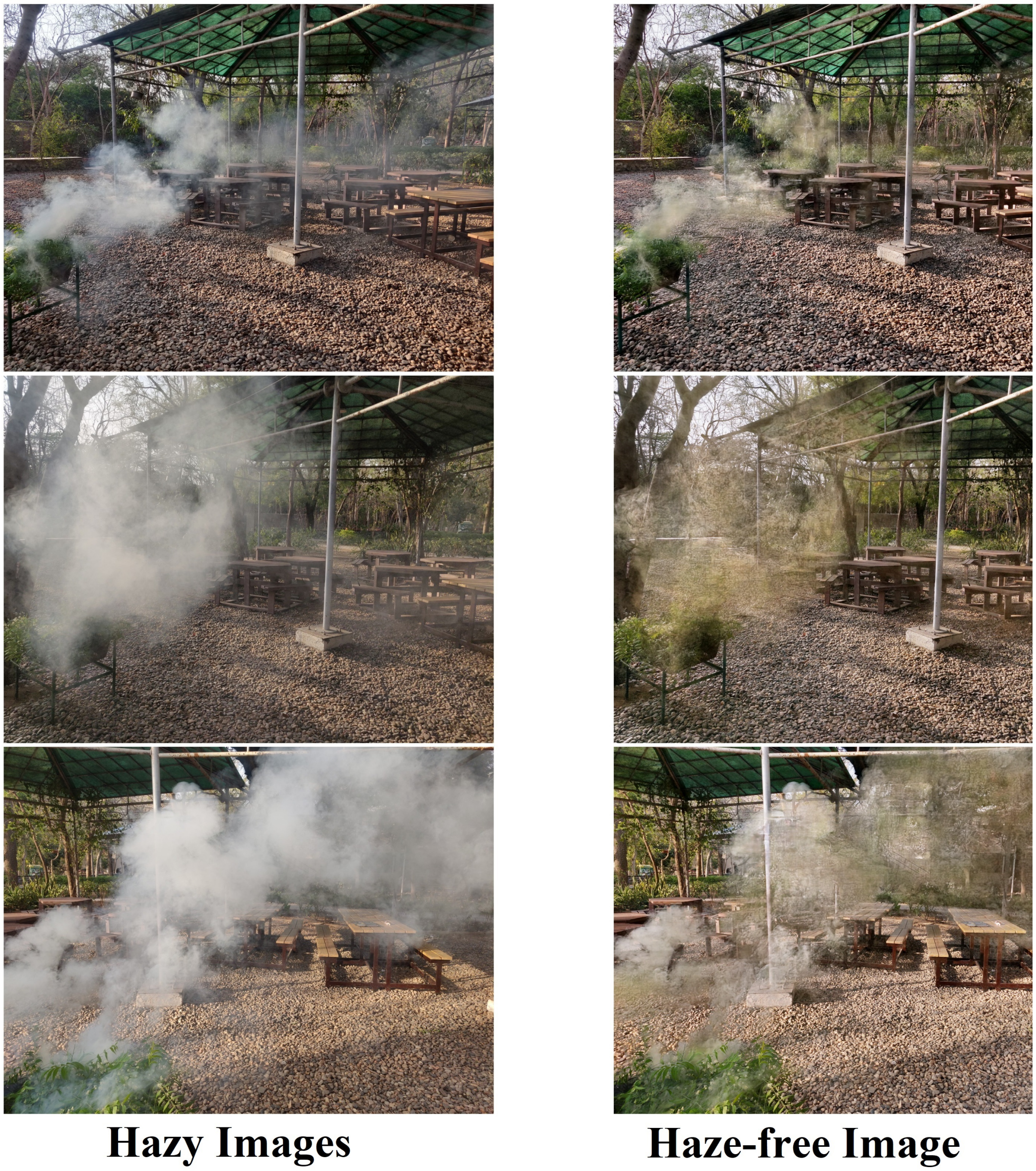


Figure 3.6: Qualitative results of real world non-homogeneous image

3.3.6 Other Experiments

In other experiments, we used a different method and added attention mechanism to check the effectiveness of the method. We conducted four other experiments: i) (ResNet + RCAB + Attention): used Pre-trained ResNet as encoder and RCAB with parallel attention in the decoder part ii)(Simple UNet): used Simple UNet architecture iii)(UNet + Attention): used UNet architecture with parallel attention in the decoder part iv)(UNet + Attention) + (RCAB) : used UNet Architecture and attention mechanism in first branch and RCAB in second branch.

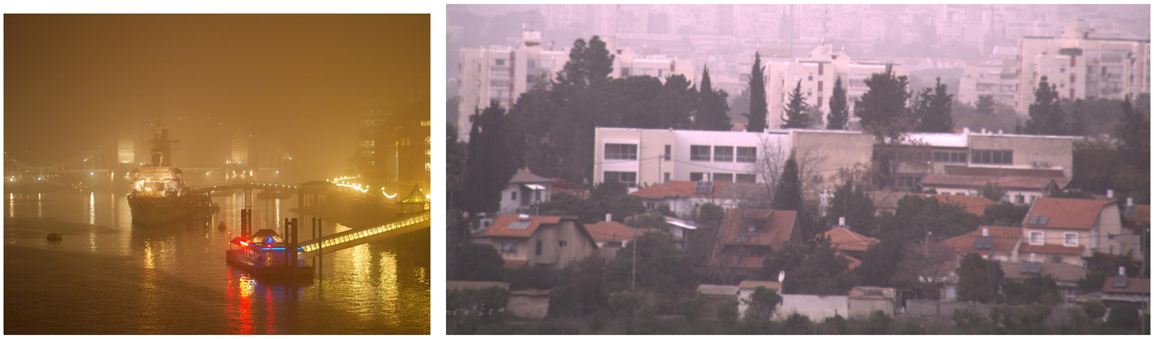
Experiment ii and iii in Table 3.5 demonstrate that by applying the attention mechanism to UNet, we achieve a significant improvement in results over the simple UNet model. This shows the importance of the attention mechanism in boosting the performance. However, if we use the two-branch network with UNet architecture without pre-trained weights, we don't see much improvement in the results as shown in Table 3.5. This shows the effectiveness of pre-trained weights in improving the performance.

Table 3.5: Result of other Experiments : The best outcomes are highlighted in **bold**.

Method	NTIRE2020		NTIRE2021	
	PSNR	SSIM	PSNR	SSIM
(ResNet + RCAB + Attention)	**19.19**	0.599	**19.29**	0.725
(Simple Unet)	17.53	0.657	17.88	0.830
(Unet + Attention)	18.22	**0.673**	18.90	0.855
(Unet + Attention) + (RCAB)	18.351	0.664	18.885	**0.857**

Chapter 4
Vari-color Image Dehazing

Due to poor weather, the quality of images captured in the presence of atmospheric particles is frequently hampered by low visibility and chromatic casts. In Fig 4.1, we can see an example of this type of image. Most methods can focus on improving visibility, but existing image dehazing methods fail to restore color balance from colored hazy images. In this chapter, we will discuss networks that concentrate on restoring color and improving hazy images' visibility.



((a)) ((b))

Figure 4.1: Example of multi-color hazy images

4.1 Network Structure

The network structure is divided into two parts 1) Restoring Color (RC) Module and 2) Enhancement of Visibility(EV) Module. Network structure is as shown in fig 4.2.

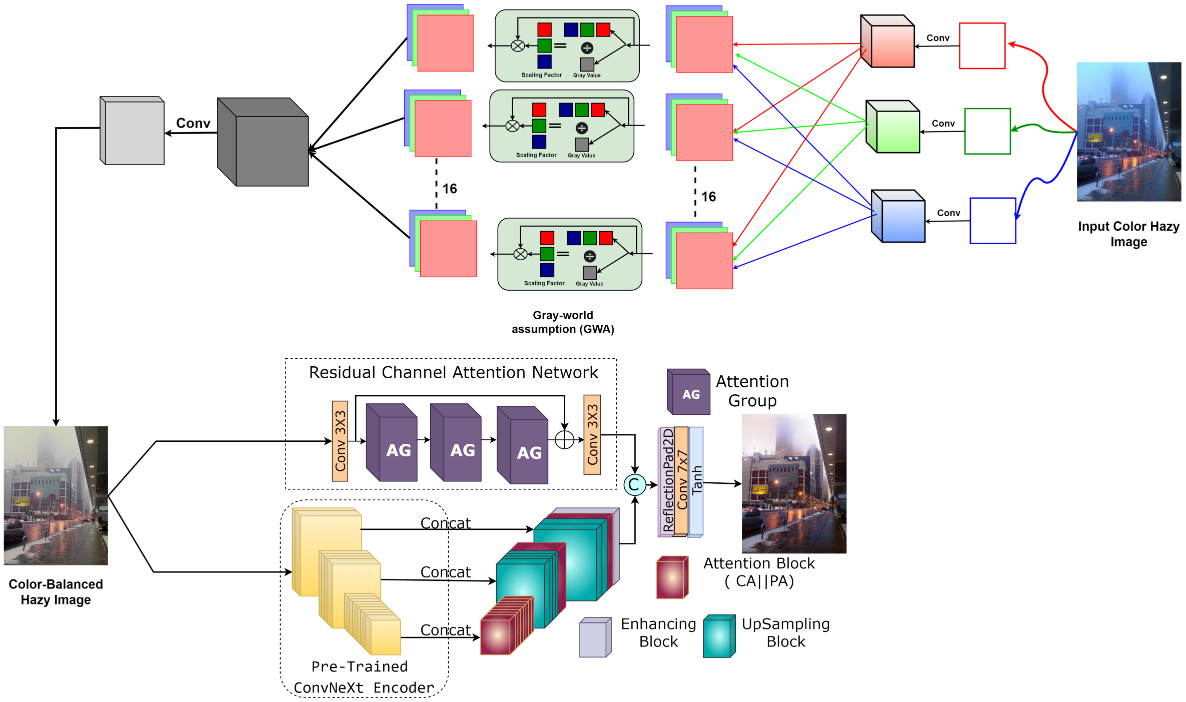


Figure 4.2: Architecture of our vari-color image dehazing

4.1.1 Restoring Color(RC) Module

As shown in Fig 4.1, we can see an image dominated by a particular color cast. Gray-world assumption (GWA)[31] has been used in previous literature[2, 21, 3, 33] to restore color balance in a multi-colored hazy image. According to GWA, The average radiance of the scene is low contrast. One of the easiest ways to preserve color balance in an image is to scale the pixel values by a factor (avgG/ avgC), where avgG is a global average of an image and avgC represents the channel-wise average. In the RC network, hazy input image (I) passes through independent convolution with each channel to get different color maps C_p.

$$C_P^i = \{f_i^{3\times3}(I^R);\;\; f_i^{3\times3}(I^G);\;\; f_i^{3\times3}(I^B)\} \tag{4.1}$$

Here, $f^{3\times3}$ represents the convolution with window size 3×3, R, G, B represents color channels, i describes the number of convolution filters used, and i $\in$\{1:16\}. We applied color correction on each color map's C_p to find improved feature mapsF_r.

$$F_r^i(x) = C_p^c(x) \times \frac{G_p}{I_p(c)};\;\; c \in \{r, g, b\} \tag{4.2}$$

Where x indicates the pixel location in the image, G_p and $I_p(c)$ represents the image's global average and a channel-wise average of each RGB channel of the image. After finding color maps, we concatenated all improved feature maps F_r together, along with the channel axis, followed by two convolution blocks to create the color-balanced hazy image.

4.1.2 Enhancing Visibility(EV) Module

Inspired by the varicolored[15] method, we are using our proposed network, which contains the ConvNeXt pre-Trained model and Attention mechanism for enhancing image visibility. As we see in Fig.4.2 we use our two branch network as an Enhancing module. The restoring color module's output is considered the input of the EV module. In terms of network structure, we have not modified the EV module.

4.2 Loss Function

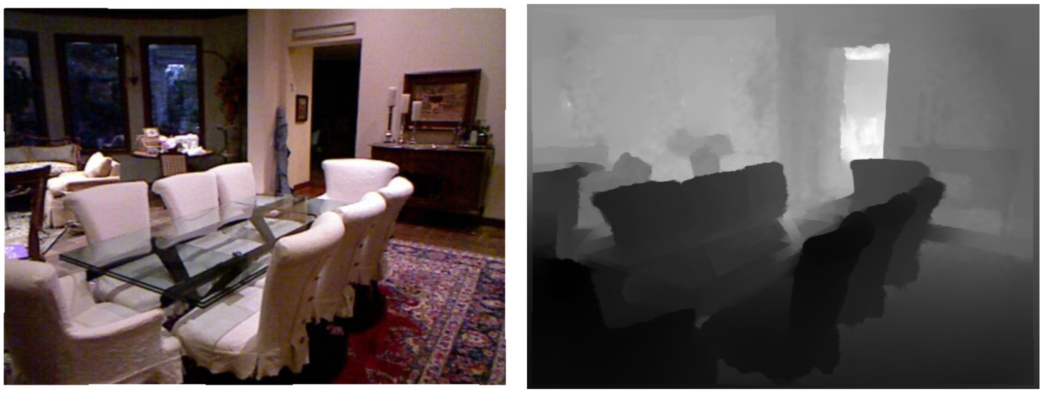
$$L = \gamma_1 L_{l1} + \gamma_2 L_{MS-SSIM} + \gamma_3 L_{perc} + \gamma_4 L_{Smooth-l1} + \gamma_5 L_{MS-SSIM_c}. \tag{4.3}$$

L comprises of five different components: i) L1 loss(L_1) , ii) structural similarity loss ($L_{MS-SSIM}$), iii) perceptual loss (L_{perc}), iv) smooth L1 loss ($L_{Smooth-l1}$) and v)structural similarity loss ($L_{MS-SSIM_c}$). Hyper-parameters $\gamma_1, \gamma_2, \gamma_3, \gamma_4$ and γ_5 are used to balance losses. Here, i),ii) & iii) losses is for reconstructing Haze-free image and vi) & v) losses is for reconstructing color balanced hazy images. We already discussed the specifics of these losses in section 3.2.

4.3 Experiments and Results

4.3.1 Dataset

In the available dataset for the dehazing task, we don't have any dataset that contains different colored hazy images with its ground truth. Even very few proposed methods can take care of this chromatic cast effect. To handle this chromatic cast effect, we need a dataset containing colored hazy images with ground truth values. As discussed in Chapter 1, the atmospheric scattering model (ASM) depends on two parameters 1) scene depth and 2)atmospheric light intensity. That means we need a dataset that contains depth information about images. The NYU-Depth[35] dataset contains 1449 indoor images with depth information. A sample of the dataset is shown in Fig 4.3.



((a)) Original Image ((b)) Depth Image

Figure 4.3: Sample image of NYU-Depth[35] dataset

We can use depth information from this dataset, and then we need an Atmospheric light value. Ten colours that are commonly seen in an atmospheric environment are considered for atmospheric light value. The color-set that we used in our dataset is shown in Fig4.4.

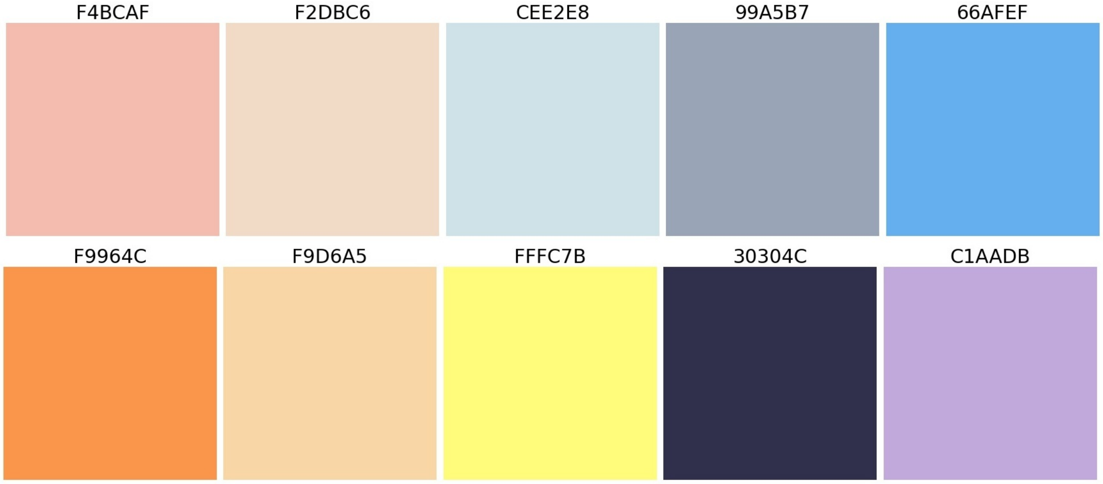


Figure 4.4: The colours we employed for atmospheric lighting

Let's consider ASM as follows.

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

In the above eq, we already have $J(x)$ and A, which are ground-truth values and atmospheric light values respectively.

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

Here we already have depth information about the scene and we know that β is used for the density of haze. We consider three different β values, which are 1, 3, and 5. Now we put all parameters A, $J(x)$, $d(x)$, and β values, so we got hazy images which are $I(x)$. We selected random 450 images from NYU-Depth[35] dataset for creating the dataset. For each image, we have ten color values and three beta values. So we have 450 * 10 * 3 = 13500 images for training our network. We can see the generated sample dataset in Fig. 4.5. We chose colour value #CCCCCC to create ground truth for colour balanced hazy images.

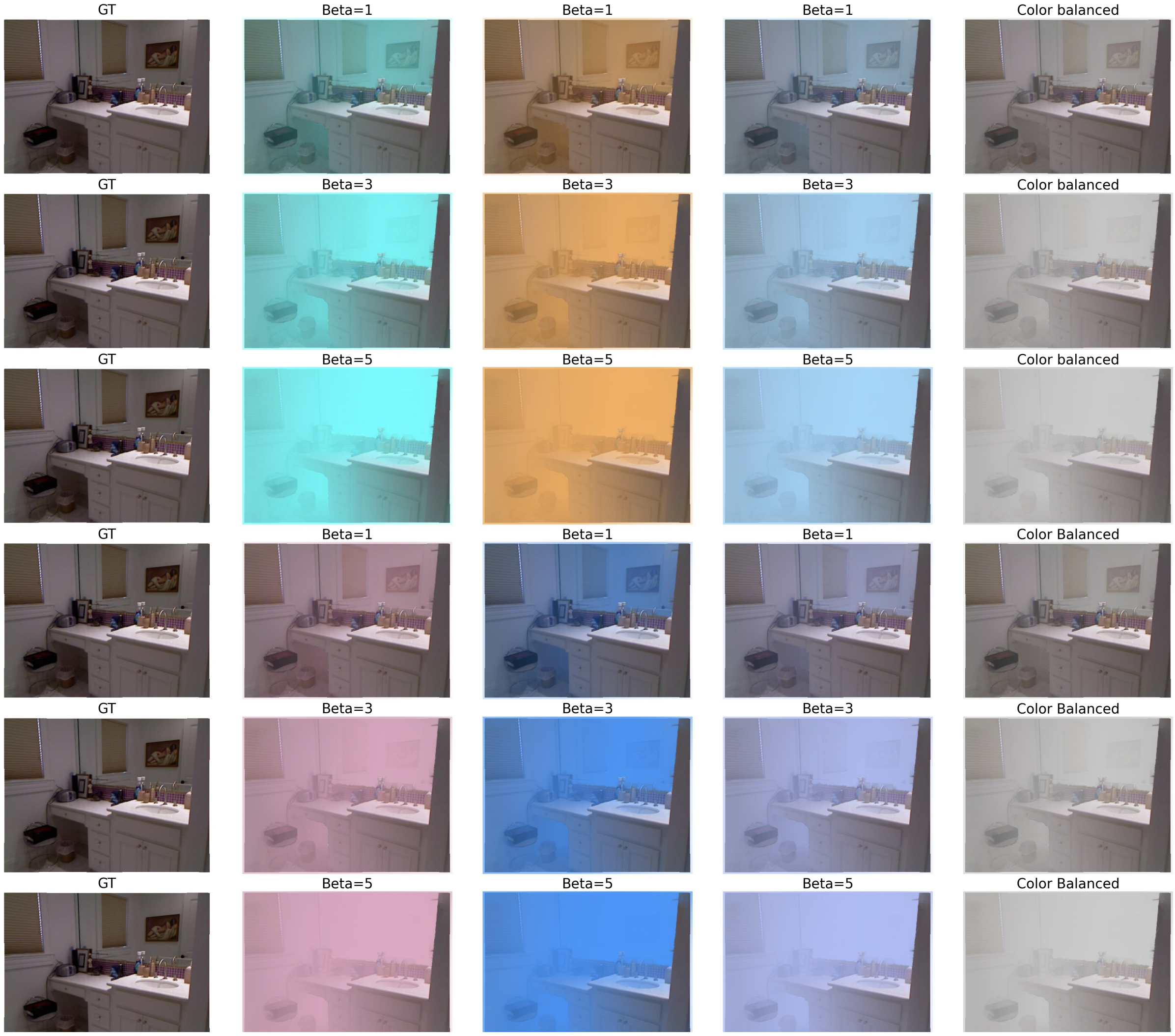


Figure 4.5: Sample of our training dataset

4.3.2 Training Details

As shown in Fig 4.2, we passed a colored hazy image to the Restoring Color(RC) Module, which produces a color-balanced hazy image. The color-balanced hazy image is then passed to the Enhancing Visibility(EV) Module, which produces the final haze-free images. We provided the ground truth of a color-balanced and haze-free image here. Both modules' output is used for backpropagation. Moreover, we extracted randomly cropped patches of size 256 × 256 from the images for training purposes.

We chose 0.0001 as the initial learning rate. Loss function hyper-parameters $\gamma 1, \gamma 2, \gamma 3, \gamma 4$ and $\gamma 5$ are 1.0, 0.7, 0.1,1.0 and 0.75, respectively. The Pytorch library is used to implement our technique. All of the tests are conducted using Nvidia Tesla P100 GPUs. We use the peak signal-to-noise ratio (PSNR), and the structural similarity index (SSIM) measures for quantitative evaluation.

4.3.3 Comparison with state-of-art Methods

We compare our multi-color dehazing method with state-of-the-art methods: TPAMI-11 [20],ECCV-16 [32],ICCV-17[23],NIPS-18[39],CVPRW-18[17],CVPRW-19 [16] and Varicolored[15], We train our model with the NYU Depth[35] dataset, which is also used in the varicolored method. As a result, we can directly take the results of other methods from the varicolored[15] paper.

Table 4.1: Quantitative comparison on D-Hazy[1]

Method	**PSNR**	**SSIM**
TPAMI-11 [20]	12.5876	0.7060
ECCV-16 [32]	12.8203	0.7231
ICCV-17[23]	12.4110	0.7177
NIPS-18[39]	15.5456	0.7726
CVPRW-18[17]	15.4130	0.6490
CVPRW-19 [16]	18.8167	0.8179
Varicolored[15]	23.3142	0.8901
Ours	**24.81**	**0.916**

The D-Hazy[1] dataset contains 1449 hazy images with their ground truth value. The D-Hazy dataset is a synthetic dataset constructed by adjusting atmospheric light A and attenuation coefficient (β). Table 4.1 displays the results of other methods as well as ours. The table 4.1 clearly shows that our method achieves significantly better results than other existing techniques.

Table 4.2: Quantitative comparison on SOTS indoor[24]

Method	**PSNR**	**SSIM**
TPAMI-11 [20]	16.6215	0.8179
ECCV-16 [32]	17.5731	0.8102
ICCV-17 [23]	19.0868	0.8512
CVPRW-19 [16]	21.5600	0.8600
Varicolored[15]	28.2688	0.9511
Ours	**28.51**	**0.968**

In contrast to D-hazy, the SOTS-indoor [24] dataset is made up of 500 synthetic images of indoor scenes with varying attenuation coefficients (beta). All 500 images were considered for evaluation. The quantitative results of existing techniques and our method are shown in Table 4.2. We can see from table 4.2 that our method outperformed existing methods.

4.3.4 Real World Multi-colored Image Testing

Visual testing was performed on real-world hazy images. We have three columns, as shown in Fig. 4.6. The first column contains the colored input image, the second column contains the output of the Restoring Color (RC) Module, which is a color balanced hazy image, and the third column contains the output of the Enhancing Visibility (EV) Module, which is our haze-free image. The images in Fig 4.6 have various haze colors such as yellow, blue, orange, and grey. Many previous techniques can remove haze, but they cannot address the issue of color restoration. In contrast, our method performs both the color restoration and the haze removal tasks. By looking at Fig 4.6, we can see that our model removes this color from the image and also performs well in the haze removal task.

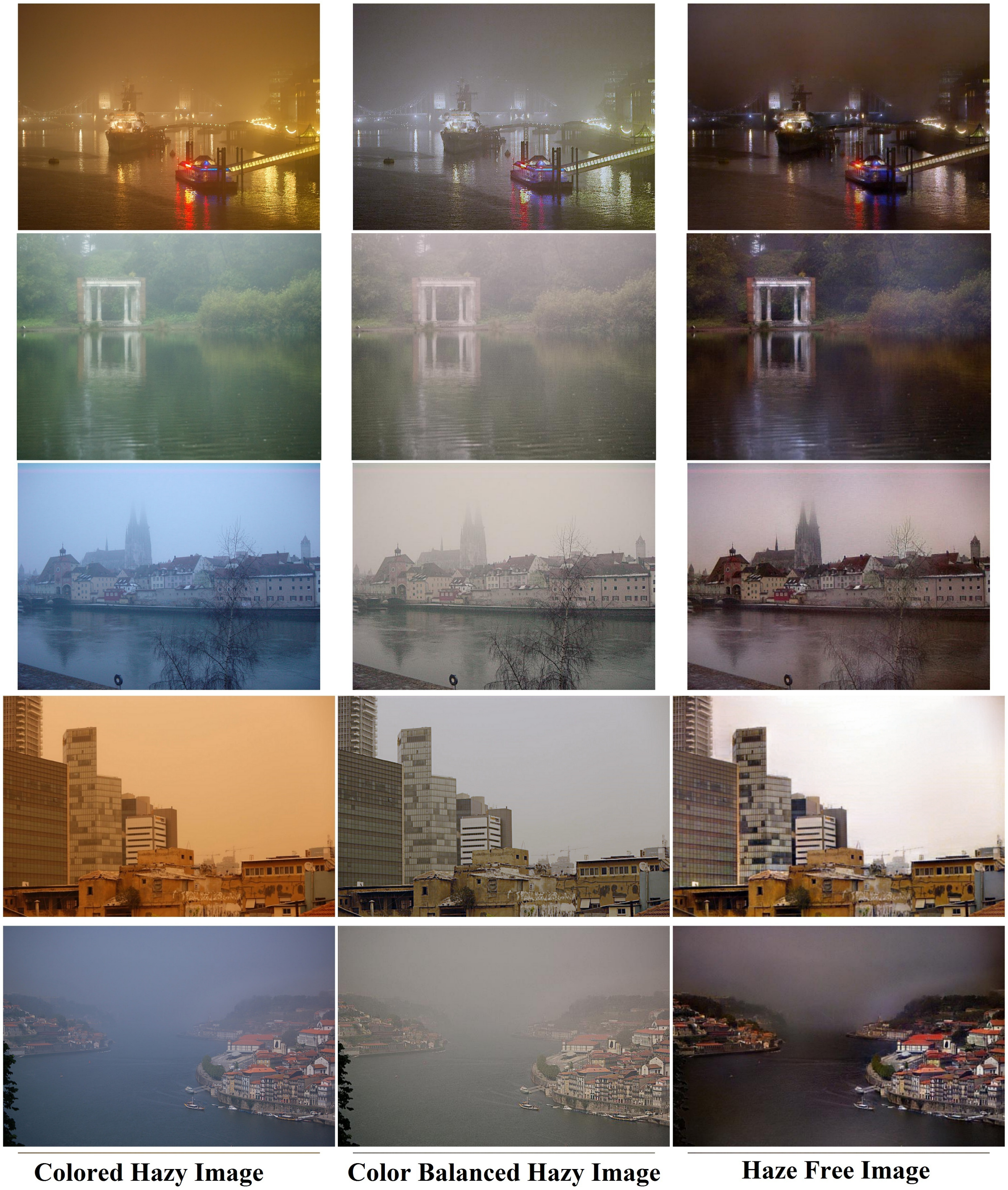


Figure 4.6: Qualitative results of real world colored image

CHAPTER 5
Conclusions

In this thesis, we have proposed a two-stream deep architecture to remove non-homogeneous haze from image, based on attention mechanism. In the first stream, we have used pre-trained ConvNext model along with attention mechanism. The pre-trained ConvNext model extracts significant and robust feature from the data. On the output of ConvNext, we have used channel attention in parallel to the pixel attention. The parallel interconnection enables our network to learn about the non-homogeneous nature of haze at pixel as well as in channel levels, independently. We have concatenated the learned features with the output of a second stream, which is residual channel attention network. The second stream extracts information directly from the data. Further, the residual connection enables a smoother information flow towards the output. The produced results emphasize the importance of attention mechanism when they are interconnected in parallel. We also proposed multi-color image dehazing to address color casting issues. We conclude from the results of real-world colored images that our module effectively handles chromatic cast issues as well as image dehazing tasks.

References

[1] C. Ancuti, C. O. Ancuti, and C. De Vleeschouwer. D-hazy: A dataset to evaluate quantitatively dehazing algorithms. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 2226–2230, 2016.

[2] C. O. Ancuti and C. Ancuti. Single image dehazing by multi-scale fusion. *IEEE Transactions on Image Processing*, 22(8):3271–3282, 2013.

[3] C. O. Ancuti, C. Ancuti, and P. Bekaert. Effective single image dehazing by fusion. In *2010 IEEE international conference on image processing*, pages 3541–3544. IEEE, 2010.

[4] 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.

[5] 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/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 444–445, 2020.

[6] C. O. Ancuti, C. Ancuti, R. Timofte, and C. De Vleeschouwer. O-haze: A dehazing benchmark with real hazy and haze-free outdoor images. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 867–8678, 2018.

[7] C. O. Ancuti, C. Ancuti, R. Timofte, L. V. Gool, L. Zhang, and M.-H. Yang. Ntire 2019 image dehazing challenge report. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, IEEE CVPR 2019, 2019.

[8] C. O. Ancuti, C. Ancuti, R. Timofte, and C. D. Vleeschouwer. I-haze: a dehazing benchmark with real hazy and haze-free indoor images. In *arXiv:1804.05091v1*, 2018.

[9] C. O. Ancuti, C. Ancuti, F.-A. Vasluianu, and R. Timofte. Ntire 2020 challenge on nonhomogeneous dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 490–491, 2020.

[10] C. O. Ancuti, C. Ancuti, F.-A. Vasluianu, and R. Timofte. Ntire 2021 nonhomogeneous dehazing challenge report. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 627–646, 2021.

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

[12] D. Berman, T. Treibitz, and S. Avidan. Non-local image dehazing. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1674–1682, June 2016.

[13] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016.

[14] S. D. Das and S. Dutta. Fast deep multi-patch hierarchical network for nonhomogeneous image dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 482–483, 2020.

[15] A. Dudhane, K. M. Biradar, P. W. Patil, P. Hambarde, and S. Murala. Varicolored image de-hazing. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4564–4573, 2020.

[16] A. Dudhane, P. W. Patil, and S. Murala. An end-to-end network for image de-hazing and beyond. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2020.

[17] D. Engin, A. Genç, and H. Kemal Ekenel. Cycle-dehaze: Enhanced cyclegan for single image dehazing. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 825–833, 2018.

[18] R. Fattal. Single image dehazing. In *ACM SIGGRAPH 2008 Papers*, SIGGRAPH ’08, pages 72:1–72:9, New York, NY, USA, 2008. ACM.

[19] R. Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[20] 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, 2010.

[21] S.-C. Huang, B.-H. Chen, and W.-J. Wang. Visibility restoration of single hazy images captured in real-world weather conditions. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(10):1814–1824, 2014.

[22] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[23] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, pages 4770–4778, 2017.

[24] 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, 2018.

[25] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie. A convnet for the 2020s. *arXiv preprint arXiv:2201.03545*, 2022.

[26] S. Mandal and A. N. Rajagopalan. Local proximity for enhanced visibility in haze. *IEEE Transactions on Image Processing*, 29:2478–2491, 2020.

[27] E. J. Mccartney and F. F. Hall. Optics of the atmosphere: Scattering by molecules and particles. *Physics Today*, 30:76–77, 1976.

[28] S. Narasimhan and S. Nayar. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):713–724, 2003.

[29] S. K. Nayar and S. G. Narasimhan. Vision in bad weather. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 820–827. IEEE, 1999.

[30] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia. Ffa-net: Feature fusion attention network for single image dehazing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11908–11915, 2020.

[31] E. Reinhard, M. Adhikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer graphics and applications*, 21(5):34–41, 2001.

[32] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang. Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*, pages 154–169. Springer, 2016.

[33] W. Ren, L. Ma, J. Zhang, J. Pan, X. Cao, W. Liu, and M.-H. Yang. Gated fusion network for single image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2018.

[34] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.

[35] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *European conference on computer vision*, pages 746–760. Springer, 2012.

[36] R. T. Tan. Visibility in bad weather from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2008.

[37] J.-P. Tarel, N. Hautiere, 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, 2012.

[38] H. Wu, J. Liu, Y. Xie, Y. Qu, and L. Ma. Knowledge transfer dehazing network for nonhomogeneous dehazing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 478–479, 2020.

[39] X. Yang, Z. Xu, and J. Luo. Towards perceptual image dehazing by physics-based disentanglement and adversarial training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[40] Y. Yu, H. Liu, M. Fu, J. Chen, X. Wang, and K. Wang. A two-branch neural network for non-homogeneous dehazing via ensemble learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 193–202, 2021.

[41] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.

[42] H. Zhao, O. Gallo, I. Frosio, and J. Kautz. Loss functions for image restoration with neural networks. *IEEE Transactions on computational imaging*, 3(1):47–57, 2016.

[43] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.