Orthogonal Transform based Generative Adversarial Network for Image Dehazing

by

Mantra H Sanathra
202011041

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, 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.

Mantra H. Sanathra

Certificate

This is to certify that the thesis work entitled "Orthogonal Transform based Generative Adversarial Network for Image Dehazing" has been carried out by **Mantra H Sanathra (202011041)** for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under our supervision.

Dr. Ahlad Kumar
Thesis Supervisor

Dr. Manish Khare
Thesis Co-Supervisor

Acknowledgments

I would like to thank my supervisors, Dr. Ahlad Kumar and Dr. Manish Khare for their enthusiasm, patience, insightful remarks, valuable information, practical guidance, and never-ending ideas, which have helped me considerably during my research. I was able to successfully accomplish this research thanks to their vast knowledge, extensive experience, and professional expertise in Machine Learning. The thesis would not have been possible without their assistance and direction. I'd also like to express my gratitude to the institute as a whole for creating a competitive and supportive environment in which I was able to produce excellent research results.

I would like to express my gratitude to my family and friends for their constant support during these challenging times.

Contents

Abstract **v**

List of Tables **vi**

List of Figures **vii**

1 Introduction **1**

2 Related Work **3**

2.1 Haze formation formula 3

2.2 Based on prior knowledge 4

2.3 Based on learning 5

3 Motivation **7**

3.1 Krawtchouk Moments 7

3.1.1 Computation of Krawtchouk Moments 7

3.1.2 Representation in Matrix Form 8

3.1.3 Basis function of Krawtchouk Moments 9

3.1.4 Basis Ordering and its Importance 10

4 Proposed Method **12**

4.1 Architecture Structure 12

4.1.1 Colour Space Transformation: RGB to YCbCr 12

4.1.2 Krawcthouk Convolution Layer (KCL) 15

4.1.3 Frequency Cube (f^Y) 15

4.1.4 Architecture for f^Y_{low} 15

4.1.5 Architecture for f^Y_{high} 16

4.1.6 Inverse Krawtchouk Convolution Layer (IKCL) 17

4.1.7 Discriminator 18

4.1.8 Colour Space Transformation: YCbCr to RGB 18

5 Experimental Work 19

5.1 Datasets . . . 19

5.2 Loss Functions . . . 19

5.2.1 VGG Loss . . . 20

5.2.2 Smooth L_1 Loss . . . 21

5.2.3 GAN Loss . . . 21

5.3 Implementation Details . . . 21

5.4 Optimization of IKCL . . . 22

5.5 Qualitative and Quantitative analysis . . . 23

5.6 Threshold analysis . . . 27

6 Limitation of our Model 28

7 Conclusions 29

References 30

Abstract

Image dehazing has become one of the crucial preprocessing steps for any computer vision task. Most dehazing methods work in the image domain, and the dehazed image is obtained by estimating the transmission map along with global atmospheric light. In this thesis, we present a novel end-to-end architecture for estimating dehazed image in the Krawtchouk transform domain. For this a customized Krawtchouk Convolution Layer (KCL) in the architecture is added. KCL is constructed using Krawtchouk basis functions which converts the image from the spatial domain to the Krawtchouk transform domain. At the end of the architecture, another convolution layer called Inverse Krawtchouk Convolution Layer (IKCL) is introduced which converts the image back to the spatial domain from the transform domain. It has been observed that the haze is primarily present in lower frequencies of hazy images. Krawtchouk transform helps to analyze the high and low frequencies of the images separately. We have divided our architecture into two branches, the upper branch deals with the higher frequencies while the lower branch deals with the lower frequencies of the image. The lower branch is made deeper in terms of the layers as compared to the upper branch to address the haze present in the lower frequencies. When compared to current state-of-the-art methods, we were able to get competitive results using the proposed Orthogonal Transform based Generative Adversarial Network (OTGAN) architecture for image dehazing.

List of Tables

5.1 PSNR/SSIM Score 23
5.2 NIQE Score 26

List of Figures

1.1 Example of Image Dehazing 1

2.1 Haze Formation Model 4

3.1 Krawtchouk Basis 9

3.2 Basis Ordering 10

3.3 Analysis of Krawtchouk Coefficients 11

4.1 Architecture of OTGAN 13

4.2 Image analysis in $YCbCr$ color space 14

4.3 Frequency Partitioning 16

4.4 Architecture for lower branch (f^{Y}_{low}) 17

5.1 Samples of real-world hazy images 20

5.2 Optimized filters of $IKCL$ 22

5.3 Qualitative comparison of various methods on SOTS-outdoor dataset 24

5.4 Qualitative comparison of various methods on real-world images . 25

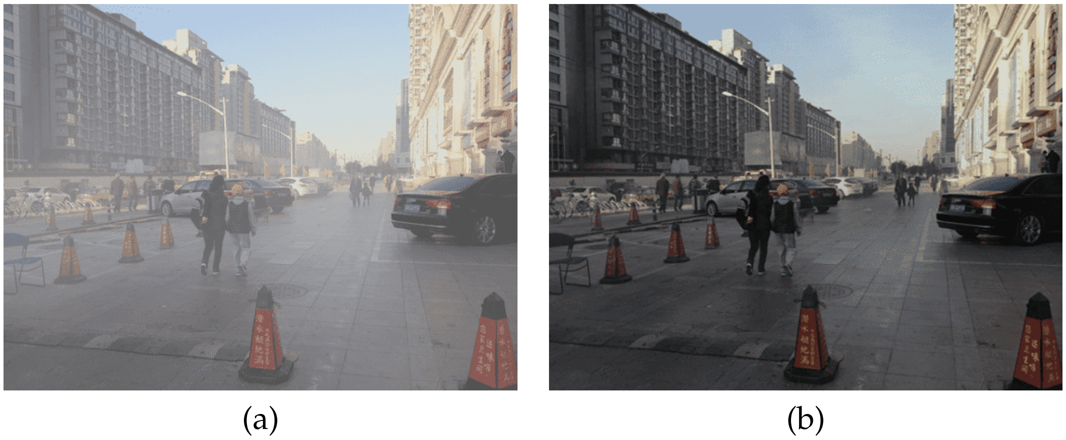
5.5 Thresold Analysis 27

6.1 Failure cases 28

CHAPTER 1

Introduction

Generally it is difficult to capture a clear photo, especially in winter season. Some amount of fog or haze is present in the atmosphere and we do not have camera sensors that can directly remove this haze to overcome this problem. Haze is a natural occurrence that reduces the image quality acquired by the camera. This is due to small particles in the atmosphere, such as dust, water droplets, and fog, which absorb and scatter light. Image dehazing (Figure 1.1) is a method for recovering a haze-free image from a hazy image in order to solve this problem. Computer vision tasks such as object detection [1], traffic surveillance, object tracking [2] require a haze-free image to perform at their best potential. As a result, haze removal becomes an important step in the preprocessing of high-level computer vision tasks.



(a) (b)

Figure 1.1: (a) Hazy Image (b) Clear Image

Earlier in [3–7], to restore the haze-free image, researchers took multiple images of the same scene. Multiple images of same scene are captured in different weather conditions but it is not always possible to get multiple images of the same scene, this encouraged them to use a single image for image dehazing. To address the image dehazing task utilising a single image, different approaches have been proposed.

Two types of image dehazing techniques exist: (a) based on prior knowledge, (b) based on learning. The first one uses characteristic differences like brightness, contrast, saturation between hazy and haze-free images and utilizes this knowledge to obtain a haze-free image. But not all images show the same characteristics which lead to some artifacts (color distortion) that makes the dehazed image look unrealistic. On the other hand, learning-based methods extract these characteristics automatically using some learning model.

In this thesis, an orthogonal transform based Generative Adversarial network (OTGAN) is proposed for image dehazing. The key aspects of the thesis are mentioned below:

- GAN based deep learning architecture for image dehazing is introduced in orthogonal transform domain. Krawtchouk moments converts the images from spatial domain to Krawtchouk domain. The architecture is trained to find the difference between the Krawtchouk coefficients of hazy image and haze-free image.

- Two custom convolution layers are designed consisting of Krawtchouk basis which are used to convert image in-between spatial domain and Krawtchouk domain; one of them is Krawtchouk Convolution Layer (KCL) used for forward transform and other Inverse Krawtchouk Convolution Layer ($IKCL$) for inverse transform. Weights of KCL are kept fixed and non-trainable, while weights of $IKCL$ are kept trainable for better adaptivity of the basis functions to the dataset.

- The proposed architecture has two branches; the upper branch consists of simple U-Net architecture, which deals with the high frequencies and the lower branch consist of pyramidal architecture that deals with the low frequencies present in the image

- Images used for training are transformed from RGB to $YCbCr$ color system, whereby only the Y channel is passed through the architecture.

CHAPTER 2

Related Work

Many researchers have proposed prior-based and learning-based approaches for image dehazing. This chapter discusses image dehazing approaches that either rely on prior information or use learning-based methods.

2.1 Haze formation formula

Figure 2.1 shows the haze formation model. The atmospheric scattering model [8] defines the haze formation model as

$$I(x) = R(x)t(x) + A(1 - t(x)) \tag{2.1}$$

Here, I stands for the image captured by the lens, R stands for the haze-free image that we are trying to recover, t stands for transmission map, which denotes the amount of light captured by the camera without any dispersion, and A stands for the global atmospheric airlight, x represents the index of pixels in the hazy image. Transmission map $t(x)$ is dependent upon the distance between camera lens and object and is calculated as

$$t(x) = e^{-\beta d(x)} \tag{2.2}$$

The distance between the object and the camera lens is denoted by $d(x)$. It can be observed from (2.2) that the transmission map $t(x)$ is inversely proportional to $d(x)$, so the objects near to the camera lens have less haze. This model is widely used by the researchers in estimating the clear images. The synthetic datasets can also be generated using the atmospheric scattering model by selecting a random value for the transmission map $t(x)$ and random airlight A. These values are then used to generate hazy images from the clear images.

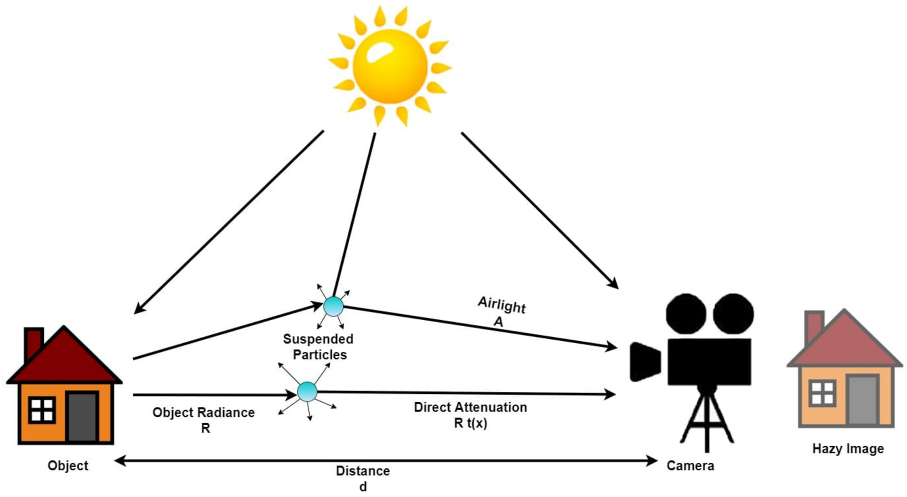


Figure 2.1: Haze Formation Model

2.2 Based on prior knowledge

Researchers used to detect characteristics difference between hazy and haze-free image like brightness, contrast, and saturation and apply this knowledge to estimate the transmission map t and global atmospheric light A and use (2.1) to get the clear image before the deep learning era. He *et al.* [9] introduced single image dehazing method using Dark Channel Prior (DCP), it is based on the fact that, outside haze-free images have some local regions with very low (near-zero) intensity values for at least one color channel. They used this observation along with the atmospheric scattering model to obtain a haze-free image directly from the estimated transmission map and atmospheric light as follows

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

Here, t_0 denotes the lower bound of the transmission map and A denotes the global atmospheric airlight. This method was not able to produce good results for regions that are similar to airlight. The observations made in DCP were used by many researchers in their work. Meng *et al.* [10] introduced an efficient image dehazing with Boundary Constraint and Contextual Regularization (BCCR), they proposed a boundary constraint for the transmission function and used it to calculate the transmission map. In Non-local Image Dehazing (NLD) by Berman *et al.* [11], utilized a non-local prior knowledge for image dehazing. They observed that only a few hundred distinct colors are required to represent a haze-free image, which is tightly clustered in RGB space. For hazy and haze-free photos, these colour clusters behave differently. The haze line in the hazy image is replaced by the colour cluster in the haze-free image, and this knowledge is utilised to estimate the transmission map and then used to recover the haze-free image.

Many methods are introduced for image dehazing in the spatial domain, Liu *et al.* [12] introduced a novel approach to dehaze image in the frequency domain . They used multi-scale wavelet decomposition [13] to convert images from spatial domain to frequency domain. It was observed that haze is present in the low frequency content of the image; wavelet decomposition produces four different sub-images where the first image contains low-frequency content and remaining images provides high-frequency content, specifically the visual features in the horizontal, vertical, and diagonal directions. The authors presented the Open Dark Channel Model (ODCM) for removing haze from the low-frequency portion of the image, and the transmission value acquired from ODCM is used to minimise noise from the high-frequency part of the image, and finally, wavelet decomposition is used to obtain a haze-free image.

Prior based methods are fast as they do not require any training, but they work on the assumptions made by the authors such as dark channel, color attenuation which are not true for all kinds of images. Even though these methods can remove the haze but the clear image does not look realistic due to some color distortion and oversaturation. This can be solved using some optimization but each image requires a different type of optimization which is not feasible. To overcome these problems, researchers started using learning-based methods which will be discussed next.

2.3 Based on learning

In Color Attenuation Prior (CAP) Zhu *et al.* [14] proposed a method which uses prior knowledge along with linear learning model to estimate the scene depth. The difference between saturation and brightness varies for hazy and haze-free images and is directly proportional to the depth map of the image, CAP utilises this knowledge for image dehazing . So the authors have used supervised linear learning model to estimate the depth map. Cai *et al.* [15] proposed a CNN based DehazeNet, architecture using different convolution layers stacked together to estimate the transmission map and further recover the haze-free image; they also introduced BReLU for accurate restoration of the image. MSCNN [16] is CNN based architecture that uses two different branches for estimating transmission maps, one of the branches estimates at coarse-scale and the other at the fine-scale.

Most of the methods used learning methods to estimate the transmission map and simply use prior knowledge to obtain the global atmospheric airlight. Shin *et al.* [17] proposed a novel optimization framework that integrates radiance and reflectance components along with structure-guided l_0 norm for further refinement. The transmission map is estimated using this reflectance map, which is then utilised for image dehazing. In All-in-one Dehazing Network(AOD-Net), Li *et al.* [18] modified the equation of atmospheric scattering model by integrating the transmission map and airlight into a single term. Instead of calculating the transmission map first, the clear image is estimated directly using lightweight CNN. Li *et al.* [19] proposed PDR-Net. The dehazed image is reconstructed using CNN, and the colour and contrast qualities of the dehazed image are enhanced using a network. Lin *et al.* [20] proposed end-to-end attention based lightweight model MSAFF-Net which uses a channel and multiscale spatial attention module, for determining the regions with haze-related features. In Densely Connected Pyramid Dehazing Network (DCPDN) Zhang *et al.* [21] proposed a method which estimates the transmission map and airlight jointly to obtain the dehazed image. Authors proposed an encoder-decoder based on the densely connected network along with pyramid pooling to estimate the transmission map and U-Net [22] is used to estimate the airlight. Discriminator based on GAN [23] framework is used to decide whether the estimated image is real or fake.

Learning-based methods achieved accurate results but a large amount of data is required during the training process. It is difficult to get ground truth images for real-world hazy images so synthetic datasets are used during training. Synthetic images cannot generate real-world scenarios, especially in darker environments. Because of this, learning-based methods are not able to dehaze real-world images completely which opens a space for further research.

CHAPTER 3
Motivation

In [12] authors performed image dehazing in the frequency domain instead of the spatial domain. Wavelet transform is used to convert the image from the spatial domain to the frequency domain. It has been observed that hazy images contain more content in the low-frequency spectrum, whereas the haze-free images have less content in the low-frequency spectrum. One of the reasons for this could be that the haze-free images are sharper and contain more edges as compared to hazy images. From this important observation, it is concluded that the haze is generally present in the lower frequency spectrum. Motivated by this observation, Krawtchouk moments are used to transform images from spatial domain to orthogonal domain in this thesis. The details about Krawtchouk moments and its analysis on hazy images is discussed next.

3.1 Krawtchouk Moments

Krawtchouk moment is widely used in the area of pattern recognition [24, 25]. They can be utilised for image dehazing and as pattern characteristics in the analysis of two-dimensional images. The role of the Krawtchouk moment in image dehazing is examined in this chapter after a quick discussion of its definition.

3.1.1 Computation of Krawtchouk Moments

Based on the discrete classical Krawtchouk polynomials [26], image analysis utilising Krawtchouk moments presented a new set of orthogonal moments associated with the binomial distribution. Krawtchouk moments of order $(m+n)$ for an image $g(x,y)$ is given as [27]

$$Q_{nm} = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} \bar{K}_n(x; p_1, N-1) \bar{K}_m(y; p_2, N-1) g(x,y) \tag{3.1}$$

with $n = 0, 1, ..., N-1$; $m = 0, 1, ..., N-1$; $g(x, y)$ is image with size of $N \times N$, $\bar{K}_m$ and $\bar{K}_n$ is set of weighted Krawtchouk polynomials, given as

$$\bar{K}_n(x; p, N) = K_n(x; p, N) \sqrt{\frac{w(x; p, N)}{\rho(n; p, N)}} \tag{3.2}$$

where

$$w(x; p, N) = \binom{N}{x} p^x (1-p)^{(N-x)} \tag{3.3}$$

and,

$$\rho(n; p, N) = (-1)^n \left(\frac{1-p}{p}\right)^n \frac{n!}{(-N)_n} \tag{3.4}$$

and $K_n(x; p, N)$ is n-th order classical Krawtchouk polynomial defined as

$$K_n(x; p, N) = \sum_{k=0}^{N} a_{k,n,p} x^k =_2 F_1\left(-n, -x; -N; \frac{1}{p}\right). \tag{3.5}$$

where $x, n = 0, 1, 2, ..., N, N > 0, p \in (0, 1)$. The hypergeometric function ${}_2F_1$ is defined as

$${}_2F_1(a, b; c; z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!} \tag{3.6}$$

where $(a)_k$ is the Pochhammer symbol given by

$$(a)_k = a(a+1) \ldots (a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)} \tag{3.7}$$

The image can be reconstructed from Krawtchouk moments using the following equation

$$g(x, y) = \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} Q_{nm} \bar{K}_n(x; p_1, N-1)\ \bar{K}_m(y; p_2, N-1) \tag{3.8}$$

3.1.2 Representation in Matrix Form

Krawtchouk moment given in (3.1) can also be implemented in matrix format. The set of Krawtchouk moments upto order $(m+n)$ in matrix form is given as

$$\mathbf{Q} = \mathbf{K}_2 \mathbf{G} \mathbf{K}_1^T \tag{3.9}$$

where $\mathbf{G}$ is the image matrix, K_1 and K_2 are Krawtchouk polynomial matrix derived from matrix $\mathbf{K}_v$ with v=1,2 as follows

$$\mathbf{K}_v = \begin{bmatrix} \bar{K}_0(0; p_v, N-1) & \cdots & \bar{K}_0(N-1; p_v, N-1) \\ \vdots & \ddots & \vdots \\ \bar{K}_{N-1}(0; p_v, N-1) & \ldots & \bar{K}_{N-1}(N-1; p_v, N-1) \end{bmatrix} \tag{3.10}$$

The inverse transformation given in (3.8) can be represented in the matrix form as

$$\mathbf{G} = \mathbf{K}_2^T \mathbf{Q} \mathbf{K}_1 \tag{3.11}$$

3.1.3 Basis function of Krawtchouk Moments

Krawtchouk moments of an image can be interpreted as the projection of the image on the basis functions, $w_{i,j}$ which is given as

$$w_{i,j} = [k_i]^T [k_j] \tag{3.12}$$

where

$$k_i = [\bar{K}_i(0; p, N-1),\ \ldots\ , \bar{K}_i(N-1; p, N-1)] \tag{3.13}$$

and

$$k_j = \left[\bar{K}_j(0; p, N-1),\ \ldots\ , \bar{K}_j(N-1; p, N-1)\right] \tag{3.14}$$

with $i = 0, 1, .., N-1$ and $j = 0, 1, ..., N-1$. The value of N and p is taken as 8 and 0.5 respectively. Here $w_{i,j}$ is matrix of size 8X8 . The basis functions from $w_{0,0}$ to $w_{8,8}$ are shown in Figure 3.1. Krawtchouk moments of an image also provides a correlation between image **F** and basis function i.e., the value of the coefficient is higher if there is a strong similarity between the basis function and the image content and vice versa.

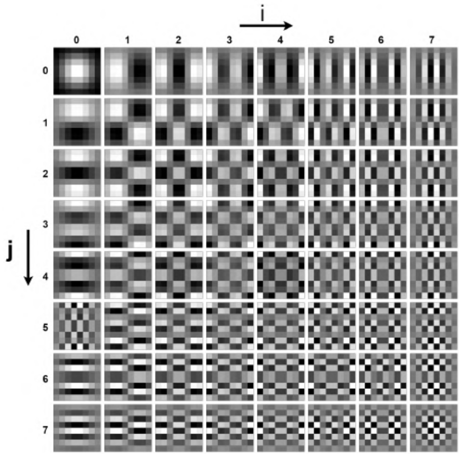


Figure 3.1: Basis function of Krawtchouk moments

3.1.4 Basis Ordering and its Importance

Krawtchouk basis functions are used as filters of KCL in the proposed architecture OTGAN. Inspired from the JPEG (Joint Photographic Experts Group) compression method [28], basis are rearranged in the zig-zag manner as shown in Figure 3.2.

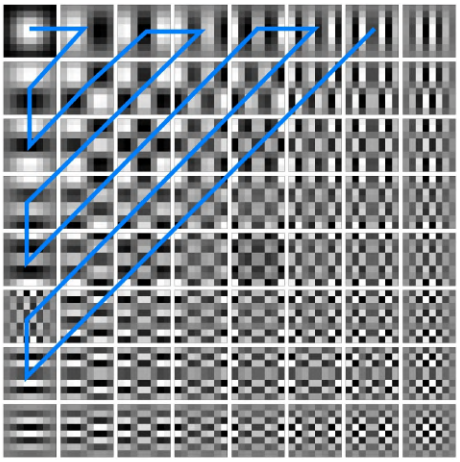
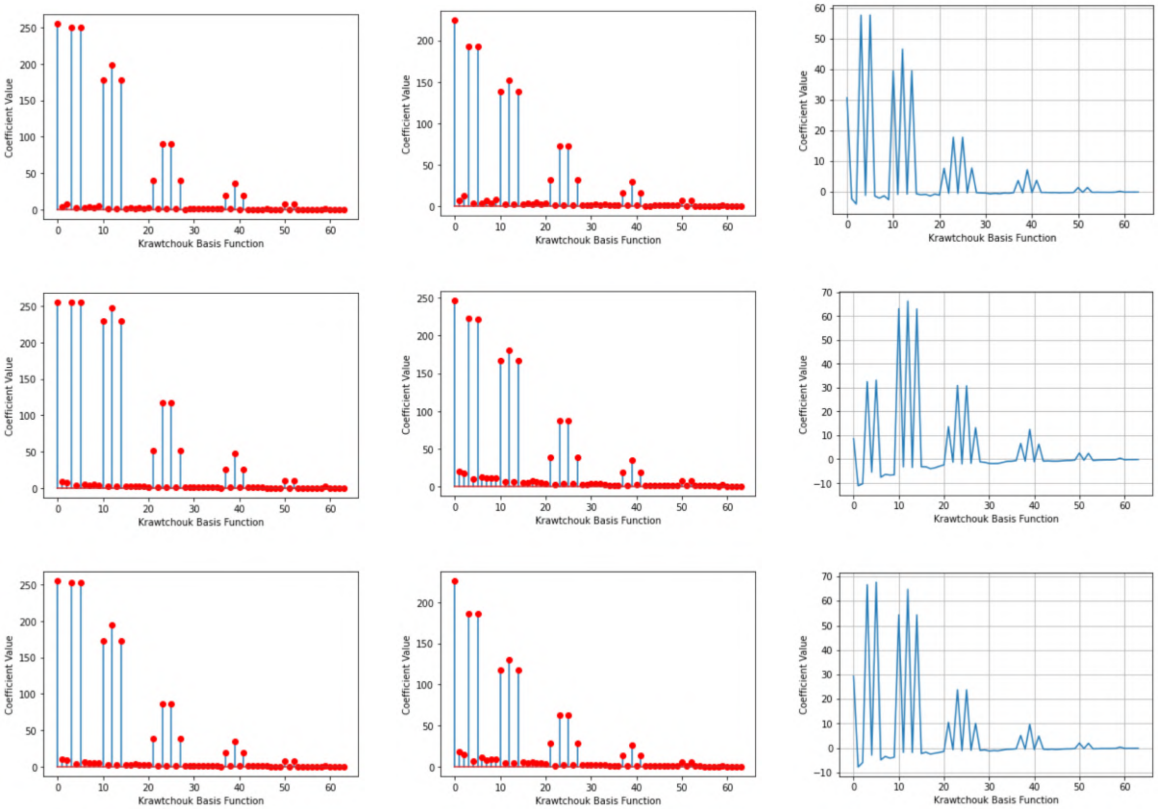


Figure 3.2: Zig-zag ordering of basis functions

We have used 64 such basis functions and represented them using w_i where $i = 0, 1, ..., 63$. Zig-Zag ordering arranges the basis functions in increasing order of frequency, i.e., frequency component increases from low to high with the increase in index i. Krawtchouk coefficients are obtained from the convolution of basis functions with the image. Average values of coefficients generated from the convolution of basis functions with three different hazy and clear images is shown in Figure 3.3. Here Figure 3.3(a)-(b) shows coefficients of three different hazy and clear image of the same scene whereas Figure 3.3(c) shows the difference between these coefficients. It can be seen from Figure 3.3(c) that there is a significant loss of Krawtchouk coefficients in basis functions with lower frequency components. Thus, in the Krawtchouk domain, the task of dehazing reduces to recovering the low-frequency Krawtchouk coefficients of a clear image from its corresponding hazy image. This observation is used in the proposed architecture discussed in the next chapter.



(a) (b) (c)

Figure 3.3: (a) Krawtchouk coefficients for hazy image (b): Krawtchouk coefficients for clear image (c) Difference in the coefficients of hazy and clear image

CHAPTER 4
Proposed Method

In this chapter, the proposed architecture, shown in Figure 4.1 is discussed in details. It consists of 8 blocks: (1) RGB to $YCbCr$, (2) Krawtchouk Convolution Layer (KCL), (3) Frequency Cube, (4) Pyramidal block for lower frequency, (5) U-Net block for higher frequency, (6) Inverse Krawtchouk Convolution Layer ($IKCL$), (7) Discriminator and, (8) $YCbCr$ to RGB .The details of the mentioned blocks are discussed next.

4.1 Architecture Structure

4.1.1 Colour Space Transformation: RGB to YCbCr

Whenever we capture any image, it needs to be stored in the electronic devices such as computers which only understand numbers. Hence, some rules need to be followed while storing the images in the memory. The color space defines this set of rules. Generally, RGB color space is used which uses Red-Green-Blue color components of an image to represent any image. The $YCbCr$ is another type of color space which represents the image using Y, Cb, and Cr components of the image. The Y component represents the Luma (brightness) component of the image, Cb and Cr represent the blue and red components related to the chroma component.

Figure 4.2 shows the hazy image along with its corresponding haze-free image in $YCbCr$ color space. It can be seen that the haze component is mainly present in the Y channel of the image. Hence, it plays an important role as compared to Cb and Cr components. Y channel of the hazy and haze-free image shows significant difference while the Cb and Cr channels do not have a significant difference. From this crucial observation, we decided to only use the Y channel for estimating the haze-free image and not changing the Cb and Cr channels. Considering this fact, first the RGB image is converted to $YCbCr$ color mode so that the hazy and haze-

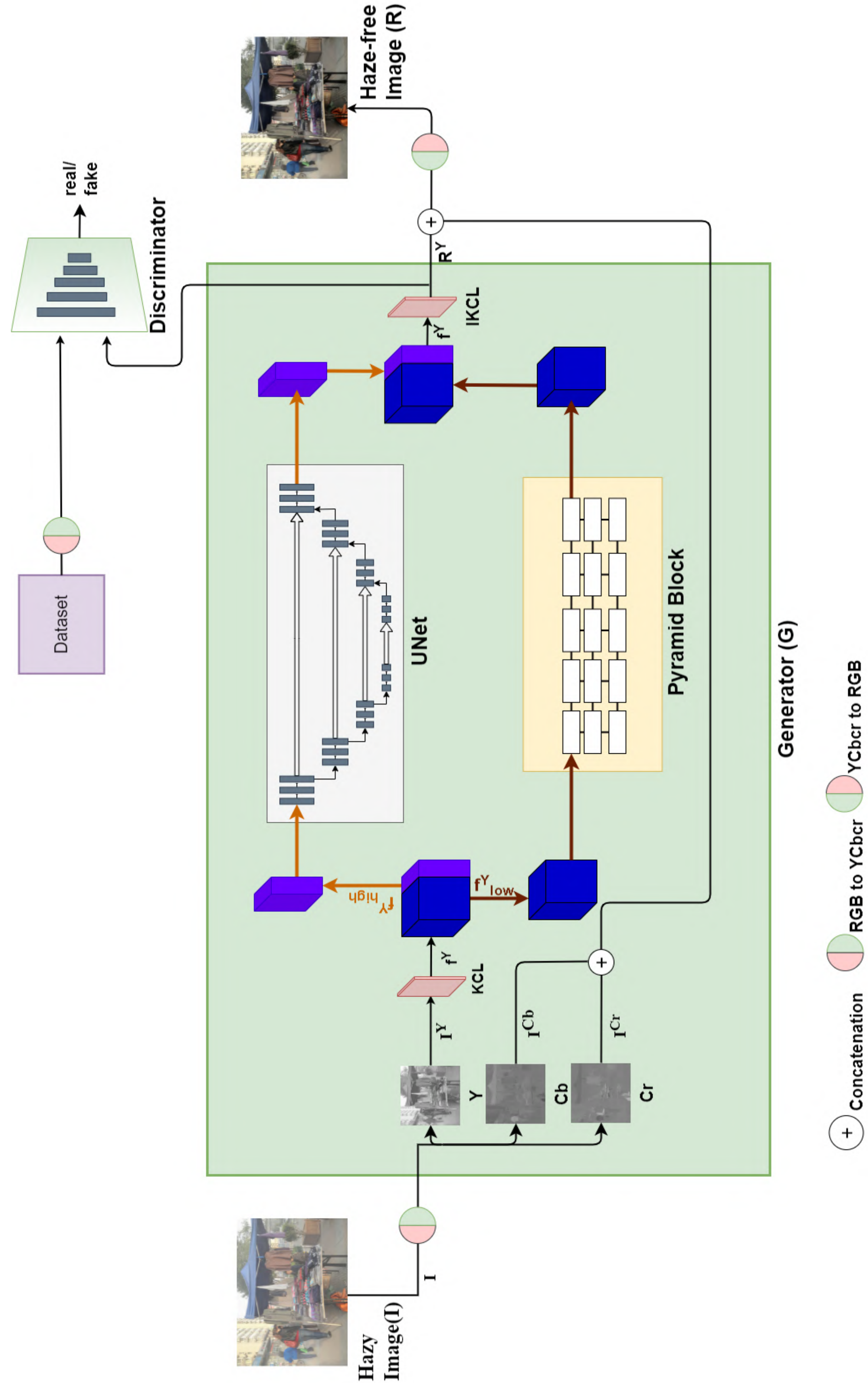


Figure 4.1: Architecture of proposed model

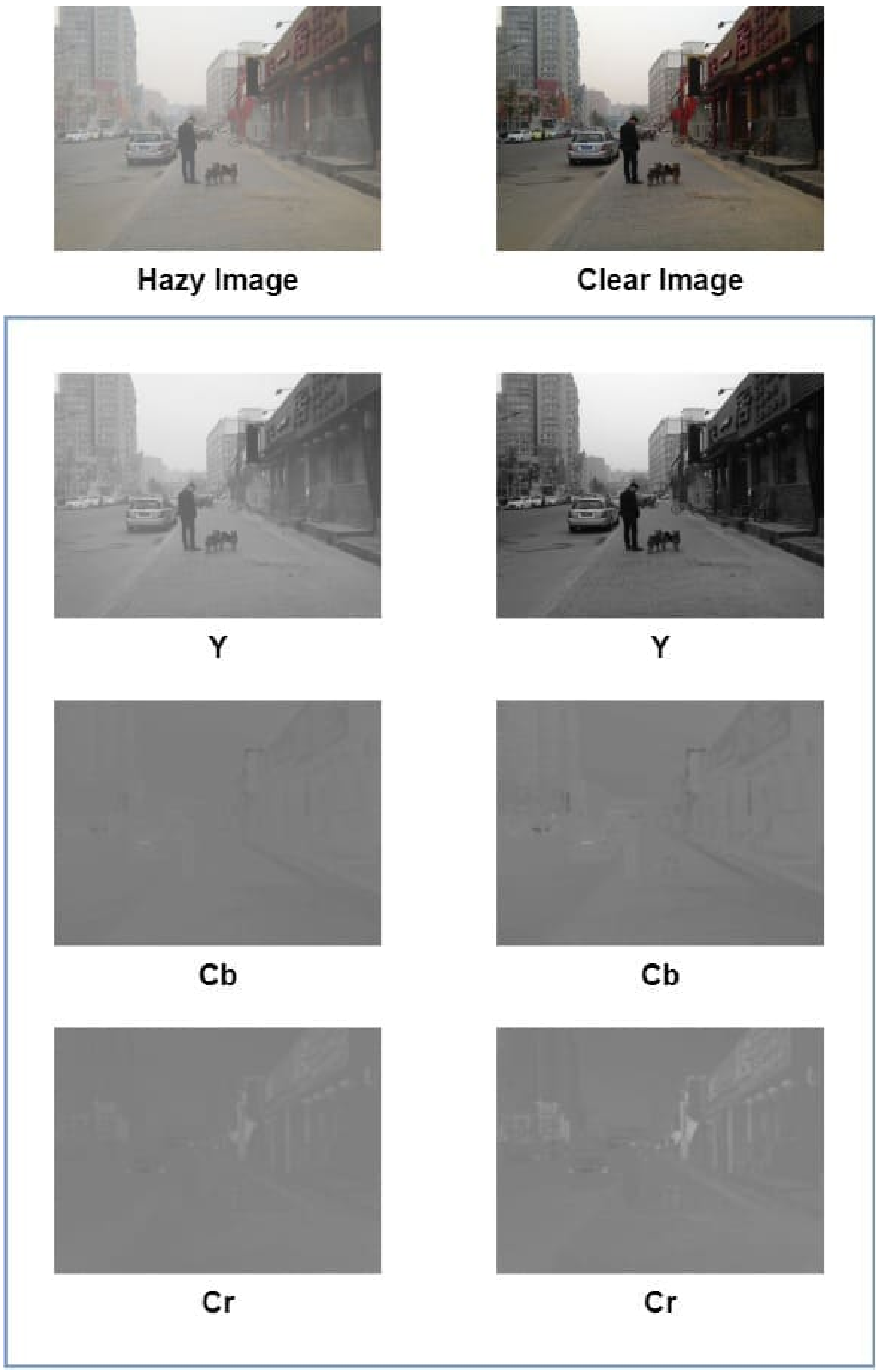


Figure 4.2: Image analysis in $YCbCr$ color space

free image pairs can be compared in $YCbCr$ space. Next, only the Y channel is passed through the proposed architecture instead of all channels. The different channel are represented as I^{Y}, I^{Cb} and I^{Cr} and are shown in Figure 4.1.

4.1.2 Krawcthouk Convolution Layer (KCL)

This layer transforms images to the Krawtchouk moments domain (orthogonal domain) from the spatial domain. Krawchouk basis function w_i of size 8×8 are treated as the filters. There are a total of 64 such filters (w_i) arranged in a zig-zag manner (Figure 3.2). The KCL layer consist of 64 features maps f_i created by performing convolution operation of w_i with I^Y as follows

$$f_i = w_i \circledast I^Y \quad \forall i \in \{0, 1, 2, ..., 63\} \tag{4.1}$$

Here, $\circledast$ represents convolution operation in which stride S is kept 1 and padding is kept as *same* for retaining the size of the image. The KCL layer is kept fixed and non-trainable during the training phase and its functionality can be compactly represented as follows

$$f^Y = KCL(f_i) \tag{4.2}$$

Here, f^Y represent the frequency cube containing all the feature maps ranging from 0 to 63. The details about the frequency cube is discussed next.

4.1.3 Frequency Cube (f^Y)

The feature maps obtained from (4.2) are used to form a frequency cube f^Y. This cube is ordered in the increasing order of the frequency content. The cube is split into two parts from a particular point T. Two parts are denoted as $f^Y_{low} = f^Y_0, ...f^Y_{T-1}$ and $f^Y_{high} = f^Y_T, f^Y_{T+1}, ..., f^Y_{63}$. The optimal value of the split point T is obtained experimentally and its value is found to be 60. The details about how to select this value is discussed in the experimental chapter. The process of partitioning is shown in Figure 4.3. The partitioned cubes f^Y_{low} and f^Y_{high} are processed separately. As discussed in Figure 3.3, the Krawtchouk coefficients have a substantial loss in lower frequencies compared to high frequencies. So, to recover the haze-free image from the hazy image, f^Y_{low} block needs a complex architecture., while simple architecture can be used for f^Y_{high} block. Next, we will discuss the network architecture for dealing with both these frequency blocks f^Y_{low} and f^Y_{high} respectively.

4.1.4 Architecture for f^Y_{low}

Taking motivation from [29], we have used a similar kind of structure for the lower part of the architecture. The detailed structure of the lower branch of

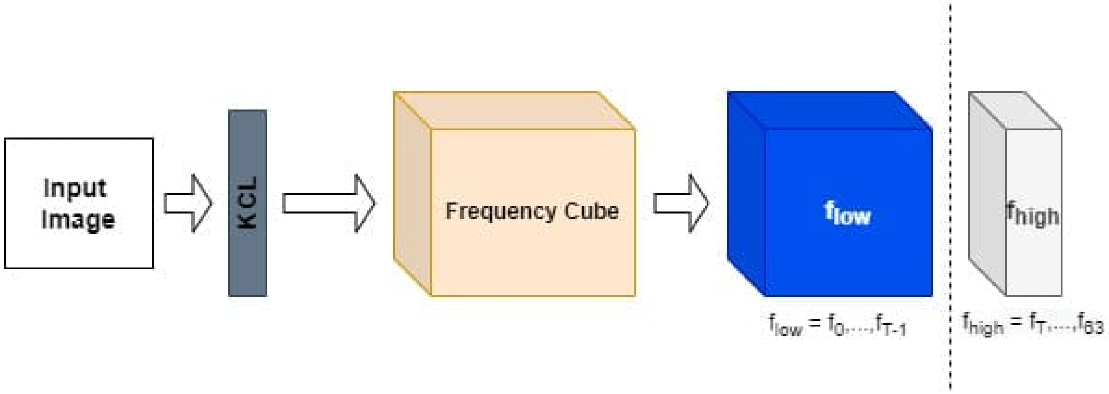


Figure 4.3: Frequency Partitioning

the proposed architecture is shown in Figure 4.4. The frequency cube f^Y_{low} obtained from frequency partitioning is sent as an input to this network which consists of six columns and three rows. The first three columns consist of down-sampling blocks and the remaining three consist of the up-sampling block. The up-sampling block increases the number of feature maps by the scale of two, while the down-sampling block decreases the number of feature maps by the scale of two. Due to this, each row which contains five dense blocks, performs an operation on a different scale while keeping the number of feature maps the same. As the feature maps of different scales have different importance, an attention mechanism is also incorporated.

Next, we will discuss the structure of the dense block shown in Figure 4.4 which is used in the architecture for f^Y_{low}. Each dense block is made up of five convolution layers, the first four of which enhance the feature map and have a skip connection to the layers before them. The final layer combines all of these feature maps so that the total number of feature maps equals the total number of input feature maps.

4.1.5 Architecture for f^Y_{high}

The higher frequency cube f^Y_{high} obtained from the frequency partitioning is sent to the upper part of the architecture. As higher frequencies do not show a substantial loss in Krawtchouk coefficients, therefore a simple UNet [22] structure is used for recovering higher frequency coefficients. The UNet structure used in the proposed architecture contains four encoders and decoders blocks. Each encoder block is constructed by stacking up convolution, batch-normalization and deconvolution blocks together. The number of channels are increased by factor of 2 for encoder blocks and decreased by factor of 2 for each decoder block. The size of

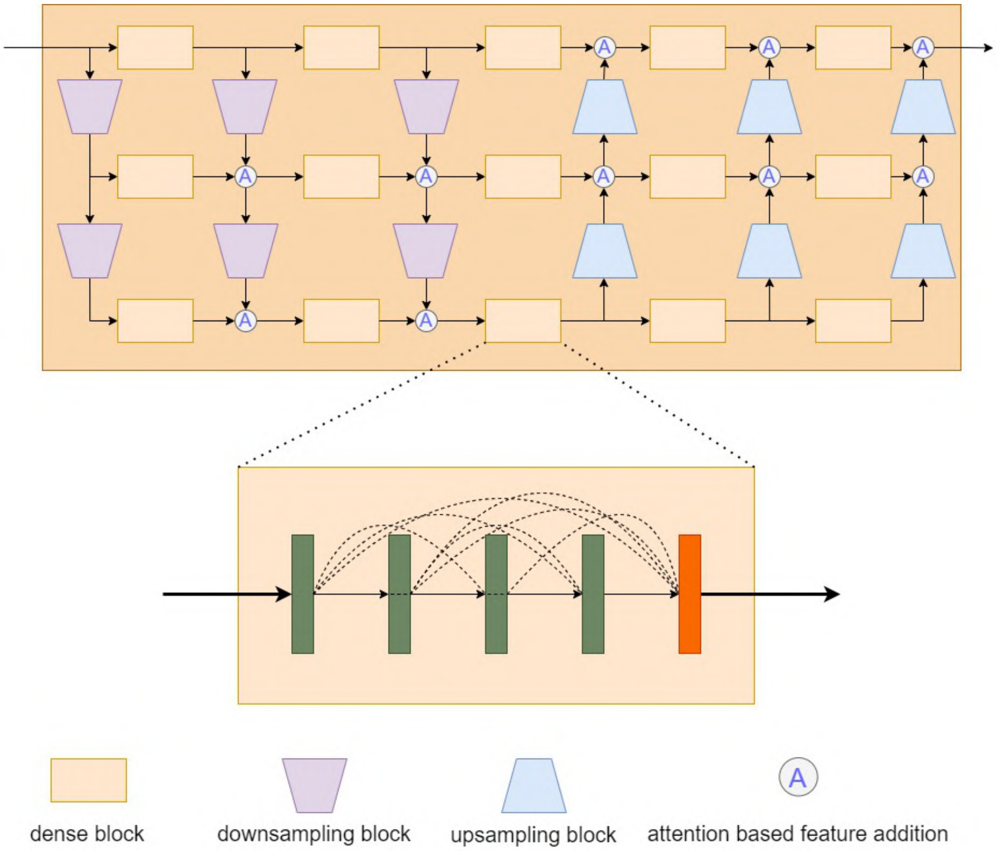


Figure 4.4: Architecture for lower branch (f_{low}^{Y})

the image is 128 × 128 for starting encoder block and is decreased by factor 2 for the successive encoder blocks; which is then increased by a factor of 2 for the successive decoder blocks. The size of the kernel is set to be 4 × 4 along with stride as 2 and padding as 1.

4.1.6 Inverse Krawtchouk Convolution Layer (IKCL)

The outputs from the lower and upper branch of the architecture are combined at the end. As the image is in Krawtchouk moment domain, it needs to be transformed into the spatial domain. The *IKCL* layer consists of a convolution layer that converts the image from the Krawtchouk moment domain to the spatial domain. The weights of the kernel are kept trainable during the training phase for providing better adaptivity of the basis functions to the dataset. This operation can be represented as follows

$$R^{Y} = IKCL(f^{Y}) \tag{4.3}$$

where, R^{Y} represents the image generated by the proposed architecture.

4.1.7 Discriminator

The generator and discriminator based GAN [23] framework is used for image dehazing. Hazy image is passed through the generator (orange box in Figure 4.1), which directly estimates the haze-free image. The Y channel of the generated image is passed through the discriminator along with the Y channel of the ground truth image. The discriminator is trained to decide whether the generated image is real or fake. The task of the generator is to produce a haze-free image that is indistinguishable from the ground truth. Discriminator and generator are not trained at the same time. The weights of discriminator are kept fixed during training of the generator, and during training of discriminator, the weights of the generator are kept fixed.

4.1.8 Colour Space Transformation: YCbCr to RGB

The image (R^Y) generated from the proposed architecture is combined with the I^{Cb} and I^{Cr} channels of the input image to get a haze-free image R which is finally transformed from $YCbCr$ color-space to RGB space for visualization.

CHAPTER 5
Experimental Work

In this chapter, experiments are carried out to verify the proposed architecture's performance, and the results are compared to state-of-the-art methods. Quantitative and qualitative experiments are carried out on synthetic images as well as the real-world images having no ground truth.

5.1 Datasets

Image dehazing is an ill-posed problem, and it's difficult to get a huge number of hazy photos as well as its haze-free image. The majority of dehazing methods rely on synthetic datasets to train their models. For creating synthetic training datasets, a depth map of haze-free images is obtained either from the existing datasets or by estimating the depth map, and then using (2.1), hazy image is generated. We have used RESIDE (REalistic Single Image DEhazing) [30] dataset which is a large scale synthetic dataset containing both outdoor (OTS) and indoor (ITS) hazy images along with its clear images. It is widely used for the training and testing of different dehazing algorithms. We used the RESIDE Outdoor Training Set (OTS) to train our model, and we tested it on the RESIDE SOTS. The SOTS dataset contains 1000 pairs of hazy and clear images of 500 outdoor and 500 indoor scenes, generated in the same way as training data is generated.We also tested our model on the RESIDE HSTS dataset, which includes synthetic hazy images as well as real-world images. Furthermore, we developed our own 200 real-world hazy images to test the proposed architecture's performance on a real-world hazy images. Figure 5.1 shows some of the real-world hazy images.

5.2 Loss Functions

In training a deep learning-based model, the selection of loss function is crucial. It has been observed through experiments that by simply using mean-square error

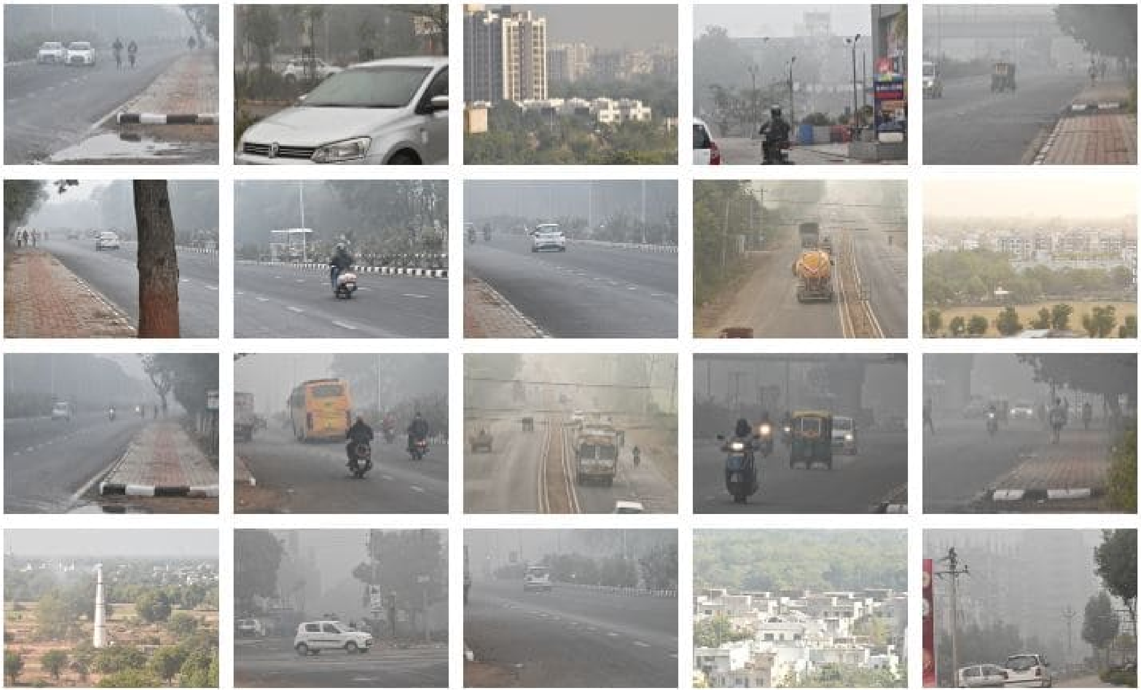


Figure 5.1: Samples of real-world hazy images

(MSE) loss is not helpful as it does not perform well with the outliers. Hence, in this thesis we have used weighted sum of three different types of losses, the details of which are as follows

5.2.1 VGG Loss

If we consider any deep neural-based image classification network, in the first few layers of the network, the feature maps obtained from the convolution layer generally contains the edges present in the image. These feature maps can be utilised as loss functions to determine the difference between the estimated and true clear image. We have used a pre-trained VGG16 model [31] trained on ImageNet [32] as the loss network. The feature maps of the last layer of the first three stages are used for defining the VGG loss as follows

$$L_{vgg} = \sum_{i=1}^{3} \frac{1}{Ch_i M_i N_i} \left\| \phi_i(\hat{R}) - \phi_i(R) \right\|_2^2 \tag{5.1}$$

where Ch represents the channel, M and N represents the size of the image, **i** represents the stage of the VGG16 network, $\phi_i(\hat{R})$ and $\phi_i(R)$, represents the features maps of the VGG16 network. Here, $\hat{R}$ and R represents the estimated and the ground truth images respectively.

5.2.2 Smooth L_1 Loss

When compared to MSE loss L_1 loss [33] is less sensitive to outliers and can prevent gradient explosion. Let $\hat{R}_i$ and R_i represents the dehazed and the original image at pixel i and N is the total number of pixels. The smooth $L_1^{(s)}$ loss [33] can be calculated as follows

$$L_1^{(s)}(\hat{R}, R) = \frac{1}{N}\sum_{i=1}^{N} \xi(\hat{R} - R) \tag{5.2}$$

where,

$$\xi(\hat{R} - R) = \xi(l) = \begin{cases} 0.5(l)^2 & if \quad |l| < 1 \\ |l| - 0.5 & otherwise \end{cases} \tag{5.3}$$

5.2.3 GAN Loss

To identify whether the created haze-free is real or fake, we used a GAN-based architecture. The discriminator D tries to tell the difference between real and fake images, while the generator G is trained to make haze-free images so that the discriminator can't tell the difference. Let $G(I)$ signify the generator's haze-free image, and R denote the dataset's real haze-free image. The GAN loss can be estimated using the following formula.

$$L_{GAN} = \underset{G}{min}\ \underset{D}{max}\ \mathbb{E}[R\ log(D(R))] + \mathbb{E}[I\ log(1 - D(G(I)))] \tag{5.4}$$

The total loss of the proposed model is obtained as a weighted sum of L_1, L_{vgg} and L_{MSE} as follows

$$L = \lambda_1 L_{vgg} + \lambda_2 L_1^{(s)} + \lambda_3 L_{MSE} + \lambda_4 L_{GAN} \tag{5.5}$$

Here, $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are regularization parameters of the loss function.

5.3 Implementation Details

Since the model is so large, training it using a full image consumes a lot of computing power and takes a long time. Therefore, we chose patches of size 128×128 at random from hazy images and matched them with patches from the haze-free image. With a batch size of 15, we used Adam's [34] optimizer for fast learning.

Learning rate is set to 0.001. Inspired from [35], the training images are converted to $YCbCr$ color mode from RGB, and only the Y (brightness) component is passed to the architecture and the remaining Cb and Cr channels are directly passed to the end of the architecture where it is combined with the Y channel of the clear image obtained from the architecture. For the loss function the values of parameters are taken as: $\lambda_1 = 0.5, \lambda_2 = 1$, $\lambda_3 = 0.04$ and $\lambda_4 = 0.05$. The model is trained for 20 epochs on NVIDIA RTX 3600.

5.4 Optimization of IKCL

Figure 5.2 shows the optimized basis functions of $IKCL$ layer after the training process of model is completed. As mentioned earlier in chapter 4.1.6, $IKCL$ layer is kept trainable during the training phase for providing better adaptivity of the basis functions to the dataset which can be seen from the figure.

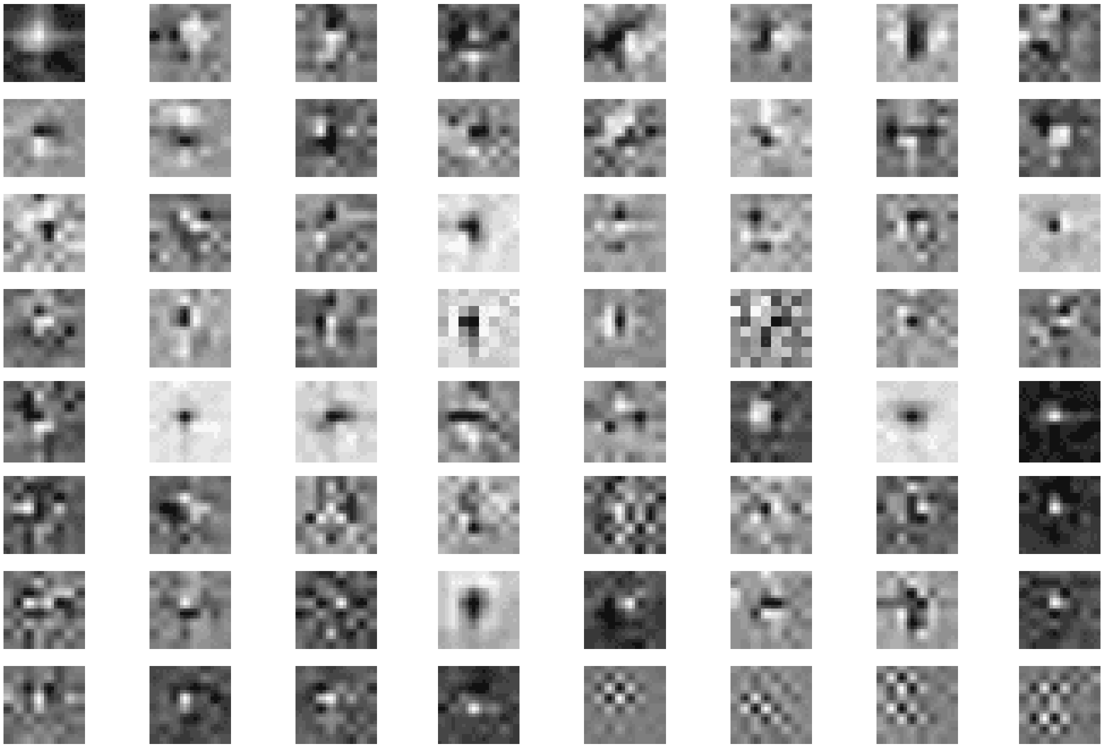


Figure 5.2: Optimized filters of $IKCL$

	SOTS(Outdoor)	SOTS(Indoor)	HSTS
DCP [9]	17.55/0.798	20.14/0.871	17.21/0.799
CAP [14]	22.28/0.912	19.06/0.835	21.53/0.866
BCCR [10]	15.48/0.782	16.87/0.789	15.09/0.737
NLD [11]	18.05/0.803	17.28/0.748	17.63/0.792
DehazeNet [15]	22.74/0.856	21.14/0.846	24.48/0.916
DCPDN [21]	19.68/0.882	15.77/0.817	20.40/0.883
AOD-NET [18]	21.34/0.924	19.37/0.850	21.57/0.921
MSCNN [16]	19.55/0.864	17.12/0.804	18.28/0.842
GFN [36]	21.48/0.837	**22.33/0.879**	22.93/0.873
Deep Energy [37]	24.08/0.933	19.25/0.832	24.44/**0.933**
OTGAN	**25.28/0.935**	21.12/0.873	**25.42**/0.929

Table 5.1: Quantitative analysis showing PSNR/SSIM scores (higher the better) for SOTS(Outdoor and Indoor) [30] and HSTS

5.5 Qualitative and Quantitative analysis

The proposed method's performance is compared to existing state-of-the-art methods in this chapter. We have compared our model with DCP [9], CAP [14], NLD [11], BCCR [10], DehazeNet [15], MSCNN [16], AOD-Net [18], DCPDN [21], GFN [36] and Deep-energy [37], the first four approaches are based on prior knowledge, while the subsequent methods are based on learning. SOTS of RESIDE is used as the testing dataset.

For quantitative examination of the dehazed images obtained from various approaches, various quality metrics such as Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index (SSIM) [38], and Natural Image Quality Evaluator (NIQE) [39] are considered as scoring metrics. Figure 5.3 shows the qualitative comparison of the dehazing methods for SOTS-outdoor dataset along with the ground truth image. It could be seen that all methods are capable of removing differing degrees of haze from the hazy image, however the results achieved using the proposed method remove haze to a greater level while also preserving the image's actual colours. The dehazing results obtained from BCCR, DCP, NLD methods are over-saturated in terms of the colors and look unrealistic. The results obtained from Dehazenet are darker as compared to our method. This can be seen from third image in sixth column of Figure 5.3, where the trees present in the image have dark color close to black while the result obtained from our method (tenth column) has true colors and it is easy to distinguish between different objects in the image. The DCP and BCCR overestimate the color of sun and we can observe ringing artifact near the sun while our method does not contain any such artifacts.

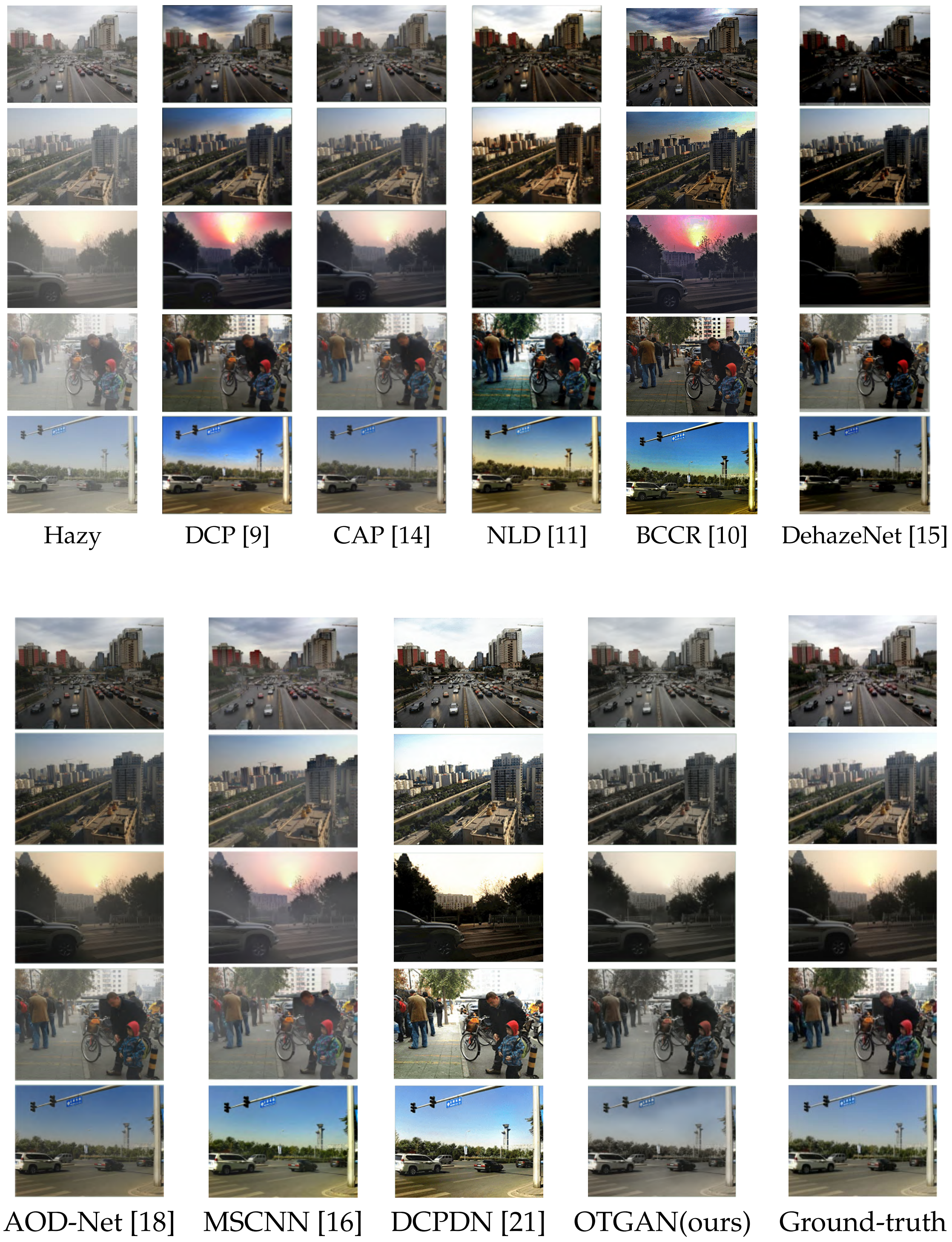
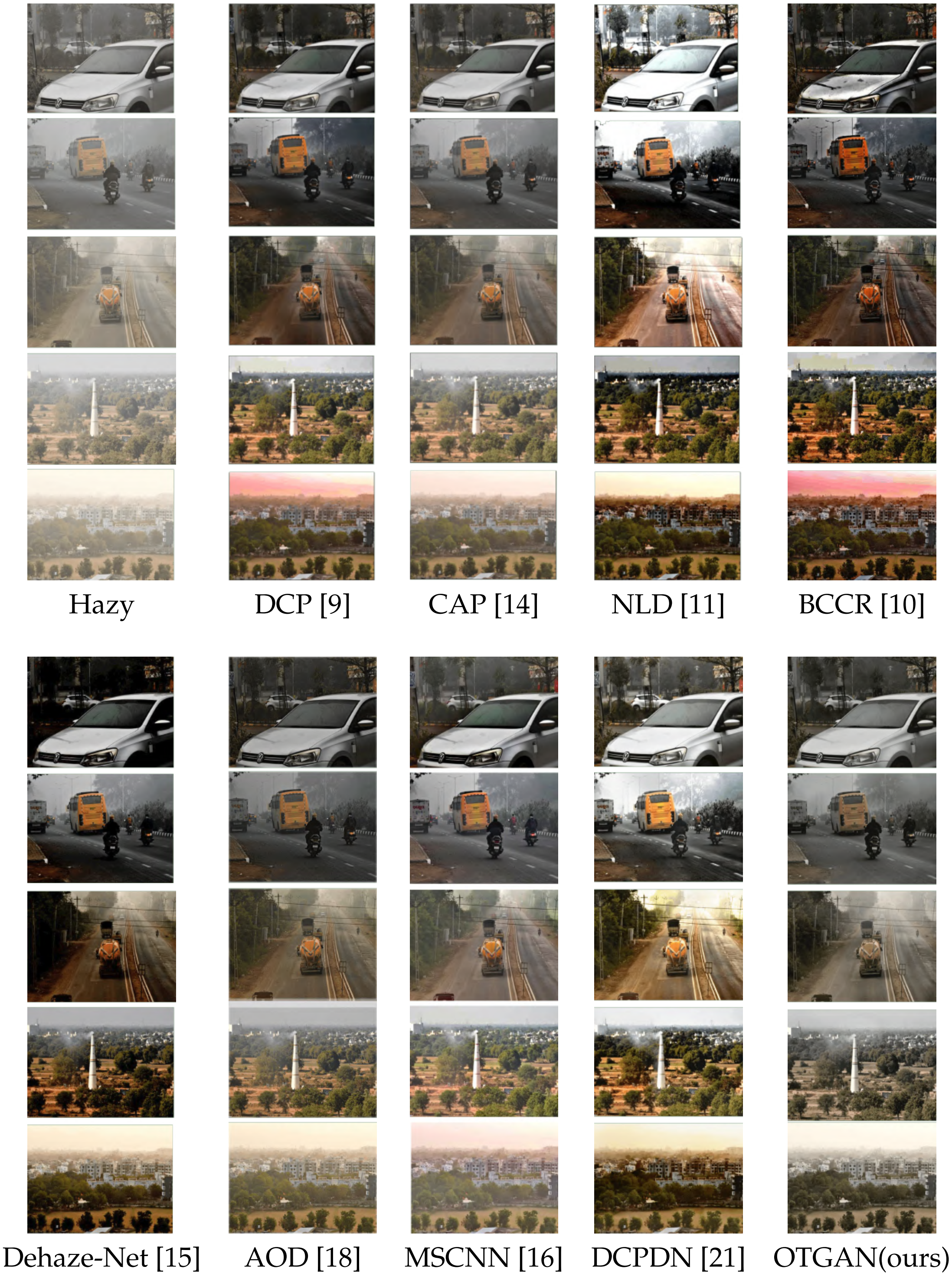


Figure 5.3: Qualitative comparison of various methods on SOTS-outdoor dataset [30]



Hazy DCP [9] CAP [14] NLD [11] BCCR [10]

Dehaze-Net [15] AOD [18] MSCNN [16] DCPDN [21] OTGAN(ours)

Figure 5.4: Qualitative comparison of various methods on real-world images

Moreover, each method produces different color of sky (see fifth row of Figure 5.3) but the color produced by our method is near to the ground truth image. Table 5.4 shows the quantitative comparison of the methods in terms of PSNR and SSIM scores for SOTS and HSTS datasets. It can be observed that our proposed method has the highest PSNR and SSIM scores for SOTS outdoor dataset as compared to other methods. The results obtained from our method on indoor dataset are not the highest but they are competitive to other methods like BCCR, NLD, DCPDN and Deep energy.

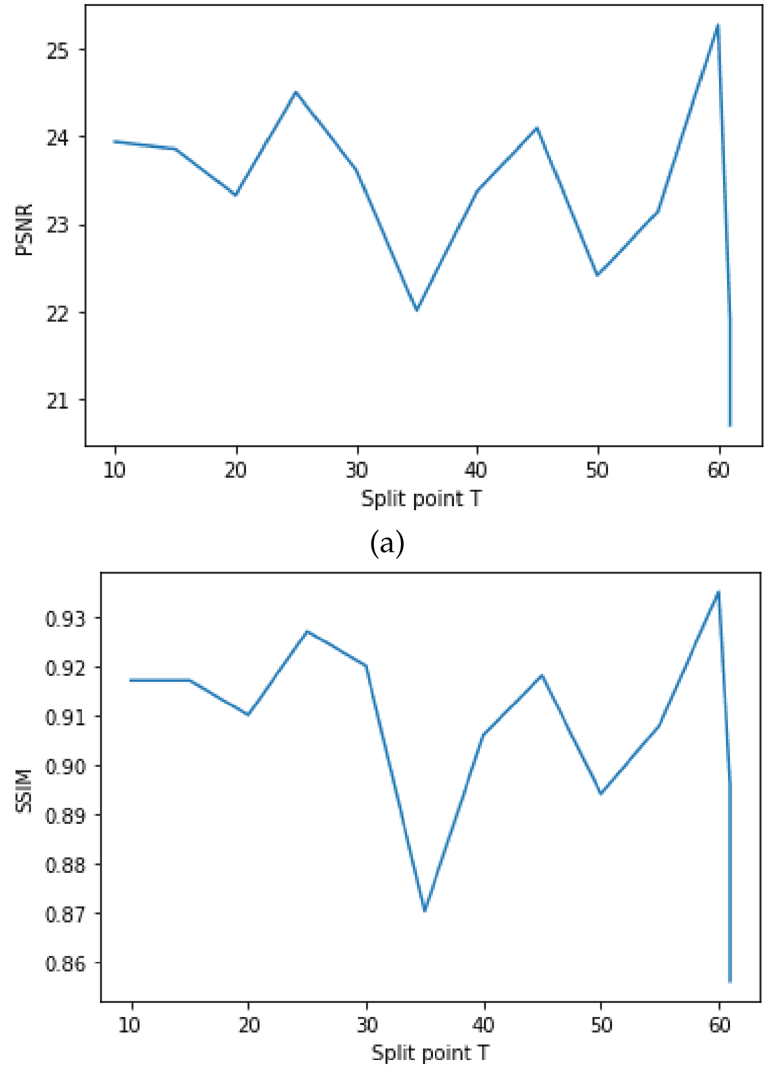
In order to validate the performance of the proposed architecture on real world images, we have chosen few images from the real world hazy dataset (Figure5.1) and compared the results of our method with other methods. Figure 5.4 shows the qualitative comparison on real world hazy images. The following observations worth noticeable are as follows: the results obtained by BCCR are darker when compared to other methods. The CAP and MSCNN methods are not able to remove most of the haze from the hazy images. Moreover, it can be concluded that all the methods struggles to dehaze the image completely, but the amount of haze removed by our method is more when compared with other methods. As the real-world images do not have the ground truth image for comparison it is not possible to evaluate PSNR and SSIM values for the real world images. A no-reference image quality metric NIQE [39] is used to measure the quality of the dehazed image. It is a no-reference metric that compares the features of the given image with Natural Scene Statistic (NSS) model. This model is constructed using natural and undistorted image corpus. A lower value of NIQE represents a better perceptual quality of the image. Table 5.5 shows the NIQE score of the proposed method along with the other methods.When compared to other approaches, the proposed method produces the lowest value of NIQE.

	NIQE
DCP [9]	9.4
CAP [14]	9.6
BCCR [10]	9.7
AOD-NET [18]	9.5
DCPDN [21]	11.9
OTGAN	**9.1**

Table 5.2: NIQE(LOWER IS BETTER) score on real world dataset

5.6 Threshold analysis

As discussed in chapter 4.1.3, the proposed architecture is divided into two branches, one for lower frequencies and the other for higher frequencies. The split point T divides the frequency cube into two parts f_{low} and f_{high}.



(a)

(b)

Figure 5.5: (a) PSNR value for different split points (T) (b) SSIM value for different split points (T)

The architecture is trained with different value of T and the PSNR and SSIM scores are calculated for each of them. Figure 5.5 shows the PSNR and SSIM scores for different values of T. It can be observed from the figure that at $T = 60$ the average PSNR and SSSIM scores achieves the highest value. Based on this observation we have selected $T = 60$ for our experimental work.

CHAPTER 6

Limitation of our Model

For the real world images suffering from severely low lighting conditions or dense haze, most existing work fails to produce good results. It has been shown that the proposed work performs better in most of the cases. However, if the above mentioned condition worsen, then the performance of the proposed work will also decrease. In particular, when the images with low light conditions are passed to our model, the dehazed image is dark and objects are not clearly visible (refer Figure 6.1(a)). The same is true for images with dense haze intensity (refer Figure 6.1(b))

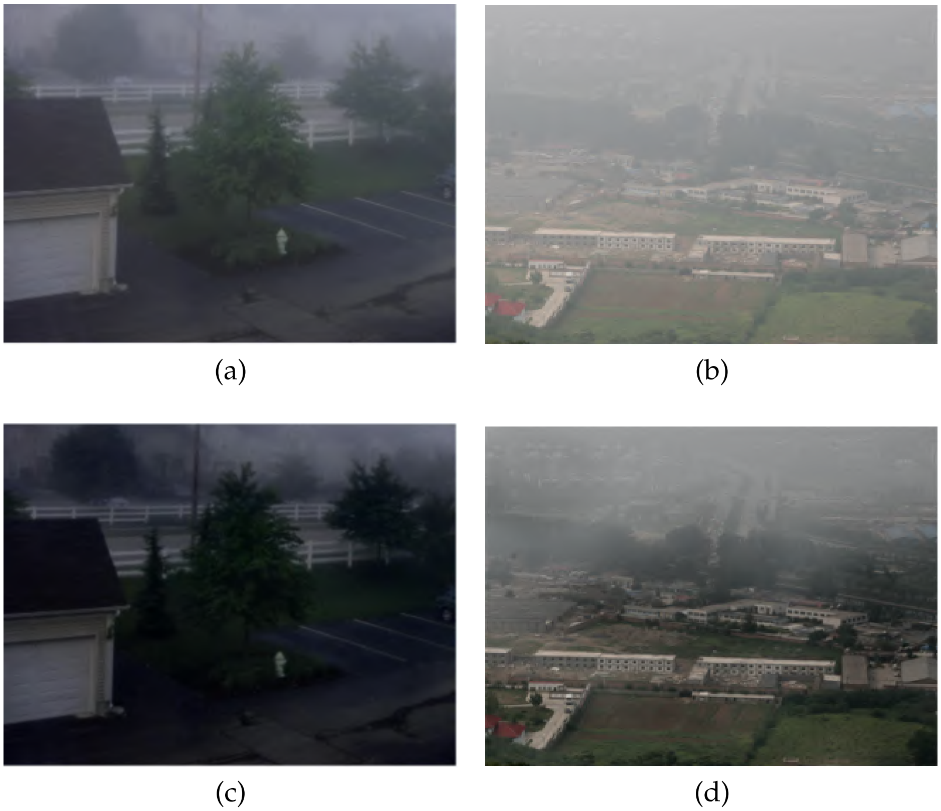


Figure 6.1: Failure cases: (a) and (b) shows the hazy inputs, (c) and (d) shows the corresponding output of our model.

Chapter 7
Conclusions

A novel end-to-end architecture has been proposed for image dehazing using Orthogonal Transform based Generative Adversarial Network, which performs image dehazing in the Krawtchouk transform domain. Instead of estimating the transmission map, the proposed model directly estimates a clear image. The Krawtchouk coefficients are used to distinguish between the image's low and high-frequency components, which are then used to recover the haze-free image from the hazy input image. When compared with existing methods, our proposed method provides competitive results. The visual comparison shows that results obtained from our method look more realistic and recovered clear images with true colors.

In the future, a dataset of real-world images and its haze-free images can be devloped to improve the dehazing quality of real-world hazy images. It is possible to employ a lightweight CNN for dehazing photos in real time.

We have submitted our work Orthogonal Transform based Generative Adversarial Network for Image Dehazing in IEEE Transactions on Multimedia.

References

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 779–788, 2016.

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[3] Srinivasa G Narasimhan and Shree K Nayar. Chromatic framework for vision in bad weather. In *Proceedings IEEE Conference on Computer Vision and Pattern Recognition. CVPR 2000 (Cat. No. PR00662)*, volume 1, pages 598–605. IEEE, 2000.

[4] Srinivasa G Narasimhan and Shree K Nayar. Vision and the atmosphere. *International journal of computer vision*, 48(3):233–254, 2002.

[5] Fabio Cozman and Eric Krotkov. Depth from scattering. In *Proceedings of IEEE computer society conference on computer vision and pattern recognition*, pages 801–806. IEEE, 1997.

[6] Yoav Y Schechner, Srinivasa G Narasimhan, and Shree K Nayar. Instant dehazing of images using polarization. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, volume 1, pages I–I. IEEE, 2001.

[7] Sarit Shwartz and YY Schechner. Blind haze separation. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1984–1991. IEEE, 2006.

[8] EJ Mccartney. Scattering phenomena (book reviews: optics of the atmosphere scattering by molecules and particles). *Science*, 196:1084–1085, 1977.

[9] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12):2341–2353, 2010.

[10] Gaofeng Meng, Ying Wang, Jiangyong Duan, Shiming Xiang, and Chunhong Pan. Efficient image dehazing with boundary constraint and contextual regularization. In *Proceedings of the IEEE international conference on computer vision*, pages 617–624, 2013.

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

[12] Xin Liu, He Zhang, Yiu-ming Cheung, Xinge You, and Yuan Yan Tang. Efficient single image dehazing and denoising: An efficient multi-scale correlated wavelet approach. *Computer Vision and Image Understanding*, 162:23–33, 2017.

[13] Xin Liu, He Zhang, Yiu-ming Cheung, Xinge You, and Yuan Yan Tang. Efficient single image dehazing and denoising: An efficient multi-scale correlated wavelet approach. *Computer Vision and Image Understanding*, 162:23–33, 2017.

[14] Qingsong Zhu, Jiaming Mai, and Ling Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE transactions on image processing*, 24(11):3522–3533, 2015.

[15] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016.

[16] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*, pages 154–169. Springer, 2016.

[17] Joongchol Shin, Minseo Kim, Joonki Paik, and Sangkeun Lee. Radiance–reflectance combined optimization and structure-guided ℓ_0-norm for single image dehazing. *IEEE Transactions on Multimedia*, 22(1):30–44, 2020.

[18] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aodnet: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, pages 4770–4778, 2017.

[19] Chongyi Li, Chunle Guo, Jichang Guo, Ping Han, Huazhu Fu, and Runmin Cong. Pdr-net: Perception-inspired single image dehazing network with refinement. *IEEE Transactions on Multimedia*, 22(3):704–716, 2020.

[20] Cunyi Lin, Xianwei Rong, and Xiaoyan Yu. Msaff-net: Multiscale attention feature fusion networks for single image dehazing and beyond. *IEEE Transactions on Multimedia*, pages 1–1, 2022.

[21] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2018.

[22] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[23] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

[24] S Padam Priyal and Prabin Kumar Bora. A robust static hand gesture recognition system using geometry based normalizations and krawtchouk moments. *Pattern Recognition*, 46(8):2202–2219, 2013.

[25] SM Mahbubur Rahman, Tamanna Howlader, and Dimitrios Hatzinakos. On the selection of 2d krawtchouk moments for face recognition. *Pattern Recognition*, 54:83–93, 2016.

[26] M Krawtchouk. On interpolation by means of orthogonal polynomials. *Memoirs Agricultural Inst. Kyiv*, 4:21–28, 1929.

[27] P-T Yap, Raveendran Paramesran, and Seng-Huat Ong. Image analysis by krawtchouk moments. *IEEE Transactions on image processing*, 12(11):1367–1377, 2003.

[28] Gregory K Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.

[29] Damien Fourure, Rémi Emonet, Elisa Fromont, Damien Muselet, Alain Tremeau, and Christian Wolf. Residual conv-deconv grid network for semantic segmentation. *arXiv preprint arXiv:1707.07958*, 2017.

[30] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE Transactions on Image Processing*, 28(1):492–505, 2018.

[31] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[32] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[33] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28, 2015.

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[35] Akshay Dudhane and Subrahmanyam Murala. Ryf-net: Deep fusion network for single image haze removal. *IEEE Transactions on Image Processing*, 29:628–640, 2019.

[36] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2018.

[37] Alona Golts, Daniel Freedman, and Michael Elad. Deep energy: Task driven training of deep neural networks. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):324–338, 2021.

[38] Peter Ndajah, Hisakazu Kikuchi, Masahiro Yukawa, Hidenori Watanabe, and Shogo Muramatsu. Ssim image quality metric for denoised images. In *Proc. 3rd WSEAS Int. Conf. on Visualization, Imaging and Simulation*, pages 53–58, 2010.

[39] Anish Mittal, Rajiv Soundararajan, and Alan C. Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal Processing Letters*, 20(3):209–212, 2013.