Image Reconstruction Under Visual Disruption caused by Rain

Tang Lai Meng

Submitted in fulfilment of the requirements for the
Degree of Doctor of Philosophy

School of Computing Science
College of Science and Engineering
University of Glasgow

July 2021

Abstract

This thesis contributes to single-image reconstruction under visual disruption caused by rain in the following areas:

1. Parameterization of a Convolutional Autoencoder (CAE) for small images [1]
2. Generation of a rain-free image using Cycle-Consistent Generative Adversarial Network (CycleGAN) [2]
3. Rain removal across spatial frequencies using the Multi-Scale CycleGANs (MS-CycleGANs)
4. Rain removal at spatial frequency's sub-bands using the Wavelet-CycleGANs (W-CycleGANs)

Image reconstruction or restoration refers to reproducing a clean or disruption-free image from an original image corrupted with some form of noise or unwanted disturbance. The goal of image reconstruction is to remove such disruption from the original corrupted image while preserving the original detail of the image scene. In recent years, deep learning techniques have been proposed for removal of rain disruption, or rain removal. They were devised using the Convolutional Neural Network (CNN) [3], and a more recent type of deep learning network called the Generative Adversarial Network (GAN) [4]. Current state-of the-art deep learning rain removal method, called the Image De-raining Conditional Generative Adversarial Network (ID-CGAN) [5], has been shown to be unable to remove rain disruption completely, or preserving the original scene detail [2]. The focus of this research is to remove rain corruption from images without sacrificing the content of the scene, starting from the collection of real rain images to the testing methodologies developed for our Generative Adversarial Network (GAN) networks. This image rain removal or reconstruction research area has attracted much interest in the past decade as it forms an important aspect of outdoor vision systems where many computer vision algorithms could be affected by rain disruption, especially if only a single image is captured.

The first contribution of this thesis in the area of image reconstruction or restoration is the parameterization of a Convolutional Autoencoder (CAE). A framework for deriving an optimum

set of CAE parameters for the reconstruction of small input images based on the standard Modified National Institute of Standards and Technology (MNIST) and Street View House Numbers (SVHN) data sets are proposed, using the quantitative mean squared error (MSE) and the qualitative 2Ds' visualization of the neurons' activation statistics and entropy at the hidden layers of the CAE. This methodology's results show that for small 32x32 pixels' input images, having 2560 neurons at the hidden layer (bottleneck layer) and 32 convolutional feature maps can result in optimum reconstruction performance or good representations of the input image in the latent space for the CAE [1].

The second contribution of this thesis is the generation of a rain-free image using the proposed CycleGAN [2]. Its network model was trained on the same set of 700 rain and rain-free image-pairs used by the recent ID-CGAN work [5]. In the ID-CGAN paper, there was a thorough comparison with other existing techniques like sparse dictionary-based method, convolutional-coding based method, etc. The results using synthetic rain training images have shown that the ID-CGAN method has outperformed all other existing techniques. Hence, our first proposed algorithm, the CycleGAN, is only compared to the ID-CGAN, using the same set of real rain images provided by the authors. The CycleGAN is a practical image's style transfer approach that falls into the unpaired category, which is capable of transferring an image with rain to an image that is rain-free, without the use of training image-pairs. This is important as natural or real rain images don't have their corresponding image-pairs that are rain-free. For comparison purpose, a real rain image data set was created. The real rain's physical properties and phenomena [6] were used to streamline our testing conditions into five broad types of real rain disruption. This testing methodology covers most of the different outdoor rain distortion scenarios captured in the real rain image data set. Hence, we can compare both ID-CGAN and CycleGAN networks using only real rain images. The comparison results using both real and synthetic rain has shown that the CycleGAN method has outperformed the ID-CGAN which represents the state-of-the-art techniques for rain removal [2]. The Natural Image Quality Evaluator (NIQE) is also introduced as a quantitative measure [7] to analyze rain removal results as it can predict the quality of an image without relying on any prior knowledge of the image's distortions. The results are presented in Chapter 6.

Subsequently, from the CycleGAN technique, the third contribution of the thesis is proposed based on the multi-scale representation of the CycleGAN, called the MS-CycleGANs technique. This proposed technique was built on the remaining gaps on rain removal using the CycleGAN. As highlighted in the rain removal paper using CycleGAN [2], the CycleGAN results could be further improved as its reconstructed output was still unable to remove the rain components at low frequency band and preserved as much original details of the scenes as possible. Hence, the MS-CycleGANs was introduced as a better algorithm than the CycleGAN, as it could train mul-

tiple CycleGANs to remove rain components at different spatial frequency bands. The implementation of the MS-CycleGANs is discussed after the CycleGAN, and its rain removal results are also compared to the CycleGAN. The results of the MS-CycleGANs framework has shown that the MS-CycleGANs can learn the characteristics between the rain and rain-free domain at different spatial frequency scales, which is essential for removing the individual frequency components of rain while preserving the scene details.

In the final contribution towards image reconstruction for removal of visual disruptions caused by rain across spatial frequency's sub-bands, the W-CycleGANs is proposed and implemented to exploit the properties of wavelet transform such as orthogonality and signal localization, to improve the CycleGAN results. For a fair comparison with the CycleGAN, both the proposed multi-scale representations of CycleGAN networks, namely the MS-CycleGANs and the W-CycleGANs, were trained and tested on the same set of rain images used by the ID-CGAN work [5]. A qualitative visual comparison of rain-removed images, especially at the enlarged rain-removed regions, is performed for the ID-CGAN, CycleGAN, MS-CycleGANs and W-CycleGANs. The comparison results among them has demonstrated the superiority of both the MS-CycleGANs and W-CycleGANs in removing rain distortions.

Contents

List of Tables

List of Figures

Acknowledgements

First and foremost, I would like to express my deep and sincere gratitude to both my research supervisors, Associate Professor Idris Lim Li Hong and Reader Paul Siebert, for giving me the opportunity to pursue this research at the University of Glasgow. I could not have completed this dissertation without their expertise, patience and support throughout this research. Their insightful feedback and valuable guidance have propelled me to achieve greater heights in my research journey.

In addition, I would like to extend my appreciation to all who have lent me a helping hand throughout my research process, in particular Mr. Teo Lim Fong, Mr. Reuben Zenas Soh Jenn Wei, Mr. Goh Wei Jie, Mr. Muhamed Isa Muhamed Nasser and Mr. Benjamin Tan. My completion of this research could not be accomplished without their support and help rendered to me.

Last but not least, I would like to express my deepest gratitude to my family members who have provided their greatest care and love during this long and difficult journey: my supportive wife Eileen who is willing to sacrifice relentlessly to provide all necessary care needed for my family and my two daughters Xue Qing and Xue Rou who have supported me throughout my studies.

Declaration

With the exception of some figures, formulas and introductory text cited in the thesis, all work in this thesis was carried out by the author unless otherwise explicitly stated.

Chapter 1

Introduction

1.1 Thesis' Outline

This thesis proposes image reconstruction techniques using deep learning, to recover clean images under visual disruption caused by rain without losing original scene's detail. A detailed study of rain characteristics and the Convolutional Autoencoder is first carried out. This is followed by the three proposed network architectures for image reconstruction based on the CycleGAN [8], MS-CycleGANs and W-CycleGANs, which have achieved significant improvements over traditional and recent deep learning algorithms, especially on removing real rain effects in images from a holistic perspective.

Bad outdoor weather condition, such as rain, snow and haze, can severely affect the majority of computer vision's algorithms, which are mostly developed to work under well-controlled environments. The most direct impact of falling rain drops, snowflakes or haze dust on outdoor images could be the degradation of its contrast or sharpness due to the change in brightness around the rain, snow or haze artifacts. In the worst case, they could cause severe distortion in the scene's content [6]. Hence, uncontrollable environmental conditions are drawing interest in computer vision, especially rain, which may be the most common but most difficult type of degradation to deal with. Hence, rain removal for image reconstruction purpose is chosen as the main focus of this thesis.

In Chapter 2, our image reconstruction's requirements based on the characteristics of rain is discussed, which play a crucial role in the determination of our proposed deep learning networks in the subsequent chapters. This is because rain produces complex intensity or brightness changes to a scene captured by a camera. The falling rain drops have a brighter appearance than

the background of the scene [6], making the background scene's appearance hazy and the objects nearer to the camera at the foreground appear blurred or distorted with low illumination [9]. Due to such complex brightness effect on images, rain detection and removal algorithms have been developed more than a decade ago to model such effects, based on the physics of rain drops [6]. One thing worth mentioning here is that most, if not all, of the rain removal algorithms till date had used test images superimposed with artificially added synthetic rain for the quantitative analysis of rain removal performance, since the ground truth or an exact rain-free version of a rain image does not exist. In addition, until recently, most image reconstruction algorithms have involved video or frame-to-frame processing, which made the rain removal task easier, due to the fact that a scene would not be occluded by rain drops in all frames of the video.

Prior to rain removal, the task of image reconstruction under visual disruption caused by rain has been studied and analyzed in Chapter 3, which involves two areas of research in image processing: image reconstruction and denoising. In recent years, Convolutional Neural Network (CNN) have demonstrated significant improvements over previous conventional image processing methods in almost all computer vision related tasks they have been applied to, which includes image reconstruction for a denoising task. The most basic CNN network architecture which can be used for both purposes is the Convolutional Autoencoder (CAE) [10], [11], [12], which is a combination of the CNN and the autoencoder. Although the CAE was developed for the purpose of initializing the network parameters of a CNN network, the CAE can be a starting point for our research as it can also be used for image denoising task due to the basic autoencoder structure in its network, if rain disruption is regarded as a form of unwanted noise. The CAE works by reconstructing an input image through non-linear compression of its data, via learning a high-level representation of the data set in a self-supervised or unsupervised manner, without the need for any data label. Thus, the CAE network is potentially capable of learning how to reconstruct a rain-free output image from a compact latent representation of the input rain image with the practical advantage of extracting both rain and rain-free features from an input rain image, without the need for rain and rain-free image-pairs, which is commonly used in many deep learning rain removal algorithms. This CAE study, coupled with our understanding of real rain physics, then led to our subsequent detailed analysis of existing and proposed algorithms in Chapters 4, 5 and 6.

1.2 Existing Techniques

To simplify our analysis of existing algorithms, we can group the various single-image rain removal techniques into two broad categories: The first category uses the conventional sig-

nal processing technique based on the rain priors models or sparse coding dictionary learning technique to remove rain effects from an image, and the second category uses deep learning approaches such as the Convolutional Neural Networks (CNNs) and the advanced Generative Adversarial Networks (GANs). As mentioned in Section 1.1, removing rain from an image instead of a video is a more challenging task because there is no spatio-temporal information in an image, unlike video-based technique which can compare successive frames to get such information [6], [13], [14], [16], [17]. In other words, the advantage in multi-frame based techniques is lacking in all single-image rain removal approaches. As this is an ill-posed problem, many single-image based methods have considered layer or signal separation [18], [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29] approaches, or derived image priors based on rain properties for detection and its subsequent removal of rain using filtering methods [18], [19], [30], [31], [32]. The most recent deep learning single-image approaches, however, have tried to overcome this natural limitation using deep-learning of both rain and rain-free features supplied by the massive training data provided during the network training. A taxonomy of the various conventional and deep learning rain removal approaches is shown in Figure 1.1.

The most common assumption of the conventional signal processing approaches, such as morphological component analysis-based dictionary learning or sparse coding methods [20], [21], [22], [23], [24], [25], [26], [27], [28], [29], and rain property-based rain priors meth-

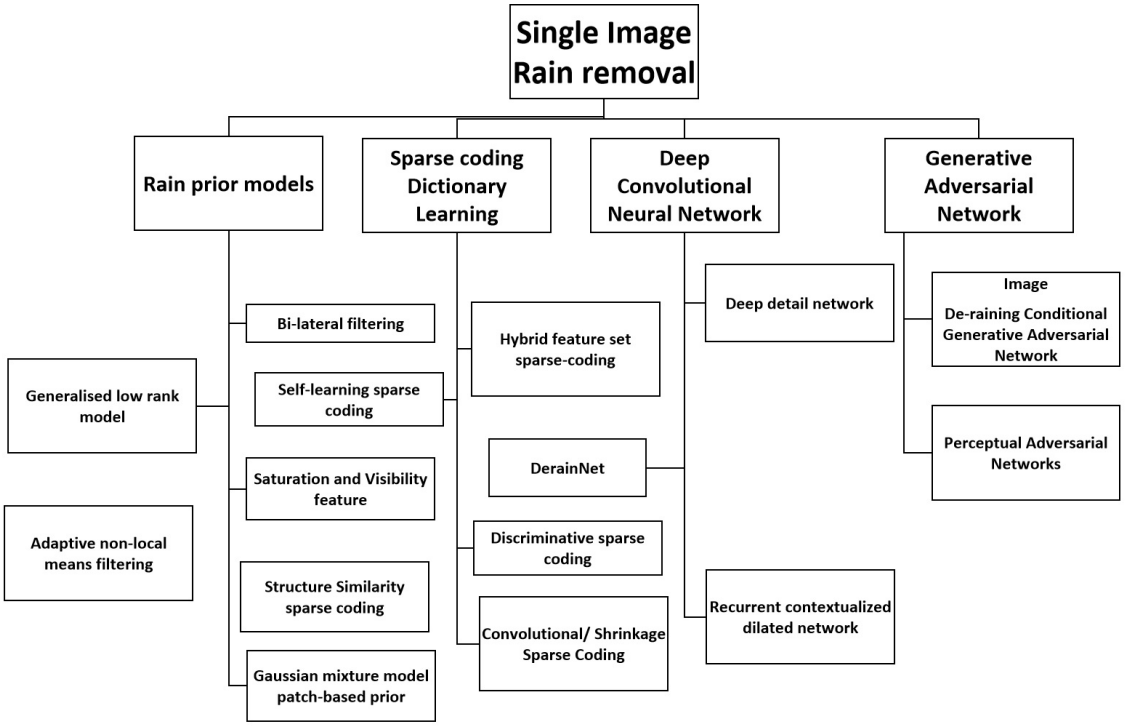

Figure 1.1: A taxonomy of various categories of single-image rain removal approaches in the literature, based on the analysis from [2].

ods [18], [19], [20], [21], [32], [33], [34], is that rain pixels are always present at the high frequency components of an image, with similar repeating patterns at a certain orientation. Although there are some successes achieved by these conventional approaches, but it was highlighted in more recent deep learning approaches using Convolutional Neural Network (CNN) and the more advanced Generative Adversarial Networks (GANs), that due to such assumptions, they may suffer from incomplete rain removal or excessive rain removal which may cause scene blurring or the introduction of unwanted artifacts [5], [35]- [38], [39].

Figure 1.2: Rain disruption on outdoor images are shown in (a) and (b). (c) and (d) are the CycleGAN's rain removal results respectively.

Figure 1.2 shows two examples of rain distorted images and their rain removal results using the proposed CycleGAN [2], to illustrate the difference before and after rain removal. The original rain images and their corresponding rain removal results by the CycleGAN have shown the potential impact that the rain removal algorithm could make for many real outdoor computer vision tasks, such as detecting the drivable path for an autonomous vehicle [40]. This is because both drivable and non-drivable paths must be detected before the autonomous system could successfully navigate around [41], regardless of its environmental conditions. In this example, such heavy rain could cause the recognition or analysis of objects in the outdoor scene to fail. Hence, there is a real need for rain removal during image reconstruction. Based on the above analysis, the proposed CycleGAN's reconstruction results are compared to the state-of-the-art deep learning approach, represented by the ID-CGAN, to demonstrate the removal of such rain distortion

to a finer detail.

This thesis aims to improve on the state-of-the-art rain removal and image reconstruction technique. In Chapter 4, both real and synthetic rain comparison results have demonstrated that the proposed CycleGAN is superior than the ID-CGAN in removing real rain distortions [2]. Another two improved CycleGAN methods in rain removal using the multi-scale approaches, are proposed in Chapters 5 and 6 respectively: the first approach is to use a MS-CycleGANs framework to remove rain distortions at different spatial frequency representations and to optimize its ability in removing rain defects using a Laplacian pyramid [42]; the second approach is to remove rain at different wavelet sub-bands using the CycleGAN, called the W-CycleGANs. The main challenge is to implement their integrated network architectures to perform a successful image-to-image translation from Domain X (image with rain) to Domain Y (image without rain), for these two implementations of the multi-scale CycleGANs, in this project.

In Chapter 5, the concept of the Laplacian Generative Adversarial Network (LAPGAN) [42] is introduced first as a GAN processing method that emulates the visual cortex processing, which divides image processing into 6 different sub-bands [42]. Using image pyramids to integrate multiple Conditional Generative Adversarial Nets (CGANs) [43], it is able to learn independent image features at multiple frequency bands and outperform the Generative Adversarial Networks (GANs) [4] in image generation. Inspired by the success of the LAPGAN in the generation of images at different frequency bands [42], the MS-CycleGANs is proposed in this thesis as a new technique to remove rain distortions independently at different spatial frequency bands based on the CycleGAN network architecture. In this framework, besides processing the entire input image at a global scale, the CycleGAN technique is also applied at each level of the Laplacian Pyramids to improve the rain removal performance of the proposed CycleGAN in Chapter 4. As shown in Figure 1.2(d), low frequency artifacts were observed in the CycleGAN's rain-removed images. One possible reason could be the lack of multi-scale decomposition of the images before applying the CycleGAN as rain degradation can happen at the low frequency spectrum of the images as well, although it was generally assumed to be at the high frequency spectrum of the images. Hence, the CycleGAN approach can be combined with the Laplacian Pyramid to form a network architecture to improve the CycleGAN's low-frequency noise removal performance from Chapter 5 onwards.

Besides the MS-CycleGANs, we have studied another multi-scale CycleGAN approach using the wavelet representation, called the W-CycleGANs, in Chapter 6. Wavelet representation has been implemented successfully in many image processing application areas such as color image enhancement task [44] and image super-resolution [45]. Our proposed technique is inspired by these wavelet-based works mentioned, by implementing the CycleGAN in the wavelet

domain, to remove rain components at different wavelet-decomposed frequency sub-bands of the Luminance channel of the Hue, Saturation and Value (HSV) color scheme. Using the same 700 training images as the CycleGAN (provided by the ID-CGAN study [5]), we have also compared both the proposed MS-CycleGANs and the W-CycleGANs networks' performance against the CycleGAN in Chapter 6, quantitatively, using the NIQE metrics which is based on the Natural Scene Statistics (NSS) model [7]. The comparison results of both real rain images from our rain data set and the images from the ID-CGAN paper [5], have shown the rain removal effectiveness of all proposed algorithms.

1.3 Contribution

In this thesis, new results of image reconstruction under visual disruption caused by rain are shown and discussed for each of the proposed algorithms. Detailed contributions in each of these areas are given as follows:

- The parameterization of a CAE using a novel methodology involving both quantitative and qualitative analysis of the latent representations of the input image at the hidden layer of the network is proposed [1].
- The CycleGAN [8] algorithm is proposed for the first time as a practical and effective approach to remove rain disruption, without the need of paired rain and rain-free training samples for learning, as required by other Convolutional Neural Networks (CNNs) and GANs methods [5], [35]- [39].
- The proposed CycleGAN algorithm is evaluated against the state-of-the-art representative, the ID-CGAN [5], and show better rain removal performance on real rain [2].
- Many existing algorithms are mainly compared with others quantitatively based on rain and rain-free image-pairs that are not natural rain images, and many of them are compared qualitatively using a few selected outdoor real rain images without a proper methodology. On the other hand, the comparison methodology proposed in this research is derived from real rain physics model [6], which identified five broad categories of real rain distortion [2].
- The MS-CycleGANs algorithm in which multiple CycleGANs are implemented within the Laplacian pyramid framework is proposed for the first time as a practical and effective multi-scale CycleGAN approach to remove rain disruption, without the need of paired rain and rain-free training samples.

- The W-CycleGANs algorithm is proposed for the first time as another effective multi-scale CycleGAN approach to remove real rain from images, including its low frequency noise.
- Both the proposed MS-CycleGANs and W-CycleGANs networks are evaluated against the proposed CycleGAN [2] using the quantitative NIQE metrics [7], and show better rain removal performance.

1.4 Motivation

The motivations behind the proposed rain removal algorithms are discussed as follow:

- We believe that a real rain image classification data set, rather than synthetically generated rain images, need to be created in this research for qualitative and quantitative evaluation of rain removal algorithm. This proposed real rain classification data set is derived based on rain physics model [6], for comparison with existing rain removal algorithms, as discussed in Chapter 2.

- The parameterization of a Convolutional Autoencoder to reconstruct a small image effectively is then discussed in Chapter 3. Through a detailed study, the effectiveness of the representations of features in the compressed latent space of the network can be studied, both qualitatively and quantitatively.

- As part of a study in rain removal, the effectiveness of a special class of Generative Adversarial Networks (GANs) [4] called the Cycle-Consistent Generative Adversarial Network (CycleGAN) [8], is explored in Chapter 4.

- The proposed multi-scale CycleGAN approaches also do not require rain and rain-free image-pairs for training, as discussed in Chapters 5 and 6. This is very practical for training the networks to remove real rain disruption for outdoor tasks, as real rain image-pairs simply don't exist in reality. This practical advantage of the various CycleGAN networks to learn the style transfer function to map an input image from rain to its rain-free version, can address the concern of other deep learning approaches because of their need for such image-pairs. This avoids the practical issues of collecting similar rain and rain-free image-pairs to train our proposed algorithms, which is non-existent. Furthermore, we feel that using synthetic rain to evaluate a GAN network's performance is not sufficient to demonstrate effectiveness in real rain removal. These approaches are compared with the

state-of-the-art, ID-CGAN [5], to demonstrate the superiority of our proposed networks in removing rain disruption on real rain images.

1.5 Organization of Thesis

Generating or inferring a rain-free image from a given real rain image can be considered an ill-posed problem, as such rain image-pairs are unavailable in reality. Hence, we address this ill-posed problem of single-image reconstruction under visual disruption caused by rain by researching into the following areas:

- Parameterization of a Convolutional Autoencoder (CAE) for small images which allows an understanding on the effectiveness of feature representation in the compressed latent space of the network
- Generation of a rain-free image using the proposed Cycle-Consistent Generative Adversarial Network (CycleGAN) [8] technique and its comparison with the state-of-the-art's ID-CGAN technique using real rain images that covers most real-world scenarios, rather than relying on synthetically generated rain images, for evaluation of the rain removal algorithms. Both the qualitative and quantitative evaluation methodologies (using NIQE [7]), are based on the proposed real rain classification data set derived from rain physics model [6], for comparison among rain removal algorithms
- The proposed rain removal techniques across spatial frequencies' sub-bands using the Multi-Scale CycleGANs (MS-CycleGANs) and the Wavelet-CycleGANs (W-CycleGANs) do not require rain and rain-free image-pairs for training, as compared to other deep learning techniques. This is very practical for training the networks to remove real rain disruption for outdoor tasks, as real rain image-pairs simply do not exist in reality

Chapter 1 gives a brief introduction to our proposed multi-scale's deep-learning approaches to address rain disruption's removal problem. This is followed by an overview of rain characteristics and existing image reconstruction methods in Chapter 2, in particular the existing rain removal techniques found in the existing literature. In Chapter 3, the methodology and results of the parameterization of a CAE are discussed. The network architecture and results of the proposed rain removal CycleGAN [8] network are then presented in Chapter 4. In Chapter 5, the network architectures and rain removal results of the proposed MS-CycleGANs, as compared to the CycleGAN, are discussed. This is followed by an illustration of the W-CycleGANs' network architectures in Chapter 6, with a comprehensive comparison, and qualitative and quantitative

analysis, using real rain test images. Lastly, Chapter 7 provides a brief summary and discussion of the research findings to conclude the thesis, and to discuss the future directions of our research.

Chapter 2

Background and Related Work

The task of image reconstruction under visual disruption caused by rain involves two areas of research in image processing: image reconstruction and rain detection and/or removal tasks. In this chapter, the background and related works in each of these areas will be discussed in detail. This review of existing techniques is essential to identify the gaps in the literature. It also provides the background for identifying CycleGAN-based networks as the more promising approach, which will be discussed in the subsequent chapters of this thesis. We will start by discussing the existing image reconstruction approaches, followed by rain physics, which has influenced how researchers tackled rain detection and/or removal tasks in image processing. Last but not least, early rain removal approaches, deep rain removal techniques and rain removal as an image translation problem will be discussed.

2.1 Image Reconstruction

In image processing, the image reconstruction task requires some data generative models to be built. The main challenge with image reconstruction is incomplete data availability to build such data generative models. However, using deep learning-based methods, object priors can potentially be learnt using the available training data to minimize image reconstruction errors [46]. Hence, an optimization-based reconstruction method that is formulated in the latent space of a generative deep neural network is growing in its popularity in the research area of deep generative models, in particular the Generative Adversarial Networks (GANs). In this chapter, we will discuss both the traditional autoencoders and these emerging GANs for the image reconstruction task. Figure 2.1 shows a summary of these two image reconstruction techniques, especially for our task of removing rain disruption from images, which is an image style transfer problem.

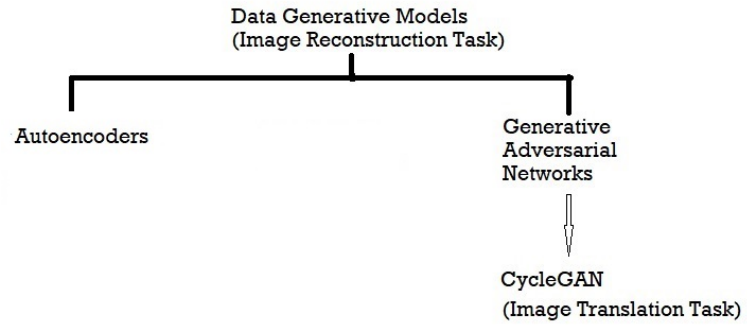

Figure 2.1: A taxonomy of data generative models.

2.1.1 Autoencoders

An autoencoder is among the most common network architecture used for image compression or denoising tasks, by reconstructing an input image through non-linear compression of data. It can learn high-level representations of a data set in a self-supervised or unsupervised manner, which makes it a basic network structure that suits our purpose of image reconstruction to remove rain effects. Since it has this advantage of self-learning capability without the need for any data label on rain images, it can be a basic network that can be improved to extract both rain and rain-free features from an input rain image, for rain removal purpose. Such an improved autoencoder can learn to extract the hierarchical, feature-based representation of an image, by combining it with a Convolutional Neural Network (CNN), to form a Convolutional Autoencoder (CAE) [10], [11], [12]. This will take advantage of both networks, so that the combined network is potentially capable of learning how to reconstruct a rain-free output image from a compact latent representation of the input rain image.

In recent years, the Convolutional Neural Network (CNN) has demonstrated significant improvements over previous conventional image processing methods in almost all computer vision related tasks they have been applied to. This is mainly due to the way CNN represents and processes visual information like the biological brain, such as the increase in receptive fields' sizes [47] in a pyramid-like, hierarchical feature-based representation [48]. However, the shortcoming of CNN is that it requires a large and labelled training data set for its supervised training. Autoencoders, on the other hand, are methods of learning higher-level representations of a data set in a self or unsupervised manner. It requires only the data and not their labels which is usually matched to each data. Hence, only a small number of training samples is needed to train an autoencoder. In other words, since the autoencoder is an unsupervised network architecture aimed at learning representation, and the CNN essentially learn hierarchical feature-based representa-

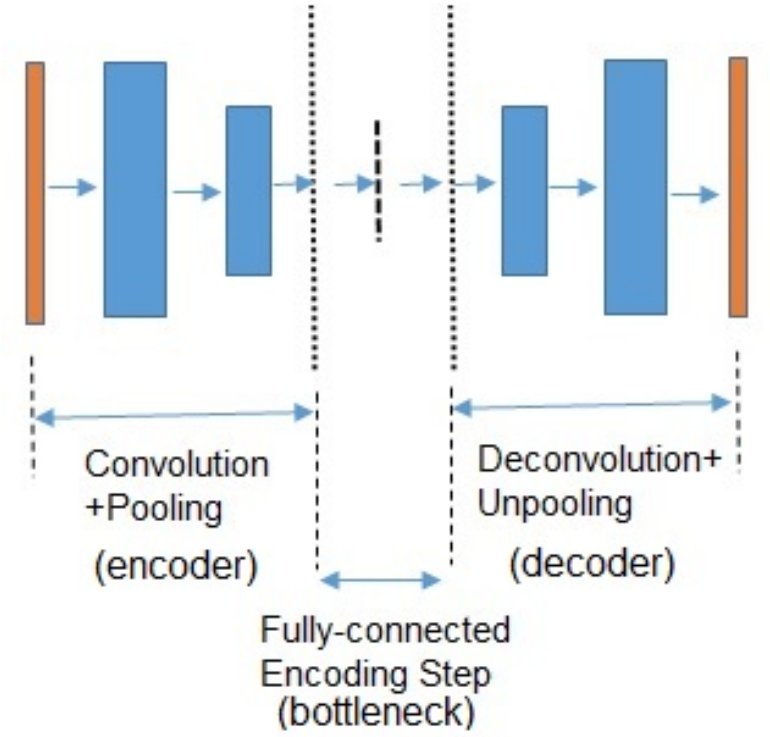

Figure 2.2: A basic Convolutional Autoencoder Model.

tion, it seems natural to combine these two techniques, to create an unsupervised, hierarchical feature-based representation learning network that emulates a biological brain [10], [11], [12].

The CAE consist of two distinct blocks: an encoder and a decoder. The input image is fed into the encoder which creates a set of information carrying units (or bits) to represent the image in the latent space at the bottleneck layer, via a convolutional network. The decoder then maps the latent space at the bottleneck layer to the output, via another convolutional network, to generate a reconstructed output image. Both the encoder and decoder networks are normally trained through the standard back-propagation method in the CAE, using some loss functions defined by the difference between both input and output of the network functions. To prevent learning an identity function, the bottleneck layer of the network is normally made quite narrow, to ensure the network must compress the input image to a smaller vector from which it must reconstruct the output image. This forces the autoencoder structure inside the CAE to learn useful features about the image. This is depicted by the architecture of a basic skeleton model of a CAE for reconstructing an input image as shown in Fig. 2.2.

Just like autoencoders, the CAE is traditionally trained to optimize a fixed criterion, such as the commonly used mean squared error, to determine which underlying features or causes are considered salient to be encoded. This reconstruction error, or mean squared error $L(x,r)$, is defined by the square of the error values between the reconstructed outputs (r) and the training

input images (x) as shown below:

$$L(x,r) = \|x-r\|^2 \tag{2.1}$$

It normally serves as the commonly used loss function to be minimized during the training for image reconstruction in an autoencoder network.

To start our analysis on a CAE, we need to explicitly define a feature-extracting function f_θ, which is a function of the parameter of the network θ, that needs to be optimized during network training. This is the encoder that allow the efficient computation of a feature vector $h = f_\theta(x)$ from an input image x. For each example of x from a data set $[x(1),..., x(T)]$, it is defined as

$$h = f_\theta(x), \tag{2.2}$$

where h is the feature vector (or representation or code) computed from x. Another closed form parameterized function, g_θ , called the decoder, maps from feature space back into input space, producing a reconstruction $r = g_\theta(h)$.

In general, autoencoders are parameterized through their encoder and decoder. The set of parameters θ (made up of weights and biases) of the encoder and decoder are learned simultaneously on the task of reconstructing the original input, i.e. attempting to incur the lowest possible reconstruction error $L(x,r)$. Hence, for a CAE, the mean squared error (MSE) is also defined as the loss function to be minimized during its network training for image reconstruction.

Due to such loss function using the mean squared error, a large enough object which can occupy a large number of pixels can have the change in its brightness to be significant enough to be implicitly specified as a salient underlying cause by the mean squared error (MSE) to be encoded from its image pixels. In other words, an autoencoder may have problem in capturing small objects in its latent space as they may not be considered salient pixels to be encoded. For example, the mean squared error used to train an autoencoder may fail to reconstruct a small object that occupy only a few pixels, as the autoencoder has limited capacity and the small number of pixels occupied by the object in the image is implicitly not being identified as being salient enough to encode, as shown in Figure 2.3 [49].

Other than this disadvantage of the autoencoders, there may be discontinuity in the latent space as well. These shortcomings of the autoencoders were addressed largely by closing up the gaps to reduce discontinuity in the latent space using a Variational Autoencoder (VAE), which may be the most useful type of image generation autoencoders. The VAE uses the architecture of an autoencoder and employs a stochastic variational inference algorithm to learn how to

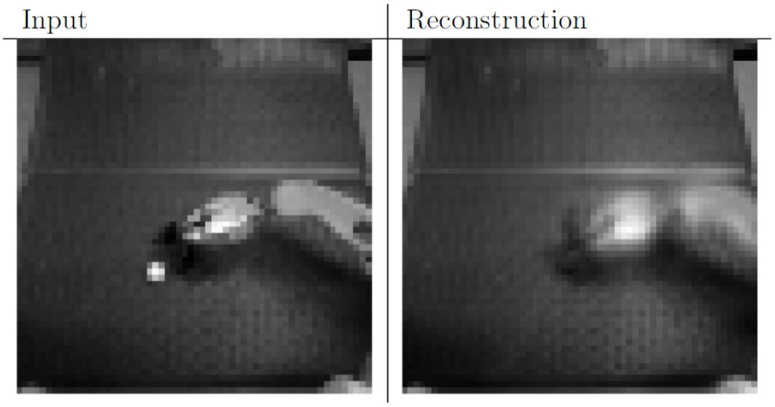

Figure 2.3: The mean squared error used to train an autoencoder for a robotic task may fail to reconstruct a small object like a ping pong ball that occupy only a few pixels, as the autoencoder has limited capacity and the mean squared error has made the training failed to identify the small ping pong ball as being salient enough to encode (reproduced from [49]).

generate different types of images with better performance than the autoencoder [50]. However, the autoencoders as a whole could be potentially improved in terms of its representation learning for better image reconstruction quality. This improvement can be extended by the state-of-the-art Generative Adversarial Networks (GANs), which is an entirely different approach to generate different types of images.

2.1.2 Generative Adversarial Networks (GANs)

GANs [4] has shown success in many image processing tasks such as image super-resolution [51], image inpainting [52] and image generation [53], [54]. Instead of imposing prior structure over learned representations like the VAE, the GANs uses a large CNN generator with sufficient capacity to learn a non-linear distribution implicitly to perform a forward mapping from the input to the output of the network. It consists of two models: a generative model, G and a discriminative model, D. The generative model learns the forward mapping to generate an output image based on the given input, such that the discriminative model is unable to distinguish those generated images from real images. As shown in Figure 2.4, the adversarial autoencoders can learn representation better than the VAE in the latent space [55]. Hence, it seems natural to use this GANs technique for our image reconstruction task.

But actually, our task is to transfer the style from a rain image domain (X) to a rain-free image domain (Y) rather than a pure image reconstruction task. Hence, it will require the GANs algorithm to have not only the ability of accurate image reconstruction but also the capability to transfer the style of an image from one domain to the other [8], [54]. This is to fulfil the goal of image reconstruction under visual disruption caused by rain, which is to remove rain effects

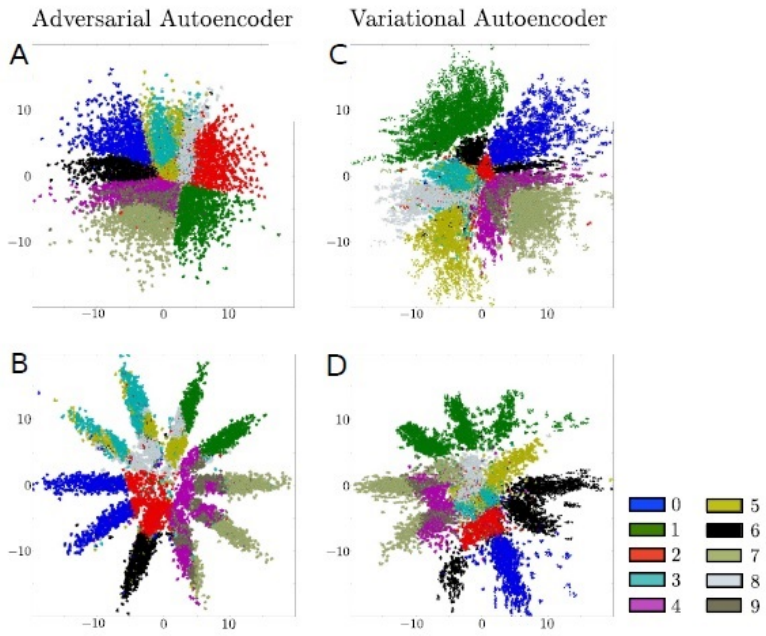

Figure 2.4: (A) & (B), (C) & (D): A comparison of the distribution of the hidden codes of both adversarial and Variational Autoencoders on MNIST in the latent space, for fitting to a 2-D Gaussian and 10 2-D Gaussians, respectively. Each color represents a different image label (reproduced from [55]).

from an input image without blurring the scene or introducing unwanted distortion [2]. In the next section, prior to removing rain distortion from images, the detrimental effects of rain on images will be analyzed, which is a topic as important as image reconstruction itself.

2.2 Visual Disruption Caused by Rain

As mentioned in Chapter 1, the performance of an outdoor computer vision system such as autonomous surveillance and navigation could be seriously degraded due to rain, which produces complex intensity or brightness changes to a scene captured by a camera. The falling rain drops always cause a brighter appearance than the background of the scene [6], making the background scene hazy and the objects nearer to the camera at the foreground appear blurred or distorted with a low illumination [9]. In this Section 2.2, rain drops' characteristics have to be understood first, before an algorithm can be devised to remove such unwanted visual effect caused by real rain.

In Section 2.2.1 to 2.2.3, the first part of our background study on conventional rain removal approach is to study the pioneering literature on removing rain in video by blurring the rain effect on images before the scene is captured by a video camera, using camera settings like the aperture size and exposure time. These settings can blur the rain drops both temporally and spa-

tially, if adjusted appropriately [6]. This pioneering work has shown that rain has similar varying intensities effects for any pixel from frames to frames, and rain couldn't cause occlusion of the scene content at all instances in the video sequences of frames. Thus, many conventional algorithms were developed later to handle rain in video.

2.2.1 Rain Models

Since the 19th century, rain phenomena's studies have been drawing interest in multiple fields such as architecture, communications and atmospheric sciences [6]. Although the purposes of these studies were not trying to solve the rain removal problem in computer vision, it became a foundation for pioneering studies on rain removal, where rain characteristics were explored and modelled in detail [6]. These pioneering in-depth and complete studies on rain's appearance on captured images have led to the derivation of rain's dynamic and photometric models which serve as the main literature explored by many subsequent rain removal algorithms. As mentioned before, the photometric model focuses on the spatial intensities' changes caused by rain while the dynamic model describes rain's properties in terms of both the spatial and temporal dimensions [6]. According to the studies, camera's parameters, rain drops' properties and environmental brightness are three critical factors that can affect rain's appearance due to rain drops' physical properties such as high terminal velocity, small diameter size and unique volumetric distribution in space [6]. Hence, the studies focused on rain detection and removal by adjusting camera's settings when the rain scene was captured [6].

Rain Properties

- Shape:

  Experimental work on dropping rain drops to investigate their shapes originated with the wind tunnel studies. A large number of experimental and theoretical investigations on the shape of rain drops falling at their terminal velocities have been reported in the literature, as early as 1883 in the area of atmospheric sciences [56]. Rain drops dropping at a high terminal velocity from the sky undergo a phenomenon called oscillations, which describes a rapid change in their shapes that subsequently cause complex intensity changes in a rain image. This phenomenon may affect light's reflection and refraction patterns at the individual rain drop which acts as a spherical lens, and can cause brightness change or even scene appearance distortion in an image [57]. Fortunately, the oscillations' effect on a rain image is mostly insignificant, as a rain drop dropping at its terminal velocity in the sky has already achieved its equilibrium shape. Rain drops that are spherical in shape are normally smaller in diameter, and those that are having oblate spheroid shape

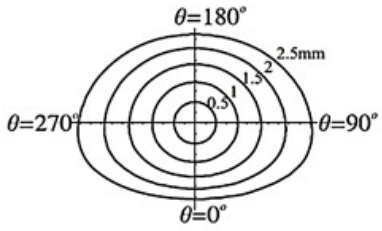

Figure 2.5: The shape model of the rain drops of various sizes (reproduced from [6]).

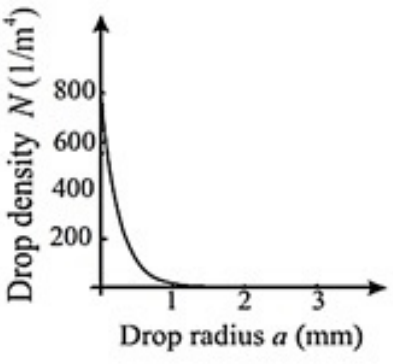

Figure 2.6: The distribution of number of raindrops per unit volume with drop size (reproduced from [6]).

are generally rain drops larger in their diameters, which can be described as a 10th order cosine distortion of a sphere [56]. The rain drop's diameter, $r(\theta)$, can be expressed as in Equation (2.3) below,

$$r(\theta) = a(1 + \sum_{n=1}^{10} c_n \cos(n\theta)), \tag{2.3}$$

where c_1 ...c_{10} are rain drop's coefficients that rely on its radius a, and θ is the polar angle of elevation [6]. Note that $\theta = 0$ is at the rain drop's dropping direction. In summary, the equilibrium shapes of various sizes of rain drops are shown in Figure 2.5 [6].

- Size:

  Rain drops' sizes can range from 0.1 *mm* to 10 *mm*. The Marshall-Palmer distribution [58] is normally used to empirically calculate the distribution of rain drops' sizes, as shown below:

$$N(a) = 8 * 10^6 e^{-8200 * h^{-0.21} a}, \tag{2.4}$$

  where $N(a)$ is the number of raindrops per unit volume that contains rain drops with radius of a metres, within a small range of fluctuation $(a, a + da)$, and h is the rate of dropping rain drops measured in *mm per hour*. This distribution of a typical rate of 30 *mm per hour* can be viewed graphically in Figure 2.6. As shown in the figure, most of the rain drops' radius is less than 1 *mm*. On the other hand, Figure 2.5 shows that such typical sizes of rain drops can be approximated by or modelled as a spherical shape, as they have minimum shape distortion.

- Velocity:

  Rain drops falling at a constant high velocity could eventually achieve their terminal velocity [59], with a velocity range between 5 - 9 m/s , if they can fall for at least 12 m without any obstruction along their paths [60]. The empirical formula derived to calculate the terminal velocity v of a rain drop with a radius a, is shown below [61]:

$$\overrightarrow{v} = 200\sqrt{a} \tag{2.5}$$

- Distribution in Space:

  The distribution of rain drops is generally assumed to be random and uniform, within a 3D volume V in space [59], [62], with a Poisson distribution $P(k)$ [59], which can be calculated as below:

$$P(k) = \frac{e^{-\bar{n}}(n)^k}{k!}, \tag{2.6}$$

  where k is the volumetric count of raindrops, and $\bar{n} = \rho V$ is the average number of raindrops n with a density of ρ.

Rain Dynamic Model

The dynamics of rain in a video can be represented by a binary field with a value of 1 and 0 to indicate rain and rain-free pixels [6]. This can be achieved by correlating the space and time information to calculate the volumetric locations of rain drops in successive frames of the video. Rain and its direction can then be detected by looking at the neighborhood of pixels with high correlation in time [6]. This is based on the constant velocity and straight-line motion property of rain drops. The rain pixels between successive frames are separated by a constant distance in the image plane in the direction of rain [6].

Rain Photometric Model

Rain drops' complex appearance with bright illumination are mainly caused by the refraction of incident radiance from the environment through the rain drops [6]. But raindrops falling at high velocity appear as rain streaks to human or camera vision due to the blurring effect of the motion [6]. This phenomenon is found to be caused by the projected duration of raindrops onto a pixel (<1.18 ms) which is much shorter than the typical exposure duration of a video camera (30 ms) [6]. Experiments on short (1 ms) vs. long exposure time (30 ms) conducted by the authors have shown that rain drops appear to be stationary, brighter and non-transparent at short

exposure time. In other words, the normal transparent appearance of rain streaks seen by human or camera are actually caused by the motion-blurring effect of rain drops. Various factors such as the brightness of the rain drops, radiance of its environment and exposure time setting of the camera can affect the rain streaks' light intensity at the camera's pixels [6]. However, the brightness or intensity increase at the pixels caused by rain drops, in terms of the mean and standard deviation of the individual RGB components' intensity level, are found to be the same for the same region of the image [17].

2.2.2 Rain Phenomena and Categories

In this section, various rain's distortions and phenomena are being categorized based on rain's properties, and both dynamic and photometric rain models as discussed earlier. Rain drops' physical properties (high terminal velocity, small diameter size and unique volumetric distribution in space), camera's parameters and environmental brightness are three critical factors that can affect rain's appearance, as shown below [6]:

$$\sigma_r(I) = \int \sigma_r(I,z)\mathrm{d}z = k_0 \frac{a^2\sqrt{\rho}}{\sqrt{v}}(L_r - L_b)\frac{\sqrt{G(f,N,z_0)}}{\sqrt{T}}, \tag{2.7}$$

where the r (rain) and b (background scene) are used to denote both L (radiance) terms. The σ_r term in the equation is used to denote rain's visibility or intensity's standard deviation over a volume of rain. The camera parameters' function G in the equation are dependent on the exposure time T, focus distance z_0, F-number N and focal length f. In addition, camera gain k_0 and distance between the camera and rain drops z can also affect rain's appearance. Lastly, rain properties denoted by ρ (rain water density), I (rain pixel's intensity), a (rain drops' radius) and v (rain drops' velocity) could also cause different phenomena of rain drops.

In addition, due to rain's high terminal velocity being reduced by both man-made and natural structures before reaching the ground, the appearance of rain drops could be changed significantly by them. As discussed before, rain drops falling at high velocity appear as rain streaks to human or camera vision due to the blurring effect of the motion [6]. But rain drops appear to be stationary, brighter and non-transparent at short exposure time or reduced velocity [6]. Hence, various rain phenomena could be analyzed and categorized into five different types based on rain's complex appearance [2]:

i.) heavy or light rain streaks;
ii.) different appearances of rain streaks due to camera settings;
iii.) rain disruption on the glass;
iv.) rain's random droplets caused by splashing at buildings; and

v.) rain's random splashing on puddles of water on ground surface.

We will make use of such derived rain phenomena to test the effectiveness of our proposed CycleGAN networks in Chapter 4 and 6.

2.2.3 Appearance of Rain Disruption on Images

Based on the rain phenomena discussion in the previous section, the appearance of rain drops in an image captured outdoor is dependent on how close the volume of rain drops is from the camera lens, z, besides the environmental brightness. Thus, to remove rain effectively from images, there is a need to tackle at least two forms of rain appearance, one at the background and the other one at the foreground [9]. Hence, we can make use of such simplified appearance of rain on images to design our proposed network to remove rain disruption from images in two network stages:

i.) first stage is to remove the foreground rain appears as local disruptions' effect caused by near-by large rain streaks; and

ii.) second stage is to remove the background rain, which is a far-away rain effect of the scene that appears hazily in the background caused by tiny rain drops' cumulative effect.

The detailed network architectures of our proposed algorithm will be discussed later in Chapters 5 and 6.

2.3 Early Rain Removal Approaches

Our focus is in reconstructing the scene distorted by rain from a single image, a more challenging task than from a video. A detailed analysis of the two categories of rain removal approaches, namely the video approach and the single-image approach, is provided. From the study, we have found that conventional, non-deep learning approaches always tackle rain removal problem based on some 'rain priors' models such as the photometric model for a single image or the dynamic model for a video. After our discussion on video's rain removal algorithms, it is followed by a discussion on removing rain disruption from an image which is a more difficult task. This part of the background study involves a detailed study on the existing single-image non-deep learning rain removal algorithms derived from rain's appearance.

2.3.1 Rain Removal from Video

Based on the three factors affecting rain's appearance discussed in the previous section [6], the authors have suggested to remove the blurring effect caused by the moving rain streaks in both spatial and temporal dimensions, by reducing the value of the camera function G or increasing the camera's exposure time T as discussed earlier. This technique is recommended for static or simple background scene only [63]. Another suggested method for rain detection and removal from video suggested by the same authors is to first detect and then segment individual rain streaks' pixels in consecutive frames of the scene, by modelling rain streaks' appearance, for subsequent rain removal. Although this method may work for dynamic scene, it is also suitable to work with simple background scene without texture only [63].

There are four different video rain detection algorithms proposed. This is summarized in Figure 2.7. In No Explicit Detection method, rain effect is simply removed by simple noise filtering techniques [64], [65] and [66], but it fails to detect rain in a moving scene. In Per-Pixel Detection algorithm, image with rain is classified into rain and rain-free pixels [17], but the temporal properties of rain will cause some misclassification. In Patch-Based Detection, the rain's photometric model is applied to detect rain regions in an image with rain, in each video frame [6]. However, a rain-free region with other moving objects could be mis-detected by this algorithm, as a hard thresholding method of classification of rain pixels is used. In Frequency-Based method [63], rain detection in video is performed by estimating rain's energy in proportion with other objects in the spatio-temporal frequency domain. Due to only global patterns of rain is considered in the frequency domain model, some challenging scenes with cluttered rain frequency will have difficulties in rain detection and its subsequent removal [63].

After rain detection, removal of rain pixels or region can then be applied. There are two main methods to remove rain in video [63]: one technique removes rain in the time domain, while the other one does so in the frequency domain. Figure 2.8 shows the algorithms that fall under these two methods in video rain removal.

One of the simplest time-based or temporal properties-based approach that can be used for a

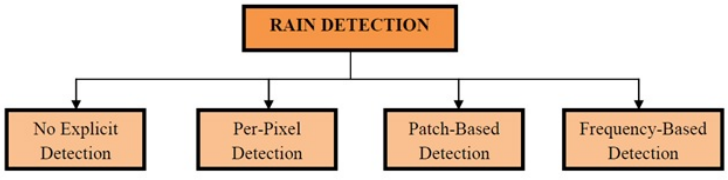

Figure 2.7: Taxonomy of rain detection in a rain video [16].

static scene without moving objects in a video uses a temporal median filter to remove rain pixels from the video scene [66]. Another simple but faster approach that uses such rain's properties in time uses a Kalman filter to estimate rain pixels' intensity to remove rain in real time [67]. Similar to the other filtering approach, such technique is limited to static scene's rain removal only. Later, another algorithm uses the chromatic or color property of rain to remove and then recover rain pixels' color values for the same pixels in consecutive frames in a video, instead of using rain's temporal properties. As discussed in the previous sections, falling raindrops will cause the rain-affected pixels to be brighter than the background pixels. This property of rain is used to recover the background's pixel color values after rain removal [68]. However, such technique is also limited to the same drawback as the other filtering methods, which is its inability to remove rain in a dynamic scene [63]. Hence, this is followed by a hybrid approach that uses both properties of rain in video. Making use of the properties of the change of pixel locations of raindrops in time and the property of rain's same intensity changes in all RGB color channels in all frames [17], K-means clustering technique is applied to the video's histogram of intensity values to detect rain pixels, followed by Gaussian blurring with α-blending applied to the detected rain pixels, to remove rain [17].

Other than rain's properties in both spatial and temporal domain, its information in the frequency domain can also be used for rain detection and its subsequent removal in video. In the Frequency-Based technique, rain pixel's frequency information can be detected by consecutive frame of the video, but its subsequent removal can be performed independently, using its neighbours in the same video frame for replacement [17]. However, the drawback of this technique is that the rain pixels cannot be removed completely, and the replaced rain pixels tend to become darker [69]. Figure 2.9(b) shows the rain removal results based on spatio-temporal frequency detection.

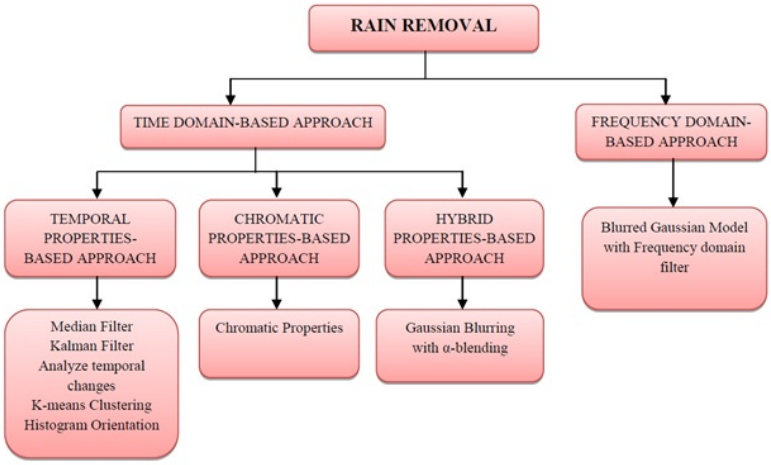

Figure 2.8: Taxonomy of rain removal in a rain video [63].

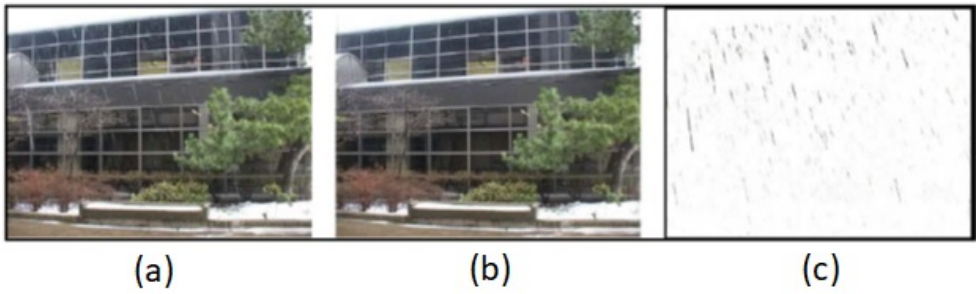

Figure 2.9: Rain detection results (c) of a windowed building with many vertical and horizontal edges, by subtracting consecutive frames of a video sequence (a) and (b). Some false detections are seen near the window frames and bushes (reproduced from [16]).

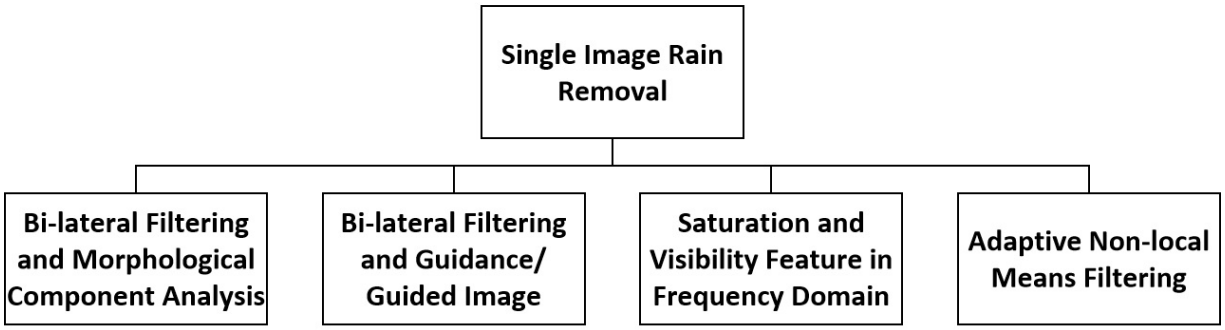

Figure 2.10: Taxonomy of early methods of removing rain from an image.

All these previous works discussed so far focus on the video-based (multi-frame, consecutive sequence of images) technique to combat the complex visual effect of rain by fitting the spatio-temporal information of rain streaks in successive images to some predicted rain physics models [31]. However, there are some limitations to this kind of approach due to the requirement to align successive frames accurately to improve their detection and removal abilities, and they couldn't be used on a single-image scenario, as highlighted in [34]. Hence, there is an increasing number of works published for removing rain from an image instead, which is also the focus of this research, as discussed in the next section.

2.3.2 Rain Removal from Single Image

Removing rain from an image instead of a video is a more challenging task because there is no spatio-temporal information in an image, unlike the video-based techniques, where successive frames can be compared [6], [13], [14], [16], [17]. A few general approaches for removing rain from an image have been proposed in [69]. Figure 2.10 shows these general approaches.

Based on the rain detection and removal algorithm for video discussed so far, we can see that there is a need to find a single-image approach and to cater for both static and dynamic scenes.

For example, moving objects can be mistaken as some large rain streaks in a video or image scene due to rain's fast-moving and large-size properties.

Morphological component analysis can be used as a rain removal method for an image corrupted with rain, using decomposition technique such as the dictionary learning and sparse coding [20]. Such image decomposition technique based on sparse representation is a common approach used by many other single-image algorithms. For example, one of the pioneering approaches has formulated rain removal from an image as a bi-lateral or guided filtering based on sparse representation or morphological component analysis [22]. However, it has the following potential limitations [23], [32]: (i) Its dictionary learning method requires a fairly complex and lengthy procedure to solve for the sparse coefficients, (ii) It may erase the texture of the background or rain-free pixels which appear in the same direction as rain streaks, (iii) Some removed rain pixels may have visual artifacts when the dictionary basis vectors are not clustered effectively, (iv) It only works on gray images.

Hence, a more recent, improved color-image sparse representation approach is subsequently proposed [23]. For removal and enhancement of the rain-free pixels, it uses the guided filtering rather than bilateral filtering technique that is coupled with a hybrid feature set like color and depth of field (DOF) information [70]. Coincidently, a similar approach, the low-rank appearance model approach, is also proposed to learn rain pixels appearing in repeated and similar patterns in an image scene, using dictionary learning and sparse coding decomposition techniques [30]. Both improved approaches are similar in their concepts. Their abilities to remove rain are also comparable [23]. Their rain removal abilities, measured as Visual Information Fidelity (VIF), are better compared to the first pioneering approach [23]. But their procedures, although improved, still require a high amount of processing time [32].

A second guidance / guided image approach is based on the white-color property of rain [33]. By separating the low frequency from the high frequency components of an image, the low frequency components can be used as a guidance image to guide the guided filter to remove rain pixels from the high frequency components. Although the processing speed of this second approach and its later refined guidance-image approach [34], is probably the fastest among all existing approaches [32], this second approach will cause the image to lose too much details and appears blurry due to effect of low pass filtering [32].

A third approach uses rain's chromatic properties (rain's visibility and saturation features) to detect and then remove rain in the frequency domain. Due to rain's high visibility and low saturation, the original color and other details in the image can be preserved after rain removal [32] in the frequency domain as rain's power spectral appears as elliptical shapes in certain orien-

tations or directions, which is easy to detect and then removed [32]. Although the processing speed of this approach is faster than first pioneering approach [23], its rain removal ability is still inferior.

The fourth approach [31] treats rain removal as a denoising problem, which is simpler than all other approaches. As rain appears as elliptical and elongated shapes oriented in the vertical directions of an image, rain can be detected by analyzing both the aspect ratio and the rotated angle of the elliptical kernel at each pixel. A non-local means filter can then be applied adaptively to the rain pixels only, to remove rain. The drawback of this algorithm is that it may cause the image to look brighter, and some blurring artifacts may occur if rain detection is not done accurately before applying the non-local means filters to the rain pixels.

Another rain removal method which uses the denoising algorithm is the bilateral filtering approach, which combines neighbouring pixels' values in a non-linear fashion, to smooth an image [70]. And more recently, the sparse and redundant representations via dictionary learning using the K-SVD algorithm is proposed as the state-of-the-art denoising algorithm to remove additive white Gaussian noise effectively [71]. But such technique may not yield satisfactory rain removal capability due to rain's complex appearance nature on images, as discussed in the previous sections.

Hence, we can conclude from the above analysis that generally rain removal from a single image using conventional, non-deep learning computer vision approaches would face the challenges of the lack of dynamic spatio-temporal information to identify and remove rain accurately, and the blurring of background scenes as a result of such rain removal processing. In the next section, we will analyze the effectiveness of various state-of-the-art deep learning rain removal algorithms.

2.4 Deep Learning Rain Removal Approaches

In Section 2.3, various conventional non-deep learning algorithms are discussed based on their rain removal methods using the properties of rain. For an image corrupted from rain drops, the pixels of certain scene are occluded at the moment the images are captured. The process of removing rain pixels can blur the actual scene content as well, particularly in a dynamic scene with object motion. Hence, the main challenge of all single-image rain removal techniques is to completely remove rain pixels and reconstruct the image without blurring the content. Based on the benefits of deep learning approaches as proposed by many contemporary literatures, we

propose to model the appearance of random rain streaks based on just a single image captured at any particular instant, using a new deep learning approach to achieve robust reconstruction of images that are affected by rain.

Other than the early algorithms developed in the past decades, which are discussed in the previous sections, various more recent single-image rain removal algorithms have been devised using signal or layer separation methods, rain filtering using image priors methods, sparse coding dictionary learning method, rain priors approaches, CNN and other advanced GANs algorithms, as discussed and shown in Figure 1.1.

Convolutional Neural Network (CNN) approaches have been recently used to remove rain from an image [35]- [38]. CNN can be trained to perform a forward image-mapping from a domain with rain (X) to another domain that is rain-free (Y), in an end-to-end manner. It has shown more superior results than the conventional approaches in removing rain from an image, including scenes of heavy rain. But as pointed out by the more advanced Generative Adversarial Networks (GANs) approaches [5], the CNN couldn't remove rain distortion as thoroughly as the GANs approaches.

The success of GANs [4] in image super-resolution [51], image inpainting [52] and image generation [53], [54] have prompted many researchers to look at this approach as a more promising approach to remove rain completely from an image without introducing unwanted distortion. Among the early GANs methods introduced for rain removal is the Image De-raining Conditional Generative Adversarial Network (ID-CGAN) [5], a rain removal approach based on the general purpose Conditional Generative Adversarial Network (CGAN) network.

The GANs consists of two models: a generative model, G and a discriminative model, D. The generative model learns the forward mapping to generate an output image based on the given input, such that it is capable of deceiving the discriminative model with its generated images. If the discriminative model is unable to distinguish those generated images from real images, the learning process is completed. Similar to GANs which has a generative model, G and a discriminative model, D, the CGAN is introduced for general purpose image-to-image translation tasks such as mapping a semantic label to a scene image, translating an image with object outlines to a color photo, and so on [54]. The advantage of the CGAN is the elimination of the need for special, hand-crafted loss function to be specified for every image-to-image translation task. Hence, the ID-CGAN is using the CGAN framework to directly tackle the single-image rain removal problem by learning a forward mapping from a domain with rain (X) to another domain that is rain-free (Y), using the discriminative model in CGAN to guide the network's learning process to ensure that the generated rain-free images by the generative model are indis-

tinguishable from the ground truth or rain-free counterparts by the discriminative model. Hence, for the ID-CGAN, the training images are also required to be made up of both rain and rain-free (ground truth) image-pairs for its network training.

In addition to the ability of the ID-CGAN to incorporate the CGAN criterion into its optimization framework to remove rain, the ID-CGAN approach also uses an additional perceptual loss function in its framework to ensure that the visual quality of its reconstructed rain-free images outperforms other advanced techniques.

2.5 Rain Removal as an Image Translation Problem

The ID-CGAN discussed in the previous section has treated rain removal as an image translation or style transfer problem. Similarly, our first proposed method in this research is to perform an image translation from rain to rain-free domains using a special GANs, the Cycle-Consistent Generative Adversarial Network (CycleGAN) [8]. But the CycleGAN does not require similar image-pairs for rain removal network training, so in this sense it has an added advantage over other CNN and GANs networks. This is possible because the CycleGAN has two generative and two discriminative models that can be trained to perform a forward image-mapping from a domain with rain (X) to another domain that is rain-free (Y), and vice versa. It learns both forward and reverse mappings, such that the additional feedback or cycle-consistency loss ensures the generated images from domain Y also appear as real as possible to the discriminative model in the X domain.

Hence, we will start the discussion on the network development of our first proposed rain removal network using the CycleGAN in Chapter 4. This was motivated by our belief that the CycleGAN, which has two generative and two discriminative models, can be trained to perform a better forward image-translation from a domain with rain (X) to another domain that is rain-free (Y), better than the ID-CGAN. Based on our results shown in Chapter 4, we have then built on the CycleGAN approach by tackling rain with different appearance and frequency bands, using novel multi-scale CycleGAN approaches, namely the MS-CycleGANs and the W-CycleGANs, to further address the gaps identified in the CycleGAN's performance [2]. These two later approaches will be discussed in detail in Chapters 5 and 6 respectively.

In the next chapter, a Convolutional Autoencoder (CAE) will first be studied to reconstruct a rain-free output image from a compact latent representation of the input rain image, as discussed in Section 2.1.1. This will enable the understanding of the optimal network capacity that

provides good image reconstruction.

Chapter 3

Parameterization of Convolutional Autoencoders for Reconstruction of Small Images

In Chapter 2, the general use of a CAE as a basic skeleton for image reconstruction based on its latent representation of the input image was discussed. Before we discuss our proposed CycleGAN algorithm in Chapter 4 which is a more complicated GANs network and sharing a similar basic encoder-decoder structure, we can study how to parameterize a CAE to reconstruct a small image effectively, to ensure our proposed CycleGAN will have sufficient network capacity to suit a large image's reconstruction task. By doing so, we can also analyze the effectiveness of the features being represented in the compressed latent space of a CAE qualitatively, as well as quantitatively using the mean squared error or image reconstruction error, to avoid overfitting or underfitting phenomena from occurring at the bottleneck layer of any network. In other words, this study is an important first step to analyze the effectiveness of our proposed algorithm later, by ensuring we have enough network capacity to learn a good image reconstruction in the latent space of the networks. In this CAE parameterization study for a small image's reconstruction task, besides looking at the quality of the images reconstructed by the networks, we need to determine the optimum number of neurons in the hidden or bottleneck layer to create an accurate latent representation of the input.

Although there exist some methods, which are used for determining the number of neurons in the hidden nodes of a conventional neural network [72], [73], [74], the research on deciding the optimum parameters such as the number of neurons in the hidden layer is still lacking to date [1]. This may pose a challenging issue while considering the complex problem of rain removal, using the proposed CycleGAN. If the number of neurons is less, as compared to the complexity of the problem data then "underfitting" may occur, when there are too few neurons

in the hidden layers to adequately detect the signals in a complicated rain data set. If unnecessary more neurons are present in the network then "overfitting" may occur. Even the most recent development of autoencoder methods for image reconstruction [49], [55] did not address such parameterization concern [1]. Thus, the first motivation behind this research is to derive an optimum set of Convolutional Autoencoders' parameters for the reconstruction of the input image, which is the fundamental purpose of image reconstruction. The experimental results of this research will be presented and discussed. Most of the material in this chapter is published in the 15th International Conference on Control, Automation, Robotics and Vision [1].

In this chapter, a framework for deriving an optimum set of CAE parameters for the reconstruction of input images based on the standard Modified National Institute of Standards and Technology (MNIST) data set is proposed. The robustness of the parameters is then verified using other data sets like the Street View House Numbers (SVHN) data set. Our results show that for small 32x32 pixels' input images, having 2560 neurons at the hidden layer (bottleneck layer) and 32 convolutional feature maps can result in optimum reconstruction performance for the CAE. In addition, the quantitative mean squared error (MSE) and the qualitative (2D visualization of the neurons' activation, the histogram statistics and estimated source entropy at the hidden layers) analysis methodology provided by this work can provide a good framework for deciding the parameter values of the CAEs to provide good representations of the input image.

Section 3.1 introduces the CAE model architecture derived for this study. Then it leads us to discuss the methodology used in our experiments both quantitatively and qualitatively to analyze the effectiveness of a CAE in preserving good feature information and reconstructing input images in Section 3.2. The experimental results of our CAE models based on the standard MNIST data set for reconstruction of small images are presented in Section 3.3. The results of the reconstruction performance of the selected CAE model with its parameters optimized, are compared with other similar input data set, the SVHN, to check the robustness of the model. Finally, this chapter is concluded with major findings for our research in Section 3.4.

3.1 CAE Models

For a CAE, we have to build a network which takes an image as an input, and reconstruct the same image as an output. Of course, the identity function would do this exactly, but we would not have learned any useful feature about the image that way. To prevent this, we will make the bottleneck layer of the network quite narrow. This means that the network must compress all the data from the image to a smaller vector from which it must reconstruct the image. This forces the autoencoder structure inside the CAE to learn useful features about the image. We will discuss the details of image compression and information theory in the next section. The

architecture of this skeleton model (Model 1) is shown in Fig. 3.1.

For this skeleton model, we will start with a network with one convolutional/pooling layer and one deconvolution/unpooling layer, called Model 1, both with filter size of 3×3. The narrow encoding layer starts with 40 neurons (N_n). This model can be trained on a Nvidia 850 GPU in about 15 minutes with no. of epochs, N_e, equals 20 and no. of feature maps, N_m, equals 32. Note that the unpooling layer is simply performing an upsampling operation, as opposed to downsampling operation for pooling layer, and the deconvolutional operation is the same as the convolutional step for symmetric convolutions upwards or downwards. This is shown in Fig. 3.2. We then compare Model 1 with Model 2, which comprises three convolutional/pooling layers and three deconvolutional/depooling layers. Since we choose to use valid borders for the convolutional and deconvolutional steps, the sizes of the layers at each step has to be calculated carefully in order to get the output image size to be the same as the input. The comparison of the two models are shown in Fig. 3.3.

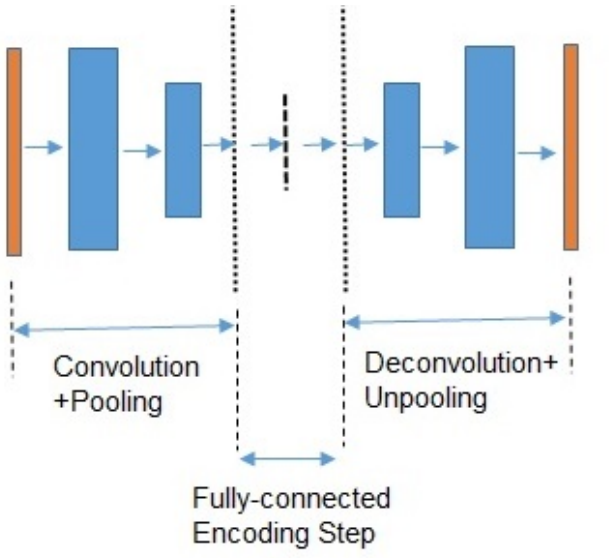

Figure 3.1: Basic Convolutional Autoencoder Model (Model 1).

	Model 1
layer_name	**Size**
0 input	Start with 1 input image of size 28x28
1 conv	To get 32 feature maps of size 26x26 (filter size 3x3, stride 1, pad 0)
2 pool	To get 32 feature maps of size 13x13
3 flatten	5408 neurons
4 encode_layer	Number of neurons (Nn=40), fully-connected
5 hidden	7200 neurons, fully-connected
6 unflatten	To get 32 feature maps of size 15x15
7 unpool	To get 32 feature maps of size 30x30
8 deconv	End with 1 output image of size 28x28 (filter size 3x3)
9 output_layer	

Figure 3.2: Model 1's architecture.

	Model 1	Model 2
layer_name	**Size**	**Size**
0 input	Start with 1 input image of size 28x28	Start with 1 input image of size 28x28
1 conv1	32 feature maps of size 26x26 (filter size 3x3, stride 1, pad 0)	32 feature maps of size 24x24 (filter size 5x5, stride 1, pad 0)
2 pool1	32 feature maps of size 13x13	32 feature maps of size 12x12
3 conv2		32 feature maps of size 8x8 (filter size 5x5, stride 1, pad 0)
4 pool2		32 feature maps of size 4x4
5 conv3		32 feature maps of size 2x2 (filter size 3x3, stride 1, pad 0)
6 pool3		32 feature maps of size 1x1
7 flatten	5408 neurons	32 neurons
8 encode_layer	Number of neurons (Nn=40), fully-connected	Number of neurons (Nn=40), fully-connected
9 hidden	7200 neurons, fully-connected	1152 neurons, fully-connected
10 unflatten	32 feature maps of size 15x15	32 feature maps of size 6x6
11 unpool3		32 feature maps of size 12x12
12 unconv3		32 feature maps of size 10x10 (filter size 3x3, stride 1, pad 0)
13 unpool2		32 feature maps of size 20x20
14 unconv2		32 feature maps of size 16x16 (filter size 5x5, stride 1, pad 0)
15 unpool 1	32 feature maps of size 30x30	32 feature maps of size 32x32
16 deconv1	End with 1 output image of size 28x28 (filter size 3x3)	End with 1 output image of size 28x28 (filter size 5x5, stride 1, pad 0)
17 output_layer		

Figure 3.3: Comparison of Model 1 and Model 2.

3.2 Methodology

In this section, the methodology used in our experiments to quantitatively and qualitatively analyze the effectiveness of a CAE, specifically in preserving good feature information and reconstructing input image, is discussed.

3.2.1 Image Compression, Information Theory and Entropy

CAE is often used in image compression to reduce the original image's data at the encoded representation. Since our inputs are images, we use Convolutional Neural Networks as encoders and decoders in a CAE. As discussed in Chapter 2, the CAE consist of two blocks: an encoder and a decoder. The input image, denoted as $f(x,y)$, is fed into the encoder which creates a set of information carrying units (or bits) to represent the image in the latent space at the bottleneck layer, via a convolutional network. For image compression, the bottleneck layer of the network is normally made quite narrow to ensure the network must compress the input image to a smaller vector from which it must reconstruct the output image. This forces the autoencoder structure inside the CAE to learn useful features about the image. If n_1 and n_2 are number of bits in the original image and encoded image respectively, the compression ratio C_R can be defined as [77]:

$$C_R = \frac{n_1}{n_2} \tag{3.1}$$

The encoded image is then used by the decoder to reconstruct an output image which can be denoted as $\hat{f}(x,y)$.

The question that naturally arises is: How many bits are required to compress or represent

an image's gray level completely? This question can be found in the theoretical framework provided by the information theory. According to this theory, information can be generated by modeling the probability of its generation process intuitively. This foundation of the information theory can be traced back to 1948, in response to the need of a communication system's design by Claude Shannon [75]. It covers the mathematical theorems for calculating signal transmission's limitations, ranging from its representation's efficiency to the transmission channel's reliability. According to this theory, we can represent each gray level of an image with L gray levels by computing the probabilities of each occurrence in the image, $p_r(r_k)$, where r_k is a discrete random variable in the range of $k = 1, 2, ..., L$, as described below [77]:

$$p_r(r_k) = \frac{n_k}{n}, \qquad k = 1, 2, ..., L \tag{3.2}$$

where n_k is the number of occurrences of the kth gray level, and n is the total pixel counts in the image, respectively. To find the average number of bits needed to represent each pixel or the average code words' length assigned to each gray-level value, L_{avg}, we can calculate the sum of the product of $I(r_k)$ and $p_r(r_k)$, where $I(r_k)$ is the number of bits used for each r_k, as shown below [77]:

$$L_{avg} = \sum_{k=1}^{L} I(r_k) p_r(r_k) \tag{3.3}$$

Hence, to encode an image with the size of $M \times N$, the required total number of bits is MNL_{avg} [77].

For a discrete random event E, the amount of information represented can be calculated using the probability of the random event, $P(E)$, which can be computed as follows [77]:

$$I(E) = \log \frac{1}{P(E)} = -\log P(E) \tag{3.4}$$

where the result always assumes a positive value or zero. In other words, there is no information transmitted when its associated event's probability is 1.0 or the event has no uncertainty. In other words, rare events which are more uncertain will require more information to be represented, and thus, a greater number of bits is needed to represent the event.

For a discrete set of possible events $\{a_1, a_2, ..., a_j\}$, the Entropy of the source [76] or the average information for each source output, H, can be defined for the source of random events with probabilities $\{P(a_1), P(a_2), ..., P(a_j)\}$, as follows [77]:

$$H = -\sum_{j=1}^{J} P(a_j) \log P(a_j) \tag{3.5}$$

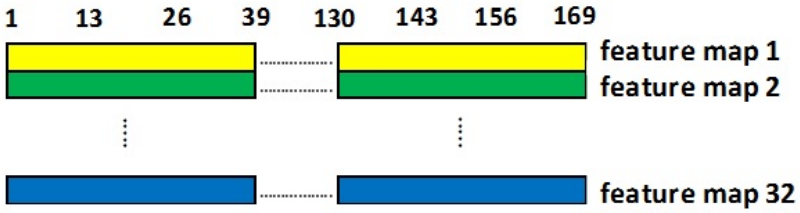

Figure 3.4: Visualization of the neurons' arrangement at layer 3 in a 2-D form.

For an image, which can be considered a "gray-level source", its gray-level histogram can be used to generate the first estimate of the source's entropy, $\tilde{H}$, as follows [77]:

$$\tilde{H} = -\sum_{k=1}^{L} p_r(r_k) \log p(r_k) \tag{3.6}$$

3.2.2 2-D Neuron Activation, Histogram Statistics and Entropy Visualization

For Model 1, the information captured by the 5408 neurons at the flatten layer (layer 3) are compressed to 40 or more neurons at the fully-connected bottleneck layer (layer 4). Since it is difficult to visualize them in a 1-D form, naturally it will be easier to visualize them in a 2-D form as shown in Fig. 3.4. There are 169 neurons arranged in each row and they correspond to the 169 neurons (or 13×13) in each of the 32 feature maps. Since each feature map at the flatten layer (layer 3) is the result of the convolution using different filters learnt followed by a subsequent pooling operation, any variation seen along the row should show the information captured by each map, which is represented by the neurons' activation states. Any variation along the column should be irrelevant for visualization of the information captured by each convolution and pooling operations at each of the feature map, as they may not show strong image structures.

To further examine the utilization of the 5408 neurons for the flatten layer (layer 3), we will also look at the entropy values of the neighboring 169 neurons for each neuron shown in each row of Fig. 3.4. The entropy values will be calculated and presented in a 2-D form as well. Also, although the neurons at the bottleneck layer (layer 4) can't be arranged by feature maps row-by-row, we will visualize and examine the encoded neurons with the assumption that the variations across neurons arranged in a 2-D form will still show some structure of activation as the neurons are still in the right order of neighboring sequence.

Finally, the histogram visualization of the stacked feature maps at both layers under study is

conducted. By looking at the statistics and histograms of each of the visualized 2-D maps and entropy images, we can then verify our conclusion on the effectiveness of the latent representation of input by our CAE.

3.2.3 Accuracy Study of Image Reconstruction using Mean Squared Error (MSE)

By using the MNIST data set as the training and validation input images, the training and validation results of our network can be measured by the mean squared error (MSE) values. The MSE is the mean of the square of the error values between the reconstructed output and the training and validation input images, and indicates the accuracy of image reconstruction of the two network models. We can then use the better model for our qualitative study in the subsequent stages.

We can also use the MSE to conclude our analysis by comparing the quality of image reconstruction (measured by mean squared error, MSE) and the efficiency of 2-D neurons' utilization in the same plots for varying both the number of neurons (N_n) and the number of feature maps (N_m). We will first define our analysis framework of using MSE as a quantitative measure of our CAE network performance. This framework is based on the autoencoder framework [78] which has an encoder function, f_θ, that extract features from an input x to compute a feature vector h = $f_\theta(x)$, which is defined as:

$$h = f_\theta(x) \tag{3.7}$$

The decoder function, g_θ, then reconstruct the input using the encoded feature vector. The reconstructed output is $r = g_\theta(h)$.

Autoencoders are parameterized through their encoder and decoder with the set of parameters θ, which is made up of weights and biases. During training, it learns the parameters of both the encoder and decoder to reconstruct the input simultaneously, and iterates towards the lowest possible reconstruction error, $L(x,r)$, which is the difference between input x and its reconstruction r. In order to estimate the data generative distribution without learning an identity function, similar low-dimensional constraint applied to the autoencoder or Principal Component Analysis (PCA) is also applied to the CAE in the regularization process. To measure the reconstruction error, MSE is defined for our experiments as follows:

$$L(x,r) = \|x - r\|^2 \tag{3.8}$$

3.2.4 Robustness Study using another data set

The final part of our methodology on image reconstruction analysis is to verify the formulation of the network parameterization guideline of the CAE using a more complex small image data set, the Street View House Numbers (SVHN). This will help to check the robustness of our hypothesis that the Convolutional Autoencoder is capturing useful information and being utilized in an efficient manner by setting the appropriate N_e (number of epochs), N_n (number of neurons at the bottleneck layer) and N_m (number of feature maps) numbers to parameterize our CAE. After the discussion of the proposed parameterization study methodology, the results are presented in the next section.

3.3 Results and Discussions

In this section, the MSE results are presented for a varying number of epochs, N_e, for the two models discussed in the previous section, Model 1 and Model 2. In addition, we will also compare the sensitivity of the MSE results to N_n for both models, to find the optimum range of the number of epochs (N_e) and the number of neurons (N_n) for our subsequent CAE experiments. The purpose of the subsequent experiments is to optimize the network structure and to understand how well the neurons are utilized in the Convolutional Autoencoders for the selected model, as well as to identify the optimum number of neurons N_n at the bottleneck layer and the optimum number of feature maps N_m. We will use the methodology discussed to analyze these results.

3.3.1 Model Selection and No. of Epochs, N_e

The effect of varying the number of epochs, N_e, on both Models 1 and 2, is shown in Fig. 3.5. We observe that the convergence rate of the MSE for both models are an exponential function over Ne, i.e. ke^{-aN_e}, where a is a constant rate of reduction and k is the initial MSE value. The values of k and a are found by performing a best-fit function for both models, and are found in the range of < 0.4 and < 0.1 respectively. But the initial MSE for Model 2 is found to be significantly higher than Model 1, as shown in Fig. 3.5. Note that the number of neurons, N_n, is fixed at 40 for the experiment. As their MSE values become smaller when N_e increases, they reach their steady state MSE values at $N_e > 15$. Hence, for all subsequent experiments, we have fixed $N_e = 20$. Note that N_m is fixed at 32 throughout our experiments.

We observe that having more neurons (and hence more parameters) are better for image reconstruction by a CAE when comparing Model 1 and 2 architecture. The flatten layer (layer 3, before the bottleneck layer) of Model 1 has a much larger number of neurons (5408, and hence a greater number of parameters), vs 32 neurons only for Model 2. Model 2 may have too much

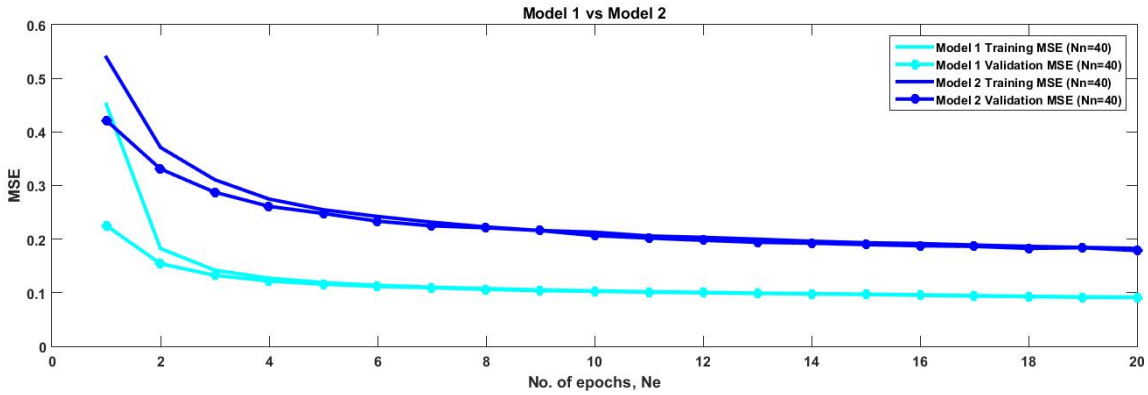

Figure 3.5: Comparison of convergence rate between model 1 and model 2 for no. of neurons $N_n = 40$ for the bottleneck layer.

information loss compared to Model 1 before being compressed.

The use of standard online back-propagation algorithm in our case, has shown that it can be difficult to reduce training error to a level near the globally optimal value, even when using more weights than training cases. But increasing the number of weights makes it easier for standard back-propagation algorithm to find a good local optimum, so using "oversized" networks can reduce both training error and generalization error. We have shown that by increasing the number of neurons in the hidden layer just before data compression plays an important role in reducing both training error and generalization error measured in MSE for a small MNIST image data set.

From these results, we can conclude that Model 1 which has 1 Conv/Pool and 1 Deconv/Depool layers is better than other models which has more Conv/Pool and Deconv/Depool layers for MNIST data set. Hence, Model 1 is selected for all our subsequent experiments.

3.3.2 No. of Neurons, N_n

In this section, we present the comparison results of the 2-D visualization of neurons' activation, the mean and standard deviation values of the 2-D neurons' activation maps, the 2-D visualization of the entropy, as well as the image reconstruction results (measured by MSE), as the number of neurons of the bottleneck layer, N_n, is varied from 40 to 2560. The N_e and N_m are fixed at 20 and 32 respectively throughout our experiments.

Fig. 3.6 shows the visualization of the stacked feature maps' neuron activation states and Fig. 3.7 shows the visualization of the histogram of the stacked feature maps at the flatten layer (layer 3) with different no. of neurons at the bottleneck layer (layer 4). As shown in Fig. 3.6, there are more activated neuron's structure or patterns as the number of neurons N_n increases from 40 to 2560, as there are lesser inactivated (black) neurons when N_n increases. Fig. 3.7(a)

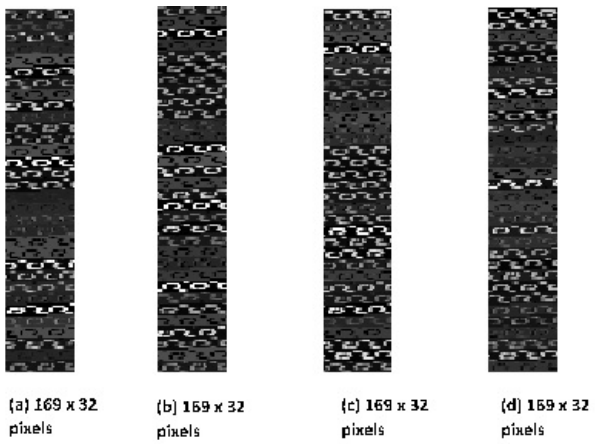

Figure 3.6: Visualization of the stacked feature maps' neuron activation states at layer 3 with different no. of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

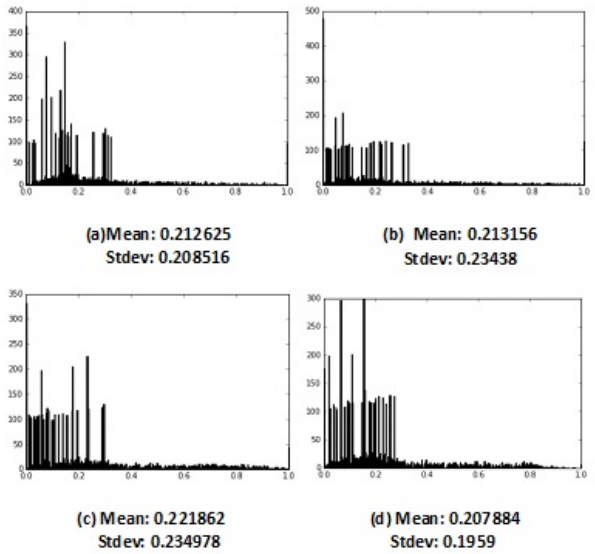

Figure 3.7: Visualization of the histogram of the stacked feature maps at layer 3 with different no. of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

to (c) show there are many inactivated neurons concentrated at the zero gray level cluster. There is no significant difference seen among their distributions, although the distributions are more evenly spread out as N_n increases from 40 to 640. But for Fig. 3.7(d), we can see that there are lesser inactivated neurons concentrated at the zero gray level cluster. Hence, the information representation ability at $N_n = 2560$ is better in information representation for our Convolutional Autoencoder model.

Next, the first estimate of the source's entropy in the neighboring 169 neurons for each neuron in the same feature map at layer 3 according to a 2-D stacked feature map is studied, as shown in Fig. 3.8. The results show the number of high entropy values growing as the number

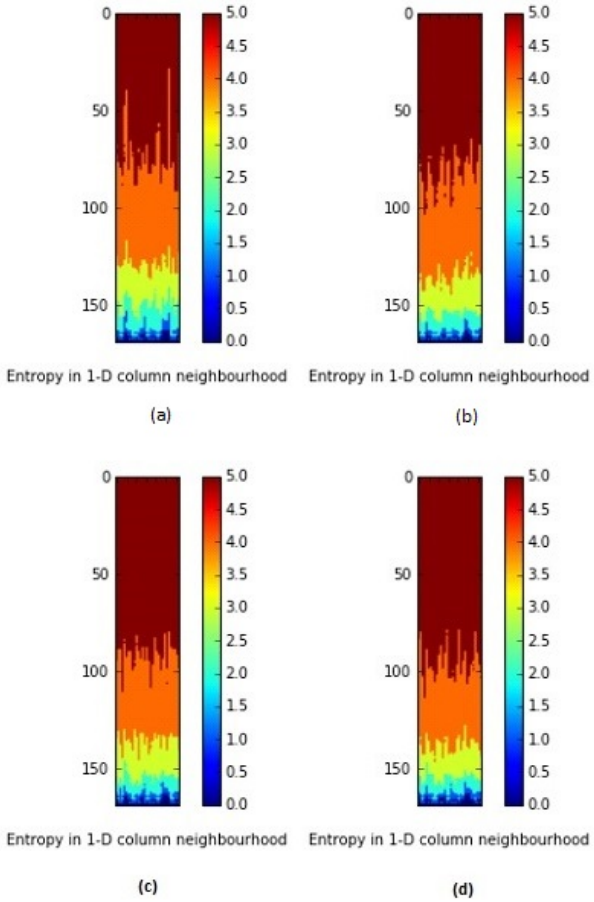

Figure 3.8: Visualization of the 169 neighborhood's first estimated entropy values of the stacked feature maps' neurons at layer 3 with different no. of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

of neurons N_n increases from 40 to 2560 at the bottleneck layer, as shown from Fig. 3.8(a) to (d). Fig. 3.8(a) shows the least number of high entropy values (computed from Equation (3.6)) due to a large number of inactivated neurons with the probability $p_r(r_k)$ values close to 1. Hence from the many high entropy values as shown in Fig. 3.8(d), $N_n = 2560$ has better information representation ability.

Similarly, we can employ the same techniques to visualize the activation of the neurons at the bottleneck layer (layer 4). The results are shown at Figs. 3.9 to 3.11. Our results show that almost all neurons are fully activated to represent the bottleneck information, with $N_n = 40$ in Fig. 3.9(a). As N_n increases from 40 to 2560, there are more neurons' activation with lesser saturation observed, in particular so for $N_n = 2560$ which shows very few highly saturated neurons at the rightmost of the grey level distribution. This may explain the improvement in training and generalization (validation) losses (measured in MSE), as shown in Fig. 3.5 previously, as N_n is increased from 40 to 2560 for Model 1.

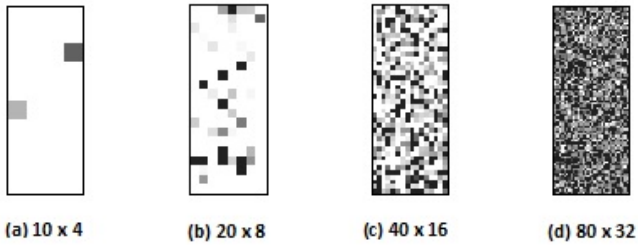

Figure 3.9: Visualization of the stacked feature maps' neuron activation states at layer 4 with different no. of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

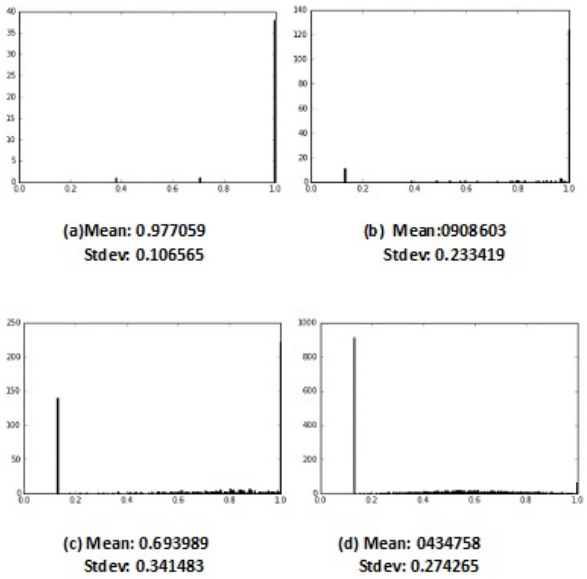

Figure 3.10: Visualization of the histogram of the stacked feature maps at layer 4 with different no. of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

This is further supported by the first estimate of the source's entropy in the neighboring neurons for each neuron in the same feature map at layer 4 presented in a 2-D form in Fig. 3.11. Fig. 3.11(a) shows many low entropy values due to many highly saturated neurons with $p_r(r_k)$ close to one, with the first estimated entropy values close to zero. As the entropy values get larger from Fig. 3.11(a) to (d), there are lesser low entropy values seen and the information captured is more clustered. This result is consistent with the conclusion from Fig. 3.8, which shows that a larger number of neurons, N_n, will result in better information representation ability. Hence, from the many high entropy values as shown in Fig. 3.11(d), it is concluded that $N_n = 2560$ will have better information representation ability.

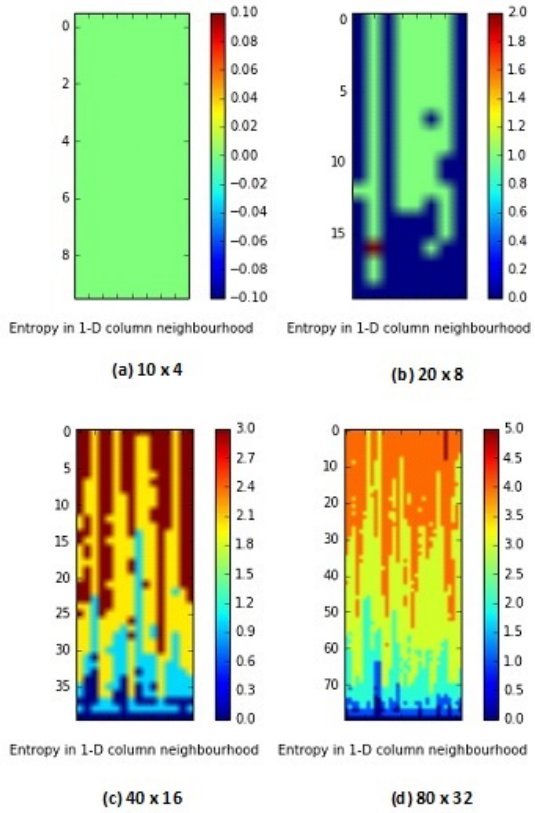

Figure 3.11: Visualization of the neighborhood's first estimated entropy values of the stacked feature maps' neurons at layer 4 with different no. of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

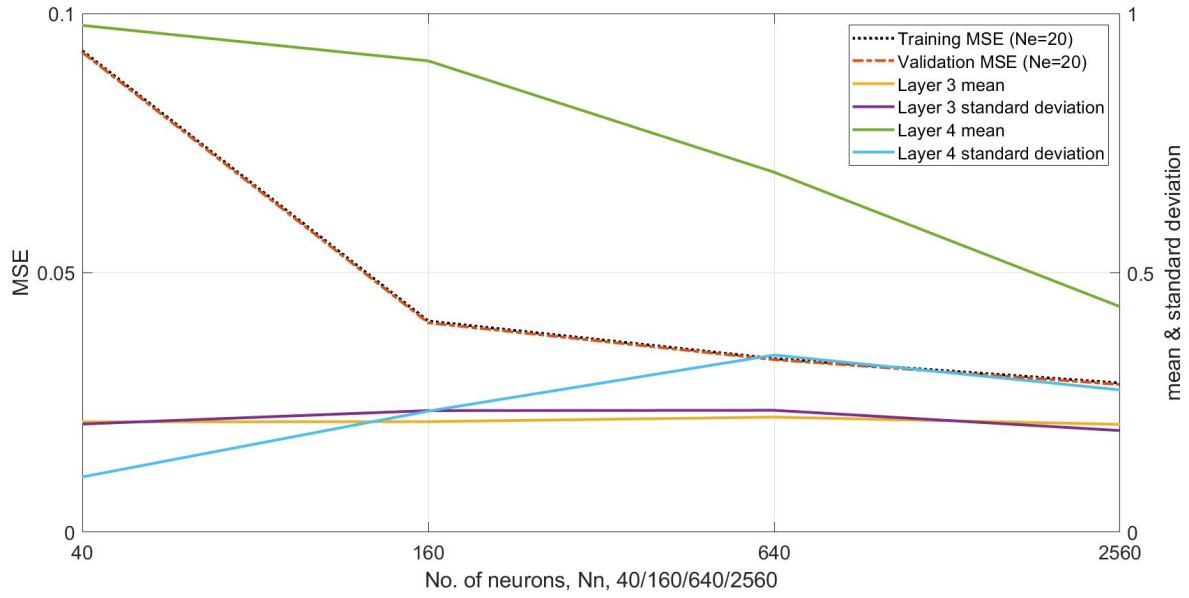

Figure 3.12: Comparison of the image reconstruction results (measured by MSE), and the mean and standard deviation values of the 2-D neurons' activation maps, as the number of neurons of the encoded layer, N_n, is varied from 40 to 2560, at both layer 3 and 4.

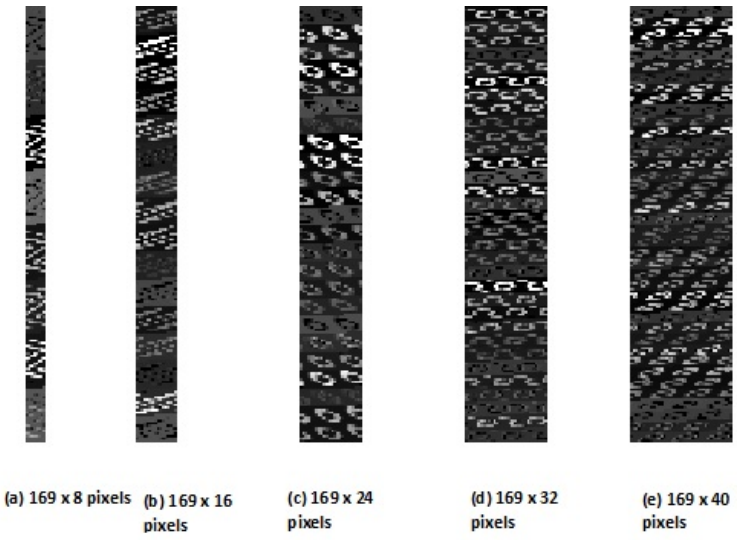

Figure 3.13: Visualization of the stacked feature maps' neuron activation states at layer 3 with different no. of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; and (d) $N_m = 32$; and (e) $N_m = 40$.

3.3.3 No. of Feature Maps, N_m

In line with our results of varying N_n, we believe that as N_m increases, we should also see better information representation by our CAE. Hence, the same techniques were used to examine the neurons' activation state for the flatten layer and bottleneck layer (layers 3 and 4) with the number of neurons at the bottleneck layer (layer 4) N_n fixed at 1200, and varying the number of feature maps N_m (N_m=8, 16, 24, 32 and 40).

Fig. 3.12 shows the comparison of the reconstruction results (measured in MSE) and the mean and standard deviation values of the 2-D neurons' activation maps, as the number of neurons of the bottleneck layer, N_n, is varied from 40 to 2560, at both layers 3 and 4. The reduction in the MSE values as N_n increases, as shown in Fig. 3.12, suggests that information representation ability at $N_n = 2560$ is better for our Convolutional Autoencoder model.

As shown by the results in Figs. 3.13 to 3.19, the visualization of the stacked feature maps' neuron activation states, the histograms of the stacked feature maps, the entropy maps and the MSE results at layers 3 and 4 (for different no. of feature maps) have all shown that N_m=32 or more will have better information representation ability as there are more activated neuron's structure or patterns and high entropy values, and lesser inactivated neurons concentrated at the zero gray cluster, as the number of feature maps N_m increases from 8 to 40. The results are consistent with our previous discussions.

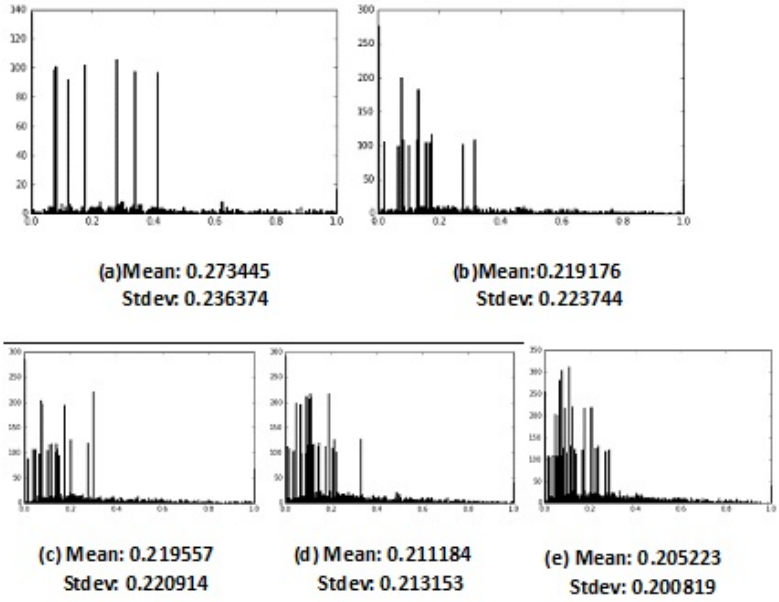

Figure 3.14: Visualization of the histogram of the stacked feature maps at layer 3 with different no. of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; (d) $N_m = 32$; and (e) $N_m = 40$.

3.3.4 Robustness Check

In this section, the formulation of the network parameterization guidelines of the Convolutional Autoencoder will be verified using a more complex small image data set, the Street View House Numbers (SVHN) data set. It helps us check the robustness of our hypothesis that the Convolutional Autoencoder is capturing useful information and being utilized in an efficient manner by setting N_n of at least 2560, N_m of at least 32 and N_e of at least 20. Fig. 3.20 shows our CAE's image reconstruction results based on the SVHN data set. It can be seen that N_n of 2560 produces the highest quality reconstructed image and is most recommended. All the results based on our methodology arrive at the same conclusion when varying N_n and N_m.

3.4 Conclusion and future work

In conclusion, the visualization of neurons' activation and entropy at layer 3 and 4 of our CAE model have proved that having N_e=20, N_n=2560 and N_m=32 will represent information well for a small input image size of 32x32 pixels.

In this chapter, we have presented our experimental setup and results of studying a proposed methodology to parameterize our CAE. The effectiveness of the methodology is verified both qualitatively and quantitatively using small input images, i.e. the standard MNIST data set. The

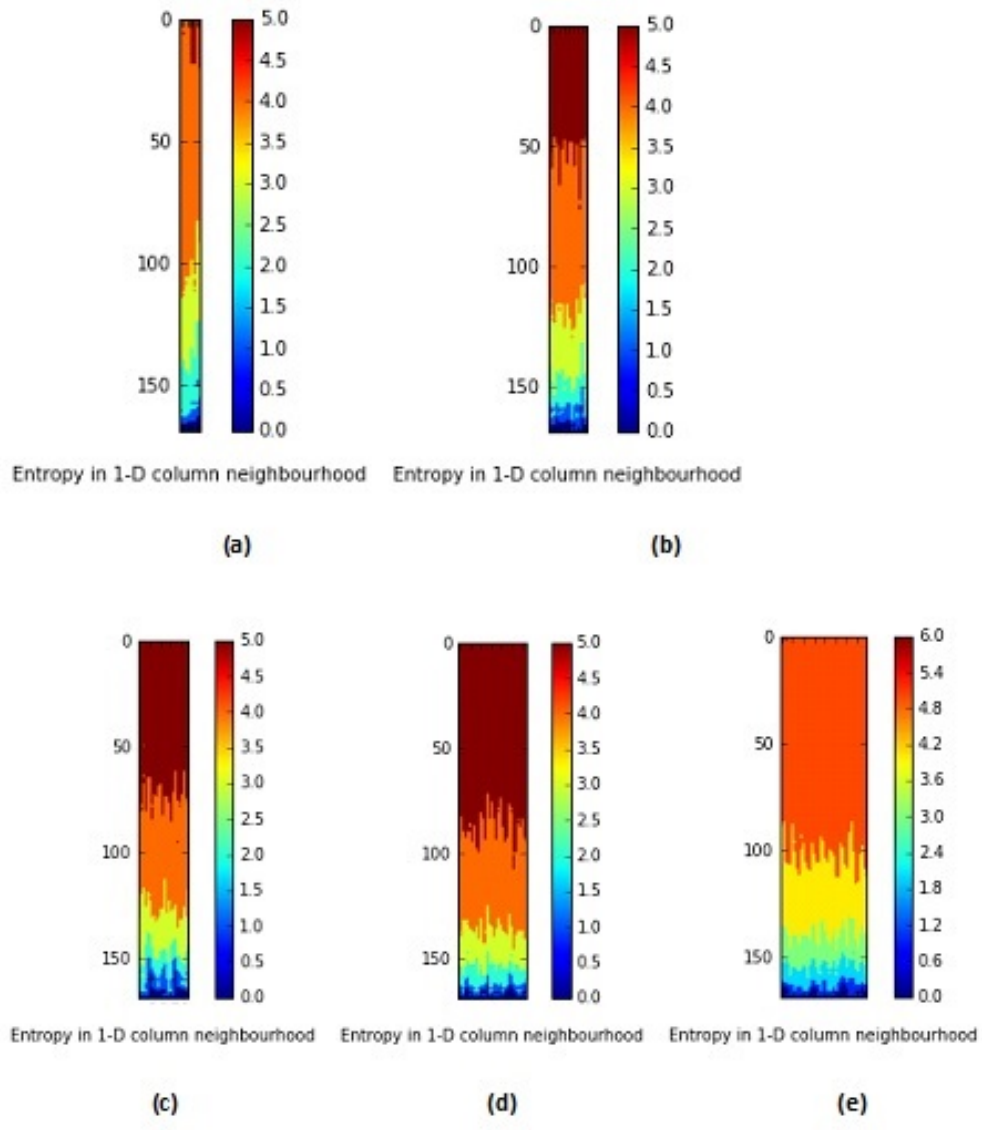

Figure 3.15: Visualization of the 169 neighborhood's first estimated entropy values of the stacked feature maps' neurons at layer 3 with different no. of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; (d) $N_m = 32$; and (e) $N_m = 40$.

Figure 3.16: Visualization of the stacked feature maps' neuron activation states at layer 4 with different no. of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; (d) $N_m = 32$; and (e) $N_m = 40$.

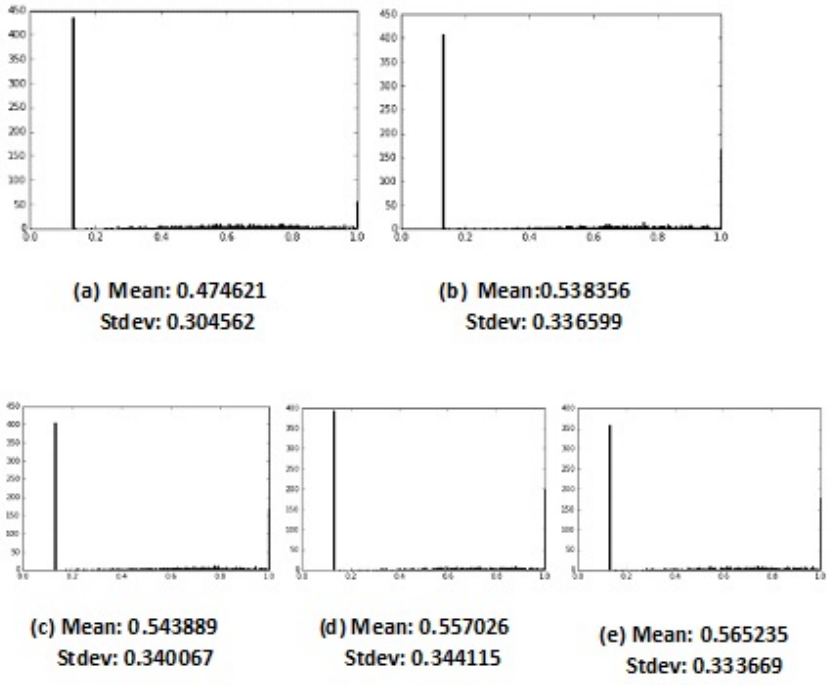

Figure 3.17: Visualization of the histogram of the stacked feature maps at layer 4 with different no. of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; (d) $N_m = 32$; and (e) $N_m = 40$.

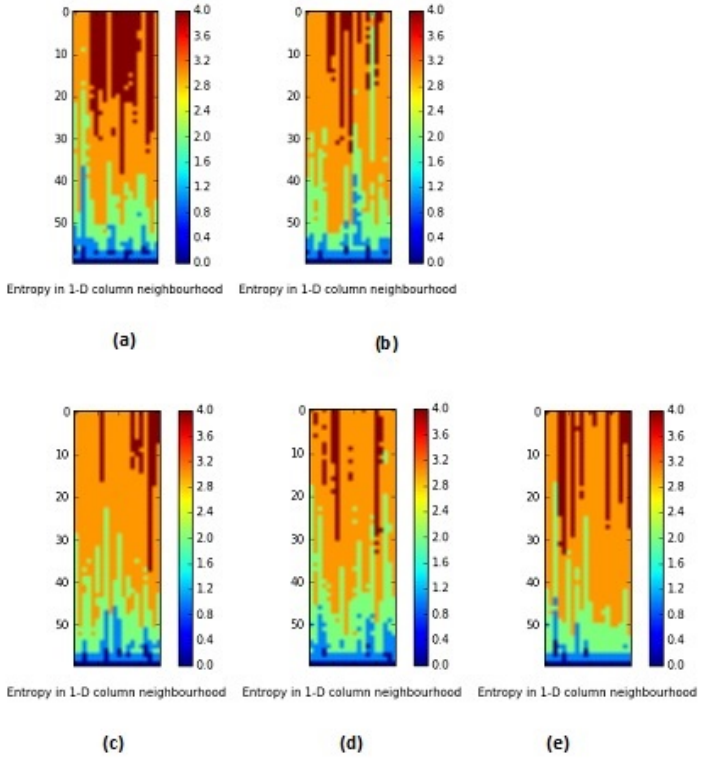

Figure 3.18: Visualization of the neighborhood's first estimated entropy values of the stacked feature maps' neurons at layer 4 with different no. of feature maps with N_n fixed at 1200: (a) $N_m = 8$; (b) $N_m = 16$; (c) $N_m = 24$; (d) $N_m = 32$; and (e) $N_m = 40$.

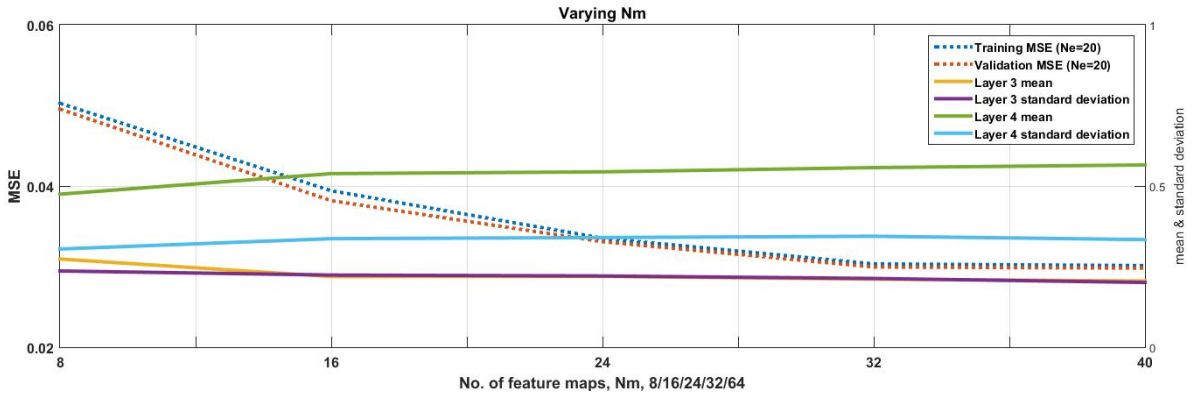

Figure 3.19: Comparison of the image reconstruction results (measured by MSE), and the mean and standard deviation values of the 2-D neurons' activation maps, as the number of feature maps of the bottleneck layer, N_m, is varied from 8 to 40, at both layer 3 and 4.

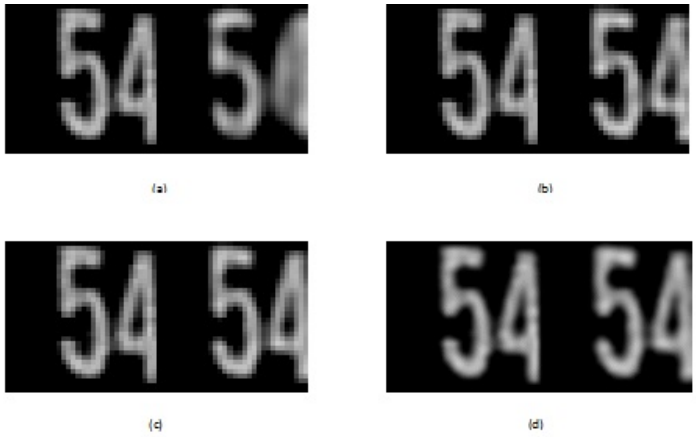

Figure 3.20: Input image and reconstructed 32×32 results of the SVHN data set with different no. of neurons at the bottleneck layer: (a) $N_n = 40$; (b) $N_n = 160$; (c) $N_n = 640$; and (d) $N_n = 2560$.

Street View House Number (SVHN) data set is used to conclude our network parameterization study on the CAE by checking the robustness of our methodology.

The results of the parameterization study of our CAE for small input images like in MNIST (28×28 pixels) and the SVHN (32×32 pixels) data sets have shown that having 2560 neurons at the hidden layer (bottleneck layer) and 32 convolutional feature maps at both the encoder and decoder can result in optimum reconstruction performance for our single conv/pool and deconv/depool layers' Convolutional Autoencoders. Our results show that using both the quantitative (MSE) and the qualitative (2D visualization of the neurons' activation, histogram statistics and estimated source entropy at the layer just before the bottleneck layer and the bottleneck layer itself) analysis methodology, as proposed by this work, can provide a good framework for deciding the optimum parameter values of the Convolutional Autoencoders to provide good representation of the input image.

As shown in our results, increasing the number of network layers for small image data sets like the MNIST and SVHN is not as effective as our single-layer model CAE. Moving forward, we may extend the study of our Convolutional Autoencoders by using a larger image's data set to look at the effect of varying the no. of network layers on the parameterization study of the CAE in the future. But for our research purpose of image reconstruction under visual disruption by rain, this study has already provided a good background understanding on how to optimize our proposed CycleGAN networks for a selected input rain image size for reconstruction purpose.

Chapter 4

Generation of a rain-free image using CycleGAN

As mentioned in Chapter 2, the success of GANs [4] in image generation [53], [54] has prompted many researchers to look at this approach as a more promising approach to remove rain completely from an image without introducing unwanted distortion. Among the early GANs methods introduced for rain removal is the Image De-raining Conditional Generative Adversarial Network (ID-CGAN) [5], a rain removal approach based on the general purpose Conditional Generative Adversarial Network (CGAN).

Similar to GANs which has a generative model, G and a discriminative model, D, the CGAN is introduced for general purpose image-to-image translation tasks such as mapping a semantic label to a scene image, translating an image with object outlines to a color photo, and so on [54]. The advantage of the CGAN is the elimination of the need for special, hand-crafted loss function to be specified for every image-to-image translation task. Hence, the ID-CGANs is using the CGAN framework to directly tackle the single-image rain removal problem by learning a forward mapping from a domain with rain (X) to another domain that is rain-free (Y), using the discriminative model in CGAN to guide the network's learning process to ensure that the generated rain-free images by the generative model are indistinguishable from the ground truth or rain-free counterparts by the discriminative model. However, for the ID-CGANs, the training images are required to be made up of both rain and rain-free (ground truth) image-pairs for its network training.

Thus, the first motivation behind this research is to use another GANs method, the proposed CycleGAN, which is an unpaired image-to-image translation technique that doesn't require such image-pairs for network training, to remove rain distortion. CycleGAN has been successfully used for image style transfer [8], without the need of image-pairs for training. Its network archi-

tecture details, followed by a result comparison with the ID-CGAN, is discussed in this chapter. Most material presented and discussed are published in the ECCV 2018 Workshop Proceedings [2].

4.1 Background

The CycleGAN method [8] follows the recent framework of the CGAN [54] to perform image-to-image translation by learning the mapping across different domains, without the use of image-pairs. Before the CycleGAN, other recent unpaired image-to-image translation approaches such as the CoupledGANs (CoGANs) [79] and cross-modal scene networks [80] are also proposed to learn the shared representation across the two domains. Such unpaired framework has also been extended with a combination of VAEs and GANs [81] to learn the common representation across different domains. Unlike these approaches, the CycleGAN is a general image-to-image translation approach that uses the idea of transitivity to supervise the learning of both forward and backward mappings [82], [83]. It uses the similar cycle-consistency loss to push both forward-backward mapping functions to be consistent with each other [8].

In this chapter, the CycleGAN's application in image-to-image translation tasks has been intuitively extended to removing rain disruption from an image as the CycleGAN has two generative and two discriminative models which can be trained to perform a better forward image-mapping from a domain with rain (X) to another domain that is rain-free (Y), without the need of image-pairs [2]. Hence, the CycleGAN is discussed based on how it can remove real rain disruptions from images, from both theoretical and experimental aspects. Our qualitative results show that it can be used as a practical image's style transfer approach for rain removal, as it is capable of transferring an image with rain to an image that is rain-free, without the use of rain training image-pairs. This is important as natural or real rain images don't have their corresponding image-pairs that are rain-free. For a quantitative measure of the effectiveness of image reconstruction with rain removal, the Natural Image Quality Evaluator (NIQE) will be introduced. This concludes Chapter 4.

4.2 CycleGAN's Network model, parameter and training data set

The CycleGAN is a practical image's style transfer approach that falls into the unpaired cat-

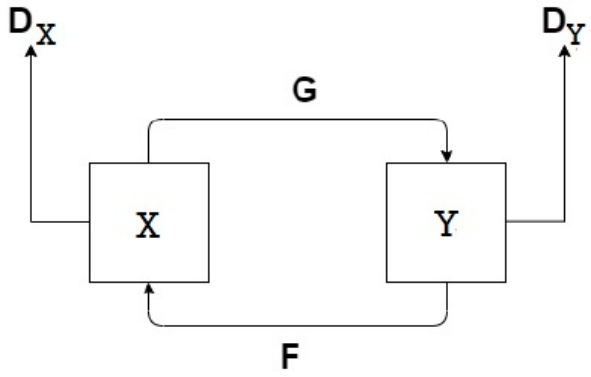

Figure 4.1: The CycleGAN model is made up of two adversarial generators $G(X)$ in the forward-mapping direction and $F(Y)$ in the reverse-mapping direction. Their associated adversarial discriminators are D_Y and D_X respectively.

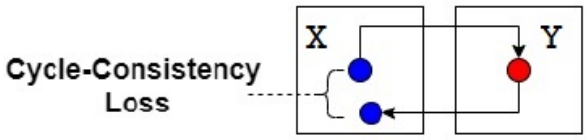

Figure 4.2: The cycle-consistency loss ensures that the results of the forward image-mapping function $G(X)$ (from blue to red), followed by a reverse-mapping function $F(Y)$ (from red to blue), can be as similar as possible to the initial input.

egory, which is capable of transferring an image with rain to an image that is rain-free, without the use of training image-pairs. The CycleGAN is ideal for our rain removal purpose because it can learn the rain statistics for removing rain, by using an image-mapping function $G(X)$ to translate the input rain images from the rain domain (X) to the generated rain-free images in the rain-free domain (Y), such that they are very similar to the statistics of the rain-free images. This can be achieved by having two generators and two discriminators in the CycleGAN, and training the CycleGAN using two types of losses: General Adversarial Networks (GANs) loss and cycle-consistency loss [8]. This is because the forward-mapping is an under-constrained problem, which has to work together with an inverse function $F(Y)$ to translate it back to the X domain, using a cycle-consistency loss function to achieve $F(G(X)) \approx X$ [8].

If the training samples are provided in both domains X and Y during the CycleGAN's network training, it aims to learn both forward-mapping and reverse-mapping functions in both domains simultaneously. Figure 4.1 shows this model which is made up of two mappings: Forward-mapping $G(X)$ in the $X \rightarrow Y$ direction together with its associated adversarial discriminator D_Y and reverse-mapping $F(Y)$ in the $Y \rightarrow X$ direction together with its adversarial discriminator D_X. In domain X of the CycleGAN, D_X is responsible for distinguishing between the given X images and the translated images $Y \rightarrow X$, while D_Y aims to discriminate between

the given images in domain Y and the translated images $X \rightarrow Y$. Hence, there are two objective terms for the CycleGAN during its network training, namely the adversarial or GANs loss ($\mathscr{L}_{\text{GAN}}$) and the cycle-consistency loss ($\mathscr{L}_{\text{cyc}}$). The adversarial loss in each domain targets to fit its data generative model to the actual data distributions, while the cycle-consistency loss ensures that the results of the forward image-mapping function $G(X)$ (from blue to red), followed by a reverse mapping function $F(Y)$ (from red to blue), can be as similar as possible to the initial input after training. This is illustrated in Figure 4.2.

The combination of this cycle-consistency loss with GAN losses in both X and Y domains results in the overall CycleGAN loss function as shown below [8]:

$$\begin{aligned}\mathscr{L}(G(X),D_Y,F(Y),D_X) =& \lambda\mathscr{L}_{\text{cyc}}(G(X),F(Y)) \\ &+\mathscr{L}_{\text{GAN}}(X,Y,G(X),D_Y) \\ &+\mathscr{L}_{\text{GAN}}(Y,X,F(Y),D_X),\end{aligned} \tag{4.1}$$

where D_Y is the associated discriminator for $G(X)$ to translate images $X \rightarrow Y$, and the same definition applies to $F(Y)$ and D_X. The cycle-consistency loss is controlled by a constant parameter λ, as shown in Equation (4.1).

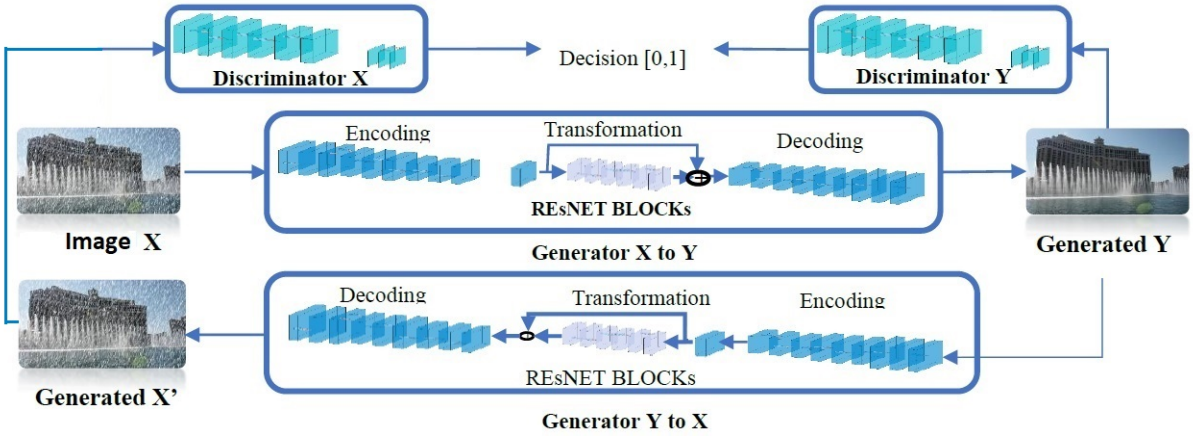

Figure 4.3: The network architecture of the CycleGAN model [8]. Only the image-to-image translation in the forward direction is shown in the figure.

Figure 4.3 illustrates the CycleGAN model [8], which is made up of two generators and discriminators, as discussed before. For our proposed CycleGAN, the generator architecture from the CycleGAN [8] is simplified to accommodate only one input image size (256x256) for the rain-removal CycleGANs network, and the same simplification is adopted for the discriminator architecture as well. It uses the same PatchGANs [8] to learn how to decide whether the image patches overlapped that are of the same size are real image patches, during network training, to reduce the discriminators' number of parameters.

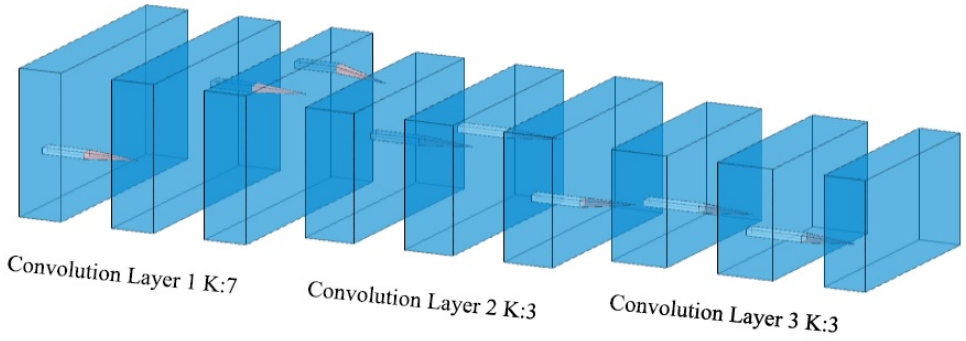

Figure 4.4: The network architecture of the CycleGAN model's encoder.

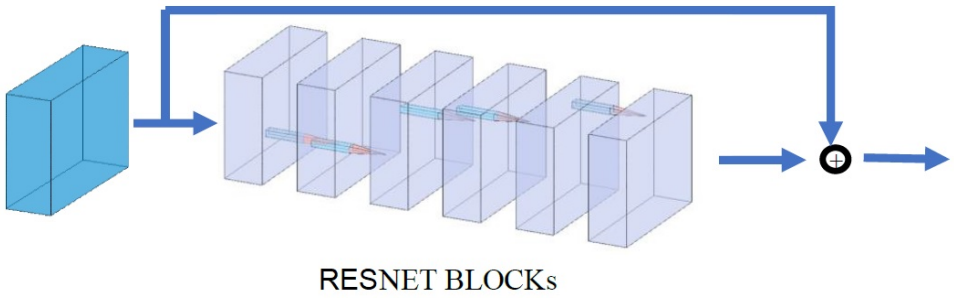

Figure 4.5: The network architecture of the CycleGAN model's transformation stage.

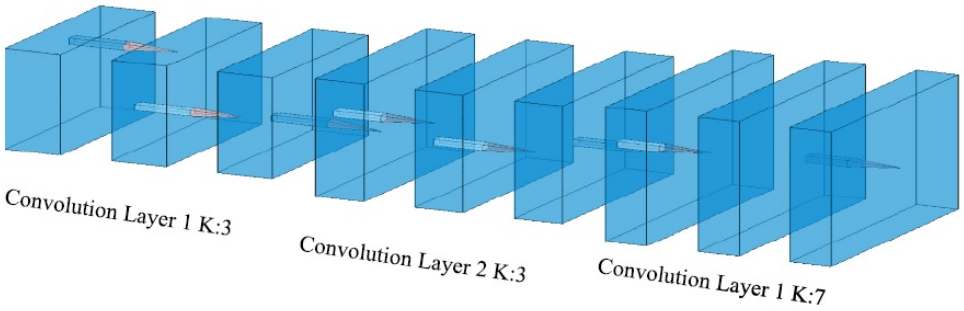

Figure 4.6: The network architecture of the CycleGAN model's decoder.

Each generator consists of 3 network stages: the encoder, transformation and the decoder. For the encoder, as shown in Figure 4.4, different features from the input RGB images are extracted by the moving filter kernels across the images. 64 feature maps of 256x256 pixels and the 64 filter kernels of 7x7 pixels are used in the first convolutional layer. The higher-level features are then extracted by the subsequent convolutional layers. The transformation stage, as shown in Figure 4.5, converts the features extracted from the encoder into a latent feature vector. There are 6 RESNET (residual) blocks in between its encoder and decoder blocks, which are adopted to accommodate input image sizes of 256x256 pixels. Rain removal and background restoration are performed by the RESNET blocks of the generators, which can help to increase

the feature extraction ability of the CycleGAN. The skip connections of the RESNET blocks of the generator can help to prevent information loss while facilitating gradient propagation in such autoencoder-like generator's network structure during training. The final decoder stage of the generator, as shown in Figure 4.6, is performing the opposite process of the encoder, using its de-convolution layers. It converts the new feature vector from the previous transformation stage back to the output RGB reconstructed image.

A Nvidia GTX 1070 is used for training our proposed CycleGAN network, and the algorithm is written in Pytorch [84]. The learning rate is set to 0.0002 for 200 epochs of training, using a λ value of 10 (see Equation 4.1), for all experiments.

The CycleGAN approach does not require rain image-pairs for its network training. But for a consistent comparison of the proposed CycleGAN rain removal approach with the ID-CGAN, the same 700 rain image-pairs provided by [5] can be used for their training after being resized to 256x256 by the CycleGAN. In addition, rain images with real distortions are used to test the performance of the CycleGAN, as explained in the next section.

4.3 Testing and evaluation results

Existing objective image quality measures require some measurement of the closeness of a test image to its corresponding reference (ground truth). These measures are either based on mathematically defined measures such as the widely used mean squared error (MSE), peak signal to noise ratio (PSNR), universal quality index (QI) [85] and structure similarity information measures (SSIM) [86], or the human visual system (HVS) based perceptual quality measures such as the visual information fidelity (VIF) [87]. Most existing literature used such generated image-pairs for quantitative comparison of their results.

Based on the five types of rain distortions listed in Chapter 2, the performance of real rain removal performance is evaluated. As ground truth reference images are not available in the test data set, the performance of the proposed CycleGAN and the ID-CGAN is evaluated visually. Although the corresponding rain-free (ground truth) images for a quantitative comparison of results using the mentioned objective image quality measures are not available, the proposed visual analysis methodology to compare the removal results using real-world rain images (Type I to V) can reveal the superiority of the proposed method, as shown in Section 4.3.1 to 4.3.5.

For a fair comparison, both the ID-CGAN and the CycleGAN were trained on the same set of 700 synthesized rain-and-ground-truth image-pairs, provided by [5]. Subsequently both

networks were tested on real rain images which fall broadly under these five types of real rain distortions. A comparison of the rain removal performance between the CycleGAN and the ID-CGAN are discussed in this chapter.

4.3.1 Type I: Different severity of rain streaks

As discussed in Chapter 2, rain drops show a wide distribution of size, volume and rate [6]. Hence, it is expected that rain properties affect the appearance of rain streaks in a wide variety of manner [6], [38]. Light rain streaks below 1 *mm* in rain drop's diameter are common; they are less visible and blur the background scene in a rain image. Heavy nearby rain streaks above 1 *mm* in rain drops' diameter are more visible and reduce the visibility by occluding the background scene. Severe distant rain with large rain drops' diameter show that their individual rain streaks are overlapping and cannot be seen, occluding the background scene in a misty manner [38]. Two Type I sample images are shown in Figure 4.7 (a) and (c) with different severity of rain streaks, and their corresponding magnified rain streaks are shown in Figure 4.7 (b) and (d) respectively. The rain removal results using the ID-CGAN and CycleGAN are shown in Figure 4.7(e) to (h) and Figure 4.7(i) to (l) respectively. The subsequent figures for other types of rain distortion are presented in the similar manner.

As shown by the results, CycleGAN removed the rain streaks of different severity equally well, while the ID-CGAN was unable to remove the rain streaks and many original rain streaks remained, especially for heavy rain. In addition, it was observed that the contrast of background scenes was enhanced with the ID-CGAN.

4.3.2 Type II: Different camera settings

As discussed in Chapter 2, camera parameters such as exposure time affect the visibility of rain [6]. By comparing rain images taken with a short exposure time of 1 *ms* and normal exposure time of 30 *ms*, the authors discovered that the short exposures produced stationary and bright raindrops and they do not appear transparent. However, at long exposures, due to fast motion, raindrops produce severely motion-blurred rain streaks. Type II distortion is typically due to a short exposure time that increases rain's visibility and produces stationary, bright and non-transparent raindrops. Due to the high speed of rain, rain drops appear as bright spheres occluding the background scene. Figure 4.8(a) to (d) shows examples of such rain degradation. The rain removal results using the ID-CGAN and CycleGAN are as shown in Figure 4.8(e) to (h) and Figure 4.8(i) to (l) respectively.

Figure 4.7: Type I distortion for different severity of rain streaks as shown in Figure 4.7(a) to (d), and their rain removal results by the ID-CGAN and CycleGAN were shown in Figure 4.7(e) to (h) and 4.7(i) to (l) respectively.

It was observed that the CycleGAN was able to remove the bright rain spheres well, although it was not trained to remove such type of defect. In comparison, the ID-CGAN was unable to remove such defect and left behind many bright rain spheres in the zoomed regions-of-interest. This may be because such real rain defect is not covered in the synthetic training data set. In addition, ID-CGAN is known to suffer from white-round rain streaks due to the high-level features from CNN network inherently enhancing white round particles [5]. Hence, the CycleGAN performed better than the ID-CGAN for the rain distortions in Type II.

4.3.3 Type III: Indoor rain images behind a glass window

Since a glass window affect the radiance or scene properties of an image, it affects the visibility of rain streaks as shown by Equation (2.7) in Chapter 2. Hence, rain streaks and its background scene viewed behind a transparent or translucent glass window should be considered separately as a different defect. The adherent rain water behind the glass window also occludes the rain streaks and its background scene. The reflection of light by, and the refraction of light through, the adherent water stain behind the glass window produces very low brightness scene captured by a camera or observed by a human, as shown in Figure 4.9 (a) to (d).

Figure 4.8: Type II distortion for different camera setting as shown in Figure 4.8(a) to (d), and their rain removal results by the ID-CGAN and CycleGAN were shown in Figure 4.8(e) to (h) and 4.8(a) to (l) respectively.

Figure 4.9(e) to (h) and Figure 4.9(i) to (l) show the results of removing rain using the ID-CGAN and CycleGAN respectively. As shown by the results, the ID-CGAN would brighten the adherent water drops as shown by Figure 4.9(e) and (f), regardless of the sizes of the drops. This may be due to the same reasons, as discussed in Type II. In comparison, the CycleGAN does not show such defects, as shown in Figure 4.9 (i) and (j). Also, as shown in Figure 4.9 (g) and (h), although the ID-CGAN managed to enhance the contrast of the low brightness background scenes, its contrast was still not as good as the CycleGAN, as shown in Figure 4.9 (k) and (l). This may be due to the nature of the learning of the cycle-consistency objective that ensures the results of the forward image-mapping function $G(X)$, followed by a reverse-mapping function $F(Y)$, can be as similar as possible to the initial input, in such low brightness situations. Although none of the algorithms was trained to remove such defects, the CycleGAN has shown that it is more superior to remove such defect and manage to enhance the contrast of the low brightness scene well.

4.3.4 Type IV: Rain velocity reduction and splashing at obstructing structures

Figure 4.9: Type III distortion for scene behind a glass window as shown in Figure 4.9(a) to (d), and their rain removal results by the ID-CGAN and CycleGAN were shown in Figure 4.9(e) to (h) and 4.9(i) to (l) respectively.

For free falling rain drops where the raindrops' velocities were suddenly reduced by a structure (e.g. the roof) of a building as shown in Figure 4.10(a) to (d), the rain streaks appeared almost stationary and bright as they are not falling at terminal velocities. This kind of distortion consists of both the usual motion blurred long rain streaks as well as the brighter and shorter streaks, as shown in Figure 4.10(a) to (d).

It is illustrated in Figure 4.10(i) to (l) that the CycleGAN was able to remove both fast and slow rain streaks, while the ID-CGAN was only able to remove the faster rain streaks. This may be due to the same reasons, as discussed in Type II. As shown in Figure 4.10(f) and (h), most of the slow rain streaks remained, in the case of the ID-CGAN. Based on these observations, the CycleGAN is more robust for a wide range of real rain defects, as compared to the ID-CGAN.

4.3.5 Type V: Splashing and accumulation of rain water on ground surface

Rain water tends to accumulate on surfaces such as the road surface or the roof of a building. Hence, distortion due to water splashing defect is common in rain images. Figure 4.11(a) to (d) show samples of such rain distortion.

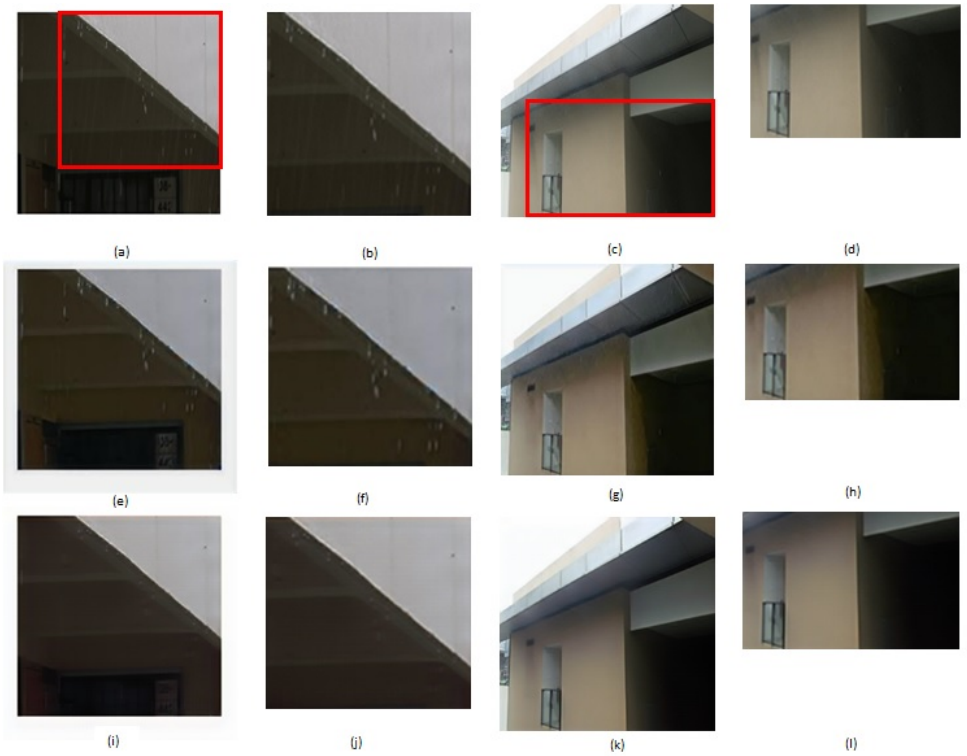

Figure 4.10: Type IV distortion for rain velocity reduction and splashing at obstructing structures as shown in Figure 4.10(a) to (d), and their rain removal results by the ID-CGAN and CycleGAN were shown in Figure 4.10(e) to (h) and 4.10(i) to (l) respectively.

Figure 4.11(e) to (h) and Figure 4.11(i) to (l) show the results of removing rain using the ID-CGAN and CycleGAN respectively. As shown by the results, the CycleGAN was able to remove water splashes and ripples of water accumulated on the surface completely. As shown in Figure 4.11 (e) and (f), the ID-CGAN has introduced many white artifacts. This was expected as the ID-CGAN was not trained to remove such a type of defect, But the CycleGAN was able to remove the defect very well with good contrast, as shown in Figure 4.11 (i) and (j). The ID-CGAN also created a large patch of bright defect on the accumulated surface water, as shown in Figure 4.11 (g) and (h). Such artifacts are not observed in the CycleGAN, as shown in Figure 4.11 (k) and (l). Based on the above observations, the CycleGAN has shown to be superior to the ID-CGAN in removing Type V rain distortion.

4.4 NIQE: a quantitative measure for real rain analysis

In the previous section, we have compared the CycleGAN's rain removal results with the state-of-the-art ID-CGAN qualitatively, using five types of real rain phenomena. To achieve a

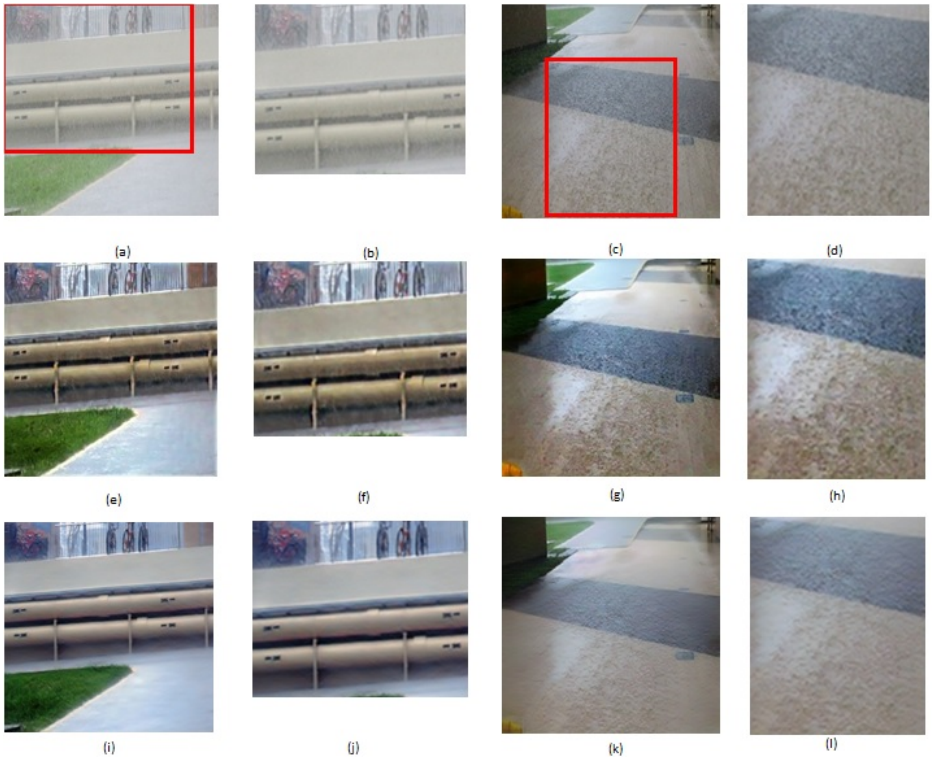

Figure 4.11: Type V distortion for splashing and accumulation of rain water on ground surface as shown in Figure 4.11(a) to (d), and their rain removal results by the ID-CGAN and CycleGAN were shown in Figure 4.11(e) to (h) and Figure 4.11(i) to (l) respectively.

more meaningful comparison, we will need to compare them quantitatively. Since there are no ground truth images for a quantitative comparison, we need to use a different quantitative approach to evaluate their rain removal capabilities. The Natural Image Quality Evaluator (NIQE) can be used as a quantitative measure [7] to analyze rain removal results as it can predict the quality of an image without relying on any prior knowledge of the image's distortion. In other words, it is a completely blind image assessment measure [7]. It can be called an 'opinion unaware' measure that is independent of the type of distortion, unlike other top-performing IQA models such as the BRISQUE (Blind/Reference less Image Spatial Quality Evaluator) that require some training samples or human perception to anticipate an image's distortion level [111]. Such robust methodology is ideal for our real rain removal's quality evaluation. This is because popular image assessment models such as the structural similarity index measure (SSIM) require the ground-truth images to guide the quantitative comparisons, which is not available, as illustrated in Figure 4.12.

NIQE is derived based on the commonly used natural scene statistics (NSS) model [7]. It learns natural images' statistics by first selecting image patches to compare them with the NSS model, which are the most distortion-free images. Using the selected patches to characterize the model, it compares both 'local sharpness' with a defined 'threshold', for model's charac-

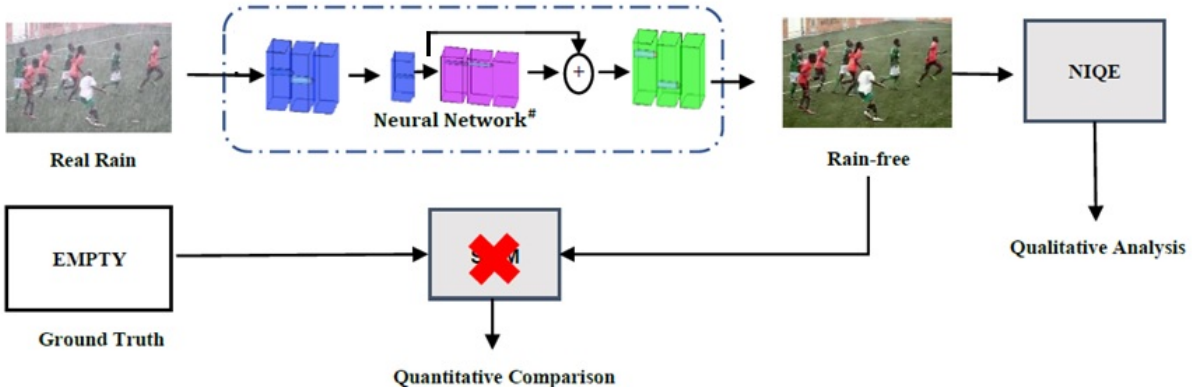

Figure 4.12: The NIQE image assessment model does not require a ground-truth image for quantitative comparison.

terization purpose [7]. For our real rain removal's quantitative comparison purpose, the lower the NIQE score achieved by the reconstructed images of the rain removal algorithms, the better their rain removal capabilities are, as they are closer to the statistics of the natural images without rain. Hence, we will evaluate all rain removal algorithms quantitatively using their NIQE scores, to decide which is the better algorithm in this research, in Chapter 6.

4.5 Conclusion and future work

Despite the success of the CycleGAN in outperforming all other state-of-the-art techniques, there is still some residual frequency artifacts observed in the CycleGAN rain-removed images. This may be addressed by a multi-scale spatial decomposition of the images before applying the CycleGAN, to remove the low frequency rain distortions. The CycleGAN approach can be combined with either the Laplacian Pyramid or the Wavelet representation, to form multi-scale network architectures to improve the CycleGAN's rain removal performance, as shown in the subsequent chapters.

Chapter 5

Rain removal across spatial frequencies using MS-CycleGANs

In this chapter, the remaining gaps on rain removal using the CycleGAN [2] will be addressed. By incorporating multiple CycleGANs into the proposed multi-scale pyramid framework, the MS-CycleGANs can learn the characteristics of translating between the rain and rain-free domain at different spatial frequency scales, which is essential for removing the individual frequency components of rain while preserving the scene details.

The details of the MS-CycleGANs' network architectures, together with their results, will be discussed in this chapter. For a fair comparison with the CycleGAN, the proposed multi-scale representations of CycleGAN networks were trained and tested on the same set of rain images used by the ID-CGAN work [5]. The comparison results of the performance between the CycleGAN and the MS-CycleGANs will be shown to demonstrate the superiority of the MS-CycleGANs in removing rain distortions.

5.1 Background

The multi-resolution or multi-scale approach [88] has been very popular for analyzing images at different spatial frequency scales. It models after the scale space theory [89] which is a more complex approach, to process images at different scales. Most of the multi-scale representations used in image processing and analysis focus on utilizing pyramid-like structures to provide a decomposed representation of the input image's features [90], [91]. This is due to the important property of the pyramids which lies in their abilities to convert global input image's features into local features that allows for 'local interaction' of the converted local features at

higher levels or coarser level of the pyramid. Hence, such representation can relate global features with the local features across pyramid levels [88].

By using different pyramid levels to unify an image's structures or local features, we can combine the multiple CycleGANs to form a MS-CycleGANs to learn rain characteristics to remove rain effectively using such signal localization property. Theoretically, the use of pyramid-like representations in a multi-scale framework will aid in improving the rain removal capability of our proposed CycleGAN network. Another way of multi-scale processing is using the scale-space approach. The scale-space approach is a signal decomposition of an image into multiple scales or frequency bands [92]. There is another form of scale-space approach where the multiple frequency scales are defined linearly or continuously [89]. For this approach, we can summarize the following desirable properties of linear scale-space:

- Shift invariance: it means spatial isotropy, where all spatial positions are treated equally.
- Scale invariance: it means spatial homogeneity, all spatial scales are treated equally
- Causality: it means there is no new feature created in scale-space

Although scale-space representation sounds very attractive for our multi-scale rain removal algorithm, a main dispute behind its construction is that if no 'rain priors' information is available on what are the suitable scales for our rain image data set, then the only reasonable method is to signify the input image at various scales or spatial sizes. In the case of a rain image, when there is a change in scale of the image, the amount of rain streak's structure information will also change according to the scales. Hence, in a multi scale or resolution rain removal system, rain streaks might subtend only a few samples on the captured image in terms of a small range of spatial frequencies. As we change the scale of an image, rain streaks' structure will grow in size on the scaled images and a correspondingly larger range of spatial frequencies may be present to describe more 'detail' on its appearance. Hence, the gross structure of rain streaks is resolved in greater detail as we change the image scale, but does not transform radically, and new detail emerges as we increase the size of the representation. This evolution of image structure in terms of more detail appearance with increasing image size forms the core concept of our multi-scale rain removal: to take advantage of the sequential evolution of detail (including rain) over scale. Such processing of image signals in a scale-independent manner can exploit the finest levels of detail, i.e. 'highest acuity information' available, for our rain removal task [93].

In our context, to develop a deep learning rain removal algorithm that detect the edges and remove a specific object, such as a rain streak, the algorithm shall be able to recognize it regardless of whether they occupy the entire field of view or just a small region in the image. In this context, we would like our rain removal algorithm to be able to remove rain regardless of the

image structures' sizes in our training or testing images, at the finest level of structure available within the images. In the final analysis, with a general framework based on multi-scale representation of an image, it is possible to devise a deep learning rain removal algorithm that operate over a range of image scales in a consistent manner. In other words, we filter the rain images in order to extract image's rain features and then learn these features from different training images. However, the types of image feature we extract and learn could practically span a very large range of spatial scales. Furthermore, the feature scales present within the training images are not usually known in advance. In this case, a multi-scale representation in the image pyramid form, which will be discussed in this chapter, would be a more suitable and practical approach for such purpose, as compared to the scale space representation.

In this chapter, the proposed MS-CycleGANs, a multi-scale pyramid based CycleGAN network for rain removal to enhance the performance of the CycleGAN, is first discussed. The network architecture consists of two stages. The first stage involves rain removal in a 3-level pyramid, using three CycleGANs, in a low-to-high rain frequency (coarse-to-fine) manner. In this way, low frequency rain is removed first at the coarse level of the image pyramid by the first CycleGAN. The background details are gradually recovered while lighter rain is being removed, as the image resolution increases, by the second and then the third CycleGAN in the network. In other words, the structure of the background of the scene at the coarse level is recovered first, followed by the fine details of the background which is reconstructed in higher resolution's pyramid levels. The second stage consists of just a CycleGAN, concatenated to the output of the first stage, to remove the residual rain. Therefore, in the MS-CycleGANs, no rain frequency of the image is estimated or the rain disruptions in a particular image region is explicitly distinguished, to prevent under or over rain removal which has happened to many of the other rain removal techniques due to such assumptions.

The main advantage of this method compared to the other pyramid based deep learning techniques lies in that there is no additional residual (input attention map) [94] being used to represent the missing background details or rain regions by the network, which leave the rain removal task that can preserve background details solely to the CycleGANs at their respective pyramid levels. In this manner, multiple CycleGANs are exploited to learn how to tackle rain components at different spatial frequency scales, without the need to target at or guide the network to a particular rain region which may not generalize well with all kinds of rain distortions, in particular the effect of real rain on images [2], as described in Chapter 2.

To validate this proposed method, the MS-CycleGANs is implemented as a 3-level Laplacian pyramid in its first stage, followed by a CycleGAN in its second stage, to form a multi-scale pyramid-based rain removal network. The purpose of its two stages is discussed in this chapter,

from image feature extraction to rain removal capability, to tackle the single-image rain distortion problem. It is worth mentioning that the MS-CycleGANs is capable of learning both rain and rain-free images' statistics in the rain domain and its corresponding rain-free domain, at different frequency scales, without requiring rain and rain-free paired training images.

This chapter provides an overview of the MS-CycleGANs' network architectures, starting from the multi-scale representation framework. The process of building the network involves the iterations of the following few phases:

1. training of MS-CycleGANs model
2. testing of the model through the reconstruction of the respective Laplacian images' at different frequency scales of the pyramid;
3. image quality check and hyper-parameters' adjustments; and
4. re-training and testing of the MS-CycleGANs.

This chapter is concluded by a qualitative comparison of the performance of the MS-CycleGANs to the CycleGAN.

5.2 Multi-Scale Representation

As mentioned in Chapter 1, the proposed MS-CycleGANs method addresses the gaps of the CycleGAN in single-image rain degradation problem by learning a mapping from an input rain image to a rain-free (ground truth) image at different frequency scales. One such possible GANs approach is the LAPGAN, which is a Laplacian pyramid-based framework that uses GANs processing method to extract image features at 6 different frequency bands for image reconstruction [42].

In this section, the image pyramid concept is first discussed, followed by the Laplacian Generative Adversarial Network (LAPGAN) [42]. The main focus is to study the LAPGAN implemented in PyTorch and its image generation results using the Canadian Institute for Advanced Research (CIFAR-10) data set. This is to demonstrate the image feature extraction and processing capability in a multi-scale approach. We end this section with an introduction of the MS-CycleGANs, to be discussed in the following sections of the chapter.

5.2.1 Image Pyramid

As a common form of multi-scale representation, image pyramid representation involves stacking a series of octave-separated images or layers in a consistent manner. A basic image pyramid can be formed by repeatedly stacking the Gaussian-filtered and down-sampled images, with the higher resolution (fine) images at the bottom and lower resolution (coarse) images at the top. As the image at the bottom will always be twice as big as the image that was directly stacked above it, the pyramid layers follow the behavior of octaves [91]. Figure 5.1 illustrates how a typical image pyramid is formed in each layer. To understand the concept of image pyramid, the Gaussian Pyramid, Laplacian Pyramid and its image reconstruction are discussed next.

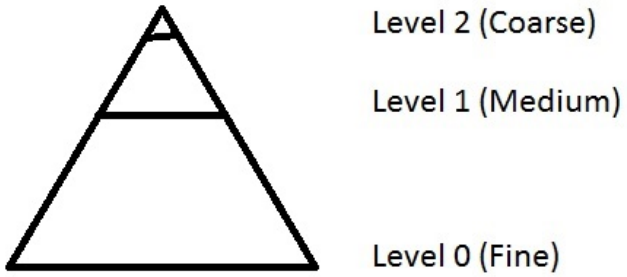

Figure 5.1: An image pyramid: each level has reduced the resolution (width and height) by a factor of two, and henceforth a quarter of the pixels, of its parent level.

Gaussian Pyramid

The Gaussian pyramid of an image can be constructed by applying the steps of Gaussian-smoothing followed by sub-sampling (or down-sampling by reducing image size by a factor of 2) each layer's image iteratively, until the minimum resolution is reached [90]. As shown in Figure 5.2, the down-sampling function reduces both the image size and its resolution. Given G_0 as the input layer and G_N as the top layer of the Gaussian pyramid, for a range of $0 < N \leq 3$, G_3 is assumed to be a result of down-sampling of G_2, through an operation of Downsample$[G_{3-1}]$, as shown below [90]:

$$G_N = Downsample[G_{N-1}] \tag{5.1}$$

This results in a bank of multi-scale low-pass filters in the pyramid.

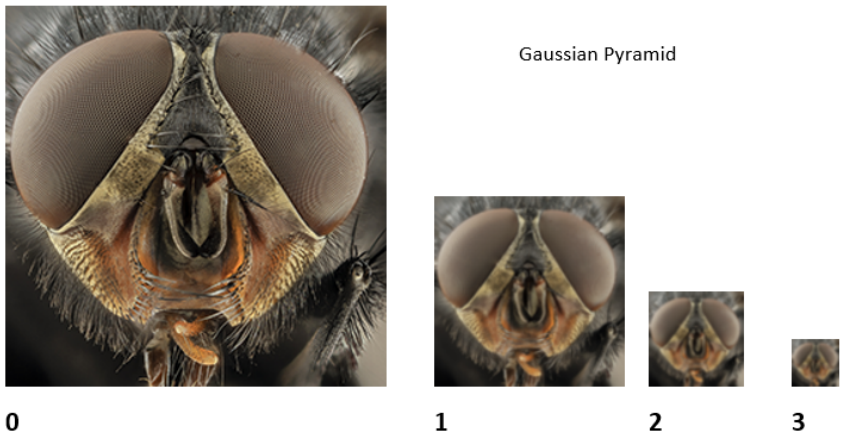

Figure 5.2: A four-level Gaussian pyramid using a bee image as the original input image. Level 0 measures 256x256, level 1 measures 128x128, level 2 measures 64x64 and level 3 measures 32x32 pixels respectively [90].

Laplacian Pyramid

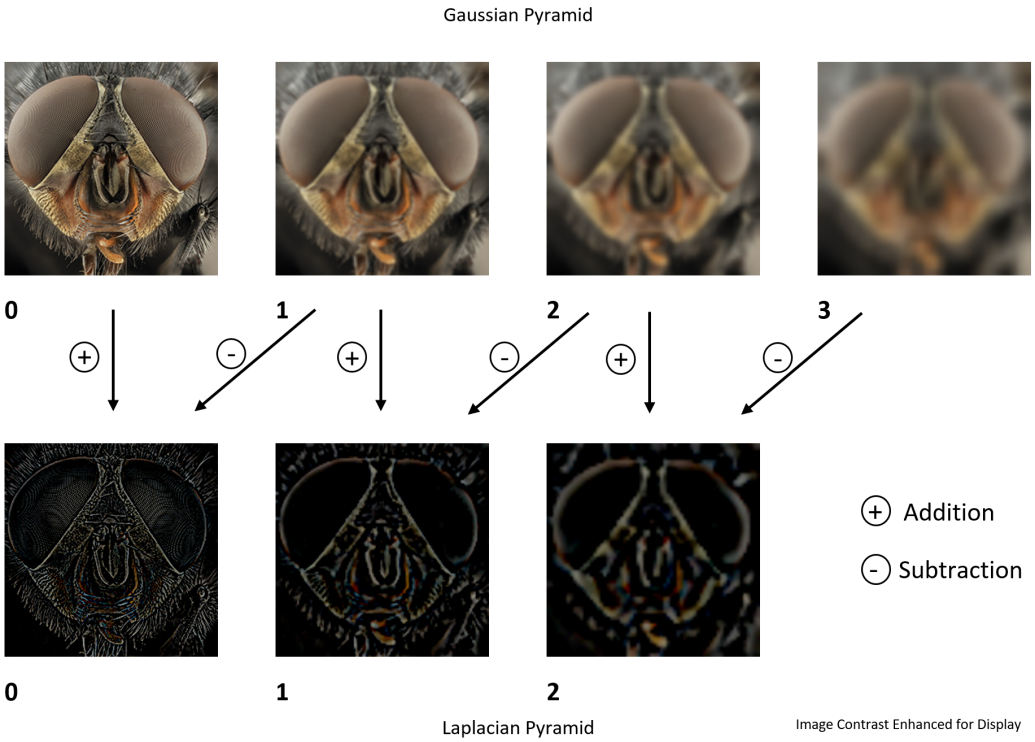

Figure 5.3: The Gaussian pyramid (top) levels up-sampled to the size of the original image, and (bottom) its corresponding levels of the Laplacian pyramid. The Gaussian image shows the low-pass filtered sample of the bee image. The Laplacian images show band-pass filtered samples of the image [90].

The Laplacian pyramid is formed by a recursive-interpolation and up-sampling operations to expand the image size by a factor of 2. The existing image levels of the Gaussian pyramid

is expanded before it is being subtracted from the previous level. This results in the Laplacian image levels of $L_N = G_N - G_{N+1}$ as shown in Figure 5.3. As each level represents the difference in Gaussian images, the image features tend to be enhanced through this operation. By performing summation of the Laplacian pyramid, the original image can be recovered, as discussed in the next section.

Reconstruction

Since the last level in the Laplacian pyramid shown in Figure 5.3 and the last level in the Gaussian pyramid shown in Figure 5.2 are the same image, image reconstruction can be achieved by adding the images from the Laplacian levels of the Laplacian pyramid [95], as shown in Figure 5.4.

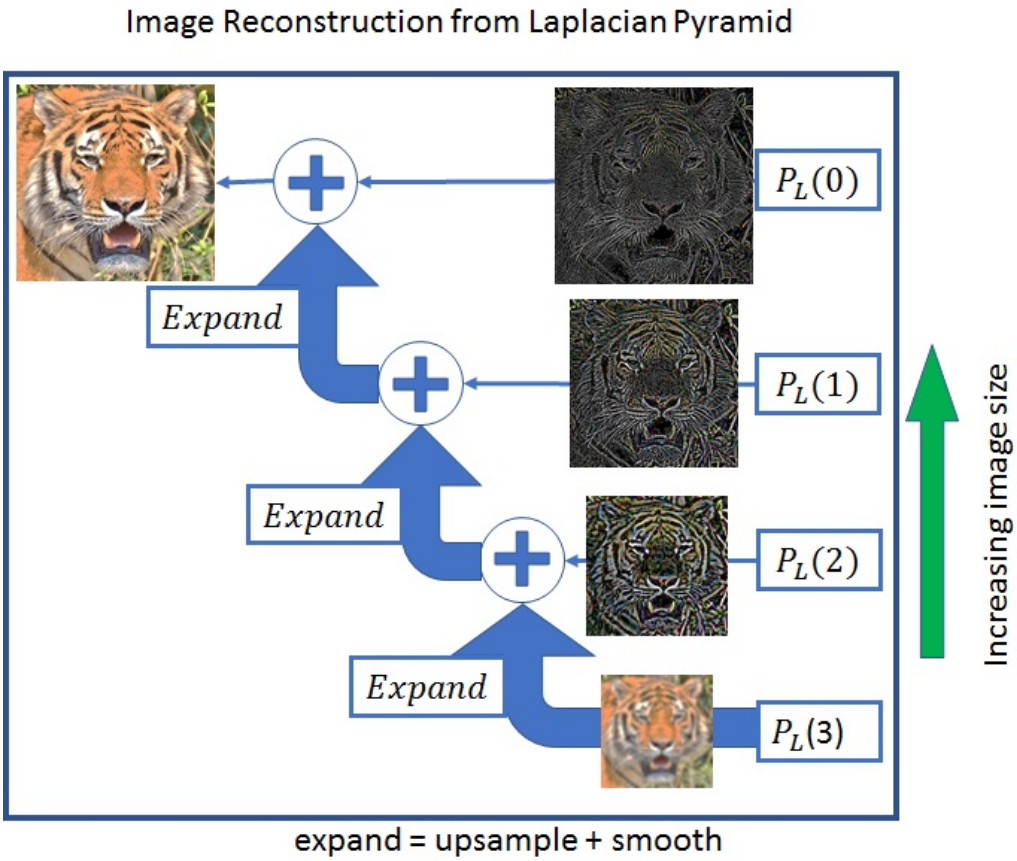

Figure 5.4: The illustration on how to reconstruct using Laplacian pyramid only. Level 3 image is the coarsest layer which is expanded and then added to level 2 image. The resulted layer is then expanded and added to level 1, and then the process is repeated until it reaches level 0 which will be the reconstructed version of the original image.

5.2.2 LAPGAN

The LAPGAN model integrates the CGAN into the framework of a Laplacian pyramid. The objective function of the CGAN is shown below [42]:

$$\mathscr{L}_{\text{CGAN}}(z,l,h,G,D) = \arg\min_{G}\max_{D}\mathbb{E}_{z\sim p_{Noise(z)},l\sim p_l(l)}[\log(1-D(G(z,l),l))] + \mathbb{E}_{l\sim p_{Data}(l,h),h}[\log D(l,h)], \tag{5.2}$$

where z is the input (random noise) to the Generator G, h is the band-pass version of input training images I for the Discriminator D and l is the low-pass version of input training images I acting as the conditioning images for both D and G at each level of the Laplacian pyramid. Note that p_{Data}, $p_{\text{Noise(z)}}$ and $\mathbb{E}$ are the Generator's distribution to be trained over samples, input noise prior and the probability of correct labels being assigned while training both the Generator and the Discriminator of the CGAN, respectively.

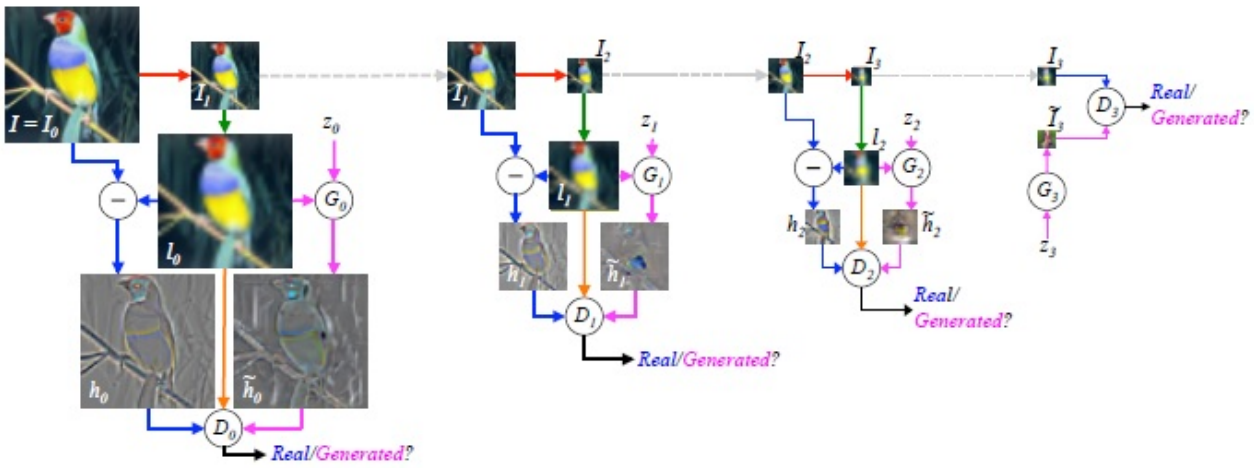

Figure 5.5: The training procedure of the LAPGAN model, reproduced from [42].

Figure 5.5, reproduced from [42], illustrates this multi-scale approach applied to the CGAN, with four levels of Laplacian pyramid used, from levels 0 to 3. As shown at the top-left of the figure, the input training image I (64×64 pixels) is used as the input image I_0 for level 0 of the pyramid. The image I_1 is the low-pass image with half of its original input size, resulted from the Gaussian-blurring and the subsequent down-sampling operations applied to I_0, as shown by the red arrow in the figure. I_1 is then interpolated and up-sampled to double its size to be the same as the original input image, as shown by the green arrow in the figure, to form l_0 which is a low-pass filtered image of I_0. Subsequently, this resultant low-pass image l_0 can serve as either a real (blue arrow's path) or a generated (fake, magenta arrow's path) input to the discriminator D_0, or fed directly into D_0, as shown by the orange arrow in the figure. For the real case of l_0, the real

input is a band-passed image h_0 formed by the subtraction between I_0 and l_0, as shown by the blue arrows in the figure. For the generated case, the generated output of the generator G_0, $\tilde{h}_0$, is a band-pass image formed by taking both the random noise input z_0 and the low-passed image l_0 as the inputs to G_0, as shown by the magenta arrows in the figure. For both real or generated cases, the real input l_0 is fed directly to the discriminator D_0 (orange arrow) to condition a better judgement by the discriminator D_0 to label the generator's output as real or fake, after 'seeing' the real inputs h_0 and l_0. This can help the generator G_0 to generate a more realistic output by an adversarial labelling of the images as real or fake by the discriminator D_0. Hence, the generator G_0 can learn to generate a more realistic band-pass image $\tilde{h}_0$ which is similar in structure with the low-pass image l_0 with the help from l_0, as well as the help from the discriminator D_0.

At levels 1 and 2 of the pyramid, the same training steps are repeated independently, using inputs I_1 and I_2 respectively. As level 3's input I_3 is a just small 8×8 pixels' image, it can be handled by a standard GANs model which is made up of G_3 and D_3.

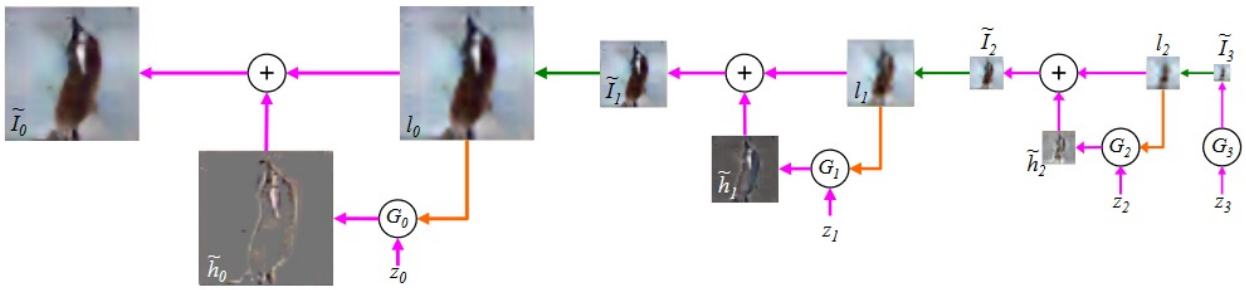

Figure 5.6: The sampling procedure of the LAPGAN model, reproduced from [42].

The image reconstruction or sampling procedure after the LAPGAN's training is shown in Figure 5.6. It can be viewed as the reverse-pyramid steps involving a series of up-sampling and summation operations for the generators' output from each level of the pyramid. Starting from level 3 or the right side of the figure, the generator G_3 generate an output image $\tilde{I}_3$ based on its noise input z_3. $\tilde{I}_3$ can be used as a conditioning input l_2 (orange arrows) for the level 2's generator G_2 after being up-sampled (green arrow), and together with its noise input z_2, it can generate $\tilde{h}_2$, as shown by the pink arrow at level 2. This output $\tilde{h}_2$ is then combined with l_2 to form $\tilde{I}_2$. The same steps are then repeated Level by level, from level 2 to level 0. The final reconstructed image, $\tilde{I}_0$, is an image similar to the training images' visual quality.

In a similar procedure as the LAPGAN, we can construct a multi-scale network architecture for our CycleGAN at each level of the Laplacian pyramids independently as described above. To test this concept, the LAPGAN was first implemented in PyTorch [84]. The implemented 3-level LAPGAN was successfully built and tested using the CIFAR-10 data set. Its image generation

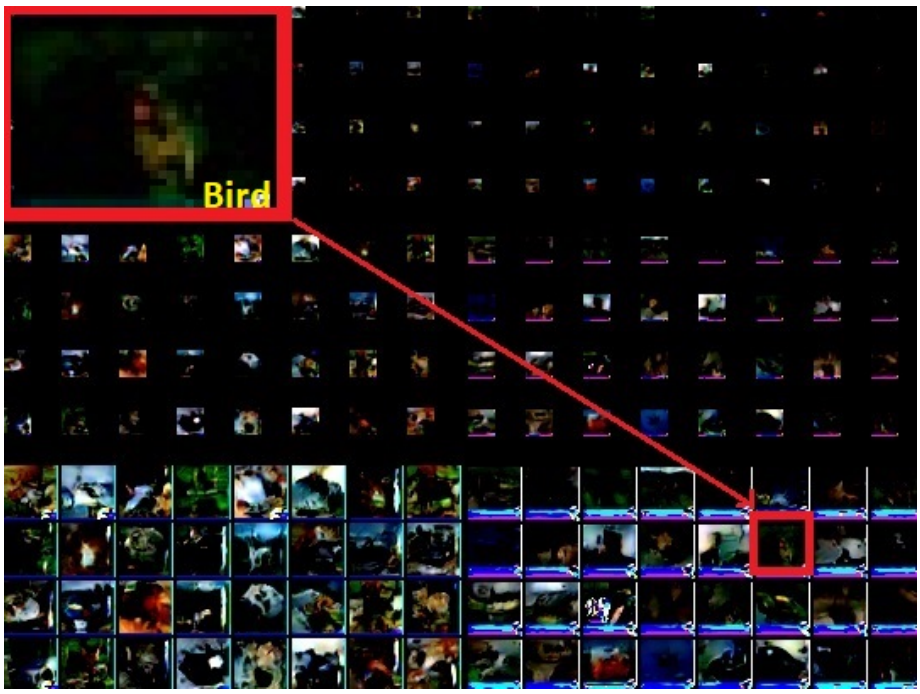

Figure 5.7: The generated CIFAR-10 images by the 3-level LAPGAN model, implemented in PyTorch. 4 rows of images are generated at each level of the LAPGAN, from coarse (top) to fine (bottom).

results for each pyramid layer is shown in Figure 5.7. From the results, it is observed that the quality of the LAPGAN images are of good visual quality, for small images' reconstruction purpose. Hence, a similar image pyramid framework for each layer's CycleGANs is used in the proposed MS-CycleGANs.

5.3 MS-CycleGANs' Network model

There are 2 stages of rain removal in the proposed MS-CycleGANs network. The first stage of the MS-CycleGANs model aims at removing different rain frequency components at different scales or pyramid levels. Low frequency rain and its effect can be removed at the coarse level, followed by its higher frequency components at the finer levels of the pyramid. The second stage is a general CycleGAN trained to remove global rain effect from the image. Figure 5.8 shows the network architecture of both stages of the MS-CycleGANs network. The MS-CycleGANs model comprises four CycleGAN models, with three of them in the pyramid at the first stage, working with image sizes of 256x256,128x128 and 64x64 pixels respectively, while the CycleGAN at the second stage is concatenated at the pyramid's output, with both of stages working with 256x256 pixels' input images. Each CycleGAN model needs to be trained concurrently to capture image features at different Laplacian pyramid levels.

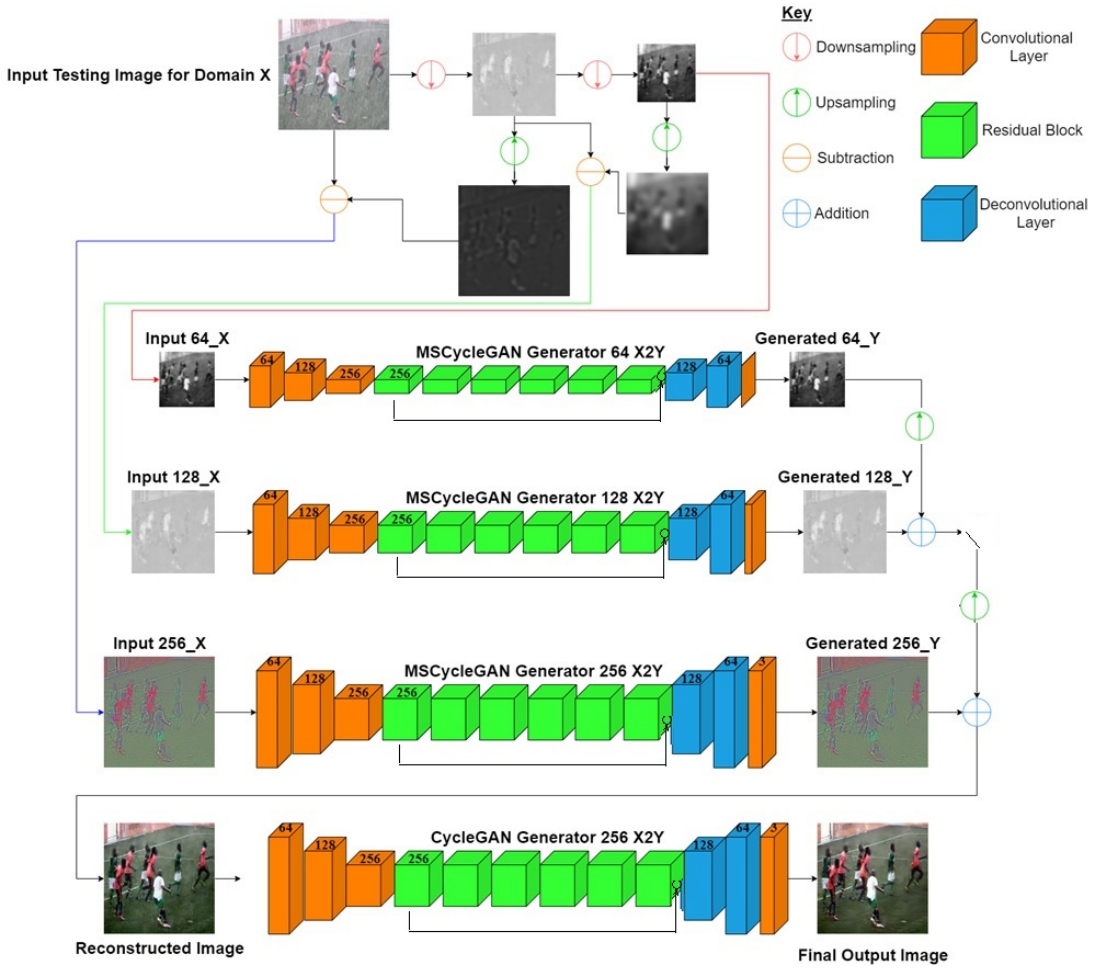

Figure 5.8: Network architecture of the MS-CycleGANs model.

The simplified CycleGAN network model discussed in Chapter 4 is adapted to suit the input image size of each of the 4 CycleGAN models used in the MS-CycleGANs framework. A Nvidia GTX 1070 is used to train the proposed integrated MS-CycleGANs network, and the algorithm is written in Pytorch [84]. This MS-CycleGANs network is made up of a 3-level (256x256, 128x128 and 64x64 pixels) CycleGANs' pyramid and a general CycleGAN at two stages. The learning rate is set to 0.0001 for 300 epochs of training, using a λ value of 10 (see Equation 4.1), for all experiments.

The MS-CycleGANs approach, similar to the CycleGAN, does not require rain image-pairs for its network training. But for a consistent comparison of the proposed MS-CycleGANs rain removal approach with the CycleGAN, the same 700 rain image-pairs provided by [5] can be used for their trainings after being resized to 256x256 by the first CycleGAN at both stages. In addition, rain images with real distortions are used to test the performance of the MS-CycleGANs, as explained in the next section.

5.4 Training and Testing

In the proposed first stage of the multi-scale pyramid framework, each of the three CycleGAN models in the pyramid is the same simplified CycleGAN model as explained earlier. They are trained to learn the rain and rain-free characteristics at different spatial frequency scales. This is essential for removing the individual frequency bands' rain components while preserving the scene details by each CycleGAN in the pyramid framework. Figure 5.8 shows the MS-CycleGANs network architecture for transferring images from rain to rain-free by each CycleGAN in the pyramid, while a similar network is trained concurrently to map reversely from the rain-free domain (Y) back to the rain domain (X), to ensure the cycle-consistency of removing and generating rain. Note that all three CycleGAN models learn to remove rain at different scales of the Laplacian pyramid of the input image concurrently. Since they are trained together at each level of the Laplacian Pyramid, the gain differences of different training images could be auto-compensated during the training process of each CycleGAN.

The first stage of the proposed MS-CycleGANs' training and testing architectures involve a combination of down-sampling and upsampling operations of input images. The purpose of this stage of the model is to remove rain gradually by each of the three CycleGANs, from large to small rain streaks; and restore background scene, from coarse to fine details. The CycleGAN at the coarse level of the pyramid is designed to remove low frequency rain, and recover the structure of the background scene. Each generator architecture of its CycleGAN consists of 6 RESNET (residual) blocks in between its encoder and decoder blocks, and is adapted to accommodate input image sizes of 256x256 pixels. The RESNET blocks of the generators help to increase the feature extraction ability to restore more background details while removing finer rain streaks. The skip connections of the RESNET blocks of the generator can help to prevent information loss while facilitating gradient propagation in such autoencoder-like generator's network structure during training. The three losses of each CycleGAN's two generators and two discriminators as shown before in Equation 4.1: forward-mapping, reverse-mapping and their cyclic losses, are being minimized concurrently during their trainings, with the help of their respective discriminators, thus ensuring consistent image gains from each level.

The second stage of the MS-CycleGANs model is made up of the CycleGAN adapted to accommodate input image sizes of 256x256 pixel. Its purpose is to remove the residual rain effect not being removed by the first stage. This may be required as rain in atmosphere can be simplified into two forms, one at the background and the other one at the foreground [9], as discussed in Chapter 2 before. The background rain is a far-away rain effects of the scene that appears hazily in the background caused by the tiny rain drops' cumulative effect, while the foreground rain appears as local disruptions' effect caused by near-by large rain streaks. While

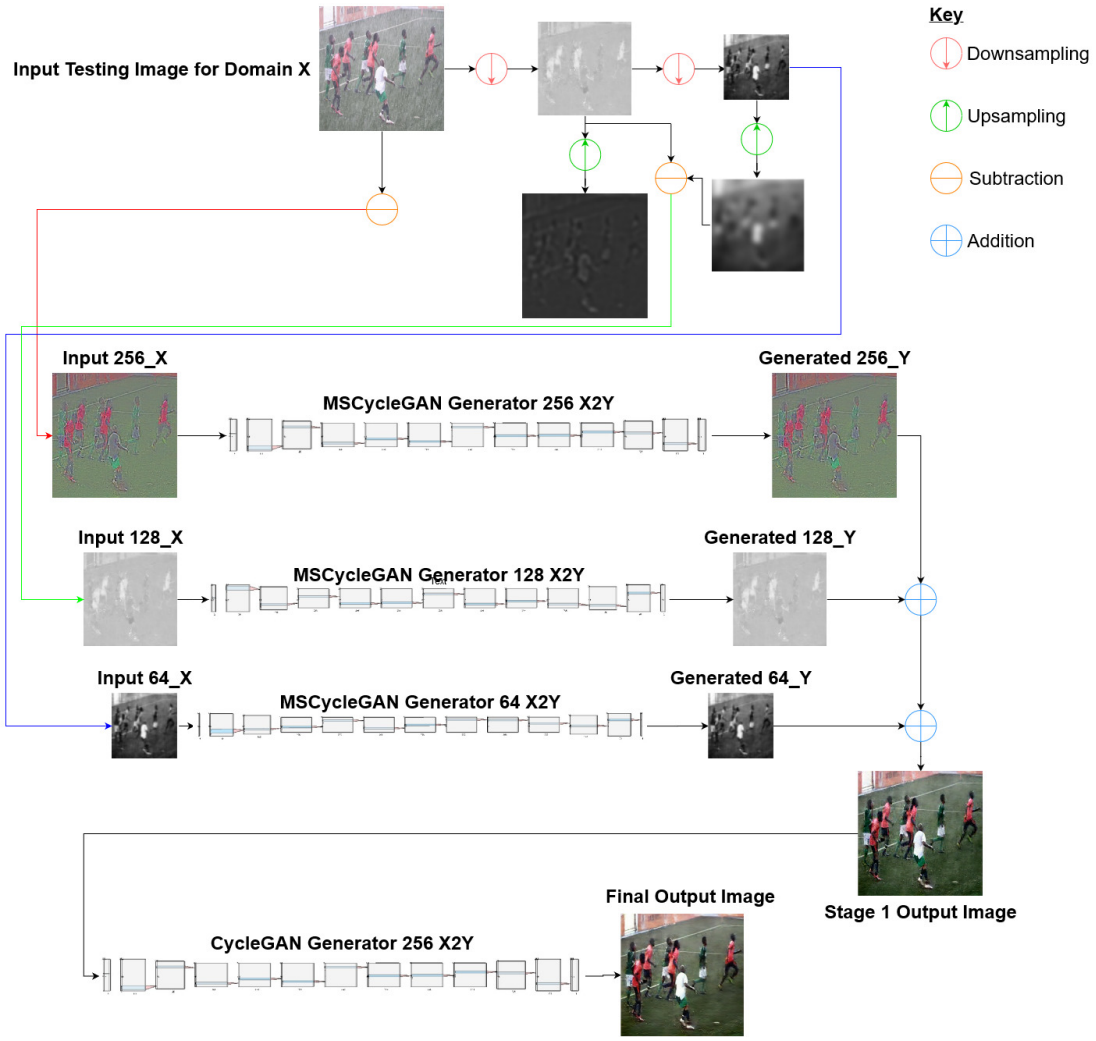

Figure 5.9: Testing of the trained MS-CycleGANs model from the trained CycleGANs.

the nearer one may be removed by the first stage of the MS-CycleGANs by the different scales of the pyramid, the far-away rain effect at the global image level of the background may need a conventional CycleGAN to remove it directly in a global manner, since the rain removal CycleGAN paper [2] has shown great results in removing such rain. Hence, we have concatenated both stages to form a two-stage MS-CycleGANs design, as shown in Figure 5.8.

After training of the MS-CycleGANs model, the testing operation involves making use of all the four trained CycleGAN models to yield the final reconstruction results, to get back a final rain-free image. Figure 5.9 depicts the testing operation. It shows the reconstruction of input test images from the trained CycleGANs. While each of them works with 256x256, 128x128 and 64x64 pixels respectively in the Laplacian pyramid at the first stage, the second stage's CycleGAN works with 256x256 pixels' input images only. The test process also runs the down-sampling for the input rain image to obtain the half-resolution or half-sized down-sampled images at each image scales, for the CycleGAN to obtain its corresponding outputs, which is

similar to the training operation. But the testing process also involves up-sampling its output at the current level and adding it back to the previous finer level's outputs, level-by-level at each image scales, to reconstruct the final rain-free image.

5.5 MS-CycleGANs' rain removal results

Figures 5.10 to Figure 5.12 show the reconstructed rain removal results from the CycleGAN and MS-CycleGANs. The images in the second row are the zoomed in regions of the first row for a clearer comparison. As shown by the figures, the MS-CycleGANs have shown its capability in removing a significant amount of rain streaks, in particular at high-frequency sub-bands. The vertical and diagonal rain streaks' characteristics were learnt by the MS-CycleGANs at each band-passed level, and the models managed to preserve the other vertical and diagonal structural details of the scenes after rain removal. This can be observed in the figures when comparing the original real rain images and their rain-removed counterparts. The MS-CycleGANs has shown its abilities in retaining the distinct and sharp structural details of the scenes after rain removal as well.

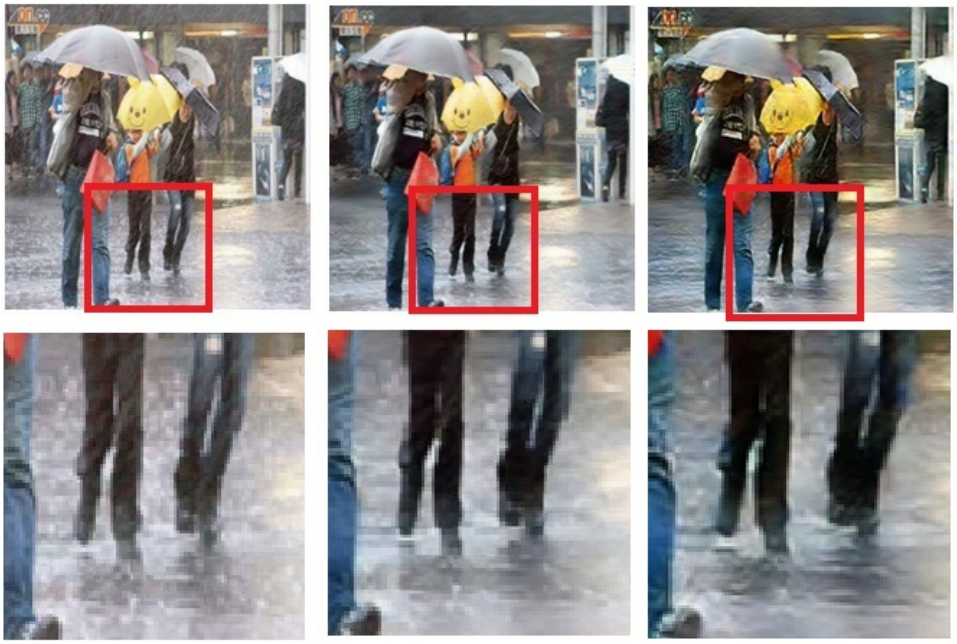

Figure 5.10: First input real rain image, CycleGAN rain removal results and the MS-CycleGANs' rain removal results (from left to right), together with their corresponding zoomed in regions shown below them.

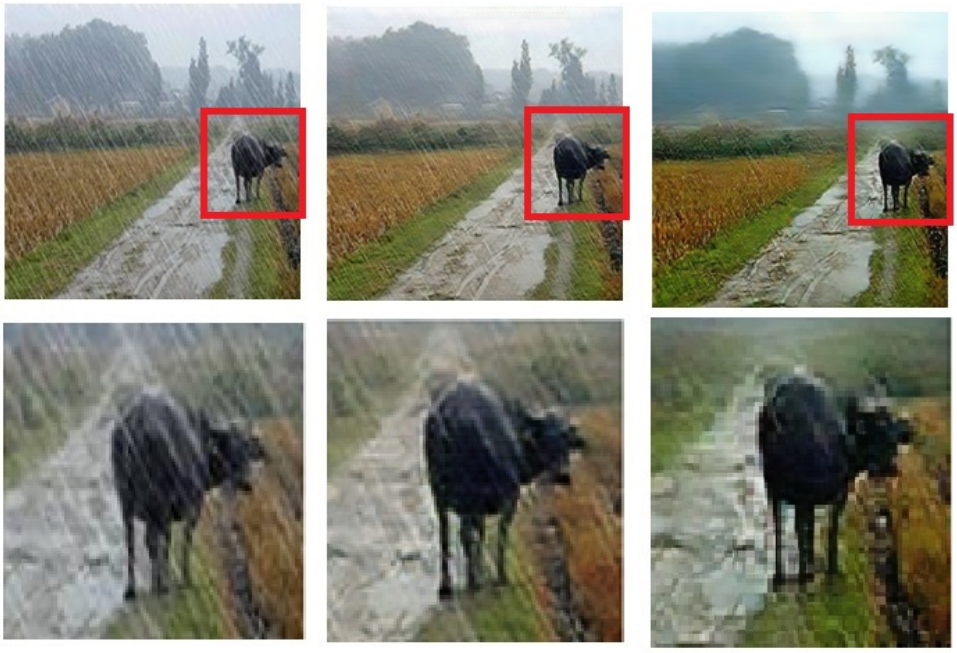

Figure 5.11: Second input real rain image, CycleGAN rain removal results and the MS-CycleGANs' rain removal results (from left to right), together with their corresponding zoomed in regions shown below them.

Figure 5.12: Third input real rain image, and CycleGAN's and MS-CycleGANs' rain removal results (from left to right), together with their corresponding zoomed in regions below them.

As shown by Figure 5.10's comparison result, the MS-CycleGANs has successfully removed the rain streaks with very little rain residue when comparing it to the original rain image. This is evident with zoomed-in image below it, which has been almost completely clear of rain. Fur-

thermore, the details of the objects at the foreground and the background are found to be retained as well. We believe that the first stage multi-scale CycleGANs at different frequency bands have removed the dynamic rain at the foreground, while the second stage's CycleGAN have removed the low-frequency static rain, have contributed to such performance. Hence, we can say that the MS-CycleGANs can preserve the scene details with almost perfect rain removal, as compared to the CycleGAN.

Figure 5.11 also show the comparison of the performance of the MS-CycleGANs with the CycleGAN. For the MS-CycleGANs trained using the same synthetic data sets, it has shown very good rain removal performance. Most of the heavy diagonal rain streaks in the image have been removed completely, besides the low-frequency rain components. The CycleGAN's reconstructed image has suffered from discolouration compared to the original rain image, while the objects' details in the image have not been well-preserved also with some obvious defect. The same analysis can also be applied to Figure 5.12.

To summarize this section, the qualitative rain removal results of the multi-scale representation using the MS-CycleGANs has been demonstrated to be superior to the CycleGAN. Both its rain removal capability, as well as the preservation of object and background details, are better than the CycleGAN. The quantitative NIQE comparison results will be discussed together with the W-CycleGANs' results analysis in the next chapter.

Chapter 6

Rain removal at spatial frequency's sub-bands using W-CycleGANs and real rain's performance analysis

In Chapter 5, we have discussed how the CycleGAN performance can be improved by the MS-CycleGANs, using the framework of a pyramid-like, multi-scale implementation of multiple CycleGANs at each of the frequency bands. Another way to process images at multi-scales for rain removal can be implemented in the wavelet domain [96]. Although Fourier transform is a popular method to transform images to the frequency domain, there is no signal localization and orthogonality in its frequency domain. Over the past decades, wavelet transform has been very popular in application areas such as super-resolution [97], denoising [98] and color enhancement [44]. Our proposed Wavelet-CycleGANs (W-CycleGANs) makes use of such desired properties of the wavelets for image reconstruction. Coincidently, another wavelet-CycleGAN technique was also proposed for denoising the satellite images using wavelet's sub-band cycle-consistent adversarial network called WavCycleGAN [99]. But our proposed rain removal method in this chapter is different, as the WavCycleGAN technique is limited to removing noise from gray-level noisy satellite images, which are computationally lighter and less complicated.

The details of the W-CycleGANs' network architectures, together with their results, will be discussed in this chapter. For a fair comparison with the CycleGAN, this proposed multi-scale representations of CycleGAN networks were trained and tested on the same set of rain images used by the ID-CGAN work [5]. The comparison results of the performance between the CycleGAN, the MS-CycleGANs and the W-CycleGANs has demonstrated the superiority of the both the MS-CycleGANs and W-CycleGANs in removing rain distortions, both qualitatively and quantitatively.

6.1 Background

Fourier transform is a great tool in signal processing but it does not represent abrupt changes caused by rain efficiently as its basis function used is a sine wave, which is a continuous function that continue indefinitely. Hence, its representation is not localized in terms of the pixel location of the rain disruption in an image. Therefore, to reconstruct an image corrupted with rain that have abrupt changes, we need to use some basis functions that are well localized in pixel locations, and orthogonal to each other, like the wavelets.

In this section, some basic wavelet concepts will be covered to understand why it is a better suited tool for analyzing an image corrupted with rain. Real world signals in an image always have mostly smooth regions, but with some abrupt changes in signal contrast at the edges of objects or fine structures of a scene. In terms of rain removal, the abrupt changes in spatial locations of an image can also be caused by rain that fall into any random frequency range of an image, as discussed in the previous chapters. Hence, there is a need to extract and remove such localized rain corruption based on its spatial-frequency information extracted at different sub-bands.

The Haar functions used in the wavelet transform for the proposed W-CycleGANs are an example of the orthogonal family of wavelet functions. It is an orthogonal basis of piecewise constant functions, constructed by dilation (scaling) and translation [100]. As shown in Figure 6.1, its scale factor is inversely proportional to its frequency, i.e. a signal $W(x)$ scaled by a factor of 2 results in a reduction of its original interval by half, or by an octave. In other words, it has a band-pass characteristic in the frequency domain. As for its translation, as shown in the same figure, a delay of 1 is introduced along the spatial location axis of the image. The wavelet needs to be translated in order to align it with different features in an image. By having the basis functions scaled and translated, they can capture both slowly varying changes as well as abrupt changes, for the reconstruction of an image corrupted with rain.

As discussed and shown in Figure 6.1, the wavelet decomposition operations from $W(x)$ to $W(2x)$ is dilation, and from $W(2x)$ to $W(2x-1)$ is the translation. The next level of operation is made up of $W(4x)$, $W(4x-1)$, $W(4x-2)$ and $W(4x-3)$, each with an interval of length $\frac{1}{4}$ [100]. The Haar's functions can be represented by:

$$W_{jk}(x) = W(2^j x - k), \tag{6.1}$$

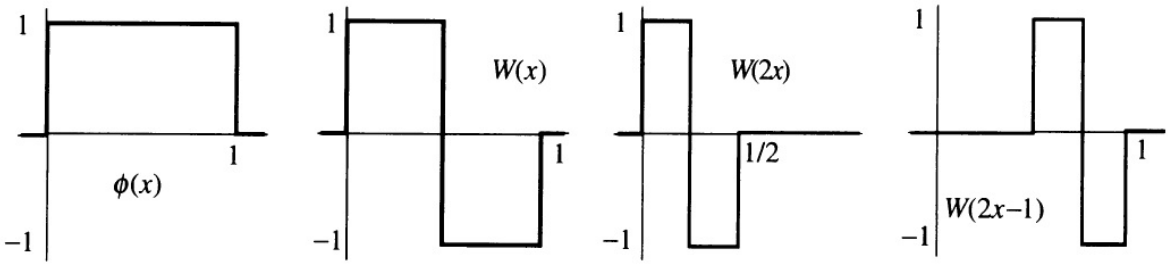

Figure 6.1: Scaling function $\phi(x)$, wavelet $W(x)$, and the next level of detail of the Haar wavelet (reproduced from [100]).

together with $\phi(x)$. This forms the Euclidean norm $L^2[0,1]$ when the index range for j is ≥ 0 and for k is $0 \leq k < 2^j$ [100].

The four functions as shown in Figure 6.1 are piecewise constant functions that are orthogonal to each other, and the combination of them can represent every function that is constant on each quarter-interval [100]. This mutually-orthogonal property of a wavelet, together with its signal localization property, have made the wavelets ideal for our proposed W-CycleGANs, as we need to reconstruct an image corrupted with rain which consists of abrupt changes.

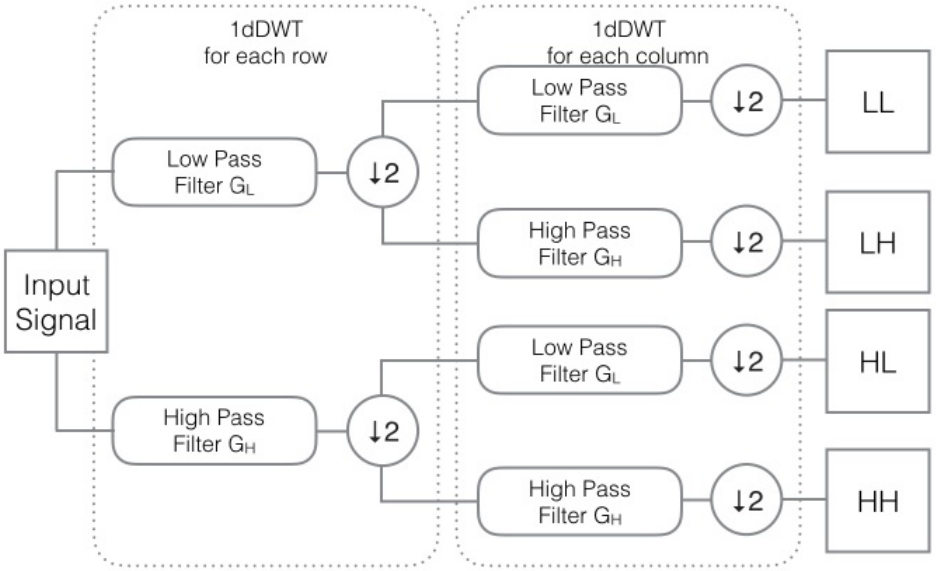

Figure 6.2: The procedure of 1-level 2D DWT decomposition results in four sub-bands' coefficient arrays LL, LH, HL and HH (reproduced from [97]).

The Discrete Wavelet Transform (DWT) is normally used in image processing for image denoising and compression applications as it can represent many natural images with fewer coefficients, or sparser representation. DWT based on the Haar wavelets can transform an input image into four wavelets coefficient arrays LL, LH, HL and HH, as shown in Figure 6.2. The

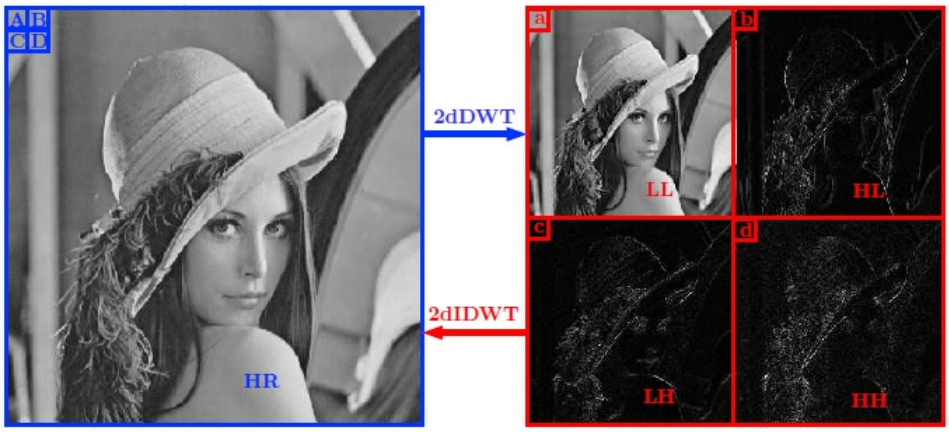

Figure 6.3: The 2D DWT and 2D IDWT operations and their resultant coefficient arrays from an image HR. LL, LH, HL and HH are the four sub-bands' coefficient arrays correspondingly after 2D DWT decomposition (reproduced from [97]).

length of the coefficients in each sub-band is half of the number of coefficients in the preceding stage. In DWT, a scale-space representation of the image is achieved using digital filtering techniques, where an image is passed through a bank of special low pass and high pass filters to yield low pass and high pass sub-bands, as shown in Figure 6.2. These filters also have the ability to reconstruct the sub-bands, while cancelling any aliasing that occurs due to down-sampling. To get back the reconstructed image from these four wavelets coefficient arrays, in a similar notation, its inverse operation, called the IDWT, has to be performed, as shown in Figure 6.3.

Recently, in the area of deep neural networks, wavelet transform has been used and achieved encouraging results. For example, a multi-scale wavelet architecture has been proposed as a deep neural network's approach in an image reconstruction task [101]. Later, a multi-scale wavelet transform method was used to remove rain disruption from an image recursively [102]. Another work [104] then combines original rain images with rain removed features for more details, and adopts a multi-scale loss function to perform coarse-to-fine restoration process.

Our proposed multi-scale wavelet-based method for rain removal is motivated by [44], which performs wavelet transform at the Hue, Saturation, and Luminance (HSV) color space, instead of the usual RGB color space. The luminance component (V), also called the brightness values, can then be decomposed by a one-level Discrete Wavelet Transform (DWT), using Haar as the wavelet function, into four frequency sub-bands which are orthogonal to each other. The rain components in the high frequency sub-bands can then be learnt by the CycleGAN, so as to remove rain distortions at various spatial directions. Finally, the rain-free V can be obtained through the inverse wavelet transform, and converted back to RGB enhanced image by re-combining it with the H and S channels.

Hence, the main motivation behind this wavelet idea is two-fold:

- Wavelet coefficients inherently encourages signal localization [101], just as the multi-scale image pyramid approach discussed earlier.
- We can decompose the image into orthogonal sub-bands but at the same time use the information from other sub-bands to predict extra coefficients for the rain-free images.

This chapter provides an overview of another multi-scale CycleGANs' implementation, the W-CycleGANs, to improve the CycleGAN's performance. It's network architecture, followed by its training and testing methods, will be discussed in this chapter. The process of building the network involves the iterations of the following few phases:

1. Training of W-CycleGANs model
2. Testing of the model through the reconstruction of the respective wavelet coefficient arrays at different frequency scales and orientation
3. Image quality check
4. Re-training and testing of the W-CycleGANs

This chapter concludes with a qualitative and quantitative evaluation and analysis of the performance of the W-CycleGANs, MS-CycleGANs, CycleGAN and the state-of-the-art, ID-CGAN.

6.2 W-CycleGANs Network model

The motivation of using the Haar wavelet [96] in our proposed rain removal framework has been discussed in Section 6.1. DWT can decompose images into four orthogonal coefficient arrays, with smaller frequency bandwidths and spatial widths. One of them is a low frequency band (LL), which is a coarse approximation of the source image, and the other three are higher frequency bands (LH, HL, HH), containing image local details at the horizontal, vertical and diagonal orientations.

As discussed in Chapter 2, a rain image consists of two types of appearances: background (far) and foreground (near) rain. Because the foreground rain appears mainly at the high frequency sub-bands, the first stage of the combination of "Wavelet-CycleGAN" approach can help

to remove such rain component by using the popular Haar wavelet to decompose the source image into the four sub-bands first, followed by removing them in an end-to-end mapping using only the CycleGAN in the wavelet domain. For the other hazy rain component as a result of the cumulative effect of far-away rain streaks, a CycleGAN at the second stage is a better approach [2] to remove them. By combining these two stages in such a manner, a multi-scale CycleGAN's model is formed and all parameters of our network model are trained together in a single framework. Hence, in this chapter, a novel multi-scale wavelet-based CycleGAN network, named W-CycleGANs, has been proposed for rain removal from images.

The first stage of the W-CycleGANs combines the benefit of wavelet transform with the CycleGAN to remove rain in multiple high-frequency sub-bands. Only the rain components and the ground truth in the rain images' luminance (V-channel) in the HSV color space [103] is decomposed into multi-scale sub-bands, via a wavelet transform, since the intensity values of images are solely determined by the V-channel values. The high frequency sub-bands of the transformed image are then forwarded to its corresponding CycleGAN to remove rain components at different frequency and orientation. In other words, the wavelet transform has effectively performed a down-sampling operation on the source image for further sub-bands' processing by the CycleGAN.

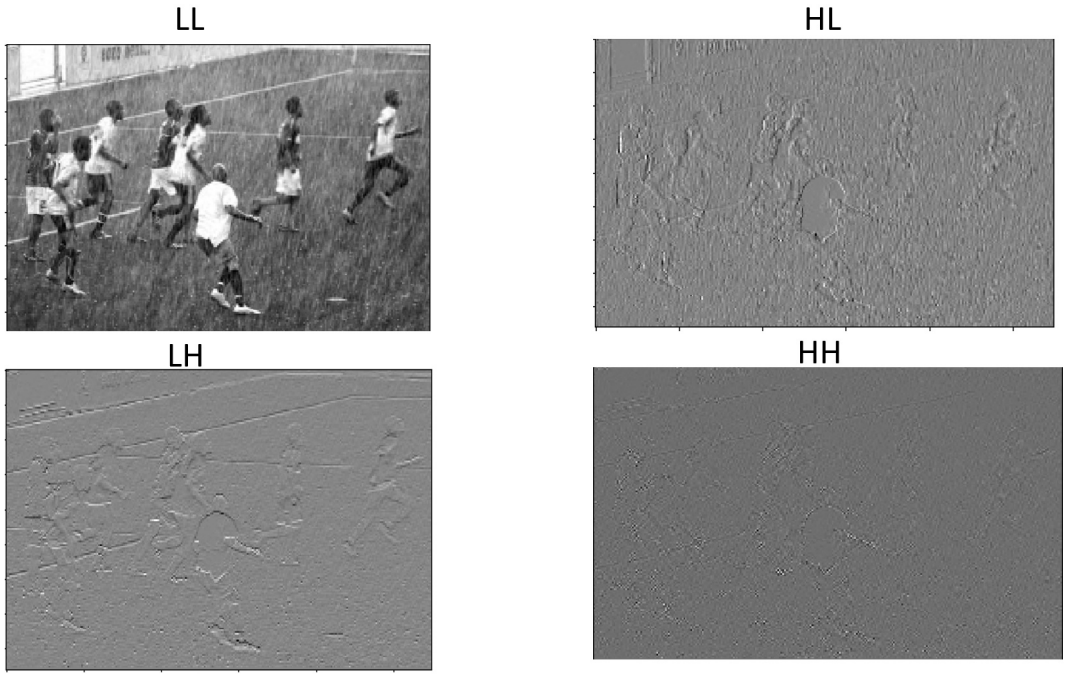

Figure 6.4: W-CycleGANs model involves the discrete wavelet transform to decompose an image into 4 orthogonal frequency sub-bands or coefficient arrays in its first stage.

Instead of decomposing rain images into multi-level hierarchical sub-bands [104] to exploit

hierarchical features to improve rain removal performance, our W-CycleGANs is built on just a single-level's 2D discrete wavelet transform (DWT) to take advantage of spatial-frequency localization. As shown in Fig. 6.4, the 2-D wavelet decomposition results in four frequency sub-bands. Since the LL represents the approximation sub-band, it contains more background or scene information. LH, HL and HH sub-bands contain more rain details and edge information, at different orientations. Hence, this decomposition can help the CycleGAN to learn both the rain and rain-free characteristics of the scene. The second stage of the W-CycleGANs is just composed of a CycleGAN, concatenated to the output of the first stage, to recover more details and further enhance restoration performance, similar to the second stage of the MS-CycleGANs discussed in Chapter 5.

Figure 6.5 shows the two generators and their associated adversarial discriminators of the CycleGAN, at the first stage of our proposed W-CycleGANs network in the rain domain (X). Using the adversarial and cycle-consistency loss functions described in the previous chapter, discriminator Y can 'help' the training of the generator X to Y, such that the generated output by the generator at domain Y (rain-free) can 'fool' the discriminator. Similar trainings of the generators Y to X and discriminator X are also happening at the Y (rain-free) domain, although it is not shown here.

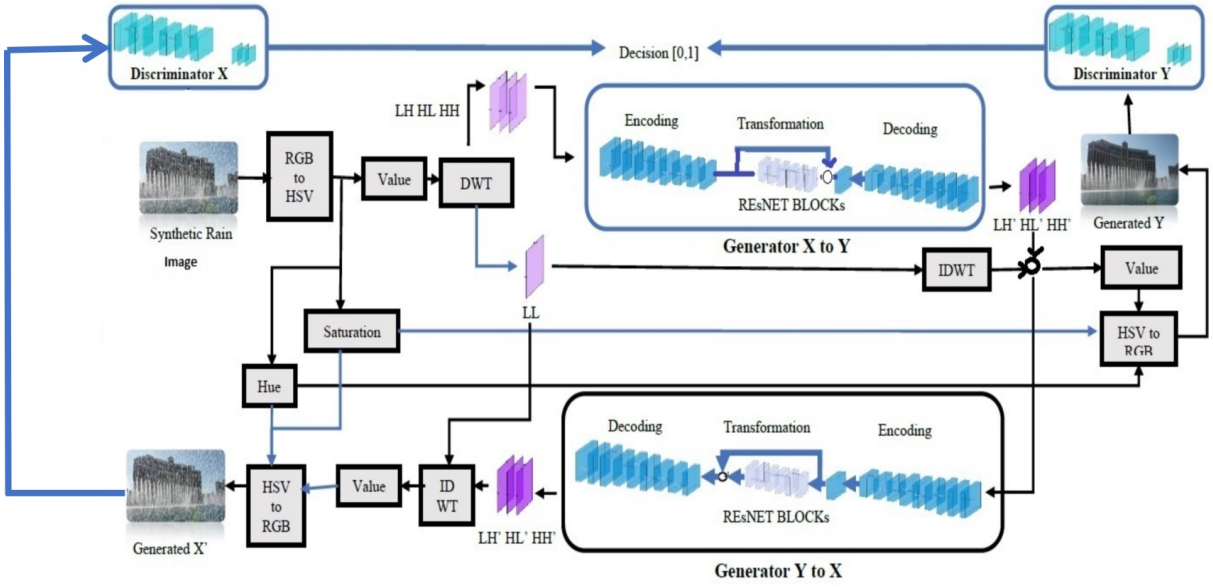

Figure 6.5: The first stage of the W-CycleGANs model.

6.3 Training and Testing

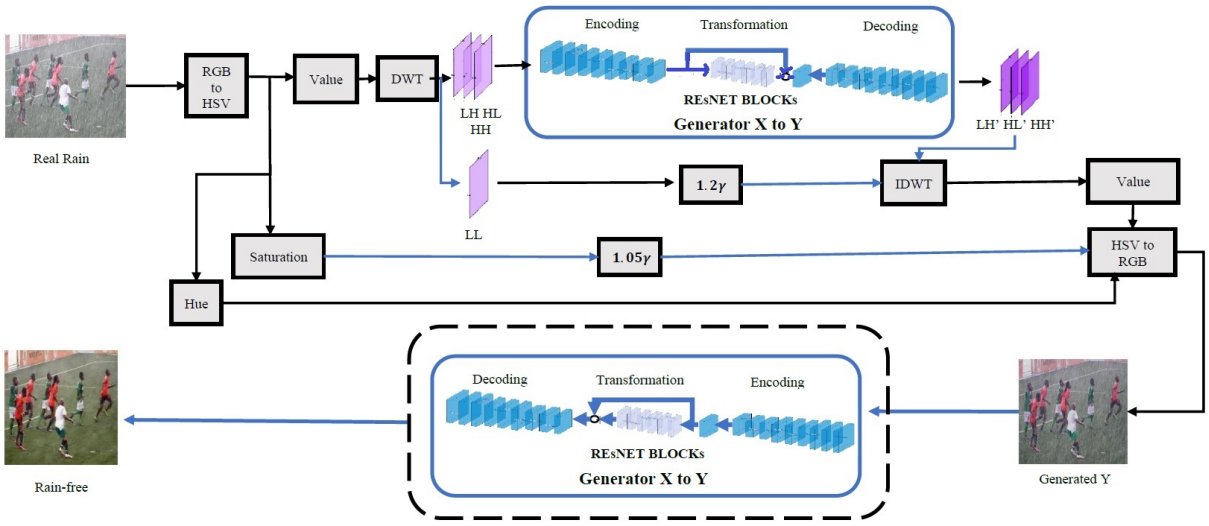

Figure 6.6: The W-CycleGANs testing model comprises two CycleGAN models at two stages, where the first stage can remove rain in the wavelet domain, and the second stage can remove rain in the spatial domain.

A Nvidia GTX 1070 is used to train the proposed W-CycleGANs network, and the algorithm is written in Pytorch [84]. A learning rate of 0.0001 is set for 200 epochs' training. For all experiments, λ is set at the value of 10 (see Equation (4.1)).

The proposed W-CycleGANs requires a series of down-sampling operation by discrete wavelet transform (DWT) and upsampling operation by inverse discrete wavelet transform (IDWT) at its first stage's rain removal process which involves only the V-channel in the HSV color space, as shown in Figure 6.6. The data set used for training is the same as the ID-CGAN's work [5]; they are 700 rain-free and 700 rain image-pairs (a total of 1400 training images). Every image will undergo a DWT operation to get into 4 sub-bands at array sizes of 128x128 pixels each and an IDWT operation which is the reverse process to restore the images after the first CycleGAN's rain removal.

The high-frequency sub-bands' coefficient arrays are then forwarded to its first stage CycleGAN to remove their foreground rain components, together with the contrast-enhanced low-frequency sub-band (LL) using the gamma correction technique [105], [106] to restore its contrast, for subsequent images' reconstruction. This is followed by a second stage CycleGAN's rain removal for the residual (background) rain removal. This final stage of the W-CycleGANs is just composed of a CycleGAN, concatenated to the output of the first stage, to recover more details and further enhance restoration performance.

All the 700 image-pairs of rain-free and rain data set will undergo training concurrently at both stages' CycleGANs, as shown in the same figure. After training, the test operation involves

making use of each of the trained models to yield the final reconstruction results, to get back a final rain-free image. As mentioned before, the same set of 700 rain image-pairs provided by [5] are used to train all the CycleGAN, MS-CycleGANs and W-CycleGANs models. After training, real rain images are used to test all networks for their rain removal performance. Their results are analyzed in this chapter both qualitatively and quantitatively. The qualitative analysis involves visual comparison of their rain-removed images, especially at the enlarged rain-removed regions. The quantitative check involves the Natural Image Quality Evaluator (NIQE) values' comparison as the analysis method because it involves real rain, which has no ground-truth images, as discussed in Chapter 4.

6.4 Gamma Correction

The term *gamma* is historically introduced to describe the non-linearity of the output voltage of a display device to the change in image brightness [107]. But this term is later generalized to the term *gamma correction* used in image processing for the camera to correct for light sensitivity difference between human eyes and a camera's sensors. An image pixel's intensity data captured by a camera's sensors is linearly related to its luminance levels, while human vision responds non-linearly to the same brightness changes [108]. Hence, the inverse of this gamma correction process performed by the camera may be necessary for our image reconstruction tasks because the first stage of the W-CycleGANs process images at the V (intensity) channel of the HSV color space. This non-linearity of human eyes to brightness level of an image is caused by human eyes' higher sensitivity to a small change in brightness in the darker regions of an image compared to the brighter areas [108]. The gamma correction performed on the camera's output image due to such non-linearity of human vision's sensitivity to brightness changes can be approximated by a power law function [108]. It can be introduced to the camera's output by simply applying a power law transform to the camera's output I, as shown below [108]:

$$I_{gc} = I^{\gamma} \tag{6.2}$$

where I is the camera's image normalized to the range [0,1], γ is the gamma correction value and I_{gc} is the gamma-corrected camera image.

Figure 6.7 shows this non-linear characteristic plot for γ value between 0.45 and 2.2, which are typically used in the camera. The corrected images of an example image at different gamma levels are shown in Figure 6.8. These figures show that an imaging device such as a camera always perform a non-linear gamma correction or mapping of its output intensity value to provide a finer perceptual detail at darker region. Hence, to reconstruct images from our trained

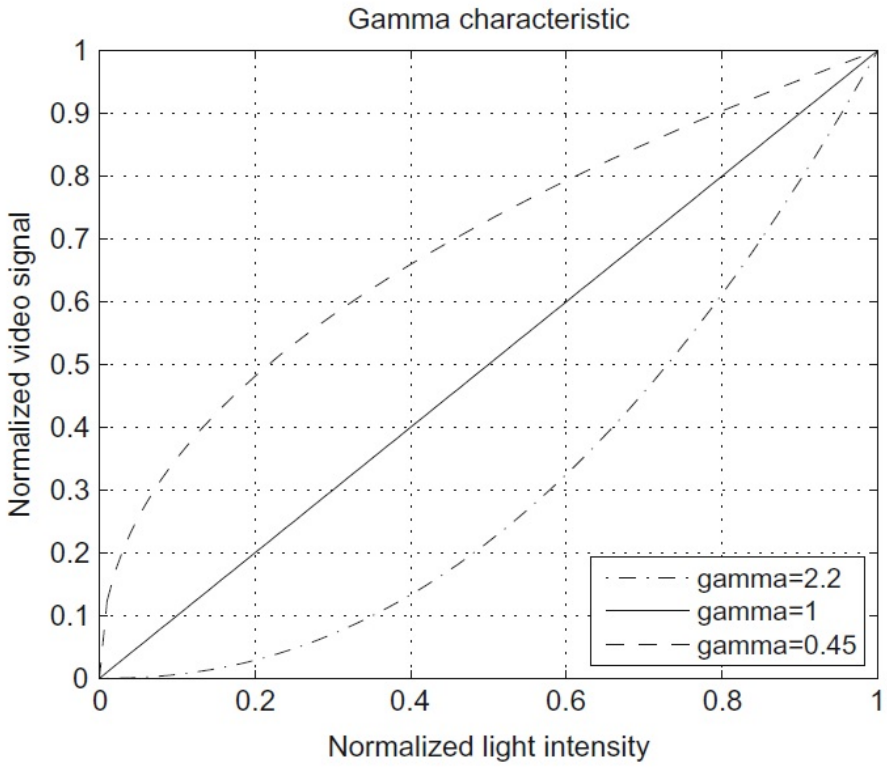

Figure 6.7: The non-linear characteristic plots for $\gamma = 0.45$, 1 and 2.2 (reproduced from [108]).

Figure 6.8: Corrected images of an example image at different value of γ (reproduced from [108]).

W-CycleGANs network with rain test images (see Figure 6.6), we need to remove these non-linearities introduced by the camera's gamma correction, prior to our second processing stage in the W-CycleGANs, immediately after the first stage's CycleGAN has removed rain components from the higher frequency sub-bands. If we know the original gamma value of the test images, we can simply inverse this gamma correction process from the low frequency LL sub-band of the V (intensity) channel of the input image, to remove these non-linearities introduced by the camera's gamma correction, as shown below [109]:

$$I = I_{gc}^{\frac{1}{\gamma}} \tag{6.3}$$

(a) (b) (c)

Figure 6.9: Brightness effect of different value of γ: (a). 0.9; (b) 1.0; and (c) 1.1.

As shown in Figure 6.9 (a), when the γ value applied to an image to inverse the gamma correction process is less than 1, the image becomes darker, and less details are visible. When γ is more than 1, as shown in Figure 6.9 (c), the image becomes brighter and more details are revealed. When γ equals to 1, as shown in Figure 6.9 (b), there is no effect on the image.

However, knowledge of the gamma correction used by the imaging device is not possible to be obtained, as most of our rain test images are provided from the ID-CGAN paper [5]. In addition, once the rain corruptions which made the whole rain image looked brighter, as discussed in Chapter 2, are removed from the W-CycleGANs, the reconstructed image will appear to be darker than original image. Hence, we will need to apply a γ of more than 1 to restore the original brightness of the image. But the main drawback of gamma correction is that it does not have a specific gamma value for every image, as each of them may have different luminance requirement. In other words, to invert the images' gamma correction performed by the camera, we need to find a way to determine the unique gamma value that can be applied to the W-CycleGANs' first stage of processing during network testing.

Although there exist some gamma correction techniques which could obtain the γ value for each different image's gamma correction, such as the Adaptive Gamma Correction (AGC) [110] which rely on different images' probability density function (PDF) and cumulative distribution

function (CDF) in histogram analysis to decide the gamma value used to enhance the luminance in each dimmed image, our experimental results show that it will over-enhance the rain image of the W-CycleGANs and make it looked saturated, as shown in Figure 6.10. This may be due to the image's dark region contrast being enhanced by the AGC [110].

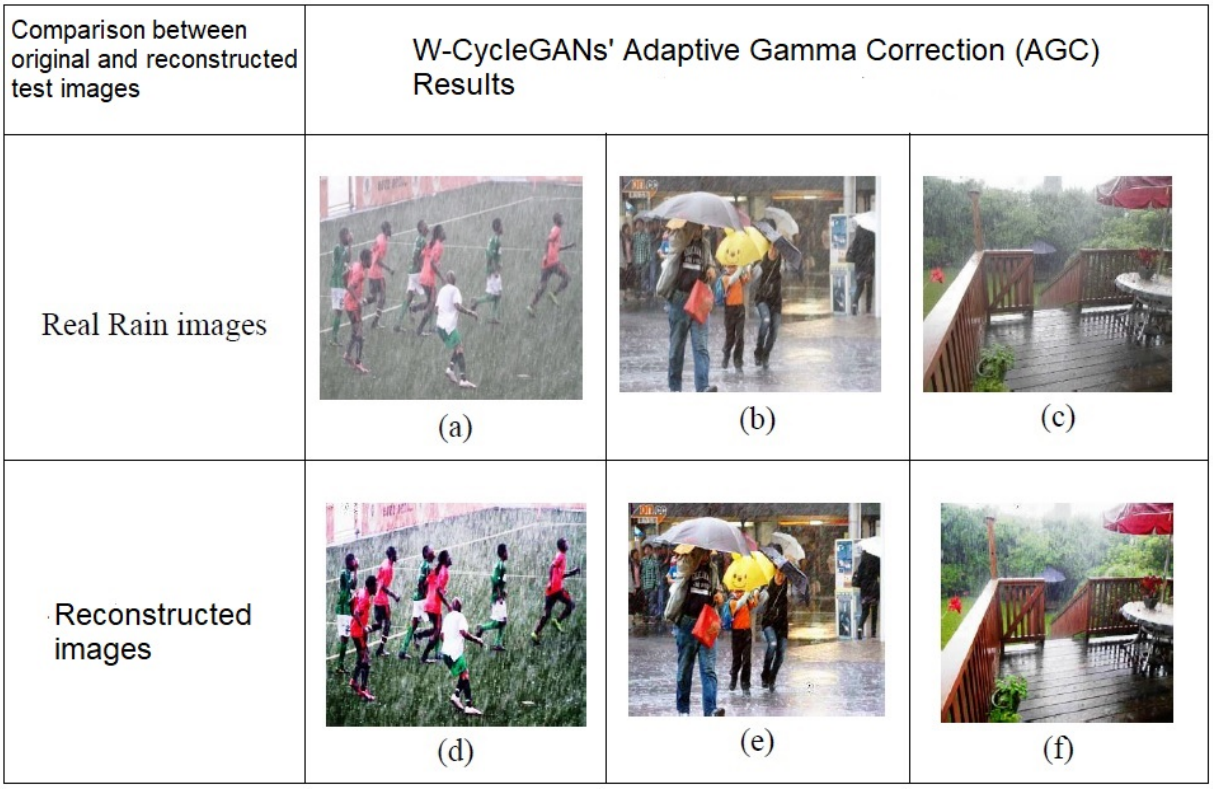

Figure 6.10: W-CycleGANs' Adaptive Gamma Correction (AGC) image reconstruction results after first stage processing.

In conclusion, we plan to enhance the reconstructed image's contrast by directly defining a specific gamma correction value in the proposed W-CycleGANs, after running multiple experiments with different γ values to determine the optimum γ value. The experiments and their results will be shown and discussed in the next section.

6.5 W-CycleGANs' rain removal results

In this section, the reconstructed rain removal results from the CycleGAN and W-CycleGANs are compared qualitatively first, followed by a quantitative analysis of all algorithms, including the ID-CGAN and the MS-CycleGANs, using NIQE scores and real rain test images. Before the results' comparison, it was mentioned in Section 6.4 that specific gamma correction value

in the proposed W-CycleGANs has to be defined for optimizing the reconstructed image's contrast. This can be achieved by running multiple experiments with different γ values to determine the optimum γ value. Hence, the experiments' results involving different γ values will first be shown and discussed in Section 6.5.1.

6.5.1 Determining γ Value for W-CycleGANs' Gamma Correction

Before we show the W-CycleGANs' rain removal results, we need to decide the optimum gamma correction value, as discussed in Section 6.4. Figure 6.11 shows the gamma correction experimental results for a range of γ after its first stage of processing, to enhance the brightness of reconstructed images. From the results, $\gamma = 1.2$ is selected as the reconstructed image shows optimum image brightness with visible image details, although its overall brightness and color can still be improved. Hence, another experiment was carried out on gamma correction on the Saturation (S) channel after second stage's processing, as shown in Figure 6.12. This is because the S-channel of the reconstructed images can be enhanced with gamma correction to optimize their overall brightness and color. The optimum γ value of 1.05 is selected as the reconstructed results have the optimum brightness and color.

6.5.2 W-CycleGANs' Results

After the selection of the gamma correction value, both the qualitative and the quantitative rain removal performance of the W-CycleGANs is compared to other algorithms in this section. The quantitative NIQE results will also be compared to the MS-CycleGANs and IDCGAN, using the five types of real rain test images [2] for such purpose.

Figures 6.13 and 6.14 show the CycleGAN and the W-CycleGANs' rain-removal results, with zoomed in regions shown directly below them, for easier comparison. The visual qualitative comparison demonstrates the W-CycleGANs' capability in removing a significant amount of rain distortions, with the scene details well-preserved. We believe that the vertical and diagonal rain streaks' characteristics, as well as the other vertical and diagonal structural details of the scenes at each sub-band, are learnt by the W-CycleGANs. This may be the contribution from the first stage's wavelet-CycleGAN to the rain streaks' removal in the high frequency sub-bands, followed by the second stage's CycleGAN to the remaining sub-bands' rain removal.

We can validate further the performance of both the MS-CycleGANs and the W-CycleGANs quantitatively, using five types of real rain phenomena, as proposed in [2]. Based on these five

Different gamma value, γ	Output of respective Gamma Correction in the testing images		
Real Rain images	 (a)	 (b)	 (c)
1.1	 (d)	 (e)	 (f)
1.2	 (g)	 (h)	 (i)
1.3	 (j)	 (k)	 (l)
1.4	 (m)	 (n)	 (o)

Figure 6.11: W-CycleGANs' image reconstruction results using different γ for V-channel at its first stage.

types which are listed as follows, the effectiveness of our proposed approaches are checked:

i. heavy or light rain streaks;

ii. different appearances of rain streaks due to camera settings;

Different Gamma Value for S	Output of respective in the testing images		
No correction	 (a)	 (b)	 (c)
1.03	 (d)	 (e)	 (f)
1.04	 (g)	 (h)	 (i)
1.05	 (j)	 (k)	 (l)
1.1	 (m)	 (n)	 (o)

Figure 6.12: W-CycleGANs' final image reconstruction results after its two stages of testing using different γ values for S-channel, on top of its V-channel's correction of using γ value of 1.2.

iii. rain disruption on the glass;

iv. rain's random droplets caused by splashing at buildings; and

v. rain's random splashing on puddles of water on ground surface.

Figure 6.15 shows the NIQE comparison scores for five real rain test images (Type I to V)

Figure 6.13: Original real rain image, CycleGAN rain removal results and the W-CycleGANs' rain removal results (from left to right), together with their corresponding zoomed in regions below them, from left to right.

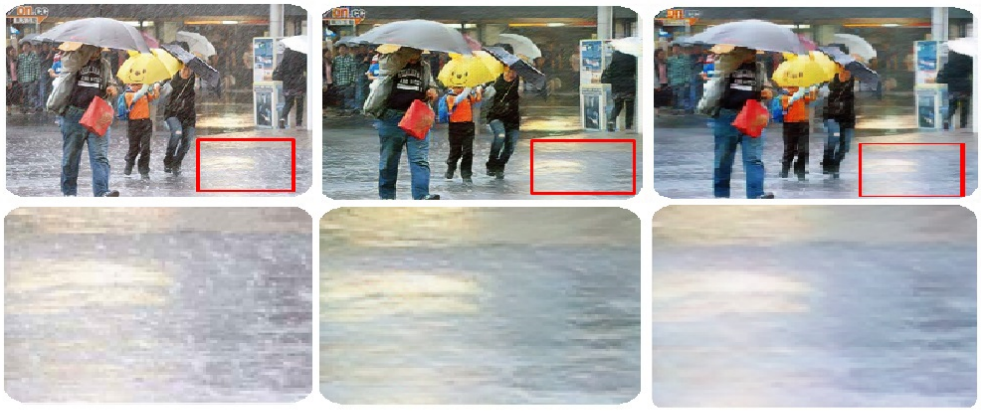

Figure 6.14: Original real rain image, CycleGAN rain removal results and the W-CycleGANs' rain removal results (from left to right), together with their corresponding zoomed in regions below them, from left to right.

and the CycleGAN's, MS-CycleGANs' and W-CycleGANs' reconstructed images (from left to right respectively). Both the MS-CycleGANs and the W-CycleGANs have shown good rain removal results, in terms of their lower NIQE quantitative values, as compared to the CycleGAN's and the original rain's scores. Qualitatively or visually, both algorithms' rain-removed images also looked more natural with lesser distortion. Both algorithms are able to remove rain from the images that are classified into the five types almost completely, including the low-frequency rain components. The objects and backgrounds in their reconstructed images are also preserved with no obvious defect or loss of information after the rain removal. In comparison, the CycleGAN's results show not only incomplete rain removal capability, but also have the same discolouration problem discussed before. To summarize the comparisons shown in the figure, both the MS-CycleGANs and the W-CycleGANs have not only successfully removed the rain streaks with

very little rain residue remained, the objects at the foreground and the background scene are found to be retained with the original color and structural details, when compared to the original rain image.

Figure 6.15: The NIQE scores for five real rain test images (Type I to V), compared to CycleGAN's, MS-CycleGANs' and W-CycleGANs' reconstructed images (from left to right respectively).

Table 6.1: NIQE scores for original real rain images compared to ID-CGAN's, CycleGAN's, MS-CycleGANs' and W-CycleGANs' reconstructed images. The NIQE's mean (first row) and standard deviation (second row) values are derived from a test data set comprises 50 selected test images covering all five types of real rain phenomena.

	Original Rain	ID-CGAN	CycleGAN	MS-CycleGANs	W-CycleGANs
NIQE (mean)	5.41	4.89	4.77	4.56	4.72
NIQE (standard deviation)	1.93	1.34	1.21	1.07	1.20

To compare all algorithms' quantitatively, including the ID-CGAN, we have carefully selected 'sharp' test images from a pool of real rain images to test their reconstructed images. This test data set is selected from hundreds of real rain images from the five types of rain distortions, to ensure a good spread of real rain phenomena and to exclude any abnormal distortion caused by the camera's defocus or poor resolution, which may affect the NIQE score in the first place. This will help us to achieve a meaningful comparison of all algorithms statistically. Table 6.1 shows that the CycleGAN, MS-CycleGANs and W-CycleGANs are all showing lower NIQE mean scores compared to the ID-CGAN and original real rain images. The results show that the MS-CycleGANs has the best performance out of all algorithms. This quantitative result is important for us to conclude that NIQE is a good evaluation metric for real rain removal's performance comparison, without using the synthetically generated ground truth image-pairs. In addition, we can quantitatively determine that the MS-CycleGANs is the better multi-scale CycleGANs algorithms, compared to the W-CycleGANs. The lower NIQE scores for MS-CycleGANs may be contributed by the image enhancement capability of the multi-scale Laplacian pyramid, compared to a single-level wavelet decomposition by the W-CycleGANs.

6.6 Conclusion and future work

To conclude the results presented in this chapter, both multi-scale representations of the CycleGAN, namely the MS-CycleGANs and the W-CycleGANs, have demonstrated very good rain removal performances both qualitatively and quantitatively, compared to the CycleGAN and the state-of-the-art ID-CGAN. This brings us to an important juncture to conclude the thesis in the next chapter.

Chapter 7

Conclusion and future directions

7.1 Conclusion

In this thesis, a number of fundamental contributions are made in the area of removing real rain effects from images as follows:

- A methodology for the parameterization of a CAE to ensure optimum latent representations or image reconstruction performance is proposed.
- The CycleGAN [8] is proposed for the first time as a practical and effective way to reconstruct images under visual disruption caused by real rain, without the need of paired rain and rain-free training data for learning to address the disruption problem posed by real rain, as compared to other CNNs and GANs methods [5], [35]- [39]. The distinct advantage of the CycleGAN lies in its training simplicity. Due to the impracticality of collecting aligned rain and rain-free image-pairs to train rain removal GANs for real outdoor task in computer vision, and that synthetically generated rain image data sets may not faithfully representing real rain characteristics, the CycleGAN is proposed as a practical way to reconstruct images under visual disruption caused by rain without the need for rain and rain-free input image-pairs.
- A comparison methodology based on rain physics for comparing the effectiveness of real rain removal in five types of rain distortion is proposed in this thesis, based on a rain physics model [6]. This methodology can be applied to the majority of outdoor rain conditions.
- Using the limited synthetic training data set for training the networks that was provided by the researchers who published the ID-CGAN methodology [5], the CycleGAN rain re-

moval performance was evaluated on real rain data set against the ID-CGAN, the state-of-the-art rain removal method. The results have demonstrated that the proposed CycleGAN approach is more robust and effective in removing all rain defects from images compared to the ID-CGAN network. The CycleGAN is able to remove all five types of rain distortion, while preserving image scene details. In comparison, the ID-CGAN was unable to remove bright and short rain streaks, leaving behind many white artifacts and was not able to enhance images with low brightness or contrast.

- Although paired domain examples are widely used for quantitative evaluation on existing rain removal techniques due to the non-existence of associated rain-free images for real rain images, our analysis has gone beyond such synthetically added rain streaks' removal quantitative study. For a fair comparison with other deep learning techniques, all proposed algorithms are trained on the same set of 700 image-pairs, the effectiveness of the algorithms are evaluated on real rain. To our best knowledge, no one has ever performed a detailed study on quantifying real rain removal using the NIQE quantitative measure, based on various real rain phenomena which represent the majority of outdoor rain conditions [2]. In other words, we have used these rain categories to evaluate the effectiveness of our rain removal algorithm in this research, both qualitatively and quantitatively.

- We have developed a robust rain removal methodology for real rain, without any rain priors model assumptions or targeted rain regions estimated, which may cause incomplete rain removal and loss of scene details, to combat the complex visual effects of real rain. In this thesis, we have provided the motivation of using a multi-scale CycleGAN approach to remove the total visual effects of rain, based on its appearance in the near-field or far-field. We think that a holistic approach which consists of a two-stage rain removal process is necessary to capture all rain's statistics before removing them. Hence, both the multi-scale MS-CycleGANs and W-CycleGANs algorithms are proposed as effective ways to remove real rain disruption from images in this thesis. The proposed multi-scale frameworks are evaluated against the state-of-the-art representative, the CycleGAN and the ID-CGAN, both qualitatively using visual check and quantitatively using NIQE quantitative measure, and show better rain removal performances. Besides, both multi-scale approaches are also practical methods, as they can learn the rain and rain-free image's statistics without using paired training samples, as required by other methods.

7.2 Future Works

With more and more outdoor unmanned vehicles and security surveillance systems using computer vision and artificial intelligence technologies, research on removing unwanted rain distortion from digital images without blurring scene content is on the rise, especially for video applications.

The initial problem statement of this research is for removing rain disruption in a single image. Since most real-world outdoor computer vision is using video application, it will be more practical to use the CycleGAN technique in removing outdoor weather disruption for a video task, rather than just for single-image rain removal purpose. Hence, we need to extend the CycleGAN algorithm in removing weather distortion to outdoor rain video. Although switching from image to video processing using the CycleGAN may involve tedious programming change, we could generalize the program to extract and process individual frames from any video with different codec format (MP4, MOV, etc.) using the CycleGAN, and gain the advantage of video processing as opposed to single-image processing. Video processing provides additional temporal information of rain streaks to the CycleGAN algorithm to further improve the CycleGAN performance to cover a wider range of applications like removing weather disruption in an outdoor character recognition task, etc. Besides, it could also help us identify individual rain streaks easily for effective evaluation of different algorithms. Hence, for future works described in Section 7.2.1, the CycleGAN approach can be extended to removing rain in video, by making use of the additional temporal information of rain in video.

So far, our research has mainly focused on removing rain from a single image which is a more challenging task than rain removal from video because the pixels of certain scene are occluded or blurred by rain pixels at the moment the images are captured. But without the availability of the spatio-temporal information of rain pixels in a video, the process of removing rain pixels may cause partial rain pixels' removal and blurring of scene content in certain complex scene situations. Hence as part of the future work, we recommend to extend this practical approach from a single image rain removal to video rain removal, to make full use of the spatio-temporal information in the video, to make the whole rain removal process complete. One recommendation is to use a deep recurrent network such as the Recurrent Neural Network (RNN), combined with the CycleGAN approach, to form a Recurrent CycleGAN network to remove rain completely in video applications.

7.2.1 Recurrent CycleGAN

Our initial video processing experiment results using the CycleGAN has shown that the CycleGAN is able to remove mist-type heavy rain at the scene's background well and enhance the scene details as well. But large rain streaks, especially near rain streaks dropping at non-terminal velocity (e.g., from the roof of a building), would cause the rain removal algorithm using CycleGAN to fail. Hence, we could further improve the existing CycleGAN network to tackle video degradation for a real outdoor recognition task by exploiting the spatio-temporal information of individual rain streaks. This would result in a combination of the recurrent neural network and the CycleGAN which we name it as the Recurrent CycleGAN.

There may be a few reasons why video processing showed that the CycleGAN failed to remove real rain streaks but could remove the mist-type rain distortion. Firstly, the CycleGAN is still the same network trained using the synthetic rain streaks samples provided by the IDCGAN [5], for an apple-to-apple comparison with other algorithms. As discussed in earlier chapters, synthetic hand-drawn rain streaks may not represent real rain statistics well. The CycleGAN has an advantage over other networks in its ability to learn the underlying relationship between the rain and rain-free domain without the need of paired rain and rain-free domain examples. Although it could understand the statistical differences between the rain and rain-free domain well to remove a wide range of rain disruptions as shown by our results, the training data set provided in [5] by Zhang et al. was not synthesized according to the five types of real rain images proposed in this thesis. Hence, to remove real large rain streaks as shown in the videos, we need to train the CycleGAN using real rain data set instead of synthetic data set.

Secondly, the distinct advantage of video processing as opposed to single-image processing lies in its ability to help us identify the individual rain streaks using the additional temporal information of high-speed moving rain streaks provided by video. This could help us identify individual rain streaks easily in video and evaluate the effectiveness of our algorithms. Lastly, the CycleGAN rain removal algorithm was trained for single-image rain removal purpose. Hence, it could not make use of the additional temporal information provided by high-speed moving rain streaks during its network training. This could enable us to address the main weaknesses of the CycleGAN network for further improvement, using the extra spatio-temporal information in video. Due to the lack of ground-truth images for a quantitative comparison of rain removal algorithms, we could make use of the NIQE values of successive frames in the videos to compare different rain removal algorithms quantitatively.

For a rain video clip made up of a sequence of rain images, Recurrent Neural Networks (RNN) can improve the rain removal capability of the CycleGAN by conditioning its image

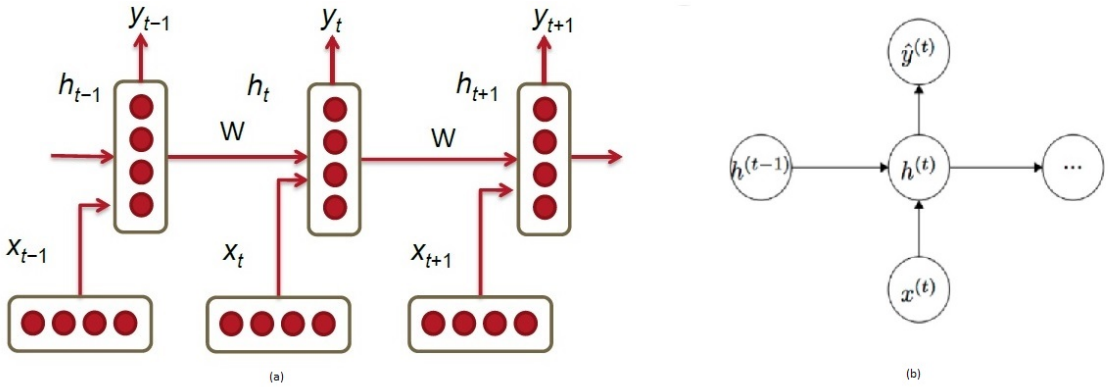

Figure 7.1: a.) The Recurrent Neural Network architecture with 3 time-steps shown. b.) The inputs and outputs to a neuron of a RNN. Note that x_t and y_t are the input and output image vectors of the RNN at time t respectively. W is a weight matrix consists of $W^{(hx)}$ which is the weights matrix used to condition the input image vector x_t, and $W^{(hh)}$ which is the weights matrix used to condition the output of the previous time-step, h_{t-1} (reproduced from [112]).

generators on all previous images in the sequence of the video. Conventional non-deep learning video rain removal techniques, as discussed in Chapter 2, rely only on a finite number of previous images in the sequence to remove rain disruption. Figure 7.1(a) shows a RNN with three consecutive time-steps in the video, for illustration of RNN architecture [112]. Each rectangular box represents a hidden layer of RNN at a particular time-step, t. Each of the hidden layer which is made up of a number of neurons, performs a linear mapping on its inputs, followed by a non-linear sigmoid operation $\sigma(.)$. At each time-step, the next predicted output $\hat{y}_t$ and output features h_t (Equation 7.1) of each hidden layer are produced by the previous step's output h_{t-1} and the next image vector, x_t, along with their associated weights $W^{(hh)}$ and $W^{(hx)}$ respectively, which are used to condition them. Figure 7.1(b) illustrates each RNN neuron's inputs and outputs. Note that $\hat{y}_t$ is the next predicted image given the output of the previous step (i.e. h_{t-1}) and the last observed image vector x_t.

$$h_t = \sigma(W^{(hh)}h_{t-1} + W^{(hx)}x_t) \tag{7.1}$$

Figure 7.2 shows our proposed Recurrent CycleGAN network architecture for both domains X and Y at time t. As shown in the figure, the recurrent CycleGAN model's Generator X to Y works by taking an input image from domain X (rain domain) which is Input X at time-step t, and the same generator's previous output image, Generated X at time-step $(t-1)$, to condition the same generator. The generator will then transform the given rain image from domain X to an image in the target domain Y (rain-free domain) at the same time-step t. The same process is happening at domain Y for the Generator Y to X.

The Recurrent CycleGAN networks have two important features: 1.) they take the form of recurrent structures which can refine their extracted features progressively in each recurrence;

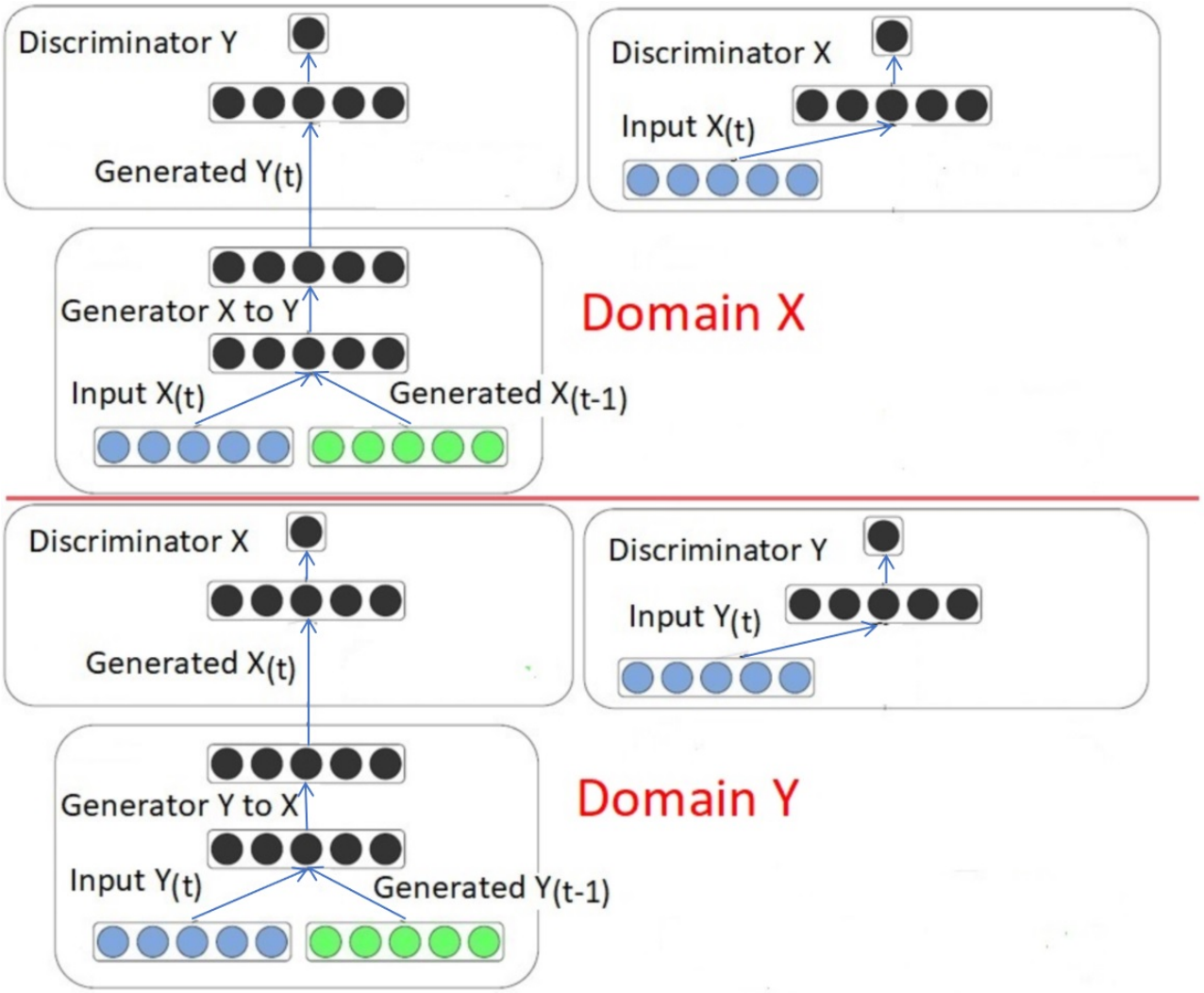

Figure 7.2: The Recurrent CycleGAN network architecture for both domains X and Y, at time step t.

2.) The weights at time-step t are conditioned on the previous time-step's output image Generated $X_{(t-1)}$. For each recurrence, the output image Generated $Y_{(t)}$ should have better rain removal capability as it can produce a more successful mapping that is absent in unpaired video data set. This previous output image acts as the additional conditioning image for the Recurrent CycleGAN. The extra rain information in Generated $X_{(t-1)}$ is needed to direct the data generation process, or to control the mode (rain or rain-free) of the data being generated. Notice that in the networks shown in Figure 7.2, Domain Y work exactly the same way as Domain X, except that they work with input images in the rain-free domain.

In conclusion, both the multi-scale and the Recurrent CycleGAN networks can be applied on areas of shortcomings of the CycleGAN where the CycleGAN could fail to remove normal large rain streaks and near rain streaks dropping at non-terminal velocity (e.g., from the roof of a building) due to the few reasons discussed previously. By exploiting both the multi-scale

spatial information and the additional temporal information of individual rain streaks in video, we are confident that it could help us remove a wide range of weather disruption on videos and images successfully. Besides, video processing could make use of the additional temporal information provided by high-speed moving rain streaks in the videos to evaluate individual real rain streaks to compare different rain removal algorithms both qualitatively and quantitatively. This methodology could overcome the current limitation of the literature in comparing individual rain removal algorithm's performance objectively due to the lack of ground truth for real rain images.

Bibliography

[1] Tang, L., Lim, L. and Siebert, P., "Parameterization of a Convolutional Autoencoder for Reconstruction of Small Images," pp. 1426-1431, doi: 10.1109/ICARCV.2018.8581254, 2018.

[2] M. Tang, H. Lim and S. Paul, "Removal of Visual Disruption Caused by Rain Using Cycle-Consistent Generative Adversarial Networks," in ECCV 2018 Workshop Proceedings, Part V, 2018.

[3] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proc. IEEE, 86(11):2278-2324, 1998.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, et al., "Generative Adversarial Nets," in NIPS, 2014.

[5] Zhang, He, Sindagi, Vishwanath and Patel, Vishal M., "Image De-raining Using a Conditional Generative Adversarial Network," in arXiv preprint arXiv:1701.05957, 2017.

[6] Garg, K., and Nayar, S. K., "Vision and Rain," in IJCV, pp. 3-27, 2007. http://dx.doi.org/10.1007/s11263-006-0028-6, 2007.

[7] Mittal, A., R. Soundararajan and A. C. Bovik, "Making a Completely Blind Image Quality Analyzer," IEEE Signal Processing Letters. Vol. 22, Number 3, March 2013, pp. 209–212, 2013.

[8] Jun-Yan Zhu, Taesung Park, Phillip Isola and Alexei A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," in arxiv, 2017.

[9] S. Li, W. Ren, J. Zhang, J. Yu and X. Guo, "Single image rain removal via a deep decomposition-composition network," in Computer Vis Image Underst, 186:8–57, 2019.

[10] Masci, J., Meier, U., Ciresan, D. and Schmidhuber, J., "Stacked convolutional autoencoders for hierarchical feature extraction," in Artificial Neural Networks and Machine Learning-ICANN 2011, pages 52-59, Springer, 2011.

[11] Tan, S. and Li, B., "Stacked convolutional autoencoders for steganalysis of digital images," in Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA), pages 1-4, IEEE, 2014.

[12] Leng, B., Guo, S., Zhang, X. and Xiong, Z., "3d object retrieval with stacked local convolutional autoencoder," in Signal Processing, 112:119-128, 2015.

[13] J. Bossu, N. Hauti‘ere and J. Tarel, "Rain or snow detection in image sequences through use of a histogram of orientation of streaks," in International journal of computer vision, 93(3):348-367, 2011.

[14] Garg, K., and Nayar, S. K., "Detection and removal of rain from videos," in Proc. CVPR, 1, 528-535, 2004. http://dx.doi.org/10.1109/CVPR.2004.1315077, 2004.

[15] N. Brewer and N. Liu, "Using the shape characteristics of rain to identify and remove rain from video," in Joint IAPR International Workshops on SPR and SSPR, pages 451-458, 2008.

[16] Barnum, P. C., Narasimhan, S. and Kanade, T., "Analysis of rain and snow in frequency space," in IJCV, 86(2-3), 256-274. http://dx.doi.org/10.1007/s11263-008-0200-2, 2010.

[17] Zhang, X., Li, H., Qi, Y., Leow, W. K. and Ng, T. K., "Rain removal in video by combining temporal and chromatic properties," in Proc. ICME, July, pp. 461-464, 2006. http://dx.doi.org/10.1109/ICME.2006.262572, 2006.

[18] Y. Li, R. T. Tan, X. Guo, J. Lu and M. S. Brown, "Rain streak removal using layer priors," in IEEE Conference on Computer Vision and Pattern Recognition, pp. 2736-2744, 2016.

[19] Yu Li, Robby T. Tan, X. Guo, J. Lu and Michael S. Brown, "Single Image Rain Streak Separation Using Layer Priors," in IEEE Transactions on Image Processing, 2017.

[20] Fu, Y. H., Kang, L. W., Lin, C. W. and Hsu. C. T., "Single-frame-based rain removal via image decomposition," in Proc. IEEE Int. Conf. Acoustics, Speech & Signal Processing, May 2011, Prague, Czech Republic. http://dx.doi.org/1453-1456.10.1109/ICASSP.2011.5946766, 2011.

[21] Li-Wei Kang, Chia-Wen Lin, Che-Tsung Lin and Yu-Chen Lin, "Self-learning-based rain streak removal for image/video," in IEEE International Symposium on Circuits and Systems (ISCAS), in IEEE Conference Publications, 2012, pp. 1871-1874, doi: 10.1109/ISCAS.2012.6271635, 2012.

[22] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu, "Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition," in IEEE Transactions on Image Processing (TIP), vol. 21, no. 4, April 2012, pp. 1742-1755, doi: 10.1109/TIP.2011.2179057, 2012.

[23] Duan-Yu Chen, Chien-Cheng Chen and Li-Wei Kang, "Visual Depth Guided Color Image Rain Streaks Removal Using Sparse Coding," in IEEE Transactions on Circuits and Systems for Video Technology (TCSVT), vol. 24, issue 8, 2014, pp. 1430-1455, doi:10.1109/TCSVT.2014.2308627, 2014.

[24] De-An Huang, Li-Wei Kang, Min-Chun Yang, Chia-Wen Lin and Y. Wang, "Context-Aware Single Image Rain Removal," in IEEE International Conference on Multimedia and Expo (ICME), IEEE Conference Publications, 2012, pp. 164-169, doi: 10.1109/ICME.2012.92, 2012.

[25] De-An Huang, Li-Wei Kang, Y.-C.F. Wang and Chia-Wen Lin, "Self-Learning Based Image Decomposition with Applications to Single Image Denoising," in IEEE Transactions on Multimedia (TMM), vol. 16, issue 1, 2014, pp. 83-93, doi: 10.1109/TMM.2013.2284759, 2014.

[26] S.-H. Sun, S.-P. Fan and Y.-C. F. Wang, "Exploiting image structural similarity for single image rain removal," in IEEE Int'l Conf. Image Processing, pages 4482-4486, 2014.

[27] Y. Luo, Y. Xu and H. Ji, "Removing rain from a single image via discriminative sparse coding," in ICCV, 2015.

[28] C.-H. Son and X.-P. Zhang, "Rain removal via shrinkage of sparse codes and learned rain dictionary," in IEEE ICME, 2016.

[29] H. Zhang and V. M. Patel, "Convolutional sparse coding-based image decomposition," in British Machine Vision Conference, 2016.

[30] Yi-Lei Chen and Chiou-Ting Hsu, "A Generalized Low-Rank Appearance Model for Spatio-temporally Correlated Rain Streaks," in IEEE International Conference on Computer Vision (ICCV), IEEE Conference Publications, 2013, pp. 1968-1975, doi:10.1109/ICCV.2013.247, 2013.

[31] Jin-Hwan Kim, Chul Lee, Jae-Young Sim and Chang-Su Kim, "Single-Image Deraining Using an Adaptive Nonlocal Means Filter," in IEEE International Conference on Image Processing (ICIP), IEEE Conference Publications, 2013, pp. 914-917, doi:10.1109/ICIP.2013.6738189, 2013.

[32] Soo-Chang Pei, Yu-Tai Tsai and Chen-Yu Lee, "Removing Rain and Snow in a Single Image Using Saturation and Visibility Features," in IEEE International Conference on Multimedia and Expo Workshops (ICMEW), IEEE Conference Publications, 2014, pp. 1-6, doi:10.1109/ICMEW.2014.6890551, 2014.

[33] Jing Xu, Wei Zhao, Peng Liu and Xianglong Tang, "Removing rain and snow in a single image using guided filter," in IEEE International Conference on Computer Science and Automation Engineering (CSAE), in IEEE Conference Publications, vol. 2, 2012, pp. 304-307, doi: 10.1109/CSAE.2012.6272780, 2012.

[34] Jing Xu, Wei Zhao, Peng Liu and Xianglong Tang, "An Improved Guidance Image Based Method to Remove Rain and Snow in a Single Image," in Computer and Information Science, vol. 5, May 2012, pp. 49-55, doi:10.5539/cis.v5n3p49, 2012.

[35] X. Fu, J. Huang, X. Ding, Y. Liao and J. Paisley, "Clearing the Skies: A deep network architecture for single-image rain removal," in arXiv eprints Sep., 2016.

[36] Fu X, Huang J, Zeng D, Huang Y, Ding X and Paisley J, "Removing rain from single images via a deep detail network," in IEEE Conference on Computer Vision and Pattern Recognition, 2017.

[37] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo and S. Yan, "Joint rain detection and removal via iterative region dependent multi-task learning," in CoRR, vol. abs/1609.07769, 2016, http://arxiv.org/abs/1609.07769, 2016.

[38] W. Yang, Robby T. Tan, J. Feng, J. Liu, Z. Guo and S. Yan, "Deep Joint Rain Detection and Removal from a Single Image," in Computer Vision and Pattern Recognition, CVPR, 2017.

[39] Wang, Chaoyue, et al., "Perceptual Adversarial Networks for Image-to-Image Transformation," in arXiv preprint arXiv:1706.09138, 2017.

[40] S. Zhou, J. Gong, G. Xiong, H. Chen and K. Lagnemma, "Road detection using support vector machine based on online learning and evaluation," in Intelligent Vehicle Research Center, Beijing, 2010.

[41] T. Birdal, and A. Ercil, "Real-Time Automated road lane and car detection for autonomous driving," in Sabanci University, Faculty of Engineering and Natural Science, 2007.

[42] Denton, Emily, and et al., "Deep Generative Image Models using a Laplacian Pyramid of Adversarial Networks," in arXiv preprint arXiv:1506.05751, 2015.

[43] M. Mirza and S. Osindero, "Conditional Generative Adversarial Nets," in CoRR, abs/1411.1784, 2014.

[44] G.Saravanan, G Yamuna and R.Vivek, "A Color Image Enhancement based on Discrete Wavelet Transform," in IJCA Proceedings on National Conference on Emerging Trends in Information and Communication Technology, 2013.

[45] S. Huang, F. Cheng and Y. Chiu, "Efficient Contrast Enhancement Using Adaptive Gamma Correction with Weighting Distribution," in IEEE Transactions on Image Processing, vol. 22, no. 3, pp. 1032-1041, March, 2013.

[46] A. Varun, B. Sayantan and A. Mark, "Compressible Latent-Space Invertible Networks for Generative Model-Constrained Image Reconstruction," in arvix:2007.02462, 2020.

[47] Guclu, U. and van Gerven, M. A., "Deep neural networks reveal a gradient in the complexity of neural representations across the brain's ventral visual pathway," in arXiv preprint arXiv:1411.6422, 2014.

[48] Kruger, N., Janssen, P., Kalkan, S., Lappe, M., Leonardis, A., Piater, J., Rodriguez-Sanchez, A. J., and Wiskott, L., "Deep hierarchies in the primate visual cortex: What can we learn for computer vision," in Pattern Analysis and Machine Intelligence, IEEE Transactions on, 35(8):1847-1871, 2013.

[49] I. Goodfellow, Y. Bengio and A. Courville, "Deep Learning," in Book in preparation for MIT Press, http://www.deeplearningbook.org, 2016.

[50] D. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in arXiv:1312.6114, 2014.

[51] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang et. al., "Photo-realistic single image super-resolution using a generative adversarial network," in arXiv preprint arXiv:1609.04802, 2016.

[52] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, "Context encoders: Feature learning by inpainting," in arXiv preprint arXiv:1604.07379, 2016.

[53] A. Radford, L. Metz and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in arXiv preprint arXiv:1511.06434, 2015.

[54] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou and Alexei A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in CVPR, 2017.

[55] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow and B. Frey, "Adversarial autoencoders," in International Conference on Learning Representations (ICLR), arXiv:1511.05644, San Juan, 2016.

[56] Beard, K.V. and Chuang, C.H., "A new model for the equilibrium shape of raindrops," in Journal of Atmospheric Science, 44(11):1509-1524, 1987.

[57] Garg, K. and Nayar, S.K., "Photorealistic rendering of rain streaks," in ACM Trans. on Graphics (also Proc. of ACM SIGGRAPH), 25:996-1002, 2006.

[58] Marshall, J.S. and Palmer,W.M.K., "The distribution of raindrops with sizes," in Journal of Meteorology, 5:165-166, 1948.

[59] Manning, R.M., "Stochastic Electromagnetic Image Propagation," in McGraw-Hill, Inc., 1993.

[60] Wang, P. K. and H. R. Pruppacher, "Acceleration to terminal velocity of cloud and raindrops," in J. Atmos. Sci., 16:275-280, 1977.

[61] Gunn, R. and Kinzer, G.D., "Terminal velocity for water droplet in stagnant air," in Journal of Meteorology, 6:243-248, 1949.

[62] Wang, T. and Clifford, S.F., "Use of rainfall-induced optical scintillations to measure path-averaged rain parameters," in Journal of the Optical Society of America, 8:927-937, 1975.

[63] Tripathi, A.K. and Mukhopadhyay, S., "Removal of rain from videos: a review," in SIViP 8, 1421–1430, 2014, https://doi.org/10.1007/s11760-012-0373-6, 2014.

[64] Narasimhan, S. G. and Nayar, S. K., "Contrast Restoration of Weather Degraded Images," in IEEE Transaction on Pattern Analysis and Machine Intelligence, 25(6): 713-724, 2003.

[65] Shariah, S., Marhaban, M. M, Ramli, A. R. and Mohd. Noor, S. B. B., "Rain Detection and Removal using a New Algorithm," in Proceedings of the National Geoscience Conference, 2011.

[66] Starik, S. and Werman, M., "Simulation of Rain in Videos," in Proceedings of Texture, the 3rd International Workshop on Texture Analysis and Synthesis, 2003.

[67] Wan-Joo Park and Kwae-Hi Lee, "Rain Removal using Kalman Filter in Video," in International Conference on Smart Manufacturing Application, 2008.

[68] Liu, P. X., Jing, L. J., Tang, X. and Zhao, W., "A Rain Removal Method using Chromatic Property for Image Sequence," in Proceeding of the 11th Joint Conference on Information Sciences, 2008.

[69] Raima Hassim and Abdullah Bade, "Taxonomy of rain detection and rain removal techniques," in Transactions on Science and Technology, 2(2): 28 - 35, 2013.

[70] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in Proceedings of IEEE International Conference on Computer Vision, pp. 839-846, 1998.

[71] M. Elad and M. Aharon, "Image denoising via sparse redundant representations over learned dictionaries," in IEEE Trans. on Image Processing, vol. 15, pp. 3736-3745, 2006.

[72] K. Ghana Sheila and S. N. Deepak, "Review on Methods to Fix Number of Hidden Neurons in Neural networks," in Mathematical Problems in Engineering, vol. 2013, Article ID 425740, 11 pages, doi:10.1155/2013/425740, 2013.

[73] Karsoliya S., "Approximating number of hidden layer neurons in multiple hidden layer BPNN architecture," in Int J Eng Trends Technol 2012;3(6):713-7, 2012.

[74] Berry, M.J.A. and Linoff, G., "Data Mining Techniques," in NY: John Wiley and Sons, 1997.

[75] Claude E. Shannon, "A Mathematical Theory of Communication," in Bell System Technical Journal, 27(3): 379-423, doi:10.1002/j.1538-7305.1948.tb01338, 1948.

[76] Haykin, S., "Neural Networks and Learning Machines," in Prentice Hall, 3rd Edition, 2009.

[77] R.C. Gonzalez, R.E. Woods and S.L. Eddins, "Digital Image Processing Using MATLAB," in 3rd Ed., Pearson-Prentice Hall, 2020.

[78] Bengio, Y., Courville, A. and Vincent, P., "Representation learning: a review and new perspectives," in IEEE Trans. Pattern Anal. Machine Intell. 35, 1798-1828, 2013.

[79] M. Liu and O. Tuzel, “Coupled generative adversarial networks,” in NIPS, pages 469–477, 2016.

[80] Y. Aytar, L. Castrejon, C. Vondrick, H. Pirsiavash and A. Torralba, “Cross-modal scene networks,” in arXiv preprint arXiv:1610.09003, 2016.

[81] M. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” in arXiv preprint arXiv:1703.00848, 2017.

[82] T. Zhou, P. Krahenbuhl, M. Aubry, Q. Huang and A. A. Efros, “Learning dense correspondence via 3d guided cycle consistency,” in CVPR, pages 117–126, 2016.

[83] C. Godard, O. Mac Aodha and G. J. Brostow, “Unsupervised monocular depth estimation with left-right consistency,” in arXiv preprint arXiv:1609.03677, 2016.

[84] Adam Paszke, Sam Gross, Soumith Chintala and Gregory Chanan, "PyTorch," in GitHub repository, GitHub, 2021. https://github.com/pytorch/pytorch

[85] Z. Wang and A. C. Bovik, "A universal image quality index," in IEEE signal processing letters, vol. 9, no. 3, pp. 81-84, 2002.

[86] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," in IEEE transactions on image processing, vol. 13, no. 4, pp. 600-612, 2004.

[87] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," in IEEE Transactions on Image Processing, vol. 15, no. 2, pp. 430-444, 2006.

[88] A. Rosenfeld, "Multiresolution Image Processing and Analysis," in volume 12, Springer-Verlag Berlin Heidelberg, 1984.

[89] T. Lindeberg, "Scale-space theory: a basic tool for analysis of structures at different scales," in Journal of Applied Statistics, 21(2). s.l.:s.n., pp. 224-270, 1994.

[90] Kutulakos, K. and Guerzhoy, M., "Image Pyramids," in csc320 lecture notes, University of Toronto, 2019. http://www.cs.toronto.edu/%7ejepson/csc320/notes/pyramids.pdf

[91] Edward H Adelson, Charles H Anderson, James R Bergen, Peter J Burt and Joan M Ogden, "Pyramid methods in image processing," in RCA engineer, 29(6):33-41, 1984.

[92] Witkin, A., "Scale-Space Filtering," in Proceedings of the International Joint Conference on Artificial Intelligence. s.l.:ACM Inc., pp. 1019-1021, 1983.

[93] Cyganek, B. and Siebert, J., "An introduction to 3D computer vision techniques and algorithms," in Chichester: Wiley, pp. 165 - 192, 2010.

[94] Y. Zheng, X. Yu, M. Liu and S. Zhang, "Residual Multiscale Based Single Image Deraining," in BMVC, 2019.

[95] Dey, S., "Hands-on Image Processing with Python," in United Kingdom: PACKT Publishing Limited, 2018.

[96] B. Vidakovic and P. Mller, "An Introduction to Wavelets," in Springer New York, 1999.

[97] T. Guo, H. S. Mousavi, T. H. Vu and V. Monga, "Deep Wavelet Prediction for Image Super-Resolution," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, 2017.

[98] V. Gupta, R. Mahle, and R. S. Shriwas, "Image denoising using wavelet transform method," in 2013 Tenth International Conference on Wireless and Op- tical Communications Networks (WOCN), pages 1–4, IEEE, 2013.

[99] J. Song, J. Jeong, D. Park, H. Kim, D. Seo and J. Ye, "Unsupervised Denoising for Satellite Imagery using Wavelet Subband CycleGAN," in arXiv:2002.09847, 2020.

[100] G. Strang, "Wavelet transforms versus Fourier transforms," in Bulletin of the American Mathematical Society, Volume28, pp. 288-305, 1993.

[101] P. Liu, H. Zhang, K. Zhang, L. Lin and W. Zuo, "Multi-level wavelet-cnn for image restoration," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, pages 773–782, 2018.

[102] W. Yang, J. Liu, S. Yang and Z. Guo, "Scale-free single image deraining via visibility-enhanced recurrent wavelet learning," in IEEE Transactions on Image Processing, 28(6):2948–2961, 2019.

[103] Pascale, Danny, "A Review of RGB color spaces...from xyY to R'G'B'," in BabelColor, 2003.

[104] J. Zhao, J. Xie, R. Xiong, S. Ma, T. Huang and W. Gao, "Pyramid Convolutional Network for Single Image Deraining," in CVPR Workshops, 2019.

[105] Alessandro CAPRA, Alfio CASTORINA and Silvia CORCHS, "Dynamic Range Optimization by Local Contrast Correction and Histogram Image Analysis," in Digest of Technical Papers International Conference on Consumer Electronics, pp. 309-310, doi: 10.1109/ICCE.2006.1598434, February, 2006.

[106] G.Padma Priya and T.Venkateswarlu, "An optimal Gamma Correction Based Image Contrast Enhancement Using DWT-SVD," in International Journal of Research in Advent Technology (IJRAT), VOLUME-6 ISSUE-8, pp. 2184-2190, August, 2018.

[107] Alvy Ray Smith., "Gamma Correction," in Computer Graphics (CG) Memos, Technical Memo 9, Sep, 1995.

[108] David R. Bull, "Digital Picture Formats and Representations," in Communicating Pictures, 2014.

[109] Hany Farid, "Blind Inverse Gamma Correction," in IEEE Transactions on Image Processing 10(10):1428-33, February, 2001, DOI: 10.1109/83.951529, 2001.

[110] Y. Chiu, F. Cheng and S. Huang, "Efficient contrast enhancement using adaptive gamma correction and cumulative intensity distribution," in IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, 2011, pp. 2946-2950, doi: 10.1109/ICSMC.2011.6084119, 2011.

[111] Mittal, A., A. K. Moorthy and A. C. Bovik. "No-Reference Image Quality Assessment in the Spatial Domain," in IEEE Transactions on Image Processing. Vol. 21, Number 12, December, pp. 4695–4708, 2012.

[112] Milad Mohammadi, Rohit Mundra and Richard Socher, "Deep Learning for NLP," in CS224d lecture notes, Stanford University, 2015. https://cs224d.stanford.edu/lecture_notes/notes4.pdf