Deep Learning-based Techniques for Image and Video Restoration

Thesis submitted to the
Indian Institute of Technology Guwahati
for the award of the degree

of

Doctor of Philosophy
in
Computer Science and Engineering

Submitted by
Prasen Kumar Sharma

Under the guidance of
Dr. Arijit Sur

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
March 7, 2022

Abstract

The efficiency of several real-time vision tasks, such as *outdoor surveillance*, *satellite optical image analysis*, and *autonomous vehicle navigation systems*, significantly relies on the visual quality of the images and videos. It severely degrades when presented with noisy or corrupt data, *e.g.,* images taken in adverse rainy or hazy weather conditions. Therefore, it is of utmost importance to propose robust and effective methods that remove the noise and restore the visual quality of the degraded images and videos. Most of the existing best-published works are built upon various prior-based frameworks that suffer from blocky visual artifacts.

Recently, efforts have been afoot towards data-driven approaches due to their improved performance over prior-based schemes. It has been evident from the existing literature that learning-based methods for low-level vision tasks are not only superior in terms of performance but also deployment-friendly for executing real-time applications. With this motivation, this thesis presents efficient data-driven methods for the following low-level vision tasks: (a) *single image de-raining*, (b) *single image de-hazing*, and (c) *video de-raining*. In what follows are the four significant contributions of this dissertation.

In the first contributory chapter, a deep learning-based scheme has been proposed for the task of single image de-raining. The designed methodology exploits the spatial domain aspects of the rain-streaks due to their pseudo-periodic nature. It has been experimentally shown that processing over the luminance channel of the rainy image alone may lead to a remarkable performance gain over correlated color-space. Further, the presented method utilizes efficient pixel upscaling

over conventional schemes to evade blocky artifacts in the de-rained images.

In the second contributory chapter, transformed domain characteristics of the rain streaks in the image are exploited for de-noising. There are two contributions in this chapter. In the first work, it has been experimentally shown that rain-streaks, due to their additive pseudo-periodic nature, leaves some traces in the uncorrelated discrete Fourier domain, which can be utilized by the deep models. Towards this, a learning-based approach has been presented that takes Fourier domain coefficients of the rainy images as input and estimates the Fourier domain coefficients of the de-rained images. In the second work, we have proposed a novel learning-based scheme that utilizes a combination of spatial and correlated transformed domain characteristics of the rain-streaks in an image. In particular, the discrete Haar wavelets have been exploited to retain the various aspects of the rain-streaks along with different directions. The proposed dual-domain learning has shown remarkable performance gain over exclusive domain learning.

In the third contributory chapter, a scale-space invariant *Convolutional Neural Network* (CNN) has been presented for the task of single image de-hazing. Unlike rain-streaks, the haze in an image exponentially varies with the depth of the pixels. It has been observed that the *Laplacians of Gaussian* (LoG) exhibits a variety of edgy structures at different scales in the hazy image. The invariance above has been achieved by exploiting the LoG cues as a supervised cost function to optimize the proposed deep model. The proposed scheme has been tested against 14 best-published works using 15 image quality metrics to demonstrate its efficacy.

In the final contributory chapter, the task of video de-raining has been addressed. Unlike the image, video de-raining has an additional complexity of retaining the temporal smoothness in the de-rained videos. Existing approaches tend to separate the spatial and temporal en-

hancement modules. However, in this work, a unified deep CNN has been presented that simultaneously optimizes the spatial and temporal characteristics of the de-rained videos. For this, the proposed method has been engineered with a multi-contextual design to capture a wide variety of spatial features and a 3D convolution-based sub-module to optimize temporal consistency. The presented work has been verified against 10 existing methods using multiple image quality metrics.

Finally, the thesis is concluded by summarizing the significant contributions and proposing some relevant future research directions.

Declaration

I certify that:

a. The work contained in this thesis is original and has been done by me under the guidance of my supervisor.

b. The work has not been submitted to any other Institute for any degree or diploma.

c. I have followed the guidelines provided by the Institute in preparing the thesis.

d. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.

e. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the thesis and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.

Prasen Kumar Sharma

Copyright

Signature of Author.....

Prasen Kumar Sharma

Certificate

This is to certify that this thesis entitled **"Deep Learning-based Techniques for Image and Video Restoration"** being submitted by **Prasen Kumar Sharma**, to Department of Computer Science and Engineering, Indian Institute of Technology Guwahati, for partial fulfillment of the award of the degree of Doctor of Philosophy, is a bonafide work carried out by him under our supervision and guidance. The thesis, in our opinion, is worthy of consideration for award of the degree of Doctor of Philosophy in accordance with the regulation of the institute. To the best of our knowledge, it has not been submitted elsewhere for the award of the degree.

.....

Dr. Arijit Sur
Associate Professor
Department of Computer Science and Engineering
IIT Guwahati

Dedicated to

My parents &

My late maternal grandparents &

Brother

Whose blessing, love and inspiration paved my path of success

Acknowledgments

I want to express my deepest gratitude to my thesis supervisor, Dr. Arijit Sur, for his valuable guidance and suggestions. Thank you so much for picking me up as your student at the start of my career. You have always motivated me to aim higher. Thank you for the freedom you gave, being with me during my failures, and teaching me how to be positive and never lose hope. I'd also like to express my sincere gratitude to all my Ph.D. committee members: Dr. Prabin Kumar Bora, Dr. Sanasam Ranbir Singh, Dr. Rashmi Dutta Baruah, for their valuable comments and suggestions to improve my work. A special thanks to my M.Tech thesis supervisors: Dr. Rajiv Ranjan Sahay, and Dr. Chittaranjan Mandal, for introducing me to this fascinating field of research in Computer Vision and Deep Learning. I want to extend my special thanks to Dept. of Computer Science and Engineering, IIT Guwahati, and Mr. Bhriguraj Borah, Mr. Nanu Alan Kachari for their valuable and unconditional technical support.

I am thankful to my seniors Shuvendu Rana, Sibaji Gaj, Satish Kumar, Brijesh Singh, Sathisha B., Neelakshi Sarma, Abhishek Mehta, and Akash Anil for their initial support and guidance. I am also thankful to my lab mates, Anirban Lekharu and Avinash Kumar Chouhan, for their unconditional support and encouragement. I'd also like to extend my sincere thanks to my juniors Priyankar Jain, Shashank Huddedar Anil, Sujoy Ghosh, Udbhav Chugh and many others for their unconditional support.

Most importantly, I want to thank my parents, my late maternal grandparents and my brother. It's their love and sacrifices, which has made this possible. The values they put, their thoughts, and suggestions will remain in me, forever.

Contents

CONTENTS

List of Figures

LIST OF FIGURES

List of Tables

LIST OF TABLES

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

List of Acronyms

BN *Batch Normalization*

CNN *Convolutional Neural Network*

ResNet *Deep Residual Network*

DenseNet *Dense Convolutional Network*

GMM *Gaussian Mixture Model*

DFT *Discrete Fourier Transform*

IDFT *Inverse Discrete Fourier Transform*

DWT *Discrete Wavelet Transform*

HOG *Histogram of Oriented Gradients*

DoF *Depth of Field*

DCP *Dark Channel Prior*

LoG *Laplacians of Gaussian*

DoG *Difference of Gaussian*

GAN *Generative Adversarial Network*

ILSVRC *Imagenet Large Scale Visual Recognition Challenge*

MSE *Mean Squared Error*

MS-SSIM *Multi-scale Structural Similarity Index*

PReLU *Parametric ReLU*

PSNR *Peak Signal to Noise Ratio*

ReLU *Rectified Linear Unit*

PReLU *Parametric Rectified Linear Unit*

RGB *Red Green Blue*

SSIM *Structural Similarity Index*

HVS *Human Visual System*

TV *Total Variation*

LR *Low Resolution*

HR *High Resolution*

MSE *Mean Squared Error*

FSIM *The Feature Similarity Index*

ILSVRC *Imagenet Large Scale Visual Recognition Challenge*

SpEED-QA *Spatial Efficient Entropic Differencing for Image and Video Quality*

ISBI *International Symposium on Biomedical Imaging*

NIQE *Naturalness Image Quality Evaluator*

PIQE *Perception based Image Quality Evaluator*

LPIPS *Learned Perceptual Image Patch Similarity*

BRISQUE *Blind/Referenceless Image Spatial Quality Evaluator*

Haar PSI *Haar Wavelet-based Perceptual Similarity Index*

GMSD *Gradient Magnitude Similarity Deviation*

BLIINDS *BLind Image Integrity Notator using DCT-Statistics*

TanH *Hyperbolic Tangent*

UQI *Universal-Image-Quality Index*

VIF *Visual Information Fidelity*

List of Symbols

$\mathbf{x}_n$	The noisy image
$\mathbf{y}_c$	The noise-free image
$\mathbf{r}_s$	The rain-streak map
$\mathbf{t}_m$	The transmission map
$\mathbf{l}_a$	The atmospheric light
β	The scattering coefficient (unless specified)
$\mathbf{d}(p)$	Depth at pixel p
$\mathbf{V}_r$	The rainy video
$\mathbf{V}_c$	The rain-free video
$\mathbf{f}_r$	The rainy frame in $\mathbf{V}_r$
$\mathbf{f}_c$	The rainy-free frame in $\mathbf{V}_c$
$\mathbf{f}_m$	The rain-streak map corresponding to $\mathbf{f}_r$ frame in $\mathbf{V}_r$
μ	The mean
σ	The standard deviation
$\mathcal{L}$	Loss function

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

Chapter 1

Introduction

Images and videos are graphical representations of our everyday life. The camera simulates the human eye across various domains spanning from everyday photography to remote sensing and astronomy, medical imaging, to cellular microscopy. In each case, there exists an object or scene that we wish to observe, analyze and remember throughout our lives. However, an image or video may not always consist of desired objects and scenes in their most refined form. They may be severely affected by unpredictable impairments due to the natural noise present in the real-world scene. Further, the efficiency of most real-time applications, such as *surveillance*, *satellite optical image analysis*, and *autonomous vehicles navigation*, *etc.,* heavily relies on the visual quality of images and videos. Their performance severely degrades when presented with a noisy or corrupt image and video.

For *e.g.,* in adverse weather conditions, such as rainy, hazy, or snowy, it may be profoundly challenging for an autonomous vehicle to figure out what lies ahead and lead to a crashed navigation system[1] [2], ultimately life-threatening. To explain, Machiraju *et al.* [2] demonstrated that the `AutoPilot` [1] model predicted different steering angle for the same scene with different noise scenarios, as shown in Figure 1.1. Therefore, it becomes necessary to propose robust and efficient

[1]Korean Competition Shows Weather Still a Challenge for Autonomous Car (https://news.ycombinator.com/item?id=8699438)

without fog with fog

Figure 1.1: *Steering angle prediction in the absence/presence of fog by an autonomous vehicle navigation system [1] for the same scene [2].*

methods that remove the noise and restore the visual quality of the degraded images and videos to avoid the aforementioned fatal flaws.

These low-level vision tasks are challenging to solve due to their inherent ill-posed nature. Many existing methods have proposed prior-based schemes to convert such ill-posed problems into well-posed. However, such methods lack efficiency and generalization by neglecting the other auxiliary information present in the real-world data, eventually resulting in blocky visual artifacts. Over the last decade, data-driven approaches have been successfully implemented in many ill-posed computer vision tasks [3, 14, 18] with superior efficiency and generalization capability.

This thesis proposes novel learning-based methods for the restoration of images and videos. In particular, we present the data-driven approaches for (1) *single image de-raining*, (2) *single image de-hazing*, and (3) *video de-raining*. In what follows, we briefly describe the overview of these problems.

1.1 Image Restoration

Given a noisy image $\mathbf{x}_n \in \mathbb{R}^{C\times M\times N}$, image restoration aims to estimate the de-noised image $\mathbf{y}_c \in \mathbb{R}^{C\times M\times N}$ using a mapping function f as $\mathbf{y}_c = f(\mathbf{x}_n)$.

For *e.g.,* in the case of single image rain-streak removal (*alias de-raining*), one may model the above relationship as

$$\mathbf{y}_c = \mathbf{x}_n - \mathbf{r}_s, \tag{1.1}$$

Rainy Rain-map De-rained

Figure 1.2: *Graphical demonstration of single image rain-streak removal problem.*

where, $\mathbf{r}_s \in \mathbb{R}^{C\times M\times N}$ is corresponding rain-streak map, and C, M, N denote the channel and spatial dimension of the image. Given $\mathbf{x}_n$, the goal of the single image de-raining problem is to estimate the de-rained image $\mathbf{y}_c$. Figure 1.2 shows the qualitative demonstration of the above defined task.

Another example of image restoration problem is single image haze removal (*alias de-hazing*), where the hazy image $\mathbf{x}_n$ can be expressed as

$$\mathbf{x}_n = \mathbf{y}_c.\mathbf{t}_m + \mathbf{l}_a.(1 - \mathbf{t}_m), \tag{1.2}$$

where, $\mathbf{t}_m \in \mathbb{R}^{C\times M\times N}$, $\mathbf{l}_a \in \mathbb{R}^{C\times M\times N}$ denote the transmission map and global atmospheric light, respectively. The transmission map $\mathbf{t}_m$ exponentially varies with the depth of the pixel p, and can be written as

$$\mathbf{t}_m(p) = e^{-\beta.\mathbf{d}(p)}, \tag{1.3}$$

where, $\mathbf{d}$ denotes the depth map and β is a scattering coefficient. Given the hazy image $\mathbf{x}_n$, the task of single image de-hazing aims to recover the clean image $\mathbf{y}_c$ based on the following inverse relation

$$\mathbf{y}_c = \frac{\mathbf{x}_n - \mathbf{l}_a.(1 - \mathbf{t}_m)}{\mathbf{t}_m}. \tag{1.4}$$

Figure 1.3 depicts the graphical demonstration of the single image de-hazing task.

Hazy De-hazed

Figure 1.3: *Graphical demonstration of single image haze removal problem.*

1.2 Video Restoration

The task of video restoration has an additional complexity of preserving the temporal consistency in the de-noised video. To formally address the task of video de-raining, let $\mathcal{V}_r$, $\mathcal{V}_c$ denote the rainy and its corresponding clean video, respectively. The mathematical formulation of the rain-streaks in the video can be expressed as

$$f_r^i = f_m^i + f_c^i, \tag{1.5}$$

where, $f_r^i \in \mathbb{R}^{C\times M\times N}$, $f_m^i \in \mathbb{R}^{C\times M\times N}$, and $f_c^i \in \mathbb{R}^{C\times M\times N}$ denote the i^{th} rainy, rain-streak map and clean frames, respectively in $\mathcal{V}_r$, $\mathcal{V}_c$. Figure 1.4 depicts the

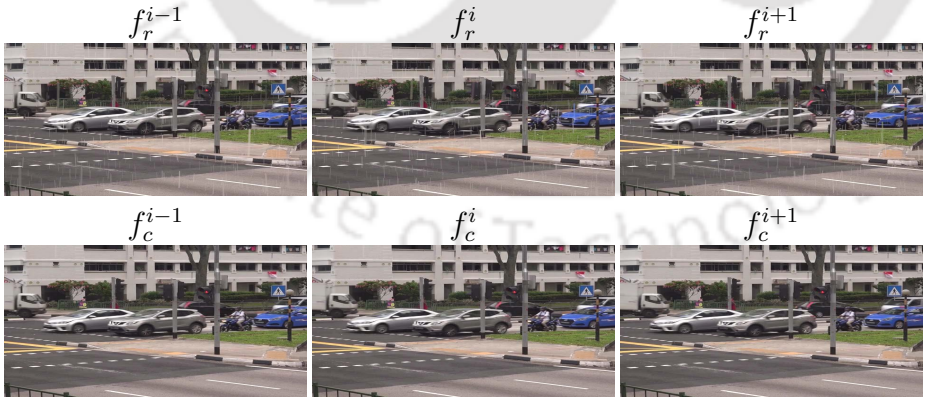


Figure 1.4: *A qualitative demonstration of the task of video de-raining.*

graphical demonstration of the video de-raining task, where the top row denotes rainy frames, and bottom row shows the de-rained frames.

Before Restoration After Restoration

Figure 1.5: *Object detection results on real-world examples of single image de-hazing. It can be observed that the de-hazed images have more detected object categories compared to their hazy counterparts.*

1.3 Applications

It has been widely established that the low-level vision tasks (*e.g.,* de-noising) may substantially improve the performance of high-level vision tasks, such as classification, object detection and semantic segmentation [19–21], as shown in Figure 1.5. With this motivation, the image and video restoration may have the following promising application domains:

- ***Autonomous Vehicles:*** Unmanned navigation may require images and videos to be rain or haze-free to achieve the desired efficiency during autonomous pilotages, such as traffic sign detection [22], vehicle tracking [23], and scene understanding [24].
- ***Satellite Imaging:*** Considered low-level vision tasks in this thesis, can be used to improve the visibility of hazy and cloudy satellite optical images for efficient spatial feature extraction, *e.g.,* roads, land, and forests [25, 26].
- ***Underwater Navigation:*** Quite a few works have established that the visibility enhancement and de-hazing of the degraded underwater images

may result in improved high-level vision tasks, such as object detection [27, 28] and diver's 2D pose estimation [21].

- ***Outdoor Surveillance and Tracking:*** Improved visibility can also be beneficial for outdoor surveillance and tracking [29].

1.4 Literature Survey

This section briefly introduces the recent developments on the considered low-level vision tasks in this thesis.

1.4.1 Image De-Raining

1.4.1.1 Layer Separation Methods

These methods are built upon the image decomposition framework, where a rainy image is split into a rain-free background image and rain-streak map. The split, in general, considers a variety of image characteristics and priors, such as (a) *sparsity*, (b) *low-rank representation*, and (c) *Gaussian Mixture Model* (GMM).

Sparsity-based methods

Gu *et al.* [30] decomposes the rainy image into two layers, one which portrays large-scale structures is approximated by sparse analysis representation and the other by synthesis sparse representation, which exhibits the finer textures in the image for the rain streaks removal. In [31], the authors divided the rain streaks into sparse and dense categories using a matrix decomposition framework. Later, Zhang *et al.* [32] learns the sparsity and low-rank representation-based convolutional kernels to determine the clear image and rain streaks.

Low-rank representation-based methods

Existing methods in this class assume that the rain-streaks follow a similar pattern and orientation within an image. Therefore, it may be beneficial to model the rain-streak map using a low-rank decomposition framework.

Based on this, earlier works in [33–38] proposed the methods that are built upon the bilateral filtering, which has been used to split the frequency parts

(*low and high*) of the rainy image. The high-frequency part is further split into rainy textures and non-rain geometric details. It is achieved by using structured dictionary learning, the *Histogram of Oriented Gradients* (HOG), eigen colors, and *Depth of Field* (DoF). In [39], the authors proposed a method that extracts the periodic noise which follows the line pattern, such as rain streaks, stripes, fences *etc.,* in an image. Yeh *et al.* [40] proposed a method of decomposition that retrieves the high and low-frequency components of a rainy image by using the Gaussian filter. To remove the rain streaks from the high-frequency part, the Canny edge detection [41] algorithm has been used. The rain streaks present in the low-frequency part are removed by using a non-negative matrix factorization technique. Whereas Wang *et al.* [42] observed that the high-frequency component of the rainy image consists of most of the rain streak part. Therefore, the rain-free details from the high-frequency component are obtained using the dictionary-based learning method.

However, it has been observed that low-rank decomposition may result in removing the important texture of the image as well, in addition to the rain-streaks. Further, it may be difficult for a low-rank representation-based model to extract rain-streaks with multiple orientations and scales.

Gaussian mixture models-based methods

With the above motivation, [43, 44] considered the apriori image processing domain knowledge, such as (a) centralized sparse representation, (b) predicted rain direction, (c) rain streak layer, and (d) GMMs, to incorporate multiple orientation and scales of the rain-streaks and estimate the rain-free background image from the corresponding rainy image.

1.4.1.2 Deep Learning-based Methods

In recent literature, many deep learning-based methods have handled the image de-raining task with outstanding efficiency. For *e.g.*, Fu *et al.* [16, 45] devised a model using a *Deep Residual Network* (ResNet) [6] that estimates the negative residual map, given a rainy image. It can be added to rainy images to get their

Light Medium Heavy

Figure 1.6: *A qualitative demonstration of the synthetic images proposed in [3] to show the different rain-streak densities in the images.*

de-rained versions, based on the following relation,

$$\begin{aligned} \mathbf{x}_n &= \mathbf{r}_s + \mathbf{y}_c, \\ \mathbf{y}_c &= \mathbf{x}_n + (-\mathbf{r}_s), \end{aligned} \tag{1.6}$$

where, $-\mathbf{r}_s$ denotes the learned negative residual map.

Shen *et al.* [46] devised a deep CNN that takes Haar [47] wavelets coefficients (DWT) along with the *Dark Channel Prior* (DCP) [48] as input to the proposed model for image de-raining. In [49], the authors introduced a model based on the conditional *Generative Adversarial Network* (GAN) [50] to estimate the high-quality rain-free images. Yang *et al.* [51] proposed a joint rain-streak detection and removal method which uses a binary map. If a value referencing a pixel in an image in the binary map is '1', it means that the corresponding pixel has been affected by the rain-streaks. They simulate heavy rain by modeling the appearance of rain streak accumulation, various shapes, and directions. Zhang *et al.* [9] proposed a model, namely IDCGAN, that utilizes a conditional GAN framework [50] with baseline model built upon an encoder-decoder [7] framework and perceptual loss function [8]. Later, in [3], the authors proposed a method, namely DID-MDN, that first classify the rain-streak map into *light, medium* and *heavy* rain-streaks and then used a *Dense Convolutional Network* (DenseNet) [52] based model to estimate the rain-streak map. A graphical demonstration of different rain densities has been shown in Figure 1.6.

1.4.2 Image De-Hazing

Over the past few years, many researchers have delved into the domain of single image de-hazing and proposed various prior and learning-based methods for the same with outstanding results.

1.4.2.1 Handcrafted Features-based Arts

One of the early yet significant contributions towards single image de-hazing was proposed by He *et al.* [48] that uses the DCP to estimate the statistical distribution in clean images taken outdoors. It works based on the assumption that at least one of the three color channels (red, green, and blue) has a low-intensity value (*known as dark pixels*) that belongs to the haze-free image. The DCP algorithm works very efficiently in non-sky regions that have dense haze but fails to recover haze-free components in case of areas similar to atmospheric light-map or with edgy structures. Yu *et al.* [53] assumed the scattering model (similar to Eq. (1.2)) as

$$\mathbf{x}_h = \mathbf{l}_a.\mathbf{y}_c.\mathbf{t}_m + \mathbf{l}_a.(1 - \mathbf{t}_m), \tag{1.7}$$

and proposed a fast bilateral filter to smoothen the fine texture of the image for single image de-fogging. The term $\mathbf{l}_a.\mathbf{y}_c.\mathbf{t}_m$ is also known as *Direct Attenuation*.

To remove the halo artifacts generated by the DCP [48], He *et al.* [54] proposed the guided image filtering method which preserves the edges. However, it fails to enhance the contrast in the haze-free images. Meng *et al.* [55] (EIDBR) modelled the de-hazing problem as an optimization task based on a weighted ℓ_1-norm based contextual regularization. Zhu *et al.* [56] devised a Color Attenuation Prior (CAP) to recover the depth of the hazy image. The transmission map is further estimated using the recovered depth scene to restore the de-hazed image. Choi *et al.* [57] utilized the measurable deviations (DEFADE) from statistical regularities in the hazy and haze-free images. Berman *et al.* [58] introduced a deterministic approach, called Non-Local Image Dehazing (NLD) based on the haze-lines, that directly estimates the haze-free images.

1.4.2.2 Deep Learning-based Methods

With the evolution of deep CNNs [59], many learning-based schemes [10, 60–65] have been introduced for single image de-hazing task.

Ren *et al.* [60] proposed a multi-scale deep CNN (MSCNN) for single image de-hazing. The proposed method in [60] directly maps the input hazy image to the transmission map using a coarse deep CNN. The haze-free image is later recovered using the inverse of Eq. (1.4). Li *et al.* [61] proposed a method, namely AOD-Net, which does not predict the transmission and airlight maps separately. Instead, it generates the haze-free image using a lightweight CNN, unifying transmission map, and airlight estimation steps within a single unit known as the *K-Estimation block*. Zhang *et al.* [10] proposed a method, called DCPDN, that estimates the transmission and atmospheric light-maps by using a pyramid [66] densely connected CNN and a U-Net [7] respectively. The haze-free image is then recovered by using Eq. (1.4). The estimated haze-free image is further enhanced using a joint discriminator. Santra *et al.* [62] proposed a CNN based Patch Quality Comparator (PQC) to estimate the de-hazed images. The method proposed in [67] uses unpaired training based on the Cycle-GAN framework [68] for the image de-hazing task. Yang *et al.* [63] leveraged the benefits of deep learning-based and prior-based methods in a single framework for haze removal problem.

1.4.3 Video De-Raining

The challenge of preserving the temporal consistency after rain-removal in the videos gives an edge over the single image rain removal methods.

1.4.3.1 Layer Separation Methods

Garg and Nayar [69] did the pioneering work on video de-raining using the dynamics and photometric properties of the rain. They have also experimented on camera parameters to determine its impact on rain detection [70]. The study of camera calibration parameters to assess its impact on rain detection was ground-

breaking. Still, it was ineffective against moving objects, often mixing them with rain-streaks, leading to false positives. Also, the proposed work could not generalize on the scale and direction of rain streaks. Zhang *et al.* [71] exploited the chromatic and temporal information of the rain-streaks. They applied K-means clustering on the intensity histogram, helping to differentiate between moving objects and rain streaks. Nevertheless, this method often leads to background smoothing. Barnum *et al.* [72] proposed a statistical model in the Fourier domain to detect and suppress the rain frequencies. However, these adjustments in the frequency domain often lead to poor results in the spatial domain, especially for dense rain streaks. For dynamic videos with dense rain streaks, Chen *et al.* [73] used the optical flow of moving objects and suggested a de-raining model. However, they could not supervise the moving camera scenes. A few of the existing schemes utilized temporal correlations and matrix decomposition framework for video de-raining, such as Kim *et al.* [4] (TCL) and Ren *et al.* [74]. However, such methods fail when the displacement of the objects in the frame is large.

Wei *et al.* [12] proposed the rain encoding (SE) using patches of Gaussian, allowing it to adapt to various rain configurations. Jiang *et al.* [75] considered the intrinsic characteristics of rain-streaks and used the alternation direction method of multipliers algorithm to solve it efficiently. Jiang *et al.* [11] proposed another method, FastDerain, to solve the problem using the directional gradient prior and applied the SALSA algorithm for solving the minimization criteria of the model.

1.4.3.2 Deep Learning-based Methods

Liu *et al.* [76] proposed a recurrent rain model (J4RNet) to remove rain-streaks as well as reduce occlusion. They further proposed another method DualFlow in [77] where they do single image de-raining, which then guides the video de-raining model, assisted by optical flow, to account for dynamic scenes. Chen *et al.* [14] devised a super-pixel segmentation-based deep CNN that estimates the rain streak location and occluded background contents using a robust frame alignment technique. In [13], authors utilized the multi-scale convolution filters

Over-coloring White-dots Artifacts

Figure 1.7: *A graphical demonstration of the over-coloring and white-dots artifacts in the de-rained results of existing works.*

(MSCSC) to capture the rain-streaks at different scales in a frame. Liu *et al.* [78] proposed the D3R-Net that depicts both rain-streaks and occlusion by integrating the hybrid model and useful motion segmentation context information.

1.5 Motivation and Objectives

Over the last few years, various methods have been proposed for the considered image and video restoration tasks from the aforementioned related developments. It should be mentioned that each method presented a novel approach towards the respective restoration based on the distribution of the noise and various characteristics of input images. However, the current literature has the following limitations:

- With respect to *single image de-raining*, the current literature has the following limitations:

    1. A majority of the existing works ([3, 9] *to name a few*) suffer from the problem of over-coloring and white-dot artifacts in the de-rained images (see Figure 1.7).
    2. Secondly, considering the pseudo-periodic nature of the rain-streaks, the transformed domain, particularly uncorrelated, has not been thoroughly explored in the case of deep learning.

Color-degradation Halo Checkerboards

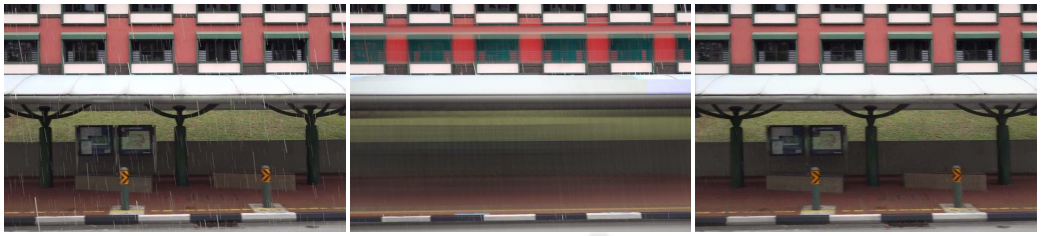
Figure 1.8: *A graphical demonstration of the color degradation, halo and checkerboard artifacts in the de-hazed results of existing works.*

- With respect to *single image de-hazing*, the existing literature has the following limitations:

  1. Most of the existing methods, to name a few [48, 55], suffer from color degradation, halo and checkerboard artifacts that prevail around the high-intensity regions, and edgy structures in the de-hazed images (see Figure 1.8).
  2. A majority of the existing learning-based schemes, to name a few [10, 18], separately estimates the transmission and atmospheric light maps, which incurs the computational cost.
  3. And most importantly, considering the exponentially varying haziness, the scale-space property of the hazy images has not been exploited in the case of deep model engineering.

- With respect to *video de-raining*, the current literature has the following issues:

  1. It is observed that existing works suffer from following shortcomings in the de-rained videos (see Figure 6.1 for a sample demonstration)-
     (a) out-of-shape transformation of the objects and motion blur (*observed in* [4]),
     (b) incomplete rain-removal (*observed in* [11]),



R_i T_i O_i

Figure 1.9: *Sample results that show the existing method TCL [4] suffer from out-of-shape reconstruction and heavy motion blur. R, and O denote the rainy, and our results. T denote TCL [4] result.*

(c) the unwanted imprints of previous frames (*observed in* [13]),

(d) inability to retain the objects that align with the rain-streaks (*observed in* [14]).

2. The existing approaches, due to the separate modules for spatial and temporal enhancement, may require more computational resources.

Motivated by the shortcomings mentioned above of the existing literature and inspired by the success of deep learning techniques in many low-level computer vision applications, the primary objectives of this dissertation are defined as follows:

- Develop a deep learning-based method that enhances the de-rained image without introducing any over-coloring and white-dot artifacts.
- Devise the deep learning-based end-to-end models that can leverage both correlated and uncorrelated transformed domain coefficients of rain-streaks in an image.
- Propose a deep learning-based end-to-end model that utilizes the scale-space information of the hazy image for single image de-hazing.
- Devise a deep learning-based approach that incorporates the exponentially increasing cost functions and multi-contextual 3D convolutions to remove the spatial and temporal inconsistencies in the de-rained videos.

1.6 Contribution of the Thesis

The major contributions of this dissertation are as follows.

1.6.1 Exploiting Efficient Spatial Upscaling for Single Image De-Raining

In the first contributory chapter, spatial domain characteristics of the rain-streaks in an image have been explored. Towards this, an adversarial learning-based framework has been presented to ensure that de-rained images are indistinguishable from their corresponding clean images. Unlike the conventional upsampling schemes that may induce blocky or popularly known as checkerboard artifacts, the proposed method utilizes the efficient sub-pixel upscaling for the better spatial reconstruction of the de-rained image without any visual artifacts. The proposed approach has been tested against best-published existing methods on benchmark datasets to demonstrate its efficacy.

1.6.2 Exploiting Transformed Domain Features for Single Image De-Raining

In the second contributory chapter of the thesis, the transformed domain characteristics of the rainy image have been investigated. To address this, we have first presented an end-to-end learning-based scheme for single image de-raining where the *Discrete Fourier Transform* (DFT) of the rainy image has been explored. It has been observed that the rain-streaks, which are pseudo-periodic and additive in nature, leave some traces in the Fourier domain. Despite the loss of spatial correlation in the Fourier domain, we have shown that if carefully crafted, such traces of noise can be learned using a deep CNN.

In the second work, we propose to combine the transform domain with the spatial for the image de-raining task. For this, we have exploited the *Discrete Wavelet Transform* (DWT) coefficients of the rainy images to expose the rain-streaks in the frequency domain. It has been observed that the Haar-wavelet

sub-bands of the rainy image preserve a variety of rain-streaks information. In addition to the spatial domain features of the rainy image, these sub-bands have been utilized by the proposed method for generating the artifacts-free clean images.

1.6.3 A Probe Towards Scale-Space Invariant Conditional GAN for Single Image De-Hazing

In the third contributory chapter of this work, the problem of single image haze removal has been addressed. Unlike the rain-streaks, the haze depicts an exponentially-varying noise along the depth in an image. It has been observed that the LoG retains a variety of edgy information in a hazy image. The proposed work has utilized these edgy structures retained by the LoG as a cost function and presented a novel deep CNN for single image de-hazing. Besides this, the proposed scheme also relies on adversarial training and perceptual loss function. The proposed approach has been tested against 14 best-published works using 15 image quality metrics to show its robustness.

1.6.4 Frame-Recurrent Multi-Contextual Adversarial Network for Video De-Raining

In the final contributory chapter, the problem of video de-raining has been addressed. Unlike images, there is an additional complexity of temporal smoothness in video de-noising that requires much attention while crafting a deep CNN. Most of the existing works separately consider the restoration of the spatial and temporal quality of the noisy video. Whereas, we propose a novel unified adversarial learning-based model that exploits the multi-contextual approach for generating the de-rained videos. It has been shown that the proposed scheme is quantization-friendly and can be deployed onto real-time devices.

1.7 Organization of the Thesis:

This PhD dissertation consists of seven chapters. The first chapter includes an introduction to considered image and video restoration tasks, followed by a literature survey, research motivations and objectives of the thesis, and contributions of the thesis. The rest of the thesis is organized as follows:

- Chpater 2 presents the elemental background of the research, including preparatory concepts of convolutional neural networks, image quality evaluation metrics, and datasets, utilized for the extensive experimentation.
- In Chapter 3, a deep learning-based method has been presented that utilizes the efficient sub-pixel upscaling instead of traditional deconvolution for spatial enhancement of the de-rained images.
- Chapter 4 describes the incorporation of both correlated (DWT) and uncorrelated (DFT) transformed domain coefficients of rain-streaks in deep CNN for single image de-raining.
- Chapter 5 presents a scale-space invariant conditional GAN that incorporates the LoG for single image de-hazing.
- In Chapter 6, a frame-recurrent unified deep CNN has been presented for an efficient video de-raining. The proposed approach unifies the spatial and temporal enhancement of the de-rained videos and incorporates multi-contextual 3D convolution for the same.
- In Chapter 7, this Ph.D. dissertation has been concluded along with the possible future works.

1.8 Summary

In this chapter, a brief overview of considered image and video restoration tasks is presented to formulate the scope of research works. Then, the respective existing

developments are briefly described. Based on the shortcomings of the existing literature, the objectives of the research are defined. Finally, a brief description of the contributions and the organization of the thesis have been presented. In chapter 2, the elemental background for this dissertation is presented for better understanding.

Chapter 2

Research Background

In this chapter, a brief overview of some preliminary concepts related to the topics of interest are presented. In particular, firstly, it includes a brief introduction to some of the image transformations like *Discrete Fourier Transform* (DFT), *Discrete Wavelet Transform* (DWT), and a filter named *Laplacians of Gaussian* (LoG). In what follows the fundamental concepts of deep *Convolutional Neural Network* (CNN), such as *VGGNet* [5], ResNet [6], *U-Net* [7]. The concepts of these CNNs are used to construct the proposed deep learning-based methods for image and video restoration. In addition, various image quality metrics for evaluating the proposed methods and corresponding datasets used for experiments are also discussed in this chapter.

2.1 Discrete Fourier Transformation

This section briefly addresses the theory of image transformation into the discrete Fourier domain (DFT). To formally describe, let i be an image, k be a kernel and f_{map} be the cross-correlation map [79, p. 329] generated by using convolution operation as

$$f_{map} = i * k, \tag{2.1}$$

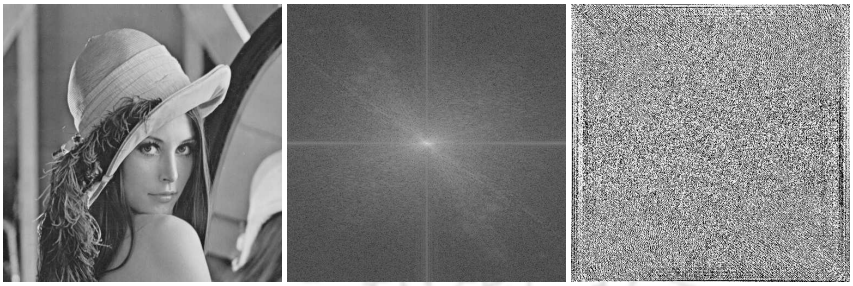


Image Magnitude Phase

Figure 2.1: *Sample demonstration of the magnitude and phase spectrums in discrete Fourier domain of in image.*

where $*$ is a convolution operator. Now, Eq. (2.1), based on the convolution theorem [80], can be expressed in the DFT domain ($\mathbf{F}$) as [81]

$$\mathbf{F}[f_{map}] = \mathbf{F}[i * k] = \mathbf{F}[i] \odot \mathbf{F}[k], \tag{2.2}$$

where $\odot$ is Hadamard point-wise multiplication operator [82]. The DFT [83] of an image i can be calculated as

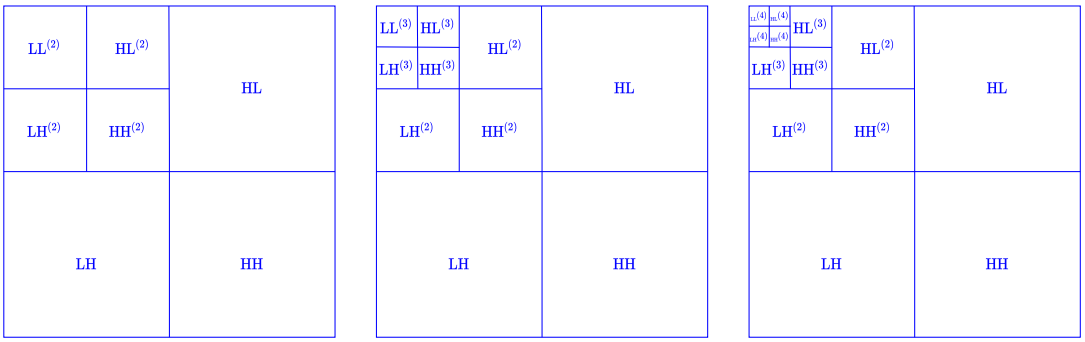
$$\mathbf{F}[u, v] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} i[m, n] \ \exp\left\{ -j2\pi . \frac{Nmu + Mnv}{MN} \right\} \tag{2.3}$$

where $j = \sqrt[2]{-1}$, $[u,\ v]$ are horizontal and vertical frequency locations in transformed domain, $\exp(\mathbf{jx}) = \cos \mathbf{x} + j \sin \mathbf{x}$ based on Euler's identity [84], for each pixel location (x, y) in spatial domain. The result of $\mathbf{F}$ on an image has complex numbers of the form $a_{u,v} + j.b_{u,v}$ where $a_{u,v}$, $b_{u,v}$ are real and imaginary coefficients respectively. For processing, we can either use the pair of real and imaginary parts or the pair of magnitude and phase parts, of $\mathbf{F}$. The magnitude ($\mathbf{M}$) and phase ($\mathbf{P}$) spectrums can be calculated as

$$\begin{aligned} \mathbf{M}_{u,v} &= \sqrt[2]{a_{u,v}^2 + b_{u,v}^2} \\ \mathbf{P}_{u,v} &= \tan^{-1}\left[\frac{b_{u,v}}{a_{u,v}}\right] \end{aligned} \tag{2.4}$$

The *Inverse Discrete Fourier Transform* (IDFT) of $\mathbf{F}$ in Eq. 2.3 to recover the original image, can be calculated as

$$\mathbf{I}[m, n] = \frac{1}{H} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \mathbf{F}[u, v] \ \exp\left\{ j2\pi . \frac{Num + Mvn}{MN} \right\} \tag{2.5}$$



(a) Two levels of 2D-DWT (b) Three levels of 2D-DWT (c) Four levels of 2D-DWT

Figure 2.2: *Sample demonstration of a multi-resolution scheme after several levels of wavelet transform.*

where $H = M \times N$. The original image can also be reconstructed from the magnitude $\mathbf{M}$ and phase $\mathbf{P}$ spectrums by first calculating the value of $\mathbf{F}$ to be used in Eq. 2.5 as

$$\mathbf{F} = \mathbf{M} \times \exp(j \times \mathbf{P}) \tag{2.6}$$

An example of magnitude $\mathbf{M}$ and phase $\mathbf{P}$ spectrums for a given input image has been shown in Figure 2.1.

2.2 Discrete Haar-Wavelet Transformation

The first DWT was invented by the renowned Hungarian mathematician *Alfred Haar*. The DWT of an image generates the refined features at a different resolution. When an input image is processed using low-pass and high-pass filters, it generates four sub-bands, namely LL, LH, HL, and HH. Further at different resolutions, it is achieved by repeated processing of input image at different scales using low-pass and high-pass filters, as shown in Figure 2.2.

- LL: The upper left quadrant comprises all coefficients generated by using the low-pass filter along the height and then along the corresponding columns by using the low pass filter again. This sub-band is denoted by LL and represents the approximated version of the original 2D input at half the resolution.

- HL/LH: The lower left and upper right sub-bands are estimated along the height and width using low-pass and high-pass filters, alternatively. The LH sub-band retains the vertical edges. Whereas, in contrast, the HL sub-band depicts the horizontal edges very clearly.
- HH: The lower right sub-band can be estimated analogously to the upper left quadrant but by using the high pass filter, which belongs to the given wavelet. It retains the edges of the original image in the diagonal direction.

2.3 Laplacians of Gaussian

Figure 2.3: *Sample demonstration of the LoG maps of an input image. (a) Image, (b)* $\mathsf{G}(\mathbf{m},\mathbf{n},k\sigma)-\mathsf{G}(\mathbf{m},\mathbf{n},\sigma)$*, (c)* $\mathsf{G}(\mathbf{m},\mathbf{n},k^2\sigma)-\mathsf{G}(\mathbf{m},\mathbf{n},k\sigma)$*, (d)* $\mathsf{G}(\mathbf{m},\mathbf{n},k^3\sigma)-\mathsf{G}(\mathbf{m},\mathbf{n},k^2\sigma)$*, (e)* $\mathsf{G}(\mathbf{m},\mathbf{n},k^4\sigma)-\mathsf{G}(\mathbf{m},\mathbf{n},k^3\sigma)$*.*

Laplacians are isotropic measure of second-order spatial derivative filters used to find the regions of rapid intensity variations (edgy structures) in an image. These filters are sensitive to the noise present in the image, and hence, it is common first to smooth the given image using the Gaussian filter with standard deviation σ - the combined filter, in general, known as *Laplacians of Gaussian* (LoG). In general, it is written as

$$\begin{aligned}\mathbf{L}(\mathbf{m},\mathbf{n}) = \nabla^2\mathbf{F}(\mathbf{m},\mathbf{n}) &= \frac{\partial^2\mathbf{F}}{\partial\mathbf{m}^2}+\frac{\partial^2\mathbf{F}}{\partial\mathbf{n}^2} \\ &= -\frac{1}{\pi\sigma^4}\Big[1-\frac{\mathbf{m}^2+\mathbf{n}^2}{2\sigma^2}\Big]\exp\Big(-\frac{\mathbf{m}^2+\mathbf{n}^2}{2\sigma^2}\Big)\end{aligned} \tag{2.7}$$

where, $\mathbf{F}$ is a 2D signal (an image in this case) with pixel location $(\mathbf{m},\mathbf{n})$. Theoretically, the *Difference of Gaussian* (DoG) can be used to closely approximate

the LoG [85, 86] with various σ values as

$$\mathsf{G}(\mathbf{m}, \mathbf{n}, k\sigma) - \mathsf{G}(\mathbf{m}, \mathbf{n}, \sigma) \approx (k-1)\sigma^2\Delta^2\mathsf{G} \tag{2.8}$$

where, $\sigma^2\Delta^2\mathsf{G}$ denotes scale-normalized LoG and k, σ are typically set to $\sqrt{2}, 1.6$ respectively [85]. The 2D Gaussian kernel is defined as

$$\mathsf{G}(\mathbf{m}, \mathbf{n}, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\Big(-\frac{\mathbf{m}^2 + \mathbf{n}^2}{2\sigma^2}\Big) \tag{2.9}$$

Figure 2.3 depicts the sample LoG output at different scales for an input image.

2.4 Convolutional Neural Networks

Recent efforts have been afoot towards deep CNNs that have shown tremendous improvement for various tasks that span across different domains such as Computer Vision [87–89], Speech Processing [90, 91], Medical Imaging [92–94] and Natural Language Processing (NLP) [95–98], etc. The ability to automatically learn task-specific features allowed CNNs to achieve the best performance on a variety of benchmarks effectively. The real-time feasibility of CNNs has been possible by recent developments in computer hardware and the availability of large-scale data. In general, a deep CNN comprises of the following modules: (a) Convolutional layers, (b) Activation functions, (c) Pooling layers, and (d) Fully connected layers. The detailed information of each of these components are as follows:

1. **Convolution layer:** The traditional convolutional layer consists of a group of neurons that are learnable given the optimization function. In literature, they are widely known as filters or kernels. Compared to the input size, each kernel is spatially small and extends its reach either channel-wise or the full depth of the input volume. It moves across the spatial dimension of the input image and generates a dot product between the filter and the input at any position. The generated two-dimensional activation map represents the

correlation between the kernel and the input image. If a convolution layer has N set of kernels, then N number of activation maps can be generated. These activation maps may or may not have redundancy among each other. These activation maps are further given as input to an activation function.

2. **Activation function:** The goal of the activation functions is to induce non-linearity during the computation of the feature maps. Further, they clamp the input feature space within some desired range in order to reduce the output range and improve learning. Some of the widely used activation functions are as follows:

   - *Rectified Linear Unit* (ReLU): $f(x) = max(0, x)$
   - *Hyperbolic Tangent* (TanH): $f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$
   - Sigmoid: $f(x) = \frac{1}{1+e^{-x}}$
   - Leaky ReLU: $f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha.x, & \text{otherwise} \end{cases}$
     where α is a constant. Usually, $\alpha = 0.001$.
   - *Parametric Rectified Linear Unit* (PReLU): $f(x) = \begin{cases} x, & \text{if } x \geq 0 \\ \alpha.x, & \text{otherwise} \end{cases}$
     where α is a hyperparameter learned together with the model parameters.

3. **Pooling Layer:** The pooling layer reduces the spatial dimension of the feature representation to avoid overfitting during training. Across each channel of the input feature map, the pooling layer works exclusively and resizes it spatially, using a variety of pooling operations such as max pooling, average pooling, etc. Pooling layers are generally applied to the feature representation produced by the activation functions of the previous layer. For example, a pooling layer with $M \times M$ size filters with the stride of 2 downsample each slice of the channel in the input feature by two along spatial dimensions. If max-pooling has been used, then the $M \times M$ window

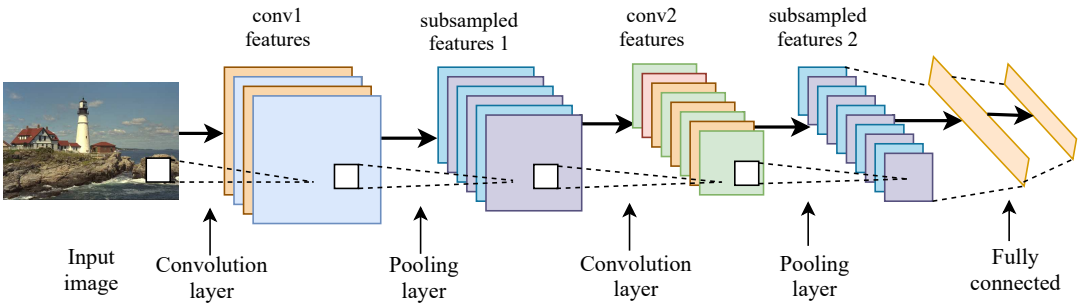


Figure 2.4: *An overview of the architecture of a generic deep CNN*

of the feature map will be represented by a single value which is maximum in that window. Whereas the average pooling considers the arithmetic mean of the window.

4. **Fully Connected Layers:** Similar to traditional neural networks, the neurons in a fully connected layer have full connections to all activations of the previous layer. Therefore, the activations of fully connected layers can be computed using a matrix multiplication followed by a bias offset.

An overview of the architecture of a traditional deep CNN has been shown in Figure 2.4. The first few convolutional layers of a deep CNN learn to extract the high-level features such as edges, lines, etc. Whereas the final convolutional layers extract the fine-grained features. These features are further refined using the fully connected layers to accomplish the desired task. In this dissertation, some of the existing popular deep CNNs - a) VGG-16, b) ResNet, and c) U-Net are utilized to design the efficient image and video restoration methods.

2.4.1 VGG-16

VGG-16 [5] is one of the earliest and most popular deep CNN developed by Simonyan and Zisserman for the task of *Imagenet Large Scale Visual Recognition Challenge* (ILSVRC) 2014 [99]. Being the runner-up of the ILSVRC 2014 challenge, the architecture of VGG-16 is shown in Figure 2.5.

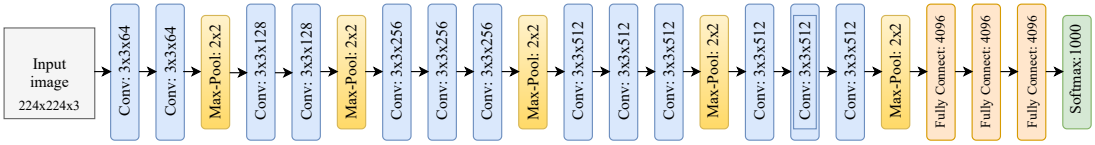


Figure 2.5: *An overview of the architecture of the VGG-16 [5] model.*

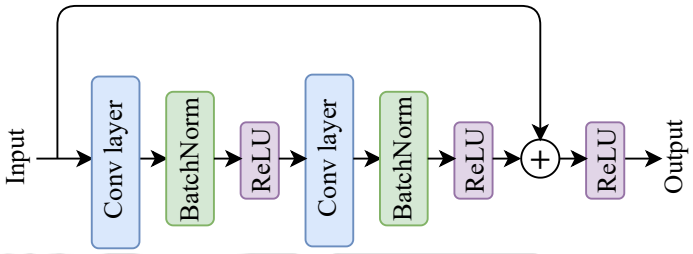


Figure 2.6: *An overview of the architecture of the Deep Residual Network block [6].*

It takes an RGB image of shape $224 \times 224 \times 3$ as input. Each convolution layer has a filter of size 3×3. All the layers in the network except the last fully-connected use ReLU as the activation function. The stride is fixed to 1 pixel, and padding is kept as 1 to retain the same spatial dimension of the input. Max pooling has been used after a set of convolutional layers with a 2×2 window and stride of 2. VGG-16 has outperformed the performance of AlexNet [19] on large-scale image recognition task [99] by achieving 7.3% Top-5 error.

2.4.2 ResNet

The depth of deep CNNs has been growing and given considerable importance in terms of the number of layers. For example, VGG-16 [5] was deeper compared to the AlexNet [19]. However, only increasing the depth of the model may not improve the performance always; instead, it may get degraded [6]. This degradation has been studied in detail by He *et al.* through *Deep Residual Network* (ResNet) [6]. The main novelty of ResNet is the incorporation of "*skip connection*," as shown in Figure 2.6. The proposed connections skip a set of layers, which does not increase the number of trainable parameters. It simply add-up

the output activations from the previous layer to the layer ahead. The authors demonstrated that it is comparatively easier to optimize the residual mapping than the original mapping. It further solved the vanishing gradient problem. ResNet was the winner of ILSVRC 2015 by showing the best performance compared to the state-of-the-art models for large-scale image classification tasks in terms of 3.57% Top-5 error.

2.4.3 U-Net

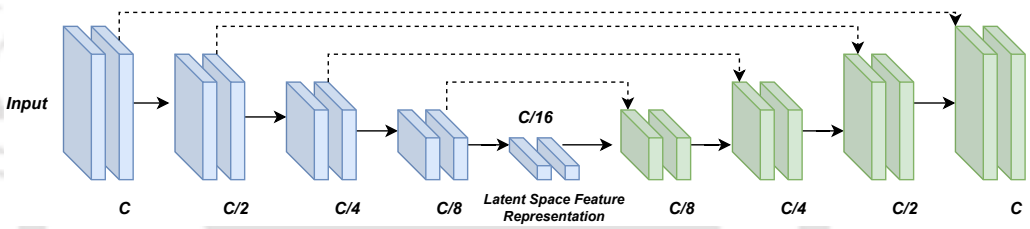


Figure 2.7: *An overview of the architecture of the U-Net [7] model.*

U-Net [7] was originally invented for segmentation of neuronal structures in electron microscopic stacks and won the *International Symposium on Biomedical Imaging* (ISBI) challenge in 2015. The authors demonstrated that the proposed model is end-to-end trainable and capable of outperforming then state-of-the-art methods. Later, a majority of the works adopted U-Net-like architectures for a variety of tasks [100–106]. The schematic network architecture is shown in Figure 2.7. It mainly comprises of a contracting path (left side) widely known as *encoder* and an expansive path (right side) also known as *decoder*. The encoder sub-module consists of series of convolutional layers where after each pair of layers, a max-pool operation has been used. Whereas, in the decoder sub-network, max-unpooling has been used for upsampling. The main idea behind downsampling the spatial dimension of the feature maps is to generate a latent space of feature representations that retains the most important aspects of the input image. The latent space representation further has been used in many other tasks for an effective computation.

2.4.4 Generative Adversarial Networks

Generative Adversarial Network (GAN) [50] is a deep learning-based framework proposed by Goodfellow *et al.* for designing generative models using adversarial training. In adversarial training, two sub-models, a generative (G) and a discriminative (D) are trained simultaneously. G learns to capture data distribution, and D predicts the probability that the given sample came from original training data or generated by G. To understand the formulation mathematically, let p_g denote the generator's distribution over data x, $p_z(z)$ denotes the input noise variables. $G(z;\theta_g)$ describes a mapping to data space, where G is a differentiable function with parameters θ_g. Let $D(x;\theta_d)$ be another differentiable function with parameters θ_d, which produces a single value. $D(x)$ denotes the probability that x has been drawn from original data rather than p_g. D is trained to maximize the probability of correctly classify both original training samples and generated samples by G. At the same time, G is trained to fool the discriminator D to minimize $log(1-D(G(z)))$. G and D play a two-player mini-max game using Eq. (2.10).

$$\min_G \max_D \mathcal{L}(D,G) = \mathbb{E}_{x\sim p_{data}(x)}\left[logD(x)\right] + \mathbb{E}_{z\sim p_z(z)}\left[log(1-D(G(z)))\right] \tag{2.10}$$

This formulation enables the generative model to learn original data distribution so that the discriminator fails to classify between the original and generated data.

2.4.5 Perceptual Loss

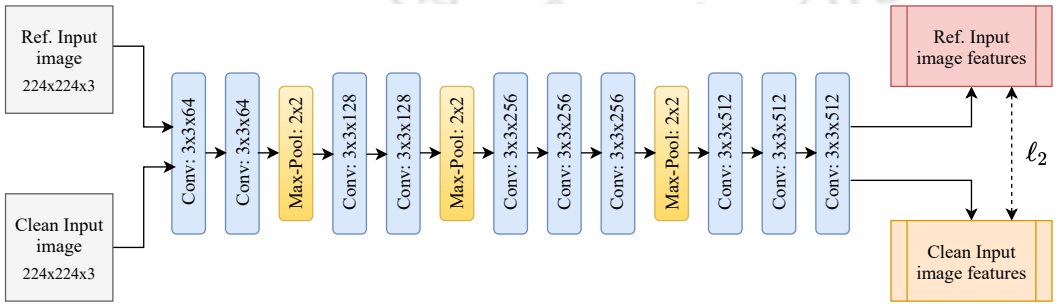


Figure 2.8: *An overview of the schematic diagram depicting the computation of the perceptual [8] loss.*

In general, the noise present in an image exhibits high-frequency nature. During image de-noising by using a traditional CNN, the use of ℓ_2 distance as a cost function may incur the loss of high-frequency details of the image along with the noise removal [107]. As a result, the de-noised images appear to be blurry and degraded. The perceptual loss function proposed by Johnson *et al.* [8] has been used in the majority of the image de-noising, and restoration problems [3, 108–113] in recent times to overcome this drawback by retaining the high-frequency details of an image. The perceptual cost function is defined as a difference between high-level features of predicted and target images extracted by using a pre-trained CNN. In general, the initial l layers of a pre-trained VGG-16 [5] model (V) have been used to extract the features. The perceptual loss function (L_P) between clean ground truth x and de-noised image y can be expressed as

$$L_P = \sum_{l} \sum_{c_i, w_i, h_i} \left\| V_l(x)^{c_i, w_i, h_i} - V_l(y)^{c_i, w_i, h_i} \right\|_2^2 \tag{2.11}$$

A qualitative demonstration has been given in Figure 2.8.

2.5 Image Quality Metrics

Processing a degraded image using restoration methods may result in improving the visual quality of the input image. One may evaluate the visual quality quantitatively in two ways- Subjective and Objective. Subjective schemes are based on human judgement and processes without specific reference parameters. Whereas the objective approaches are based on computations using explicit numerical criteria. The objective metrics can further be categorized into (a) *Full-reference metrics* that requires ground truth, and (b) *Reference-less metrics* that do not require ground truth. A few examples of objective approaches used in this work to evaluate the quality of restored images and videos are as follows:

2.5.1 Full-reference Metrics

These metrics are calculated between the original ground truth clean images and the corresponding generated de-noised images.

- **PSNR:** Given a clean image x and restored version of that image y with size $m \times n$. The *Peak Signal to Noise Ratio* (PSNR) between x and y is calculated as:

$$PSNR(x,y) = 10log_{10}\left(\frac{peak^2}{MSE(x,y)}\right) \tag{2.12}$$

where *peak* denotes the maximum possible intensity (for b-bit image, $peak = 2^b - 1$) and $MSE(x,y) = \frac{1}{mn}\sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - y_{ij})^2$

- **SSIM:** The *Structural Similarity Index* (SSIM) [15] is another metric for image quality assessment, which considers the structural information. The SSIM models the image distortion as combination of three modules - distortion in luminance (l), contrast distortion (c), and loss of correlation (s). The SSIM between image x and y is defined as:

$$SSIM(x,y) = l(x,y).c(x,y).s(x,y), \tag{2.13}$$

where

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}$$
$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}$$
$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\sigma_y + C_3}$$

$l(x,y)$ measures the similarity of mean luminance of two images. $l(x,y)$ is maximum $(= 1)$ when $\mu_x = \mu_y$. $c(x,y)$ compares the contrast of two images. It is maximum $(= 1)$ when $\sigma_x = \sigma_y$. $s(x,y)$ compares the structure using correlation coefficient between two images x and y. The SSIM$\in [0,1]$, where 0 implies no correlation between images, and 1 implies $x = y$. C_1, C_2, and C_3 are positive constants used to avoid zero denominator.

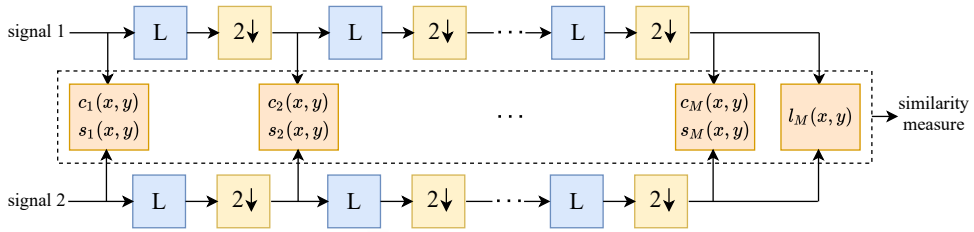


Figure 2.9: *MS-SSIM evaluation system. L: low pass filtering and 2 ↓: downsample by factor of 2.*

- **MS-SSIM:** The *Multi-scale Structural Similarity Index* (MS-SSIM) [114] takes the clean image and the restored image as input, and iteratively applies low-pass filtering and downsamples the image by a factor of 2. The original image is indexed at scale 1, and the maximum scale M. At i^th scale, the contrast comparison and structure comparison are denoted as $c_i(x,y)$ and $s_i(x,y)$, respectively. The luminance is compared only at scale M and is denoted by $l_M(x,y)$. The MS-SSIM is computed by combining these terms as:

$$SSIM(x,y) = [l_M(x,y)]^{\alpha_M} . \prod_{i=1}^{M} [c_i(x,y)]^{\beta_i} [s_i(x,y)]^{\gamma_j} \tag{2.14}$$

  The relative importance of different components is tuned by α_M, β_i, and γ_j. A graphical demonstration of its evaluation system is presented in Figure 2.9.

- **UQI:** The *Universal-Image-Quality Index* (UQI) [115] can be written as

$$Q = \frac{4\sigma_{xy}\bar{x}\bar{y}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]} \tag{2.15}$$

  where $x = x_j | j = 1, 2, ..., N$ $y = y_j | j = 1, 2, ..., N$ are original and test images, respectively. $\bar{x} = \frac{1}{N}\sum_{j=1}^{N} x_j$, $\bar{y} = \frac{1}{N}\sum_{j=1}^{N} y_j$, $\sigma_x^2 = \frac{1}{N-1}\sum_{j=1}^{N}(x_j - \bar{x})^2$, $\sigma_y^2 = \frac{1}{N-1}\sum_{j=1}^{N}(y_j - \bar{y})^2$, and $\sigma_{xy} = \frac{1}{N-1}\sum_{j=1}^{N}(x_j - \bar{x})(y_j - \bar{y})$. The range of Q belongs to $[1, 1]$. The best value of $Q = 1$ is achieved when $y_j = x_j$, $\forall_i$. The UQI can also be considered as the product of three different factors: decay in correlation, distortion in luminance, and distortion in contrast as

follows:

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} . \frac{2\bar{x}\bar{y}}{(\bar{x})^2 + (\bar{y})^2} . \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \tag{2.16}$$

- **VIF:** The *Visual Information Fidelity* (VIF) [116] measures the information fidelity for the input image in a statistical model of the *Human Visual System* (HVS) [116]. For this, the VIF utilizes two variables. The first computes the statistics between the initial and the final stage of the visual channel without distortion. Whereas the second considers the mutual information between the input of the distortion block and the output of the visual block. The VIF is estimated for a collection of $N \times M$ wavelet coefficients from each sub-band as follows.

$$VIF = \frac{\sum_{i \in subbands} I(C^{N,i}; F^{N,i} | s^{N,i})}{\sum_{i \in subbands} I(C^{N,i}; E^{N,i} | s^{N,i})} \tag{2.17}$$

  where $I(C^{N,i}; F^{N,i}|s^{N,i})$ and $I(C^{N,i}; E^{N,i}|s^{N,i})$ are the information that can ideally be computed by the brain from a specific wavelet sub-band in the reference and the test images, respectively.

- **LPIPS:** The *Learned Perceptual Image Patch Similarity* (LPIPS) [117] demonstrated that the deep neural network activations can be used as a perceptual similarity metric. For this, the authors utilized SqueezeNet [118], AlexNet [19] and VGG [5]. It can be used to measure the difference between two image patches, with the output value higher means more difference and lower means more similar.

- **FSIM:** The *The Feature Similarity Index* (FSIM) [119] measures the quality score based on the fact that the HVS understands an image mainly by processing its low-level features. With this motivation, FSIM considers the phase congruency (PC) as a significance of a local structure, and gradient magnitude (GM) to incorporate the contrast change. The input RGB image is initially converted into YCbCr color space to separate out the luminance

channel of the image. To formally address, let two images are x and y and their phase congruency can be denoted by PC_1 and PC_2, respectively. The PC and GM (G_1, G_2) maps extracted from two images x and y. FSIM can be defined and calculated based on PC_1, PC_2, G_1 and G_2. Firstly, the similarity between PC maps can be computed as

$$S_{PC} = \frac{2.PC_1.PC_2 + T_1}{PC_1^2 + PC_2^2 + T_1}, \tag{2.18}$$

where T_1 is a positive constant. Similarly, the same between the GM maps can be computed as

$$S_G = \frac{2.G_1.G_2 + T_2}{G_1^2 + G_2^2 + T_2}, \tag{2.19}$$

where T_2 is a positive constant.

Now, S_{PC} and S_G are combined together to compute the FSIM as

$$S_L = [S_{PC}]^{\alpha}.[S_G]^{\beta}, \tag{2.20}$$

where, α and β are used to adjust the relative importance of PC and GM features. Originally, in the paper, $\alpha = \beta = 1$.

- **CIEDE 2000:** The CIEDE 2000 [120] measure the difference between the color channels of the original clean and restored images. It considers the five corrections, namely: (a) a hue rotation term, to deal with the problematic blue region,(b) compensation for neutral colors (the primed values in the L*C*h differences), (c) compensation for lightness, (d) compensation for chroma, and (e) compensation for hue.

- **Haar PSI:** *Haar Wavelet-based Perceptual Similarity Index* (Haar PSI) [121] estimates the perceptual similarity between two images using features obtained by first performing Haar-wavelet decomposition. It later applies an additional non-linear mapping to the local similarities obtained from high-frequency Haar wavelet filter responses using logistic function owing to the explanation that it greatly models the thresholding in biological neurons.

The Haar PSI value of two similar images will be exactly one and the same of two completely different images will be close to zero.

- **GMSD:** Let d, r be the distorted and reference images, respectively. *Gradient Magnitude Similarity Deviation* (GMSD) [122] first calculates the horizontal and vertical directional gradients by convolving Prewitt filter along the two directions, denoted as G_x, G_y. Then the image gradient maps for r and d at pixel i can be calculated as

$$\begin{aligned} G_r(i) &= \sqrt{G_{x,r}(i)^2 + G_{y,r}(i)^2} \\ G_d(i) &= \sqrt{G_{x,d}(i)^2 + G_{y,d}(i)^2} \end{aligned} \tag{2.21}$$

It then computes the gradient magnitude similarity (GMS) as

$$GMS(i) = \frac{2.G_r(i).G_d(i) + c}{G_r(i)^2 + G_d(i)^2 + c}, \tag{2.22}$$

where c is a numerical stability constant. The GMSD map then can be computed as

$$GMSD = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(GMS(i) - GMS_\mu)^2}, \tag{2.23}$$

where GMS_μ is the mean of GMS map.

- **SpEED-QA:** *Spatial Efficient Entropic Differencing for Image and Video Quality* (SpEED-QA) relies on local spatial operations on given noisy and noise-free image frames and frame differences to estimate the perceptually relevant image/video quality features in an efficient manner.

2.5.2 Reference-less Metrics

This class of objective metrics do not require original ground truth clean images.

- **TV Error:** The *Total Variation* (TV) error denotes the total amount of noise present in an image. Given a restored image y, the TV error can be

written as

$$V(y) = \sum_{i,j} \sqrt{|y_{i+1,j} - y_{i,j}|^2 + |y_{i,j+1} - y_{i,j}|^2}. \tag{2.24}$$

The above defined version is popularly known as *isotropic* version of TV error. However, the one which is easier to minimize and widely used in practice is called as *anisotropic* version and can be described as

$$V_{aniso}(y) = \sum_{i,j} \sqrt{|y_{i+1,j} - y_{i,j}|^2} + \sqrt{|y_{i,j+1} - y_{i,j}|^2}. \tag{2.25}$$

- **NIQE:** The *Naturalness Image Quality Evaluator* (NIQE) [123] compares the restored image to a default model computed from statistics of the images of natural scenes. A smaller score indicates better perceptual quality. The quality of the de-noised image is expressed as the distance between the quality aware natural scene statistics feature model and the multi-variate Gaussian fit to the features extracted from the restored image and can be written as:

$$D(v_1, v_2, \Sigma_1, \Sigma_2) = \sqrt{(v_1 - v_2)^T \frac{(\Sigma_1 + \Sigma_2)^{-1}}{2} (v_1 - v_2)}, \tag{2.26}$$

where $v_1, v_2, \Sigma_1, \Sigma_2$ are the mean vectors and covariance matrices of the natural multivariate Gaussian model and the restored image's multivariate Gaussian model.

- **PIQE:** The *Perception based Image Quality Evaluator* (PIQE) [124] estimates the perception-based quality of the image using block-wise distortion analysis. It first computes the Mean Subtracted Contrast Normalized (MSCN) coefficient for each pixel in the image using method proposed in [125]. For each block, it then evaluates the distortion due to blocking artifacts and noise using the estimated MSCN coefficients. According to the analysis presented in [124], image quality based on the PIQE scores can be divided into five categories as mentioned in Table 2.1.

Quality scale	Score range
Excellent	0-20
Good	21-35
Fair	36-50
Poor	51-80
Bad	81-100

Table 2.1: *Image quality based on the PIQE scores.*

- **BRISQUE:** *Blind/Referenceless Image Spatial Quality Evaluator* (BRISQUE) [126] estimates the score by utilizing a support vector regression (SVR) model trained on an image database with corresponding differential mean opinion score (DMOS) values. The presented database comprises images with known distortion such as compression artifacts, blurring, and noise, and it has the pristine versions of the distorted images.

- **BLIINDS:** *BLind Image Integrity Notator using DCT-Statistics* (BLIINDS) [127] uses the natural scene statistics models of discrete cosine transform (DCT) coefficients to perform distortion-agnostic non-reference image quality assessment of the noisy/noise-free images.

2.6 Datasets

In this dissertation, following benchmarks have been used to conduct the extensive experiments against the best-published works:

- **Image De-raining:** For single image rain-streak removal, one of the popular synthetic and real-world benchmark datasets proposed by Zhang *et al.* [3] has been used. The dataset is publicly available at [1]. The training images are augmented into the disjoint patches of size 128×128, generating 192K images. The test set consists of 1201 images of size 512×512.

- **Image De-hazing:** For training of single image haze removal method, a

[1] https://github.com/hezhangsprinter/DID-MDN

synthetic benchmark provided by Zhang *et al.* [10] has been used, which consists of 4000 indoor images. In addition, 45 pairs outdoor images provided by Ancuti *et al.* [128] have also been used. The proposed model has been tested on synthetic dataset (SOTS) provided by Li *et al.* [129] which consists of 500 outdoor and indoor images in addition to the benchmark test-set[1] provided by Fattal *et al.* [17] and real-world hazy images.

- **Video de-raining:** For training of the proposed model for video de-raining task, a set of synthetic videos utilized in J4RNet [76] have been used. The incorporated training set consists of 349 rainy, clean video pairs, where each video comprises of 9 frames. A conjugated test sets from DualFlow [77], SPAC [14], and [4] have been utilized. Thus, a comprehensive comparison on over 11 test sets, namely **Test Set 1**, **Test Set Heavy**, **Test Set Light**, a_1, a_2, a_3, a_4, b_1, b_2, b_3, and b_4, has been conducted. The details of the each test set is given as follows In SPAC [14] test sets, group a consists

Test Set	**1**	**Heavy**	**Light**	a_1	a_2	a_3	a_4	b_1	b_2	b_3	b_4
No. of Videos	25	25	9	1	1	1	1	1	1	1	1
Avg. frames per video	31	31	96	168	116	125	298	256	250	219	250

of scenes shot from a panning and unstable camera, whereas the group b from a fast-moving camera (with speed range between 20 to 30km/h). The qualitative comparison on the real-world test set provided by the SPAC [14] has also been given.

2.7 Summary

In this chapter, background concepts on the convolutional neural network (CNN) have been presented. These concepts are used to design learning-based image and video restoration frameworks, which are presented in the later chapters of

[1] http://www.cs.huji.ac.il/~raananf/projects/dehaze_cl/results/

this thesis. In addition, evaluation metrics used to evaluate the methods and the datasets are presented in this chapter.

With this background, this thesis's first contribution will be discussed in the next chapter, where the task of single image de-raining will be addressed in the spatial domain.

Chapter 3

Exploiting Efficient Spatial Upscaling for Single Image De-Raining

It has been observed in Section 1.3 that low-level vision tasks, *e.g.* image, and video de-noising, can be beneficial for high-level tasks, especially in bad weather conditions. There can be several possible reasons for bad weather conditions, including heavy rainfall, haze or fog, and snowfall, *etc.* The efficiency of the outdoor vision tasks, such as autonomous vehicle navigation system, depend on the visual quality of the image and videos, which can be severely degraded by the rainfall or haze in bad weather conditions. Therefore, it becomes necessary to propose efficient and real-time friendly learning-based methods for improving the visual quality of degraded images and videos in bad weather scenarios. In this dissertation, firstly, the task of rain-streak removal in the images has been studied.

Rain-streaks exhibit pseudo-periodic additive nature in an image (Eq. (1.1)). It has been observed from the limitations of existing works mentioned in Chapter 1 that a majority of existing methods suffer from the problem of over-coloring and white-dots artifacts in the de-rained images (see Figure 1.7). In other words, most best-published works fail to reconstruct the original perceptual quality of the clean image. It may be due to the improper usage of the deconvolution layers during the reconstruction of the de-rained images in network engineering. The

deconvolution layer with stride > 1 may induce blocky visual artifacts. Furthermore, the processing of rainy images using deep CNNs in highly correlated color space, such as RGB, may not be much beneficial. It may be one of the reasons why the de-rained images of existing works that operate in RGB color-space suffer from the over-coloring problem. Also, the high-level features from deep CNN inherently capture the white round particles, so the perceptual loss [8] may enhance the white-dot artifacts in the de-rained images.

These limitations motivate us to enhance the perceptual quality of the rain-free image and, therefore the contributions made in this chapter can be summarised as follows:

- U-Net [7] based architecture has been very successful in the case of image denoising, and reconstruction tasks [9], due to its ability to preserve important features for the reconstruction of images and discard the irrelevant and noisy components. Therefore, an image de-raining model, namely HRID-GAN, has been proposed based on the U-Net framework.
- It has also been proposed to use the efficient sub-pixel convolution [130] instead of conventional deconvolution layers to avoid the blocky visual artifacts in the generated de-rained images.
- To further improve the quality of the de-rained image generated by the encoder-decoder network, a deep residual network [6] has been used.
- The cGAN based adversarial training has been incorporated in order to achieve better de-rained images.
- An ablation study has been given at the end of this chapter to demonstrate the effects of certain modules in the network with detailed comparisons.

3.1 Sub-pixel Convolution

During the downsampling of a noisy image, the most prominent features remain in the compressed form, whereas the high-frequency details such as noise are

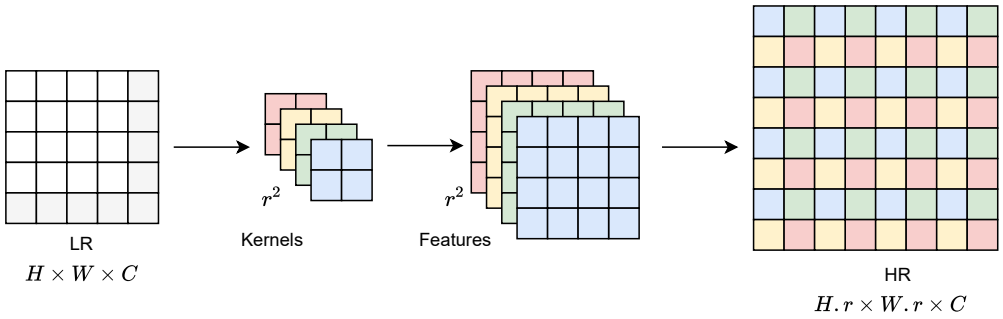


Figure 3.1: *Schematic diagram of sub-pixel upscaling.*

discarded. To reconstruct the de-noised image, the most common method used is transposed convolution, which is also popularly known as deconvolution [9]. Even though bicubic interpolation is a special case of deconvolution [130], it has been observed that transpose convolution (*stride* > 1) often induces blocky visual artifacts in the generated images [131].

The other way to upscale an image is to perform convolution operation with a fractional stride of $\frac{1}{r}$ and then perform the pixel-shuffle operation $\mathcal{PS}$ based on the following equation [130].

$$\mathcal{PS}(\mathbf{T})_{x,y,c} = \mathbf{T}_{\lfloor x/r \rfloor, \lfloor y/r \rfloor, c.r.mod(y,r)+c.mod(x,r)} \tag{3.1}$$

It can be explained in detail using a schematic diagram presented in Figure 3.1, where the input *Low Resolution* (LR) features are upscaled to a *High Resolution* (HR) feature map without utilizing stride > 1. The LR features of shape $H \times W \times C$ are upscaled to HR features of shape $H.r \times W.r \times C$, where r denotes the upscaling factor. For this, first, r^2 number of different convolution filters are used to generate the r^2 feature maps. Later, these generated r^2 feature maps are periodically shuffled using $\mathcal{PS}$ operation to get the desired HR features. In this way, the drawback associated with deconvolution can be avoided when stride > 1. In this work, instead of deconvolution, we have utilized the sub-pixel upscaling to generate artifacts-free de-rained images.

3.2 Proposed Approach

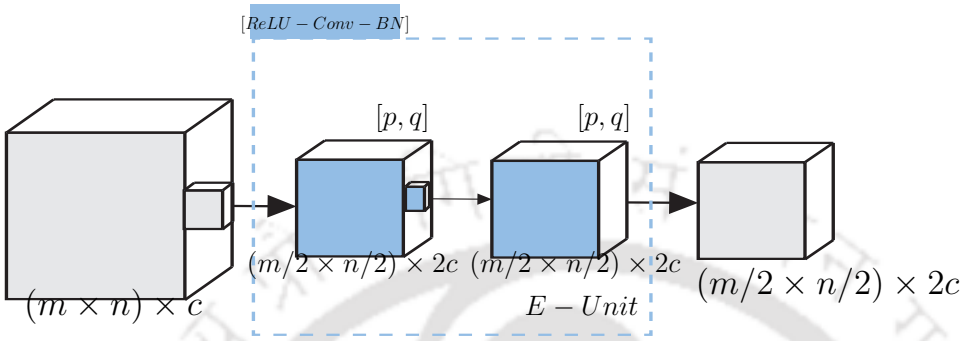


Figure 3.2: *An overview of single encoder unit (E-Unit) which cinsists of two convolution layers. An E-Unit outputs a tensor of spatial dimension downsampled (/2) with upsampled (×2) channel dimensions compared to the input given.*

3.2.1 Baseline Generator Model

The proposed baseline generator model consists of an encoder-decoder network where the decoder network comprises of conventional transpose convolution layers for up-sampling, followed by a deep residual network of 10 layers. The encoder consists of four encoding units (**E-Units**) as shown in Figure 3.2, whereas the decoder consists of four conventional deconvolution[1] layers with symmetric skip connection to avoid the loss of image details when the network goes deeper [132] and helps deconvolution to recover a better clean image [133]. The encoder network can be summarised as follows:

- **E-Unit A** comprises of 2 convolution layers with kernel size 5×5, number of kernels 64 each layer, spatial stride of 1×1 with activation function as ReLU at each layer. *Batch Normalization* (BN) [134] is applied only in second convolution layer.
- **E-Unit B** comprises of 2 convolution layers with kernel size 4×4, number of filters 128 each layer, spatial stride of 2×2 in the first layer and 1×1 in second.

[1]Transpose convolution is also referred as Deconvolution

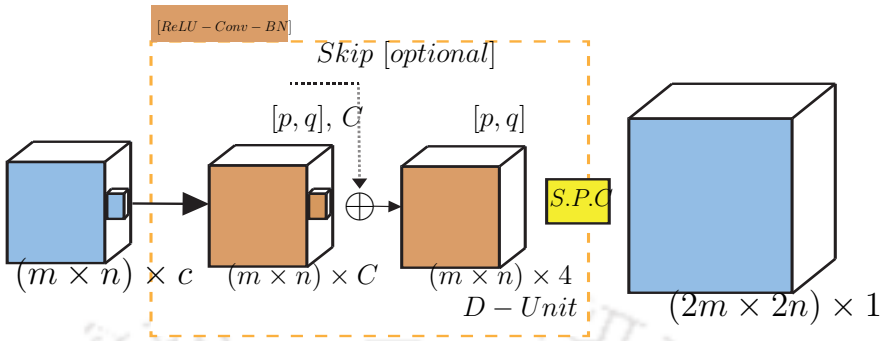


Figure 3.3: *An overview of single decoder unit (D-Unit) which cinsists of two convolution layers followed by an efficient sub-pixel re-arrangement(S.P.C) block. A D-Unit outputs a tensor of spatial dimension upsampled (×2) compared to the input given.*

- **E-Unit C** comprises of 2 convolution layers with kernel size 3×3, number of filters 256 each layer, spatial stride of 2×2 in the first layer and 1×1 in second.
- **E-Unit D** comprises of 2 convolution layers with kernel size 2×2, number of filters 512 each layer, spatial stride of 2×2 in the first layer and 1×1 in second.

Each convolution layer of **E-Units (B:D)** consists of ReLU as activation function followed by BN for faster convergence [134].

3.2.2 Generator with Efficient Sub-Pixel Convolution

The baseline generator model consists of conventional deconvolution operation which is quite computationally expensive. Several upsampling methods are available that have shown remarkable success in the recent times [135] [136]. Lu *et al.* [135] proposed Non-Convex JBU (NCJBU) by extending the well-known Joint Bilateral Upsampling (JBU) [137] with a novel non-convex optimization framework for guided depth-map upsampling. In the proposed model, the conventional transpose convolution operations are replaced with the efficient sub-pixel convolution blocks (**D-Units**) for upsampling, as shown in Figure 3.3. The overall architecture of the generator model remains the same as the baseline generator

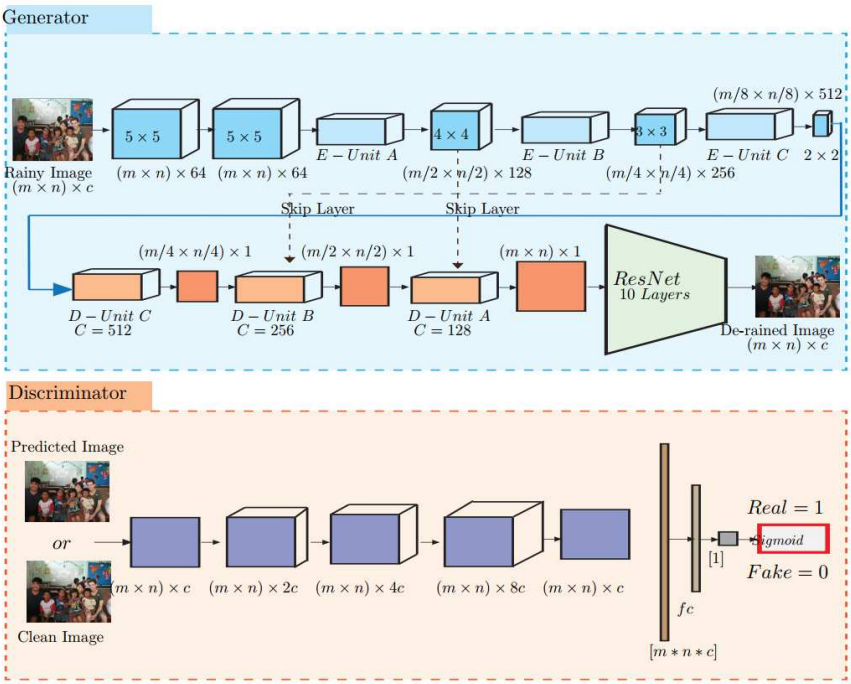


Figure 3.4: *An overview of the architecture of the proposed framework for rain streak removal from the single image.*

as discussed in the previous Subsection 3.2.1.

3.2.3 Discriminator

Following Zhang *et al.* [9], we have employed the adversarial training for single image de-raining problem. While the objective of the proposed generator model is to estimate the de-rained image from the rainy image, the proposed discriminator model is trained to differentiate whether the estimated de-rained image is real or fake. In other words, the feedback from the proposed discriminator is used to train the proposed generator more efficiently. An overview of the architecture of the proposed discriminator model, which consists of 5 convolution layers, is shown in Figure 3.4. Each layer comprises of $\mathbf{k/2}, \mathbf{k}, \mathbf{k^2}, \mathbf{k^3}$ and $\mathbf{k/2}$ convolution filters respectively where $\mathbf{k} = 2$. ReLU activation function has been used to induce the non-linearity in the model. A fully-connected neural network layer with 128

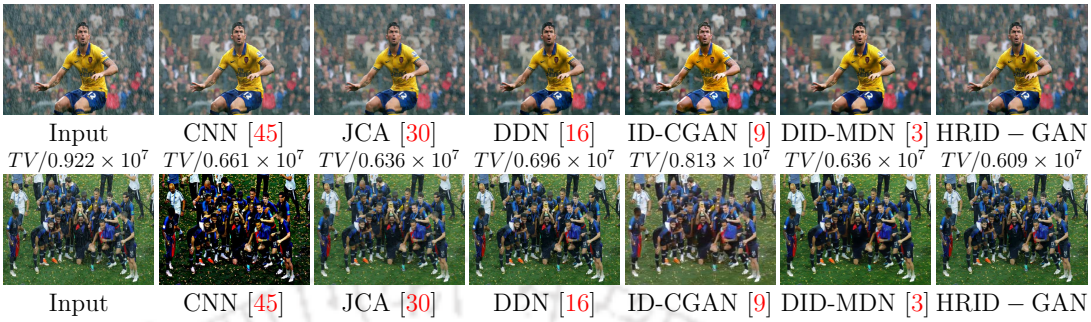


Figure 3.5: *Sample results on real-world rainy images in terms of Total Variation (TV).*

Methods	SSIM	PSNR
Input	0.7781	21.15
CNN [45]	0.8422	22.07
DDN [16]	0.8978	27.33
JCA [30]	0.8374	23.63
ID-CGAN [9]	0.8325	22.85
DID-MDN [3]	0.9087	27.95
SPANet [138]	**0.9342**	30.05
Baseline	0.9269	30.07
Generator(G)	0.9297	<u>30.62</u>
Proposed	<u>0.9308</u>	**30.78**

Table 3.1: *Quantitative comparison with existing methods on test dataset in terms of SSIM and PSNR. Best and second best results are highlighted in bold and underlined fonts, respectively.*

neurons, followed by a Sigmoid layer has been used to predict the logits.

3.2.4 Cost Function

With the evolution of deep learning-based single image restoration methods, several regression, image reconstruction and object detection cost functions have been introduced, such as [139] [140]. To formally define the cost fuction used for the optimization of the proposed model, let ψ_G be the proposed non-linear generator model which outputs the de-rained image when a rainy image $x \in [0,1]^{h \times w \times c}$ is given as input with w, h, c as width, height and channels of rainy image respectively. To retain the structural information, the *Mean Squared Error* (MSE) is

Methods	SSIM	PSNR	VIF	MS-SSIM	TV-Error	UQI	MSE
Input	0.7781	21.15	0.3734	0.7334	1.55	0.8636	0.766
CNN [45]	0.8422	22.07	0.4082	0.8384	1.25	0.8650	0.708
DDN [16]	0.8978	27.33	0.4246	0.8650	1.14	0.9526	0.124
JCA [30]	0.8374	23.63	0.3867	0.8145	1.05	0.8865	0.520
ID-CGAN [9]	0.8325	22.85	**0.5177**	0.9007	1.19	0.8922	0.513
DID-MDN [3]	0.9087	27.95	0.4552	0.8904	1.13	0.9497	0.124
Baseline	0.9269	30.07	0.4741	0.9058	**0.94**	0.9677	0.080
Generator(G)	0.9297	30.62	0.4833	0.9075	1.01	0.9674	0.072
Proposed	**0.9308**	**30.78**	0.4851	**0.9090**	1.01	**0.9682**	**0.070**

† TV-E is $\times 10^7$. ‡ MSE is $\times 10^{-3}$

Table 3.2: *Quantitative comparison with existing methods on test dataset. Best and second best results are highlighted in Bold and Underlined fonts, respectively.*

mostly used error function in denoising algorithms and can be defined as

$$\mathcal{L}_{\text{MSE}} = \frac{1}{w.h.c} \sum_{i=1,j=1,k=1}^{w,h,c} ||\psi_G(x)^{i,j,k} - y^{i,j,k}||_2^2 \tag{3.2}$$

The adversarial loss to train the proposed generator model is defined as

$$\mathcal{L}_{Adv} = -\frac{1}{N} \sum_{i=1}^{N} \log\, D(x_i) \tag{3.3}$$

where N is the number of generated de-rained images from generator. Therefore, the total generator loss is the linear combination of mean-squared error and entropy losses and can be written as

$$\mathcal{L}_G = \lambda_M.\mathcal{L}_{mse} + \lambda_A.\mathcal{L}_{Adv} \tag{3.4}$$

where λ_M and $\lambda_A$are pre-defined weights for the cost functions defined above. The objective of the proposed method is to generate the de-rained image given a rainy image as an input. The proposed generator tries to generate the de-rained image such that it is difficult for the discriminator to decide whether generated de-rained image is real clean image or fake. The proposed network tries to minimize the generator cost function $\mathcal{L}_G$.

3.3 Experiments and Results

The synthetic datasets of rainy and clean images given by the authors of [3] have been used for training and testing. Zhang *et al.*. [3] have included the medium

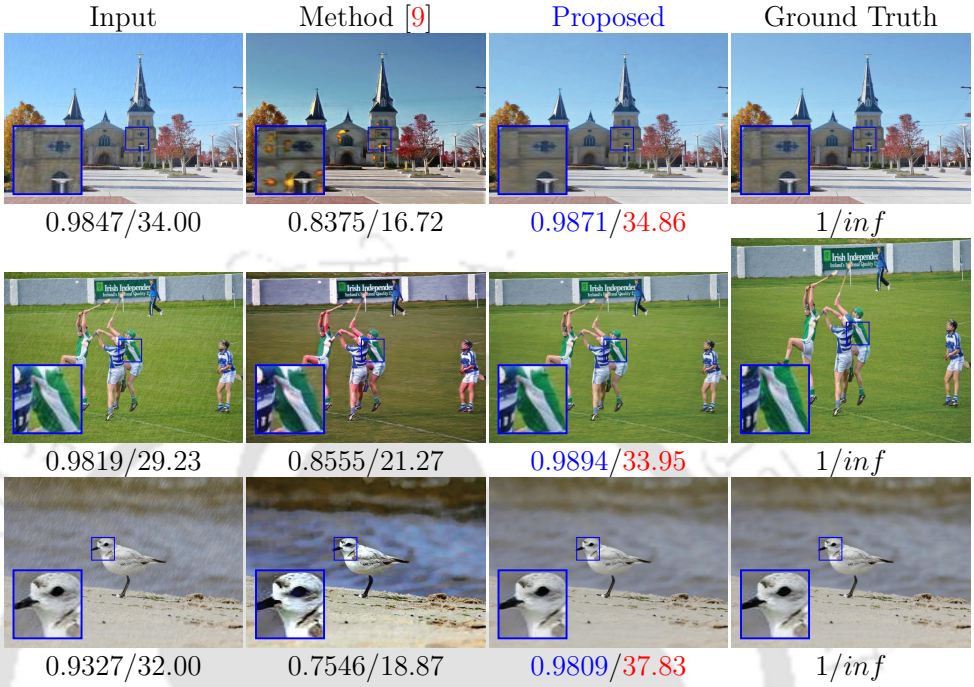


Figure 3.6: *Qualitative comparison with ID-CGAN [9] on synthesized test images in terms of* SSIM/PSNR.

rain density images in addition to images with light and heavy rain densities. We have augmented the rainy and clean images from our selected training dataset into the disjoint patches of size 128×128, thus creating a total of 1,92,000 patches. Our selected test dataset consists of 1201 rainy images with ground truth. The proposed method, along with baseline configurations, have also been tested on real-world rainy images, as shown in Figure 3.5. However, the ground truth of such images does not exist in order to compare with. To give a quantitative comparison of such images, we have used the following evaluation metrics.

3.3.1 Quality Measures

The proposed scheme along with the existing state-of-the-art methods have been evaluated on the following image quality evaluation metrics: UQI [115], SSIM [15] for measuring the similarity between de-rained and ground truth images, PSNR,

Figure 3.7: *Sample results on real-world rainy images*

VIF [116], MS-SSIM [114], MSE and TV to calculate amount of noise present in the image after de-noising. The description of incorporated evaluation metrics can be found in Section 2.5. *Color images are given as input to measure all quality metrics for every compared paper to be fair instead of luminance only as done in* [9] [30].

3.3.2 Model Parameters

In this sub-section, the dataset and parameters used during the training and testing of the proposed framework are discussed briefly. The publicly available synthetic dataset [1] [3] is used for training and testing where training set consists of 12000 images of size 512×512. For augmentation, we have cropped the training images into the disjoint patches of shape 128×128, resulting in a total of $192,000$ patches in the training set. For evaluation, the synthetic test set comprises 1.2K images of shape 512×512. To test the generality of the proposed scheme, we have also evaluated the proposed model on the real-world rainy images that do not have the ground truth clean images. The proposed model is built upon Tensorflow [141] framework and is trained on Nvidia-GTX 1080 GPU for 50 epochs. The learning rate (lr) is initially set to 0.01 and reduced by $\times 0.1$ after every $15 - 20$ epochs. The batch size of 20 and Adam [142] optimization algorithm have been used when training. The weights are set as follows: $\lambda_{\mathrm{E}} = 1$ and $\lambda_{\mathrm{adv}} = 0.01$.

[1] https://github.com/hezhangsprinter/DIDMDN

3.3.3 Comparison configurations

The proposed method has been compared with the following baseline configurations :

1. **Baseline** : We propose a baseline model described in Section 3.2.1.
2. **Generator** : The proposed generator with efficient sub-pixel convolution model as described in Section 3.2.2 with λ_{adv} set to zero.

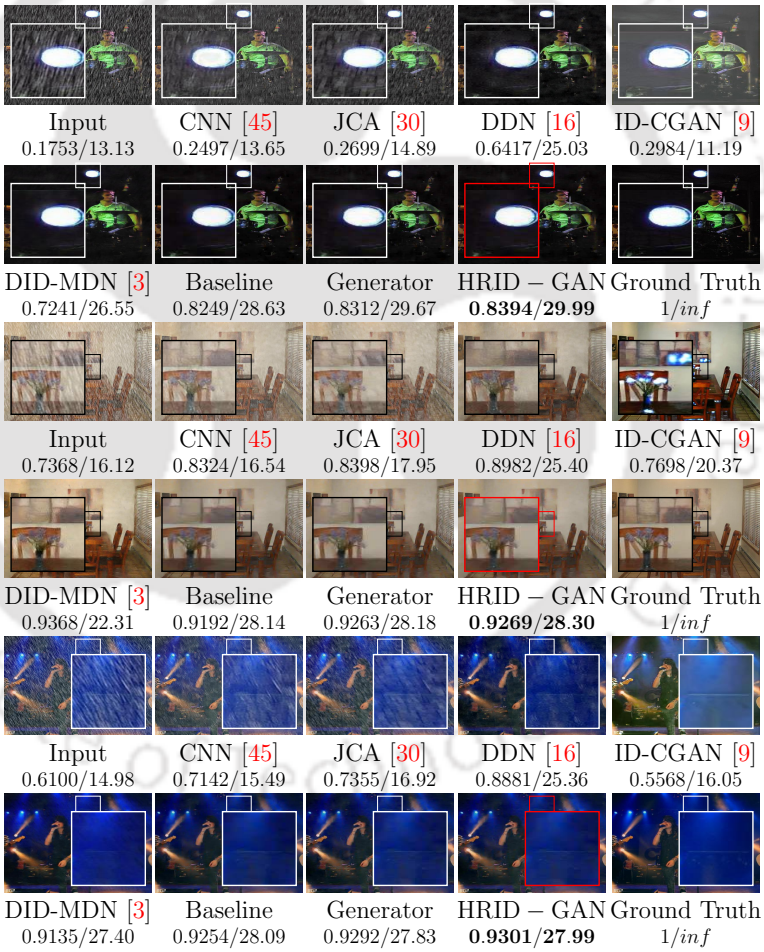


Figure 3.8: *Sample results on synthetic rainy images compared with the exising schemes in terms of SSIM/PSNR.*

The proposed method is compared with the following existing methods :

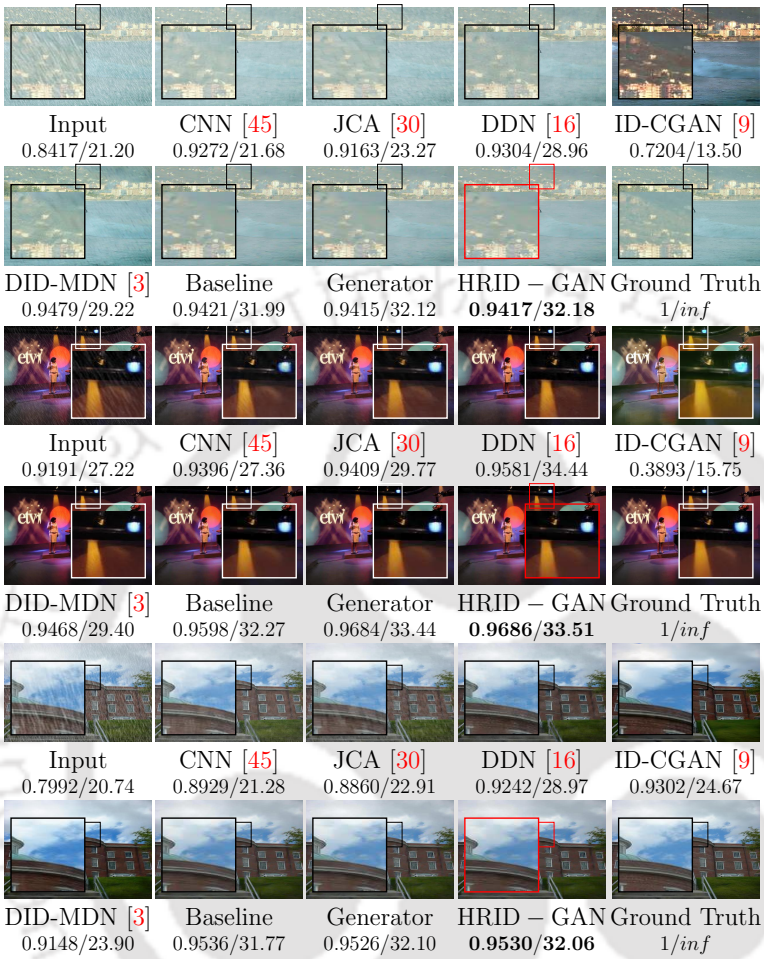


Figure 3.9: *Sample results on synthetic rainy images compared with the exising schemes in terms of SSIM/PSNR.*

1. **DDN** : A deep residual networks based architecture that requires prior image processing domain knowledge [16].

2. **DID-MDN** : A rain streak density-aware method based on densely connected CNN and a classifier [3] is the existing state-of-the-art method for single image de-raining problem.

3. **ID-CGAN** : A conditional GAN based framework which has an underlying model of encoder-decoder network [7] with perceptual loss function for training.

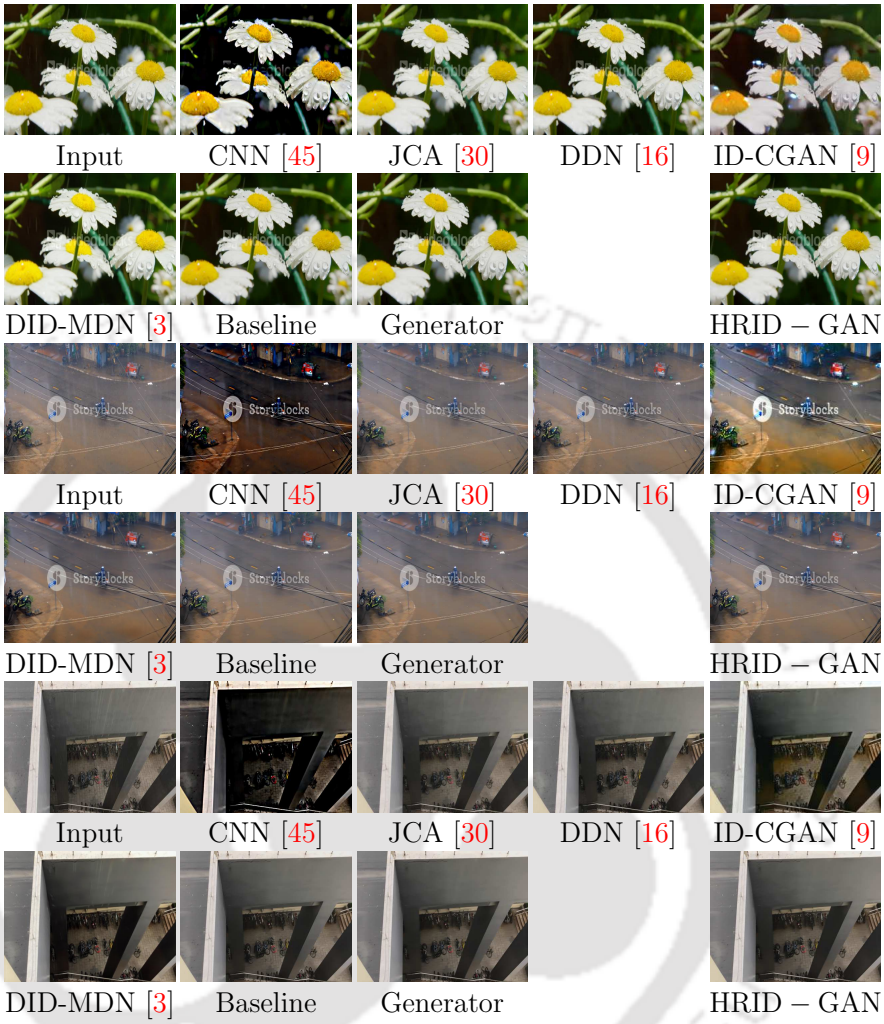


Figure 3.10: *Sample results on real-world rainy images*

4. **JCA** : A layer separation method based on convolution analysis jointly with synthesis sparse representation [30] used for rain streak removal from single images.

5. **CNN** : Clearing the skies : A deep network architecture for single image rain removal [45] is based on the ResNet similar to [16].

3.3.4 Quantitative Results

Quantitative comparison results with state-of-the-art methods [3, 9, 16, 30, 45] and baseline configurations are given in Tables 3.1 3.2. The proposed baseline generator model with conventional transpose convolution for the up-scaling operation has shown a significant improvement over recent state-of-the-art methods. The generator model (G), where transpose convolution operations have been replaced with efficient sub-pixel convolution operations, further improves the baseline model. The generator model with efficient sub-pixel convolutions (G), as shown in Table 3.2 is only trained on the mean-squared error as a loss function. The inclusion of discriminator and adversarial loss, denoted as (G+D), and mean-squared error on the proposed model (G) has further improved over recent methods and baseline configurations. The visual comparison of synthetic as well as real-world rainy images are given in the subsequent sub-sections.

3.3.5 Qualitative Results

It can be observed from Figures. 4.13, 3.5, 4.14 and 3.9 that the visual quality of the de-rained images estimated by using the proposed model is far better than the existing state-of-the-art methods. It is observed that the de-rained images generated by the existing methods [16], [30] and [45] still contains the rain-streaks. The methods [3] and [9] estimates the rain-streak free images in RGB domain. It can be seen that such methods due to high correlation among color channels suffer from over-coloring (color saturation) and white round artifacts (maybe due to the use of perceptual loss in [9]) respectively. However, the de-rained images generated by the proposed scheme do not suffer from these artifacts and degradations. The visual quality of the de-rained images may have been improved by incorporating the adversarial training in the proposed model, unlike in [45] and [16]. Following [9], learning the difference between real and fake de-rained images via adversarial training has been beneficial for single image rain-streak removal. However, the proposed model does not include the perceptual

loss function [8] due to the possibility of white round artifacts. The proposed model has made a relaxation on highly correlated color spaces, i.e., unlike [3] and [9], instead of estimating the de-rained images in colors, such as RGB or YCbCr, it estimates only the Y channel of the de-rained images. The relaxation of chrominance components Cb, Cr has been beneficial, and an improvement of $\sim 2\%$ in SSIM and $\sim 10\%$ in PSNR has been recorded over existing scheme [3]. The proposed method has also been evaluated on real-world rainy images as shown in Figures 3.5, 3.7, 3.10 and based on the TV error, it can be observed that the visual quality of the de-rained images generated by the proposed model is better than the existing methods.

3.4 Discussion

In this work, a conditional GAN based image de-raining scheme has been proposed, and it has been experimentally justified that the proposed scheme outperformed the state-of-the-art schemes. Although the proposed work has employed cGAN architecture, it is significantly different from cGAN based scheme [9], which has been proposed in the recent literature. The major differences are pointed out as follows:

1. The model used in [9] is trained in RGB color space, whereas the proposed model in this work has utilized the decorrelated YCbCr color space.
2. The model used in [9] has incorporated the perceptual loss [8] unlike the proposed model in this chapter.
3. The proposed model in this work has incorporated a 10 layer ResNet [6] unlike in [9].

However, the improvement reported by the proposed baseline model in this work compared to [9] is substantially huge in terms of both SSIM and PSNR, as shown in Table 3.2. This is mainly due to the fact that the inclusion of perceptual loss in [9] have added the white round artifacts in the generated de-rained images

when operating in RGB color space (as reported in [9]) whereas this work has retained the color information in terms of chrominance channels and has not utilized the benefits of perceptual loss on generated de-rained (grayscale) images. In addition to this, estimating the RGB values of the pixels in the predicted de-rained images as done in [3, 9] may be difficult for a network when compared to only predicting the grayscale values of the de-rained images. The proposed model has shown substantial improvement over the work reported in [16] which has utilized the ResNet [6] architecture and estimates the negative residual of the rain-streaks using the result of high-pass filter over the given input as a prior to the network. However, in this work, no such prior knowledge is required. Moreover, in addition to the residual network, this work also incorporates the encoder-decoder framework along with cGAN, which makes it more effective over [16].

While other methods have been successful in removing the rain-streaks up to some extent, most of the existing methods failed to consider the spatial resolution of the image. It is observed that after removing the rain-streaks, the obtained results suffer from the problems of over-smoothness and blurriness. To avoid these issues, unlike existing methods, this work has incorporated a sub-pixel convolution [130] to enhance the spatial resolution of the image. The use of efficient sub-pixel convolution instead of traditional deconvolution for up-sampling of the features in the proposed generator model (G) compared to the proposed baseline model initially has found to be beneficial, and an improvement of $\approx$ 0.5 dB in PSNR has been observed as shown in Table 3.2. This can be because efficient sub-pixel convolution might have improved the spatial resolution of the de-rained images by the proposed architecture, which in turn resulted in better PSNR values and a slight improvement in SSIM and other evaluation metrics. The adversarial loss, in addition to the MSE on the proposed generator, has been useful in further improving the results.

3.5 Summary

In this chapter, a conditional GAN based framework is proposed for a single image rain-streak removal task. The proposed scheme employed the computational and visual efficiency of efficient sub-pixel convolution over conventional transpose convolution for up-scaling and produced better results than existing state-of-the-art methods. It is shown how sub-pixel convolution for image de-noising tasks can replace conventional deconvolution, and noticeable improvement can be achieved if adversarial training is used in addition to traditional loss functions. The spatial resolution of the de-rained image has been achieved by the efficient sub-pixel convolution for upscaling as generated results have better resolution than the results obtained by the existing methods. Unlike existing state-of-the-art methods, the de-rained images generated by the proposed scheme do not contain white rounds artifacts and blurriness, which ensures the applicability of the proposed method on single image de-raining tasks.

This contributory chapter explored the spatial domain aspects of the rain-streaks in an image using a learning-based scheme. However, it has been mentioned that rain-streaks exhibit pseudo-periodic additive nature in an image, which may have some extra benefits in the transformed domain. In the next chapter, the transformed domain coefficients of the rain-streaks have been explored using deep CNNs.

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

Chapter 4

Exploiting Transformed Domain Features for Single Image De-Raining

The last contributory chapter addressed the visual artifacts in the de-rained images generated by using the existing best-published works. However, it has also been shown in chapter 1 that none of the existing methods utilized the transformed domain coefficients of the rain-streaks using deep CNN. In this chapter, we address the single image de-raining in the transformed domain using deep learning-based frameworks. There are two works that have been presented in this chapter, namely, (a) *exploiting rain-streaks in uncorrelated transformed domain*, and (b) *utilizing rain-streaks in correlated transformed domain.*

For investigating the behavior in the uncorrelated transformed domain, the DFT domain has been explored. Whereas, in the case of the correlated transformed domain, the DWT domain has been exploited. The main goal of this chapter is two-fold: (a) *exploit* the capability of deep CNN when presented with the purely uncorrelated input, and (b) *leverage* the added advantages of correlated transformed domain cues in addition to the spatial domain input to deep CNNs. In what follows are the respective studies in the uncorrelated and correlated transformed domain.

4.1 Image De-Raining in Uncorrelated Transformed Domain

Following the analysis presented in Section 2.1, the magnitude spectrum of the rainy image has been used to prove that rain streak information is preserved in the transformed domain and real, imaginary coefficients can be given as input to the deep network. Note that the DFT of an image consists of redundant frequencies because (a) DFT is Periodic, *i.e.* $\mathbf{F}[u, v] = \mathbf{F}[u + Nk, v + Ml]$ for all $k, l \in \mathbb{Z}$, (b) DFT holds conjugate symmetry, *i.e.* $\mathbf{F}[u, v] = \mathbf{F}^{\star}[-u+pN, -v+qM]$ for any integer p,q, for an image of size $M \times N$ [143]. These redundancies have not been removed before processing because their presence may not incur any misinterpretation.

4.1.1 Rain Streaks in DFT

When an image is transformed from the spatial to the DFT domain, the transformed signal is difficult to understand. However, the pseudo-periodic rain streaks among the pair of rainy and clean images can be visualized by using some transformation on Fourier data. The magnitude spectrums of a rainy and clean image have been calculated using Eq. 2.4 as shown in Figure 4.1. Both the images are first converted to grayscale, and their transformed DFT coefficients have been obtained using Eq. 2.3. For a better visual representation, magnitude spectrums are scaled down by a logarithmic transformation as

$$\mathbf{M}_{\mathbf{I}_*}^{\mathsf{scaled}} = \eta \times \log_e(\mathbf{M}_{\mathbf{I}_*}^{\mathsf{original}}) \tag{4.1}$$

where $\mathbf{M}_{\mathbf{I}_*}^{\mathsf{original}}$ is unscaled, $\mathbf{M}_{\mathbf{I}_*}^{\mathsf{scaled}}$ is the scaled magnitude spectrums respectively and η is a scaling parameter which is set as 12 in our experimentation. It is observed in Figure 4.1 that the transformed domain rain information can be more distinguishable if **DC** of the image signal is shifted to the center. Therefore, a **DC** shift to the center has been done by swapping all quadrants row and column-wise before calculating the magnitude spectrums. Let the transformed space of

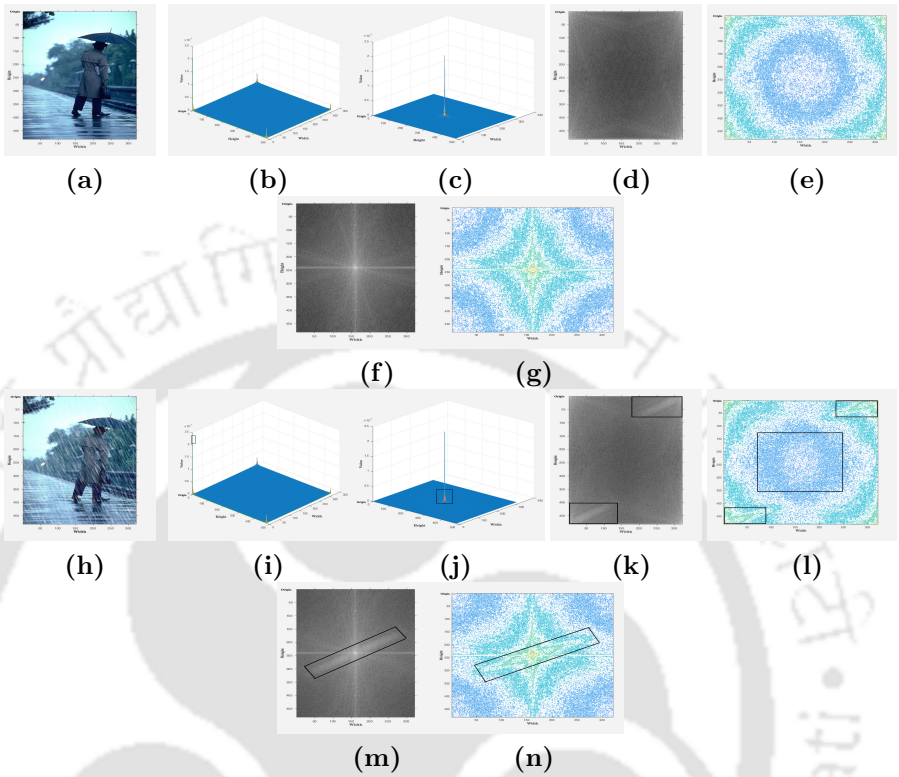


Figure 4.1: *A flow of visualizations. (a) Clean image, (b) Unscaled magnitude image of (a) with unshifted **DC** component, (c) Unscaled magnitude image of (a) with **DC** component shifted to center, (d) Scaled magnitude image of (a) with unshifted **DC** component, (e) Contour plot of (d), (f) Scaled magnitude image of (a) with **DC** component shifted to center, (g) Contour plot of (f). Similar series of figures goes for rainy image (h) to (n) in the bottom row.*

scaled magnitude image with shifted **DC** component be S^*. In Figure 4.1, the difference(here rain streaks) between rainy and clean images are noticeable in S^* space as indicated in black colored boxes. The difference between the unscaled magnitude spectrums of rainy, clean images with unshifted **DC** components is not visible to the human eye as shown in Figures 4.1i, 4.1b respectively. When we shift the **DC** component to the center as shown in Figures 4.1j, 4.1c, the difference is slightly better visible. When we scale down the magnitude spectrums using Eq. 4.1 and do not shift the **DC** components, as shown in Figures 4.1k, 4.1d, the difference is much more visible compared to previous images. We then shift

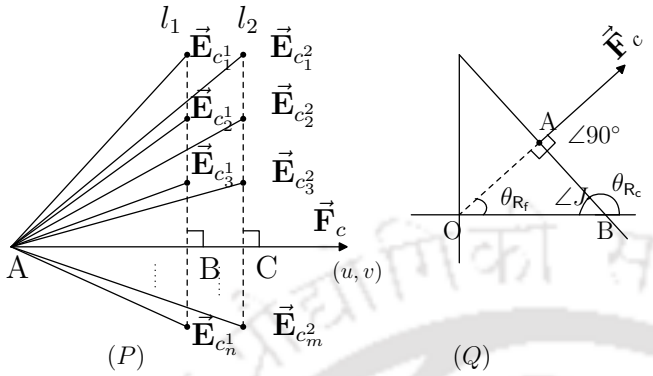


Figure 4.2: *Sinusoids depicting the rain-streaks in spatial and transformed domain.*

their **DC** components to center and scale down the magnitude images for better understanding and visualization as shown in Figures 4.1m, 4.1f. It is now easy to differentiate between a rainy and clean image in the Fourier domain, which are further transformed to S* space, to show the preservation of rain streaks. It can be observed from Figure 4.1 that there exist several high-intensity spots in the magnitude spectrum of the rainy image in S* space, bounded in the black colored box which is absent in the magnitude spectrum of the clean image. Each high-intensity spot in S* space denotes a two-dimensional sinusoidal wave which corresponds to one periodic noise based on the assumption that pseudo-periodic noise in a rainy image consists of some combination of multiple periodic noises. The high-intensity spot is nearer to the center if the frequency of its corresponding sinusoid is low and farther otherwise. Therefore we may conclude that these high-intensity spots may represent rain streaks in the transformed domain and removing these high-intensity spots from the magnitude spectrum of the rainy image may remove most of the rain streaks in the spatial domain. However, there may exist some high-intensity spots in the magnitude spectrum which may contain some rain information, but they are not very sensitive to the human eye. One of the primary goals of this work is to train a deep network by exploiting this information to predict the rain map. To formally define the rain-streak information such as *relationship between the direction* of rain-streaks in spatial

and transformed domains, consider a two dimensional sinusoid with variables (x, y) in eucledian space and (u, v) in frequency space as

$$\begin{aligned} f_{cp} &= \texttt{exp}^{j2\pi(x.u+y.v)} \\ &= cos2\pi(x.u + y.v) + j.sin2\pi(x.u + y.v) \end{aligned} \tag{4.2}$$

Considering the projection of sinunoid f_{cp} on real axis, $cos2\pi(x.u+y.v)$ will have maxima and minima denoting the rain streaks patterns when $2\pi(x.u+y.v) = n\pi$. This can be written using vector notations as $2\pi(\vec{\mathbf{x}}.\vec{\mathbf{u}}) = n\pi$ which denotes the set of equally spaced parallel lines $\mathbf{L} = \{l_1, l_2, ...l_p\}$ along the direction of $\vec{\mathbf{u}}$ where $\vec{\mathbf{u}} = (u, v)^\intercal$ and $\vec{\mathbf{x}} = (x, y)^\intercal$. To formally define the relationship between rain directions in spatial and S* space, consider the complex exponential f_{cp} as

$$\begin{aligned} f_{cp} &= \texttt{exp}^{j2\pi(x.u+y.v)} \\ &= \texttt{exp}^{j2\pi w(x.\frac{u}{w}+y.\frac{v}{w})} \\ &= \texttt{exp}^{j2\pi w(\vec{\mathbf{E}}_c^l.\vec{\mathbf{F}}_c)} \end{aligned} \tag{4.3}$$

where $w = \sqrt{u^2 + v^2}$, $\vec{\mathbf{F}}_c$ is unit vector along the direction of (u, v) and $\vec{\mathbf{E}}_c^l = (x, y)^\intercal$ is a spatial vector on the line l. The dot product $\vec{\mathbf{E}}_c^l.\vec{\mathbf{F}}_c$ represents the projection $\mathbf{AB}$ of spatial point $\vec{\mathbf{E}}_c^l$ onto the direction of $\vec{\mathbf{F}}_c$ as shown in Figure 4.2(P). All points on a straight line l perpendicular to the direction of $\vec{\mathbf{F}}_c$ have same projection. Hence $\texttt{exp}^{j2\pi(x.u+y.v)}$ represents a two dimensional planar sinusoid with frequency w and whose direction is along the vector $\vec{\mathbf{F}}_c$, i.e, $\theta_{\mathsf{R_f}} = tan^{-1}(\frac{v}{u})$. As shown in Figure 4.2(Q), let $\theta_{\mathsf{R_c}}$ denote the rain streak direction in spatial domain and $\theta_{\mathsf{R_f}}$ in S* space. $\angle J = 180° - \theta_{\mathsf{R_c}}$. Using the summation property of triangles on $\triangle OAB$, $\theta_{\mathsf{R_f}}$ can be written as $\theta_{\mathsf{R_c}} - 90°$.

4.1.2 Fourier Domain Input to Deep CNNs

The input to the deep network is one of the major concerns in this work. Intuitively, both magnitude and phase spectrums are required to model the rain streak map in the Fourier domain. It is clear from Figure 4.5 that the phase spectrum also contributes to reconstructing the rain map. In Figure 4.5, 4.5b

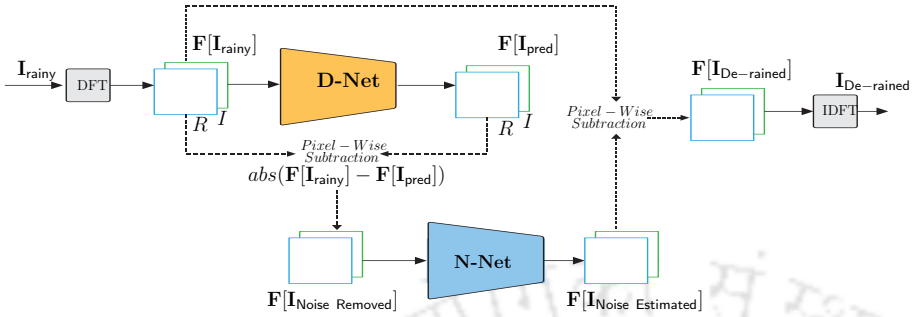


Figure 4.3: *The CNN framework of the proposed single-image rain streak removal. The details of D-Net and N-Net are given in Figure 4.6.*

is reconstructed using magnitude spectrum of clean image and phase of rainy image by using inverse discrete Fourier transformation based on the Eqs. 2.6, 2.5. Figure 4.5c is reconstructed using magnitude spectrum of rainy image 4.5a and phase of clean image 4.5e. Figures 4.5b, 4.5c still contains rain streaks when compared to original clean image 4.5d. But it can be observed that most of the rain information is captured in the magnitude spectrum, compared to phase, since 4.5b has fewer rain streaks than 4.5c. Therefore for rain streak removal, both magnitude and phase spectrums have to be given into the network[1]. To reduce the computational cost, instead of magnitude and phase information, real and imaginary coefficients are given as input to the deep network. Let $\mathbf{I}_{\text{rainy}}$ be the rainy image, $\mathbf{I}_{\text{clean}}$ be the clean image and $\mathbf{I}_{\text{rain map}}$ be the rain streak map associated with $\mathbf{I}_{\text{rainy}}$. $\mathbf{I}_{\text{rainy}}$ can be written as

$$\mathbf{I}_{\text{rainy}} = \mathbf{I}_{\text{clean}} + \mathbf{I}_{\text{rain map}} \tag{4.4}$$

When we convert the RGB rainy image to YCbCr colorspace, it is observed that most of the rain streak information exists in Y channel only. Therefore input to the network is discrete Fourier transformation of Y channel based on Eq. 2.3 as

$$\Psi(\mathbf{I}_{\text{rainy}}) = \mathbf{F}[\text{Y}_{\text{rainy}}] = \mathbf{F}_{\text{R}}[\text{Y}_{\text{rainy}}] \circ \mathbf{F}_{\text{I}}[\text{Y}_{\text{rainy}}] \tag{4.5}$$

[1]Note: Figures 4.5e,4.5a are first converted into YCbCr color space. Magnitude and phase are then calculated by performing DFT [83] on the Y channel for each image. Chrominance values of Figure 4.5a has been used to construct Figures 4.5b,4.5c. Chrominance values of Figure 4.5e has been used to construct Figure 4.5d

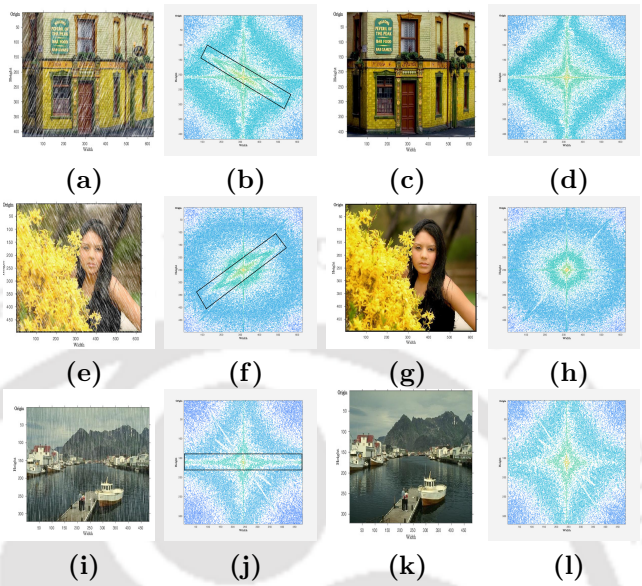


Figure 4.4: *Different rain directions and thier corresponding orientations in the contour plots of magnitude spectrums which are in S^* space. The mentioned angles are approximations. (a) Rain direction $\approx 45°$, (b) Magnitude image of 4.4a, m.a $\approx 135°$, (c) Clean image of 4.4a, (d) Magnitude image of 4.4c, (e) Rain direction $\approx 135°$, (f) Magnitude image of 4.4e, m.a $\approx 45°$, (g) Clean image of 4.4e, (h) Magnitude image of 4.4g, (i) Rain direction $\approx 90°$, (j) Magnitude image of 4.4i, m.a $\approx 0°$, (k) Clean image of 4.4i, (l) Magnitude image of 4.4k.*

where $\circ$ is concatenation operation depthwise, $\mathbf{F}$ is a two dimensional discrete Fourier transformation defined in Eq. 2.3 and $\mathbf{F}_{\mathsf{R}}$, $\mathbf{F}_{\mathsf{I}}$ are real and imaginary parts respectively. In later part of the chapter, $\mathbf{F}[\mathrm{Y}_{\mathsf{rainy}}]$ is also refered as $\mathbf{F}[\mathbf{I}_{\mathsf{rainy}}]$.

4.1.3 Noise residual in Fourier domain

The noise residual based on the Eq. 4.4 can be defined as

$$\mathbf{I}_{\mathsf{rain\ map}} = abs(\mathbf{I}_{\mathsf{rainy}} - \mathbf{I}_{\mathsf{clean}}) \tag{4.6}$$

where $\mathbf{I}_{\mathsf{rain\ map}}$ is called as noise/rain map residual and *abs* is absolute difference. Once the model predicts rain map residual, it is pixel-wise subtracted from the rainy image to get the de-rained image. We train our model to learn the rain map residual from rainy image in Fourier domain. The concept of noise residual

SSIM : 0.6982 SSIM : 0.7877 SSIM : 0.7453 SSIM : 1.0000 SSIM : 1.0000

PSNR : 18.43 dB PSNR : 22.40 dB PSNR : 17.59 dB PSNR : ∞ PSNR : ∞

(a) **(b)** **(c)** **(d)** **(e)**

Figure 4.5: *Reconstruction of images using different phase and magnitude spectums in terms of SSIM and PSNR. (a) Rainy image, (b) Magnitude of 4.5e and phase of 4.5a, (c) Magnitude of 4.5a and phase of 4.5e, (d) Magnitude of 4.5e and phase of 4.5e, (e) Ground truth.*

described in Eq. 4.6 can also be used in frequency domain since DFT holds the linearity property which states that,

$$\mathbf{F}[p+q] = \mathbf{F}[p] + \mathbf{F}[q]$$
$$\mathbf{F}[c.p] = c.\mathbf{F}[p]$$

where p, q are two dimensional discrete signals and $\mathbf{F}$ is discrete Fourier transformation with some constant c. Considering above, Eq. 4.4 in frequency domain can be expressed as,

$$\begin{aligned}\mathbf{F}[\mathbf{I}_{\text{rainy}}] &= \mathbf{F}[\mathbf{I}_{\text{clean}} + \mathbf{I}_{\text{rain map}}] \\ &= \mathbf{F}[\mathbf{I}_{\text{clean}}] + \mathbf{F}[\mathbf{I}_{\text{rain map}}]\end{aligned} \tag{4.7}$$

The rain map residual as defined in spatial domain using Eq. 4.6, can be formulated in frequency domain as

$$\begin{aligned}\mathbf{F}[\mathbf{I}_{\text{rain map}}] =& abs(\mathbf{F}[\mathbf{I}_{\text{rainy}}] - \mathbf{F}[\mathbf{I}_{\text{clean}}]) \\ =& abs(\mathbf{F}_{\text{R}}[\mathbf{I}_{\text{rainy}}] - \mathbf{F}_{\text{R}}[\mathbf{I}_{\text{clean}}]) \circ \\ & abs(\mathbf{F}_{\text{I}}[\mathbf{I}_{\text{rainy}}] - \mathbf{F}_{\text{I}}[\mathbf{I}_{\text{clean}}])\end{aligned} \tag{4.8}$$

where $\circ$ is depthwise concatenation operation. The real, imaginary coefficients are pixel-wise subtracted from the real, imaginary coefficients of rainy image to get the real, imaginary coefficients of de-rained image. The rain streaks free image in spatial domain can be reconstructed using inverse DFT on calculated real and imaginary parts.

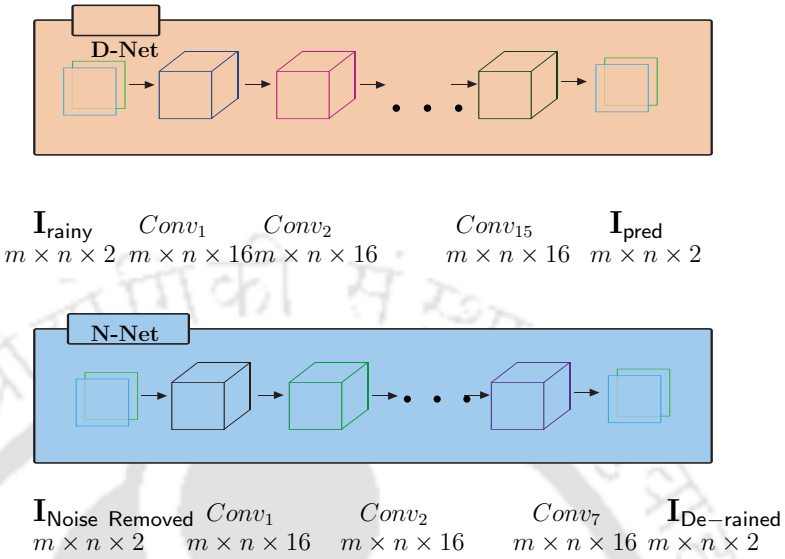


Figure 4.6: *The detailed architectures of D-Net and N-Net.*

4.2 Proposed Networks

We fix the size of input patch based on the Eq. 4.5, i.e., $\mathrm{X} \in \mathbb{R}^{128\times128\times2}$. Given X, the networks are trained to learn the parameter Θ of a function $h(\cdot|\Theta)$, which maps X to the output $\mathrm{Y} \in \mathbb{R}^{128\times128\times2}$ such that

$$\mathrm{Y} = h(\mathrm{X}|\Theta) = f_L(f_{L-1}|\Theta_L) \tag{4.9}$$

where f_0=X each operation $f_i(\cdot|\Theta_i)$ is referred as an i^{th} convolution layer of the network for $i > 0$. The proposed network denoted as $\text{RRSIFT}_{4.3}$ shown in Figure 4.3 is comprised two major modules **D-Net** and **N-Net**. The **D-Net** takes input as described in Eq. 4.5 and predict the real, imaginary coefficients of rain streak less image denoted by $\mathbf{F}[\mathbf{I}_{\text{pred}}]$ in Figure 4.3. The initial estimate of rain streak map can be calculated using below equation as

$$\mathbf{F}[\mathbf{I}_{\text{Noise Removed}}] = abs(\mathbf{F}[\mathbf{I}_{\text{rainy}}] - \mathbf{F}[\mathbf{I}_{\text{pred}}]) \tag{4.10}$$

The range of DFT coefficients is $(-\infty, \infty)$ hence it may be difficult for a deep network to estimate the coefficients of rain streak map from a rainy image. Therefore an initial estimate of rain streak is calculated using **D-Net** and given as an

input to **N-Net** to predict the complete rain streak map. The final predicted rain streak map by **N-Net** in Fourier domain is subtracted from the input of **D-Net** to get the final DFT coefficients of rain free image. The de-rained image in spatial domain can be recovered using IDFT. The detailed architectures of **D-Net** and **N-Net** are described in following subsections.

4.2.1 D-Net

D-Net as shown in Figure 4.6 is comprised of 16 convolution layers $l_{1,2,\cdots,16}$ and predicts the real and imaginary parts of Y channel of de-rained image based on the Eq. 4.9 with $L = 16$. The denoiser D-Net is parameterized as follows. l_1 has input of the shape $m \times n \times 2$ with 16 convolution filters of shape 3×3. $l_{2:15}$ comprise of 16 convolution filters at each layer with shape 3×3. l_{16} outputs the real and imaginary parts of Y channel of initially predicted de-rained image of shape $m \times n \times 2$ with 2 convolution filters of shape 3×3. Each filter has a stride of 1 and padding = *same*.

4.2.2 N-Net

N-Net as shown in Figure 4.6 is comprised of 8 convolution layers $l_{1,2,\cdots,8}$ and predicts the real and imaginary parts of rain streak map based on the Eq. 4.9 with $L = 8$. The noise estimator N-Net is parameterized as follows. l_1 has input of the shape $m \times n \times 2$ with 16 convolution filters of shape 3×3. $l_{2:7}$ comprise of 16 convolution filters at each layer with shape 3×3. l_8 outputs the real and imaginary parts of final predicted rain streak map of shape $m \times n \times 2$ with 2 convolution filters of shape 3×3. Each filter has a stride of 1 and padding = *same*. We have not used the pooling layers as it might cause in losing minor rain streak information due to down-sampling. We have not used any non-linearity in the network and have not normalized the input. The detailed discussion is given in the Section 4.4.

4.2.3 Loss functions

We calculate the loss and optimize the networks **D-Net** & **N-Net** in frequency domain. Let $\mathbf{I}_{\text{rainy}}$ be the rainy image whose ground truth image and associated rain map can be written as $\mathbf{I}_{\text{clean}}$, $\mathbf{I}_{\text{rain map}}$ respectively. Outputs of networks can be defined as

$$\begin{aligned} f^{l}_{\text{D-Net}}(\mathbf{F}[\mathbf{I}_{\text{rainy}}]) &= \mathbf{W}^{l}_{\text{D-Net}} * f^{l-1}_{\text{D-Net}}(\mathbf{F}[\mathbf{I}_{\text{rainy}}]) \\ &\quad + \mathbf{b}^{l}_{\text{D-Net}} \\ f^{l}_{\text{N-Net}}(\mathbf{F}[\mathbf{I}_{\text{Noise Removed}}]) &= \mathbf{W}^{l}_{\text{N-Net}} * f^{l-1}_{\text{N-Net}}(\mathbf{F}[\mathbf{I}_{\text{Noise Removed}}]) \\ &\quad + \mathbf{b}^{l}_{\text{N-Net}} \end{aligned} \tag{4.11}$$

where l is the layer index, $*$ be the convolution operator with $\mathbf{W}_{\text{D-Net}}$, $\mathbf{b}_{\text{D-Net}}$, $\mathbf{W}_{\text{N-Net}}$, $\mathbf{b}_{\text{N-Net}}$ as weights, biases of D-Net and N-Net with $f^{0}_{\text{D-Net}}(\mathbf{F}[\mathbf{I}_{\text{rainy}}]) = \mathbf{F}[\mathbf{I}_{\text{rainy}}]$ and $f^{0}_{\text{N-Net}}(\mathbf{F}[\mathbf{I}_{\text{Noise Removed}}]) = \mathbf{F}[\mathbf{I}_{\text{Noise Removed}}]$.

Let the actual target output of D-Net and N-Net be denoted as $\mathbf{F}[\mathbf{I}_{\text{clean}}]$, $\mathbf{F}[\mathbf{I}_{\text{rainy}}] - \mathbf{F}[\mathbf{I}_{\text{clean}}] = \mathbf{F}[\mathbf{I}_{\text{rain map}}]$ respectively. The loss functions of D-Net and N-Net can be defined as

$$\begin{aligned} \mathrm{L}_{\textbf{D-Net}} &= \sum_{i=1}^{N} \left|\left| \mathbf{F}[\mathbf{I}^{i}_{\text{pred}}] - \mathbf{F}[\mathbf{I}^{i}_{\text{clean}}] \right|\right|_2^2 \\ \mathrm{L}_{\textbf{N-Net}} &= \sum_{i=1}^{N} \left|\left| \mathbf{F}[\mathbf{I}^{i}_{\text{Noise Estimated}}] - \mathbf{F}[\mathbf{I}^{i}_{\text{rain map}}] \right|\right|_2^2 \end{aligned} \tag{4.12}$$

where N is the number of image samples in the training set. The D-Net is first trained followed by the N-Net.

4.3 Results

We have implemented both the modules in Tensorflow [141] framework. We have used the SSIM [15], PSNR as evaluation metrics implemented in MATLAB 2018a. The synthetic datasets of rainy and clean images given by the authors of [16], [3] and [51] have been used for training and testing. Fu *et al.* [16] have selected clean images from various sources such as UCID [144], BSD [145] and Google images

Dataset	Models	PSNR	SSIM [15]	FoM†
TD-Zhang *et al.* [3]	Input	21.15	77.81	49.48
	Luo *et al.* [38]	**21.44**	**78.96**	**50.20**
	Li *et al.* [44]	22.75	83.52	53.135
	Fu *et al.* [45]	**22.07**	84.22	53.145
	Yang *et al.* [51]	24.32	86.22	55.27
	Fu *et al.* [16]	27.33	89.78	58.55
	Zhu *et al.* [43]	23.05	85.22	54.135
	Zhang *et al.* [3]	27.95	90.87	59.41
	D-Net	**21.80**	**78.97**	**50.38**
	D-Net + N-Net	**22.22**	**78.46**	**50.34**
TD-Fu *et al.* [16]	Input	21.63	81.57	51.60
	Fu *et al.* [16]	27.56	91.57	59.56
	D-Net	**22.21**	**82.71**	**52.46**
	D-Net + N-Net	**22.59**	**82.31**	**52.45**
TD-Yang *et al.* [51] - H	Input	12.13	50.44	31.28
	Yang *et al.* [51]	23.45	74.90	49.17
	D-Net	**14.44**	**53.80**	**34.12**
	D-Net + N-Net	**13.87**	**52.50**	**33.18**
TD-Yang *et al.* [51] - L	Input	25.52	90.54	58.03
	Yang *et al.* [51]	36.11	97.00	66.55
	D-Net	**23.05**	**89.75**	**56.40**
	D-Net + N-Net	**23.92**	**89.51**	**56.71**

Table 4.1: *Quantitative results evaluated in terms of average SSIM [15] and PSNR (dB) on the test datasets. FoM† = $\frac{SSIM+PSNR}{2}$. SSIM [15] values shown here have been multiplied by 100.*

in order to generate rainy images with different orientations and density of rain streaks. Yang *et al.* [51] have divided the dataset into two categories, one with light and another with heavy rain density. Zhang *et al.* [3] have included the medium rain density images in addition to light and heavy. We have randomly selected our training data only from the dataset available by Fu *et al.* [16] for both the modules. We have augmented the rainy and clean images from our selected training dataset into the disjoint patches of size 128×128, thus creating a total of 1,20,000 patches. We have trained our both modules on randomly selected 1,00,000 patches and validated on 20,000. We have selected three test datasets(TD). TD-Fu *et al.* [16] is taken randomly which has 2800 rainy images. TD-Yang *et al.* [51] (Heavy[1] and Light) and TD-Zhang *et al.* [3] are given by the authors of [51] and [3] respectively. For real-world case comparison, we have

[1]Rain-streaks in this test-set may contradict with the real-rain.

adopted images provided by the authors of [138]. The quantitative comparison has been shown in the Table 4.1 on all test datasets. We have compared our models with the quantitative comparison given by Zhang *et al.* [3]. It can be observed that the proposed DFT based approach for rain streak removal achieves a comparable result with state-of-the-art approaches on the test datasets. It is important to note that while the existing architectures incorporate and learn spatial domain features in their architecture, this is the first work achieving comparable performance using frequency domain input in deep convolutional neural networks. We have shown the qualitative results on real-world rainy images in Figure 4.7. Figure 4.8 and 4.9 shows the qualitative results achieved by our proposed methods on the test dataset TD-Zhang *et al.* [3]. It is observed that complete rain streaks have not been removed from the images when compared with the results obtained by spatial domain state-of-the-art methods. However, a visual improvement along with the reduction in rain streaks can be observed in the de-rained image compared to the original rainy image.

4.4 Discussion

4.4.1 Normalization of input data

It is observed that when input data is de-correlated, a linear model is enough to fit. The nonlinearity in the model may reduce the model capability and may get underfit. In general, input values to deep networks are normalized between a certain range to reduce the range of values to be predicted by the model. In this case, the input and values to be predicted by the proposed model lie between $-\infty$ to ∞ which makes normalization of the data difficult. We tried to normalize the input data based on the individual minimum and maximum values of input images and fed into the experimental networks A, B whose quantitative results are described in Table 4.4. Model A is topologically similar to **D-Net** with each convolutional layer has hyperbolic tangent as a nonlinear activation function. The input and target are normalized between [-1,1]. While testing,

Figure 4.7: *Qualitative results on real-world rainy images.* ***Top*** *row shows rainy images, whereas* ***Bottom*** *row shows our results.* ***TVE*** *($\times 10^6$) denotes the total variation error that describes the amount of noise present in an image.*

the result is upscaled by using the minimum and maximum DFT values of the input image which incurs in the loss. Model B is similar to A except the last convolution layer which does not have any activation function and predicts the un-normalized output. Both the models suffer from the underfitting problem as can be observed from Table 4.4. The loss in the reconstructed image is due to normalization techniques and also because of nonlinearity which is described in Subsection 4.4.3. Qualitative results of Table 4.4 are given in Figure 4.10.

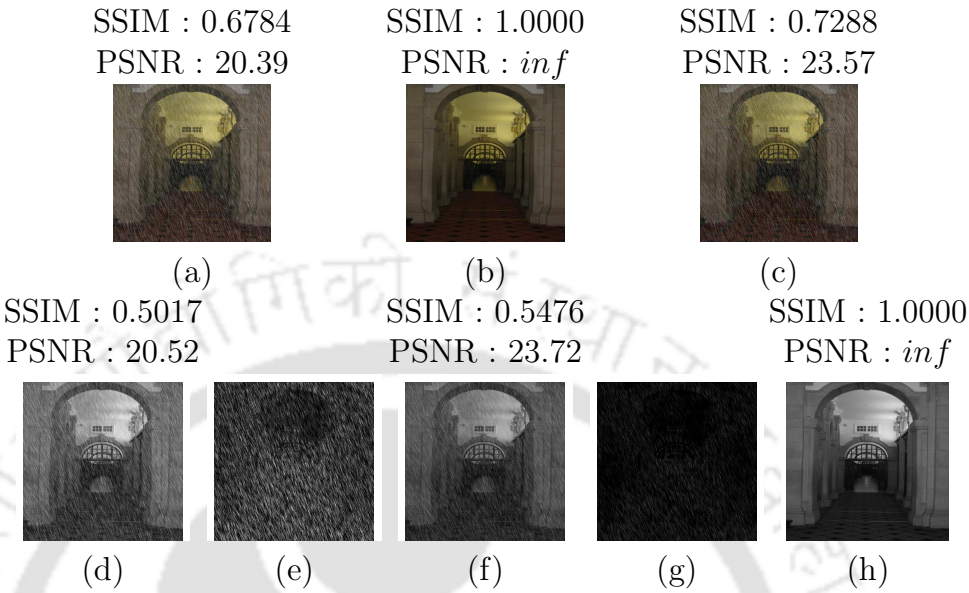


Figure 4.8: *Qualitative results on TD-Zhang et al. [3] dataset using **D-Net**. (**a**) Rainy image, (**b**) Clean image of (a), (**c**) Predicted de-rained image of (a), (**d**) Grayscale channel of (a) , (**e**) Rain present in (d), (**f**) Grayscale channel of (c), (**g**) Rain present in (f), (**h**) Grayscale channel of (b). Rain present map here has been calculated by taking absolute difference of rainy and clean images i.e., $|\boldsymbol{I}_{\text{rainy or predicted}} - \boldsymbol{I}_{\text{clean}}|$. PSNR is measured in dB.*

4.4.2 Single layer Vs. Multilayer

We carried out experiments with the different number of layers in the **D-Net** with detailed configurations given in Table 4.3. It can be observed that there is no significant improvement when the number of convolution layers is increased from 1 to 16. Therefore it may be inferred that the series of convolution layers without any nonlinearity and downsampling in between acts as a single convolutional layer.

4.4.3 Non-linearity

The nonlinearity in model is used to make the network complex so that it can learn more complex functions which is incapable for a linear model. It also hepls in filtering out the unwanted information to be sent into the subsequent

Figure 4.9: *Qualitative results on TD-Zhang et al. [3] dataset using the model* ***D-Net + N-Net****.* ***Top Row*** *:* ***(a)*** *Rainy image,* ***(b)*** *Clean image of (a),* ***(c)*** *Predicted derained image of (a),* ***(d)*** *Grayscale channel of (a) ,* ***(e)*** *Rain present in (d),* ***(f)*** *Grayscale channel of (c),* ***(g)*** *Rain present in (f),* ***(h)*** *Grayscale channel of (b).* ***Bottom Row*** *: goes the same as top. Rain streak map here has been calculated by taking absolute difference of rainy and clean images i.e.,* $|\boldsymbol{I}_{\text{rainy or predicted}} - \boldsymbol{I}_{\text{clean}}|$*. PSNR is measured in dB.*

layers in a sequential model. To note the affect of nonlinearity, we used three different activation functions namely hyperbolic tangent, sigmoidal and ReLU in an experimental models and quantitative results are given in Table 4.2. It can be seen from the Subsection 4.4.2 that single layer network behaves almost similar to multilayer in special cases, we trained three networks each with single convolutional layer and different activation functions mentioned in Table 4.2 and can be summarised as follows

1. Hyperbolic tangent transforms the DFT values from $(-\infty, \infty)$ to $(-1, 1)$

Activation	Mode	PSNR	SSIM [15]
tanh	Train	7.81	7.09
	Test	6.87	7.37
Sigmoid	Train	7.81	7.09
	Test	6.87	7.37
ReLU	Train	**18.18**	**68.63**
	Test	**16.24**	**71.51**

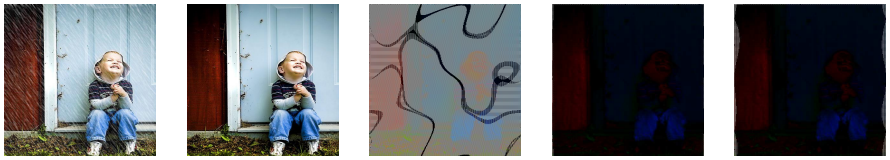
Table 4.2: *Quantitative results on the testset TD-Fu et al. [16] of experimental models with different nonlinear activation functions.*

Layer	Configuration	PSNR	SSIM [15]
1	*Conv k3-n2-s1-*l_1	22.39	82.13
4	*Conv k3-n16-s1-*$l_{1:3}$ *Conv k3-n2-s1-*l_4	22.44	82.19
8	*Conv k3-n16-s1-*$l_{1:7}$ *Conv k3-n2-s1-*l_8	22.54	82.29
16	*Conv k3-n16-s1-*$l_{1:15}$ *Conv k3-n2-s1-*l_{16}	22.21	82.71

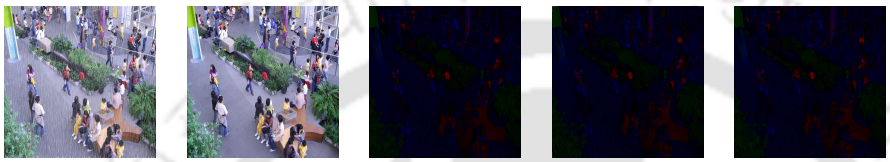
Table 4.3: *Quantitative results on the testset TD-Fu et al. [16] of experimental models with different number of layers.*

which is too much random in nature. Therefore the transformed data might not preserve any information for the model to learn and the model may underfit which has happened when we use hyperbolic tangent as an activation function as mentioned in Table 4.2.

2. Sigmoidal function transforms the DFT values from $(-\infty, \infty)$ to $(0, 1)$ which is again too much random similar to hyperbolic tangent and may result in losing the negative frequencies due to its range. Therefore it also results in underfit as mentioned in Table 4.2.
3. ReLU transforms the DFT values from $(-\infty, \infty)$ to $maximum(0, x)$ where x is the DFT value to be transformed. It can be observed that using ReLU will only result in losing negative DFT coefficients. Therefore there might be some information left after transformation which a model can attempt to learn. In this case, there is no underfitting as mentioned in Table 4.2. It is comparatively less random than hyperbolic tangent and sigmoid functions, therefore, perform better.



(a) *Rainy image* **(b)** *Ground truth* **(c)** *De-rained image after* 1^{st} *epoch* **(d)** *De-rained image after* 18^{th} *epoch* **(e)** *De-rained image after* 37^{th} *epoch all layer tanh*



(f) *Rainy image* **(g)** *Ground truth* **(h)** *De-rained image after* 1^{st} *epoch* **(i)** *De-rained image after* 4^{th} *epoch* **(j)** *De-rained image after* 6^{th} *epoch*

Figure 4.10: *Reconstruction of de-rained images in RGB colorspace after specified epoch using normalized input and activation function at,* ***top row****: each convolution layer,* ***bottom row****: except at last convolution layer.*

Epoch	Mode	PSNR	SSIM [15]	Epoch	Mode	PSNR	SSIM [15]
1	Train	5.64	2.19	1	Train	7.74	6.98
	Test	5.75	4.25		Test	6.87	7.36
18	Train	7.76	6.90	4	Train	7.71	6.91
	Test	6.86	7.27		Test	6.87	7.36
37	Train	7.70	6.74	6	Train	7.75	7.00
	Test	6.84	7.21		Test	6.87	7.37

Table 4.4: *Quantitative results on the test set TD-Fu et al. [16] of experimental models which takes normalized input and have* ***Left :*** *nonlinearities at each layer,* ***Right :*** *nonlinearities at each layer except the last.*

Therefore it can be concluded that the use of activation function in convolutional neural networks may remove some frequencies which are useful in reconstructing the image back in the spatial domain.

So far, the analysis of uncorrelated input domain to deep CNN has been presented. To summarize, it can be concluded that although, a minor improvement has been achieved using DFT domain, one may need a different transformed domain which can preserve comparatively more information than DFT domain. In this line of thought, we now shift our attention towards DWT domain which

retains the spatial correlation upto some extent.

4.5 Image De-Raining in Correlated Transformed Domain

Despite, Shen *et al* [146] used the concepts of Haar wavelets [147] and DCP [48] for image de-raining to predict the wavelet coefficients of the de-rained image from the same of the rainy image using deep CNN. However, in this part, it has been shown that wavelet sub-bands could be more suitable for predicting the rain streak map, and significant improvement can be achieved if these frequency domain cues are provided to the network in addition to the spatial domain features of the rainy image. In this line of thought, this part makes the following contributions:

- A cGAN based framework is proposed, which utilizes both spatial as well as frequency domain cues for image de-raining.
- The perceptual loss function is used to ensure the visual quality of the de-rained image.

4.6 Proposed Scheme

One may conclude that the success of an image de-noising algorithm relies on the choice of color space [148], input to the algorithm and cost function. This section presents the details of color space preferred, proposed architecture and cost function for image de-raining problem. Given a rainy image $\mathrm{R_I}$ with associated rain streak map $\mathrm{M_R}$ and clean background image $\mathrm{B_I}$ such that $\mathrm{R_I}$,$\mathrm{M_R}$,$\mathrm{B_I} \in [0, 255]^{h \times w \times 3}$ with height h and width w. Following [16], image $\mathrm{B_I}$ can be restored from $\mathrm{R_I}$ by using a conventional linear model as

$$\mathrm{B_I} = \mathrm{R_I} - \mathrm{M_R} \tag{4.13}$$

The more desirable color space for image denoising is YCbCr which is decorrelated, unlike RGB color space. Due to the pseudo-periodic additive property

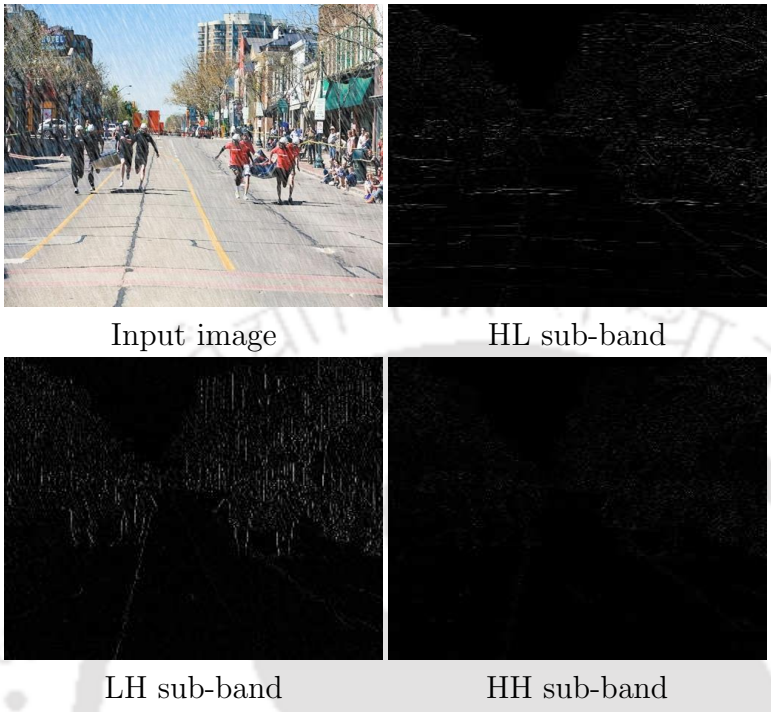


Figure 4.11: *DWT sub-bands of an image.*

and high-frequency nature of rain streaks noise, the obtained color differences Cb $\propto$ Y-B and Cr $\propto$ Y-R are smoothened, and noise remains only in luminance channel [148]. Therefore, the proposed method removes the rain streaks from the Y channel only. The frequency-domain cues are given to the proposed model in addition to the spatial domain features as input. Although image transformation from spatial to a frequency domain, in general, destroys the pixel correlation, which makes it challenging to use CNN's, it can be observed from Figure 4.11 that Discrete Wavelet Transformation (see Section 2.2), more specifically Haar wavelet, preserves the spatial correlation of the image to some extent.

The Haar wavelet transform decomposes a 2D discrete signal, such as an image, into four sub-bands that emphasize the finest resolution of the image. The approximation sub-band LL represents the background details of the image, whereas sub-band LH demonstrates the variation along the y axis, HL along the x axis and HH represents the diagonal details of the image. In general, the dyadic

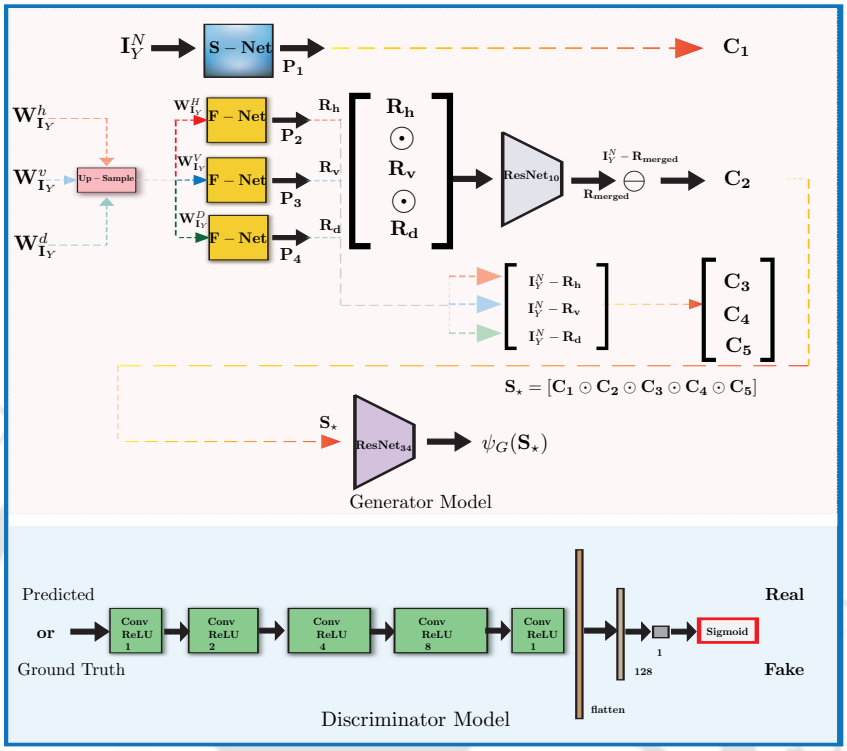


Figure 4.12: *Architecture of the proposed method for rain streak removal from single images.*

partitioning of the LL sub-band is used for detailed analysis. However, with most of the background inference eliminated, the sub-bands LH, HL and HH preserve a variety of information about rain streaks as shown in Figure 4.11. Therefore, these sub-bands are more suitable for predicting the rain streak map instead of directly mapping the luminance channel of the rainy image to the rain streak map. The other useful background details for image reconstruction are retained in the luminance channel of the rainy image, which is also given as an input along with chosen wavelet sub-bands.

To define the input given to the proposed network, let $\mathrm{I} \in [0, 255]^{h \times w \times c}$ be the rainy image with height h, width w and channel c. The rainy image is first converted into YCbCr color space whose luminance channel is denoted by $\mathrm{I}_Y \in [0, 255]^{h \times w \times 1}$. The Haar wavelet sub-bands of I_Y are then obtained such that

$W_{I_Y}^a$ represents approximation sub-band LL, $W_{I_Y}^h$ represents horizontal sub-band LH, $W_{I_Y}^v$ represents vertical sub-band HL and $W_{I_Y}^d$ represents diagonal sub-band HH where each sub-band $\in \mathbb{R}^{h/2 \times w/2 \times 1}$. The sub-bands $W_{I_Y}^h$, $W_{I_Y}^v$, $W_{I_Y}^d$ are spatially up-scaled by the factor of 2 such that $W_{I_Y}^H = UP(W_{I_Y}^h) \in \mathbb{R}^{h \times w \times 1}$, $W_{I_Y}^V = UP(W_{I_Y}^v) \in \mathbb{R}^{h \times w \times 1}$ and $W_{I_Y}^D = UP(W_{I_Y}^d) \in \mathbb{R}^{h \times w \times 1}$ where UP is Bicubic interpolation for up-scaling. The normalized I_Y, denoted by $I_Y^N \in [0,1]^{h \times w \times 1}$ is given as input to the proposed method along with $W_{I_Y}^H$, $W_{I_Y}^V$ and $W_{I_Y}^D$ which may act as the frequency domain cues to predict the luminance channel of the de-rained image.

Inspired by the GAN [50], the proposed architecture as shown in Figure 3.4 inherits the cGAN framework which consists of two following networks: Generator(G) and Discriminator(D). Given a rainy image R_I and clean image B_I, networks G and D play a 2-player minimax game based on the below equation

$$\min_G \max_D \; \mathbb{E}_{R_I \sim p_{rain}}[\log(1 - D(G(R_I)))] + \mathbb{E}_{B_I \sim p_{clean}}[\log(D(B_I))] \tag{4.14}$$

where D is trained to maximize the probability of correctly classifying the input samples whereas G is trained to generate more realistic de-rained images. The regimes of operation of G and D are described as follows.

4.6.1 Generator Network (G)

The proposed generator as shown in Figure 3.4, aims to learn multiple de-rained image candidates by exploiting spatial as well as frequency domain of the rainy image. It consists of four independent processing units P_1, P_2, P_3 and P_4. Unit P_1 takes I_Y^N as an input and process the spatial domain cues. Units P_2, P_3 and P_4 take $W_{I_Y}^H$, $W_{I_Y}^V$ and $W_{I_Y}^D$ as input respectively and process the frequency domain cues. Unit P_1 consists of a proposed sub-network called S-Net which comprises of six convolution layers with filter size 3×3, spatial stride of 1×1 with number of filters per layer $4, 8, 16, 32, 64$ and 1 respectively. Each layer in S-Net consists of BN [134] for faster convergence and ReLU activation function. The purpose of unit P_1 is

to utilize spatial features of the rainy image and outputs a clean image candidate C_1. Each of P_2, P_3 and P_4 units consist of a proposed sub-network called F-Net which comprises of four convolution layers where each layer has filters of size 3×3, spatial stride of 1×1 with number of filters at each layer are $4, 8, 6$ and 1 respectively. Each layer in F-Net consists of BN followed by ReLU. The purpose of units P_2, P_3 and P_4 is to utilize the cues in wavelet sub-bands LH, HL and HH which are more suitable for generating the rain maps and output the intermediate rain maps denoted by R_h, R_v and R_d respectively. ResNet can be more effective in improving the input signal quality [16]. Therefore, these intermediate rain maps are concatenated and feed into a 10 layers ResNet to refine further and output a merged rain map denoted as R_{merged}. Clean image candidate C_2 is obtained by pixel-wise subtracting R_{merged} from I_Y^N and the intermediate rain maps R_h, R_v and R_d are pixel-wise subtracted from I_Y^N to get the clean image candidates C_3, C_4 and C_5 respectively. Finally, the obtained clean candidates $C_{1:5}$ are concatenated and feed into a 34 layers ResNet to predict a final de-rained image.

4.6.2 Discriminator Network (D)

The objective of the discriminator network is to maximize the probability of precisely classifying the input samples into real or fake, thereby inspire the generator model to predict more realistic de-rained images. The proposed discriminator model, as shown in Figure 3.4 consists of 5 convolution layers. Each layer comprises of $1, 2, 4, 8$ and 1 filters of shape 3×3 respectively with spatial stride of 1×1 and ReLU activation. A fully connected layer with 128 neurons and ReLU activation function are used after that followed by a Sigmoid layer.

4.6.3 Cost Function

The cost function for the generator model can be defined as follows. Let $\psi_G(\mathbf{S}_\star)$ be the de-rained image estimated by the generator where $\mathbf{S}_\star = \{I_Y^N, W_{I_Y}^H, W_{I_Y}^V, W_{I_Y}^D\}$ is input to the proposed model. Let $y \in [0,1]^{h \times w \times c}$ be the ground truth image.

Methods	SSIM	PSNR	VIF	MS-SSIM	TV †	UQI	MSE ‡
Input	0.7781	21.15	0.3734	0.7334	1.55	0.8636	0.766
State-of-the-Art Methods							
CNN [45]	0.8422	22.07	0.4082	0.8384	1.25	0.8650	0.708
DDN [16]	0.8978	27.33	0.4246	0.8650	1.14	0.9526	0.124
JCA [30]	0.8374	23.63	0.3867	0.8145	1.05	0.8865	0.520
ID-CGAN [9]	0.8325	22.85	0.5177	0.9007	1.19	0.8922	0.513
DID-MDN [3]	0.9110	27.98	0.4552	0.8904	1.13	0.9497	0.124
Proposed	0.9209	30.05	0.4638	0.8943	1.01	0.9627	0.082
Proposed baseline configurations							
SF-GEN	0.9022	27.70	0.4561	0.8893	1.14	0.9234	0.126
SF-cGAN	0.9192	29.07	0.4604	0.8911	1.05	0.9326	0.107
S-cGAN-P	0.8849	25.48	0.4456	0.8678	1.04	0.9406	0.201
Proposed	0.9209	30.05	0.4638	0.8943	1.01	0.9627	0.082

Table 4.5: *Quantitative results compared with recent methods on synthesized test images. Best results are highlighted in blue color.* † TV *is* $\times 10^{7}$. ‡ MSE *is* $\times 10^{-3}$.

The MSE is used in majority of the de-noising algorithms and can be defined as

$$L_{\mathrm{E}} = \frac{1}{h.w.c} \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{c} ||\psi_G(\mathbf{S}_\star)^{i,j,k} - y^{i,j,k}||_2^2 \tag{4.15}$$

However, MSE does not correlate well with the HVS of image quality and may induce splotchy or blurred artifacts in the de-rained image [107]. Therefore, the perceptual loss function [8] is used to avoid these artifacts by preserving the contextual and high-level features of the image. For this, a pre-trained VGG-16 [5] model (V) is used for features extraction at convolution layer `conv2_2`. The perceptual loss can be defined as

$$L_{\mathrm{feat}} = \frac{1}{h.w.c} \sum_{i=1}^{h} \sum_{j=1}^{w} \sum_{k=1}^{c} ||\mathrm{V}(\psi_G(\mathbf{S}_\star))^{i,j,k} - \mathrm{V}(y)^{i,j,k}||_2^2 \tag{4.16}$$

Given the set of N de-rained images, the entropy loss from the discriminator to govern the generator can be defined as

$$L_{\mathrm{adv}} = -\frac{1}{\mathrm{N}} \sum_{i=1}^{\mathrm{N}} \log\, \mathrm{D}(\psi_G(\mathbf{S}_\star)_i) \tag{4.17}$$

Therefore the total loss for the generator can be defined as

$$L_{\mathrm{G}} = \lambda_{\mathrm{E}}.L_{\mathrm{E}} + \lambda_{\mathrm{adv}}.L_{\mathrm{adv}} + \lambda_{\mathrm{feat}}.L_{\mathrm{feat}} \tag{4.18}$$

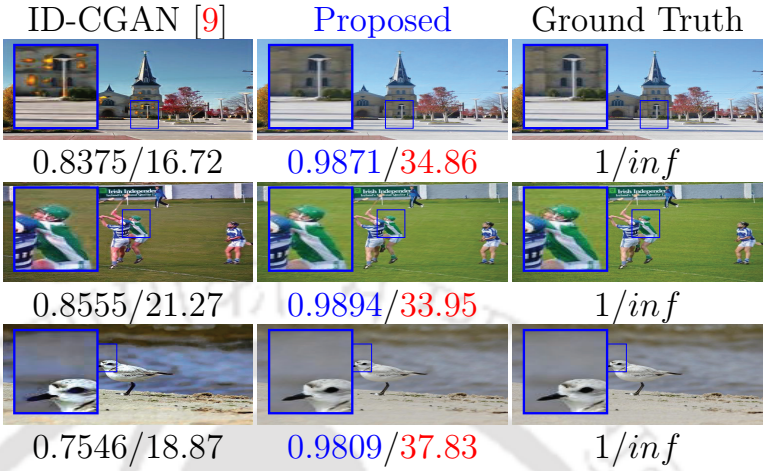


Figure 4.13: *Qualitative comparison with Method [9] on synthesized test images in terms of* SSIM/PSNR.

where λ_{E}, λ_{adv} and λ_{feat} pre-defined weights for each loss. Objective of our proposed method is to minimize L_{G}.

4.7 Experiments and Results

This section presents the details of synthetic, real-world rainy image dataset publicly available at [1] [3] and parameters used for training and evaluation of the proposed method followed by a qualitative comparison with most recent methods. The training images are augmented into the disjoint patches of size 128×128 thereby generating 192K images. The synthetic test set consists of 1201 images of size 512×512. The proposed framework is also evaluated on real-world rainy images.

[1] https://github.com/hezhangsprinter/DID-MDN

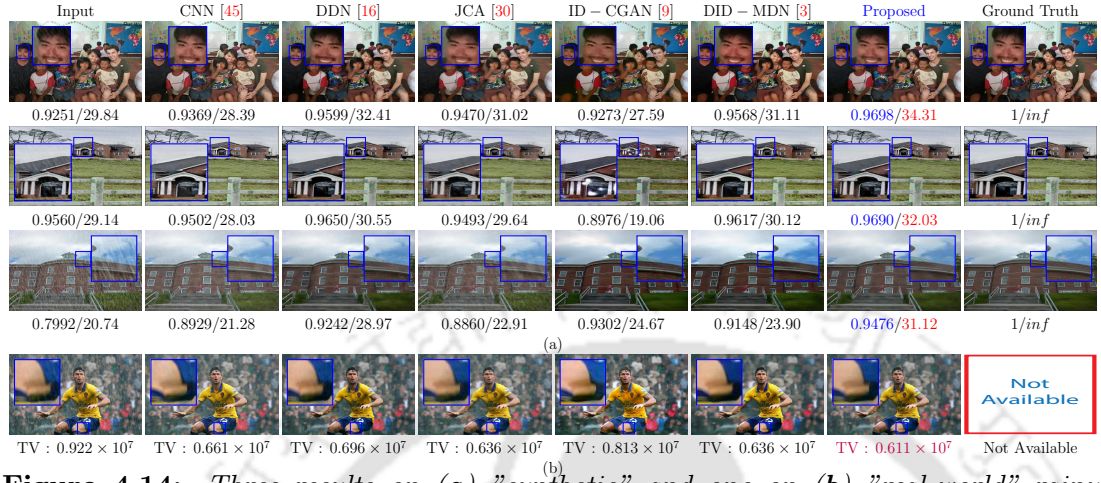


Figure 4.14: *Three results on (**a**) "synthetic" and one on (**b**) "real-world" rainy images in terms of* SSIM/PSNR *and* TV.

The entire model is trained on Nvidia-GTX 1080 GPU using the Tensorflow framework for 42 epochs with learning rate set to 0.01, batch size of 20 and Adam [142] optimization algorithm. The cost weights are experimentally set as follows: $\lambda_{\mathrm{E}} = 0.9889$, $\lambda_{\mathrm{adv}} = 0.01$ and $\lambda_{\mathrm{feat}} = 0.001$. The proposed method is compared with 5 state-of-the-art methods using 7 evaluation metrics namely SSIM [15], PSNR in dB, VIF [116], MS-SSIM [114], TV, UQI [115] and MSE. The description of incorporated evaluation metrics can be found in Section 2.5.

4.7.1 Performance Evaluation

Quantitative results compared with state-of-the-art methods [3, 9, 16, 30, 45] and baseline configurations are given in Table 4.5. The baseline configuration SF-GEN refers to the framework consists of only proposed generator model and is trained on L_{G} loss by setting λ_{adv} and λ_{feat} to zero. SF-cGAN consists of the proposed cGAN framework and is trained by setting λ_{feat} to zero. S-cGAN-P process spatial domain features only and consists of unit P_1 followed by a 34 layers ResNet which outputs the de-rained image. It is trained on L_{G} with similar cost weights as in the case of the proposed method. The proposed method has shown an impressive improvement over recent methods and baseline configurations in

terms of all evaluation metrics. A visual improvement over existing methods can be observed from Figures 4.13, 4.14. While the methods [16], [45] and [30] still contain rain streaks in the de-rained images, methods [3] and [9] suffer from the problem of over-coloring and white round artifacts in the de-rained images respectively. The proposed method does not suffer from such artifacts. Unlike in methods [16, 45], the use of perceptual loss and adversarial training may have improved the visual quality of the de-rained images by preserving the high-level features. Methods [3,9] predicted the de-rained images in RGB color space which is highly correlated. In such color spaces, the misinterpretation of a pixel value in one channel may lead to the same in others. The relaxation of Cb, Cr and the use of frequency domain cues in addition to spatial domain input have found to be beneficial, and the proposed method has shown a significant improvement of $\sim 4\%$ in SSIM and $\sim 18\%$ in PSNR over S-cGAN-P. The proposed method has performed better than existing methods on the real-world rainy image as shown in Figure 4.14 where TV represents the amount of noise present in an image and found to be least in the case of proposed method. Methods [3, 9] might have suffered from over-coloring similar to synthetic results obtained by the same. However, unlike synthetic, real-world images do not have ground truth to verify this assumption. On average, the proposed method takes about 0.168 secs on 8GB GPU to de-rain an image of size 512×512.

4.8 Summary

In the first part of this contributory chapter, a deep learning architecture has been presented for single image de-raining that takes DFT coefficients of the rainy image as input. While our proposed network D-Net predicts the initial estimate of the rain streak map, D-Net+N-Net learns the final estimate of the rain map, which can be subtracted pixel-wise from rainy image DFT coefficients to get de-rained image DFT coefficients. It has been shown that when the input is in the Fourier domain, the presence of non-linearity in the network may result

in underfitting. We have used the synthetic datasets of rainy and clean images provided by the authors of [16], [3], and [51] for single image de-raining. It has been proved that rain streaks information is preserved in the DFT domain, and deep CNN can be trained to utilize such features for image de-raining problem.

In the second part, a dual-domain conditional GAN based framework is proposed for a single image de-raining task. The proposed method utilized both spatial, frequency domain features and produced results are both quantitatively and qualitatively better than existing state-of-the-art methods. It is shown how frequency domain cues can be used along with the spatial domain features for the image de-raining task, and significant improvement can be achieved over the spatial domain baseline method. The visual quality of the de-rained image has been ensured by using the perceptual loss function. Unlike existing methods, the proposed method does not suffer from the problem of over-coloring and white rounds artifacts.

So far in this dissertation, a pseudo-periodic additive noise in terms of single image rain-streak removal has been addressed using a variety of deep CNNs. Both spatial and transformed domain aspects of the rain-streaks have been exploited. However, in the real world, noise in an image may not be as simplistic as rain-streaks. In that case, one may require to approach in a different direction by considering the underlying distribution of the noise. With this motivation, the next chapter addresses an exponential noise in terms of single image de-hazing using deep learning techniques.

Chapter 5

A Probe Towards Scale-Space Invariant Conditional GAN for Image De-Hazing

Haze in an image follows exponential distribution based on the depth of the pixel (see Eq. 1.2). Most of the methods discussed in Chapter 1 have been successful in the single image dehazing task. However, a few of such haze-free images suffer from color degradation and halo artifacts that prevail around the high-intensity regions and edgy structures.

In particular, existing methods such as [48, 55] suffer from color degradation, halo and checkerboard artifacts that prevail around the high-intensity regions, and edgy structures in the de-hazed images (see Figure 5.1). Also, A majority of the existing learning-based schemes, to name a few [10, 18], separately estimates the transmission and atmospheric light maps, which incurs the computational cost. And most importantly, considering the exponentially varying haziness, the scale-space property of the hazy images has not been exploited in the case of deep model engineering. It has also been shown Section 2.3 that the LoG retains a variety of edgy structures at different scales in an image.

In the case of image de-noising, deep learning models may consider every object in an image at the same scale-space. As a result, the de-noised images may suffer from blurriness and halo artifacts. It is based on the fact that a traditional CNN model does not aware of scale-space of an object. Therefore,

Color-degradation Halo Checkerboards

Figure 5.1: *A graphical demonstration of the color degradation, halo and checkerboard artifacts in the de-hazed results of existing works.*

this chapter makes the *first attempt* to study the behavior of a scale-space aware deep learning-based model. For this, we define a *new* loss function, which is based on the LoG of an image. It has been observed during our experimentation that the difference of LoG's between clean and de-hazed images can be used as a cost function to optimize the proposed model (see Figure 5.2). Such incorporations may help the model to learn the scale-space [85] of every object in an image. An analogy, based on the use of perceptual loss function [8] in a deep network to recover the high-frequency details of an image, may support this argument. Therefore, the contributions of this chapter can be summarised as follows:

1. A novel scale-space aware conditional GAN based method has been proposed for single image de-hazing. In addition to the adversarial training, the perceptual loss function has been used to enhance the visual quality of the de-hazed images.
2. We introduce the LoG difference between clean and de-hazed images as a cost function to optimize the proposed conditional GAN-based model and wipe out the halo artifacts in the de-hazed images by retaining the edgy structures more precisely.
3. A random data augmentation has been done when training to improve the efficiency of the proposed model further. A brief study of the same has been given in this chapter followed by an ablation study, which is presented at

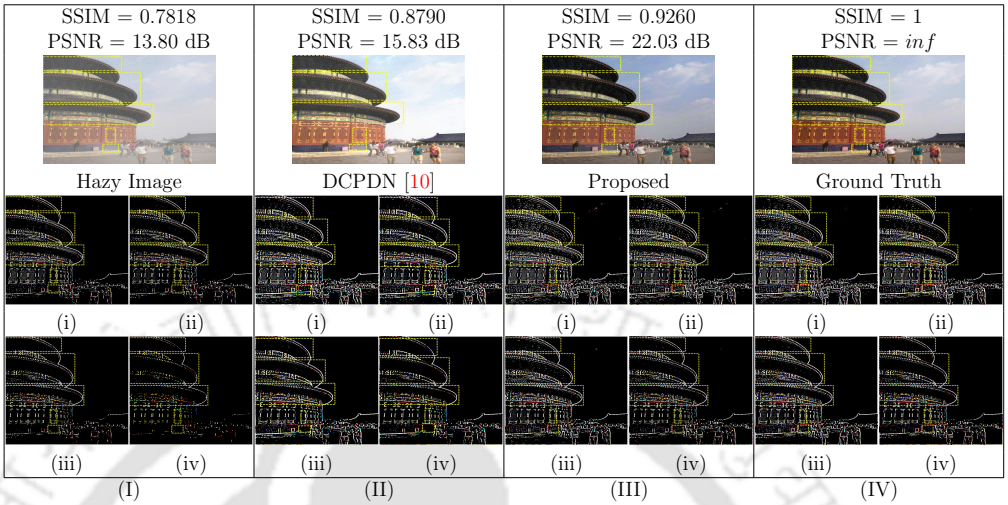


Figure 5.2: *Sample Laplacians of Gaussian (LoG) filters of (I) Hazy, (II) Dehazed by using [10], (III) Proposed and (IV) Clean images. For each in I,II,III and IV, (i)* $\mathsf{G}(\mathbf{m},\mathbf{n},k\sigma) - \mathsf{G}(\mathbf{m},\mathbf{n},\sigma)$*, (ii)* $\mathsf{G}(\mathbf{m},\mathbf{n},k^2\sigma) - \mathsf{G}(\mathbf{m},\mathbf{n},k\sigma)$*, (iii)* $\mathsf{G}(\mathbf{m},\mathbf{n},k^3\sigma) - \mathsf{G}(\mathbf{m},\mathbf{n},k^2\sigma)$*, and (iv)* $\mathsf{G}(\mathbf{m},\mathbf{n},k^4\sigma) - \mathsf{G}(\mathbf{m},\mathbf{n},k^3\sigma)$*.*

the end of this chapter, along with extensive experiments.

5.1 Proposed approach

In this section, we first present the architecture of the proposed model as shown in Figure 5.3, which is based on a conditional GAN framework followed by the cost functions incorporated for the single image dehazing task. The proposed model comprises of two main sub-networks, Generator (ϕ_G) and Discriminator (ϕ_D). The regimes of operations of ϕ_G and ϕ_D are as follows.

Generator (ϕ_G) model takes input as hazy image $\mathbf{H}$ in RGB color space and predicts the corresponding dehazed image $\bar{\mathbf{C}}$. It consists of an encoder-decoder [7] architecture which has been useful in various image restoration tasks. The encoder part consists of 6 convolutional layers, each with 64 kernels followed by BN [134] and ReLU activation function. Each kernel has a spatial dimension of 3×3 with stride and padding of 1. On the other hand, the decoder part comprises of 6 transpose convolutional layers which are also known as *Deconvolution*.

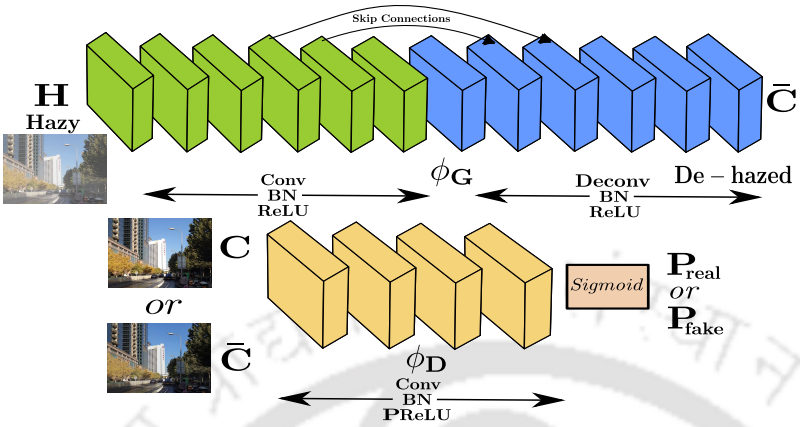


Figure 5.3: *An overview of the proposed model for the single image dehazing problem.*

Each deconvolution layer consists of 64 kernels except the last, with the spatial dimension of 3×3 and stride, padding of 1 followed by BN and ReLU. Orhan *et al.* [149] have shown that skip connections may reduce the singularities, such as elimination, overlap and those caused by the linear dependence of the nodes, that slow down the training process of deep CNN. Therefore, the skip connections have been assigned between layers 4, 5 of the encoder to layers 3, 2 of the decoder. The proposed ϕ_G directly estimates the dehazed image $\bar{\mathbf{C}} = \phi_G(\mathbf{H})$ from the input hazy image $\mathbf{H}$. Unlike Zhang *et al.* [10], the proposed scheme preserves the spatial dimension of the input ($\mathbf{H}$) and output ($\bar{\mathbf{C}}$) images, thereby achieving the shape invariant nature.

Discriminator (ϕ_D) learns to maximize the probability of precisely classifying the input samples into real or fake de-hazed images. This, in turn, helps the ϕ_G to generate natural de-hazed images. The proposed ϕ_D consists of 4 convolution layers with 8, 16, 32, and 3 kernels respectively, followed by BN and PReLU [150] activation function. Each kernel in ϕ_D has a spatial dimension of 3×3 with stride and padding of 1. The output of the ϕ_D is the mean sigmoid over the feature maps from the last convolution layer.

5.1.1 Loss function

Let $\phi_G(\mathbf{H}) \in [0,1]^{c \times w \times h}$ be the dehazed image estimated by the proposed model with c, w, h as channels, width and height respectively. The conventional *per-pixel* loss (L_E) between dehazed and ground truth ($\mathbf{C}$) images can be written as

$$L_E = \sum_{c_i, w_i, h_i} \left\| \phi_G(\mathbf{H})^{c_i, w_i, h_i} - \mathbf{C}^{c_i, w_i, h_i} \right\|_2^2 \tag{5.1}$$

In general, the noise present in an image exhibits high-frequency nature. During image de-noising by using a traditional CNN, the use of Euclidean distance as a cost function may incur the loss of high-frequency details of the image along with the noise removal [107]. As a result, the de-noised images appear to be blurry and degraded. The perceptual loss function proposed by Johnson *et al.* [8] has been used in the majority of the image de-noising and restoration problems [3,108–113] in recent times to overcome this drawback by retaining the high-frequency details of an image. The perceptual cost function is defined as a difference between high-level features of predicted and target images extracted by using a pre-trained CNN. In this case, initial five layers (l) of a pre-trained VGG16 [5] model (V) have been used to extract the features [1]. The perceptual loss function (L_P) can be expressed as

$$L_P = \sum_{l} \sum_{c_i, w_i, h_i} \left\| V_l(\phi_G(\mathbf{H}))^{c_i, w_i, h_i} - V_l(\mathbf{C})^{c_i, w_i, h_i} \right\|_2^2 \tag{5.2}$$

Adversarial training has been beneficial in many of the de-noising tasks [151–154]. In this case, the proposed generator ϕ_G can learn from its adversary ϕ_D based on the adversarial loss for a set of N training samples defined as

$$L_A = -\frac{1}{N} \sum_{i=1}^{N} log\ \phi_D(\phi_G(\mathbf{H})_i) \tag{5.3}$$

Intuitively, the use of Euclidean, Perceptual and Adversarial losses to train the proposed model may have given the visually appealing results. However, it is ob-

[1] https://github.com/pytorch/examples/tree/master/fast_neural_style

served that a few of the estimated dehazed images suffer from halo and checkerboard artifacts. The LoG filters (f) capture these finer details and can be used as a cost function based on the following equation [1].

$$L_{LoG} = \sum_{f,c_i,w_i,h_i} \left\| \mathbf{L}_f(\phi_G(\mathbf{H}))^{c_i,w_i,h_i} - \mathbf{L}_f(\mathbf{C})^{c_i,w_i,h_i} \right\|_2^2 \tag{5.4}$$

Therefore, the aggregated loss to train the proposed generator model ϕ_G can be written as

$$L_{\phi_G} = \lambda_E.L_E + \lambda_A.L_A + \lambda_P.L_P + \lambda_G.L_{LoG} \tag{5.5}$$

where, $\lambda_E, \lambda_A, \lambda_P$ and λ_G are the cost weights. Moreover, the final objective function of the proposed model for single image de-hazing task can be defined as

$$\min_{\phi_G} \max_{\phi_D} \{L_{GAN} + \lambda_E.L_E + \lambda_P.L_P + \lambda_G.L_{LoG}\} \tag{5.6}$$

5.2 Experiments and results

This section illustrates the details of the experimental setup and dataset used for the training and testing of the proposed model. A concise description of image assessment metrics chosen, followed by an ablation study and comparison with the existing methods on both synthetic and real-world hazy images are given.

5.2.1 Datasets and training details

We have chosen the training dataset provided by Zhang *et al.* [10], which consists of 4000 indoor images. In addition, we have also included 45 hazy outdoor images provided by Ancuti *et al.* [128]. During training, following [155], we have augmented the input pairs by using (1) *Random rotation*, (2) *Vertical flip*, (3) *Horizontal flip* and, (4) *Random cropping*. Each data augmentation technique has a probability of 0.5 to be applied to the input pair. Whereas, the input pair will be augmented with the expectation (p_{dat}) of 0.5. Input pair is *rotated*

[1] `Parameters of the laplacian model are not updated when training.`

with the degree randomly chosen between [1°, 359°]. *Cropping* is done with the size $u \times u; u \in [8, 256]$ randomly chosen at a random location in the input pair. For each input pair when training, a randomly selected transformation between {*Horizontal flip, Vertical flip*}, in addition to *Rotation* and *Cropping*, is applied in random order. For testing, we have used synthetic dataset (SOTS) provided by Li *et al.* [129] which consists of 500 outdoor and indoor images. We have also evaluated our proposed work on the benchmark test set[1] provided by Fattal *et al.* [17] and real-world hazy images.

The proposed network is trained on a Nvidia Tesla GPU using the Torch framework [156] for 104 epochs. For training, we have experimentally chosen $\lambda_E = \lambda_A = \lambda_P = \lambda_G = 1$ for the losses in estimating the dehazed image. With the batch size of 5 images, Adam [142] optimization algorithm with a fixed learning rate of 2×10^{-4} has been used when training. The training samples are resized to 496×496.

5.2.2 Evaluation metrics

The proposed scheme has been compared with existing approaches using following 16 full-reference and no-reference image quality metrics: *Full-reference* - *Structural Similarity Index* (SSIM) [15], *Peak Signal to Noise Ratio* (PSNR), *Visual Information Fidelity* (VIF) [116], *Universal-Image-Quality Index* (UQI) [115], *Learned Perceptual Image Patch Similarity* (LPIPS) [117], *Mean Squared Error* (MSE), *Multi-scale Structural Similarity Index* (MS-SSIM) [114], *The Feature Similarity Index* (FSIM) index [119], CIEDE 2000 [120], *Haar Wavelet-based Perceptual Similarity Index* (Haar PSI) [121], *Gradient Magnitude Similarity Deviation* (GMSD) [122] and *Spatial Efficient Entropic Differencing for Image and Video Quality* (SpEED-QA) [157]. *No-reference* - *Total Variation* (TV) [158], *Naturalness Image Quality Evaluator* (NIQE) [123], *BLind Image Integrity Notator using DCT-Statistics* (BLIINDS)[2] [127], and *Blind/Referenceless Image*

[1] http://www.cs.huji.ac.il/~raananf/projects/dehaze_cl/results/
[2] Images are resized to 512×512 to reduce the computation time.

Spatial Quality Evaluator (BRISQUE) [126]. In this chapter, the behaviour of these evaluation norms are described by using following symbols: ▲ (denotes higher is better) and, ▼ (denotes lower is better). The description of incorporated evaluation metrics can be found in Section 2.5.

Measure	Behaviour	Input	DCP [48] TPAMI '11	EIDBR [55] ICCV '13	CAP [56] TIP '15	DEFADE [57] TIP '15	MSCNN [60] ECCV '16	NLD [58] CVPR '16	AOD-Net [61] ICCV '17	DCPDN [10] CVPR '18	PQC [62] TIP '18	PLD [63] ECCV '18	DSIE [64] CVPRW '19	Proposed
SSIM	▲	0.8148	0.8254	0.7665	0.7584	0.8072	0.8389	0.8210	0.9062	0.8722	0.8791	0.8889	0.7580	0.8941
PSNR	▲	15.95	17.33	15.47	18.12	18.89	19.48	17.97	20.43	18.04	19.28	19.45	15.57	20.57
VIF	▲	0.7348	0.5529	0.5766	0.5047	0.5380	0.5492	0.6003	0.6638	0.7502	0.7892	0.8150	0.4969	0.6890
MSE	▼	2.084	1.566	2.850	1.216	1.238	0.822	1.413	0.695	1.279	1.021	0.938	2.062	0.642
UQI	▲	0.7765	0.8581	0.8210	0.7565	0.8314	0.8649	0.8511	0.8962	0.8434	0.8552	0.8653	0.8015	0.8754
LPIPS	▼	0.1038	0.1604	0.1855	0.1348	0.1425	0.1155	0.1465	0.0795	0.0885	0.0751	0.0792	0.2320	0.0728
MS-SSIM	▲	0.9263	0.8859	0.8399	0.8796	0.8948	0.9101	0.8920	0.9435	0.9435	0.9524	0.9521	0.8296	0.9456
TV-Error	▼	0.7802	1.1151	1.7031	0.9753	1.3706	1.0559	1.4894	1.1302	0.8623	1.2920	1.1211	1.3097	1.1720
NIQE	▼	2.9053	2.8033	3.2193	1.5827	2.5663	2.4603	3.1550	2.4240	2.2705	2.7443	2.7819	4.0628	2.1452
FSIM	▲	0.9448	0.9445	0.8769	0.9405	0.9441	0.9598	0.9361	0.9579	0.9635	0.9665	0.9701	0.8830	0.9687
CIEDE 2000	▼	23.80	15.50	18.82	17.88	16.30	14.83	16.04	12.11	17.66	14.61	14.22	21.08	11.96
Haar PSI	▲	0.8616	0.7866	0.6382	0.8219	0.8180	0.8582	0.7633	0.8768	0.8418	0.8804	0.9037	0.6552	0.8752
GMSD	▼	0.0522	0.0643	0.1162	0.0643	0.0561	0.0438	0.0677	0.0462	0.0572	0.0425	0.0333	0.1137	0.0325
BRISQUE	▼	15.79	23.43	27.21	24.18	22.13	22.63	22.32	12.95	23.40	16.06	15.19	17.25	14.67
SpEED-QA	▼	8.66	13.88	28.61	17.84	16.53	15.21	17.33	12.99	11.96	12.00	11.15	25.23	10.17
fom	▲	-	0	0	1	0	0	0	3	0.6	1.8	3	0	5.6

Table 5.1: *Quantitative comparison on the SOTS (Outdoor) dataset. Best and second best results are shown in blue and red colors respectively. A figure of merit (fom) decides the final score as number of (0.6×Best+0.4×Second Best)/Total Metrics. TV-Error is* 10^7.

5.2.3 Ablation study

This sub-section presents an ablation study of the proposed method. We have compared the proposed model with the baseline configurations (**M**-X, where X denotes the proposed model is trained using only loss X) and **M**-NDA refers to the proposed model trained without using adopted data augmentation based on the Eq. 5.6. It can be observed from the Table 5.3 that the inclusion of L_P in addition to L_E and L_A has shown a significant improvement. Further, the use of L_{LoG} has contributed an average improvement of $\sim 1.72\%$, $\sim 3.76\%$ in SSIM and PSNR, respectively, over the model **M**-$L_E + L_A + L_P$. A noticeable increment of $\sim 2.18\%$, $\sim 6.69\%$ in SSIM and PSNR, respectively, is further observed when the data augmentation techniques, summarised in Sub-section 5.2.1, are used during the training.

Measure	Behaviour	Input	DCP [48]	EIDBR [55]	CAP [56]	DEFADE [57]	MSCNN [60]	NLD [58]	AOD-Net [61]	DCPDN [10]	Cycle-Dehaze [67]	MAMF [65]	MS-PPD [159]	PQC [62]	PLD [63]	DSIE [64]	Proposed
NIQE	▼	4.5196	3.6763	5.6602	3.3444	3.6283	4.4219	4.7583	4.1259	5.9101	4.5506	8.6962	5.5536	4.4316	4.5568	8.1091	**4.3041**
BRISQUE	▼	18.49	21.80	22.55	19.11	22.98	20.29	20.19	18.73	20.57	13.23	21.62	19.73	19.78	22.42	21.03	**18.65**
BLINDS II	▲	5.87	7.20	11.34	6.44	10.08	10.27	10.65	9.70	14.00	10.48	12.18	12.18	10.48	6.50	12.27	**13.44**

Table 5.2: *Quantitative results on the Benchmark images provided by [17].*

Baseline	Input	M-L_E	M-L_P	M-$L_E + L_P$	M-$L_E + L_A$	M-$L_E + L_A + L_P$	M-NDA	Proposed
SSIM	0.8148	0.7780	0.8220	0.8321	0.7937	0.8602	0.8750	0.8941
PSNR	15.95	17.78	16.80	17.83	17.49	18.58	19.28	20.57

Table 5.3: *Quantitative comparison of the proposed method with the baseline configurations on the SOTS (Outdoor) test set.*

5.2.4 Comparison with State-of-the-Art Methods

Evaluation on synthetic dataset. Tables 5.1, 5.4, and 5.5 present the quantitative comparison of the proposed scheme with 14 state-of-the-art methods using 15 image quality metrics as mentioned in earlier Sub-section 5.2.2. Based on the proposed figure of merit (fom) in Tables 5.1 and 5.4, it can be observed that the proposed scheme has shown a significant improvement over the existing methods [10, 61–63]. Despite the fact that SSIM value achieved by [61] on SOTS (outdoor) test set is $\sim$ 1.35% higher, the proposed scheme outperforms [61] by a noticeable margin of $\sim$ 86% in overall ranking (fom). One of the important aspect of the single image haze removal problem is color restoration. To evaluate this, we have employed CIEDE which essentially measures the color difference between two images. As reported in Table 5.1, the proposed scheme has outperformed the existing methods [10, 48, 56, 58, 60–63] with the *lowest* CIEDE value of 11.96. Qualitative analysis on outdoor and indoor test sets, as shown in Figures 5.5, 5.7 respectively, proves the supremacy of the proposed scheme over other

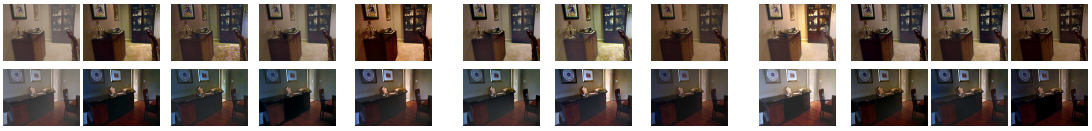


Figure 5.4: *Subjective comparison of the proposed method with the existing state-of-the-art schemes on the SOTS (Indoor) test images.*

Measure	Behaviour	Input	DCP [48] TPAMI '11	EIDBR [55] ICCV '13	CAP [56] TIP '15	DEFADE [57] TIP '15	MSCNN [60] ECCV '16	NLD [58] CVPR '16	AOD-Net [61] ICCV '17	DCPDN [10] CVPR '18	PQC [62] TIP '18	PLD [63] ECCV '18	DSIE [64] CVPRW '19	Proposed
SSIM	▲	0.6942	0.8595	0.7682	0.8171	0.7565	0.7955	0.7775	0.8260	0.7283	0.8513	0.8445	0.6791	0.8598
PSNR	▲	11.97	20.04	16.13	18.97	17.20	17.12	17.29	19.07	13.19	20.25	20.28	13.78	19.68
VIF	▲	0.6858	0.7179	0.6782	0.6468	0.6656	0.7305	0.7808	0.6698	0.7148	0.7785	0.7765	0.4563	0.6945
MSE	▼	4.856	0.776	2.024	0.983	1.531	1.592	1.427	1.144	3.723	0.722	0.728	3.114	0.914
UQI	▲	0.6435	0.8492	0.7681	0.7879	0.7326	0.7577	0.7698	0.8019	0.6774	0.8228	0.8144	0.7029	0.8406
LPIPS	▼	0.1999	0.1033	0.1584	0.1108	0.1449	0.1116	0.1432	0.1014	0.1531	0.0771	0.0779	0.2690	0.1128
MS-SSIM	▲	0.8771	0.9266	0.8767	0.9095	0.8924	0.9269	0.8968	0.9199	0.8915	0.9432	0.9409	0.7858	0.9267
TV-Error	▼	0.4808	0.7683	0.9635	0.6685	0.8218	0.6657	0.9738	0.6626	0.6302	0.8343	0.7430	1.0559	0.8786
NIQE	▼	1.8977	1.2869	1.7616	1.3165	1.6364	1.7327	1.7397	1.6264	2.3240	1.6780	1.7007	3.7639	0.9852
FSIM	▲	0.9112	0.9418	0.8978	0.9309	0.9289	0.9473	0.9203	0.9410	0.9253	0.9569	0.9573	0.8637	0.9613
CIEDE 2000	▼	34.73	11.85	18.76	15.24	18.91	21.39	18.34	16.14	29.00	12.41	13.48	23.36	12.14
Haar PSI	▲	0.7599	0.8313	0.6751	0.8138	0.7764	0.8514	0.7275	0.8336	0.7180	0.8663	0.8797	0.6332	0.8155
GMSD	▼	0.0879	0.0668	0.1129	0.0793	0.0805	0.0656	0.0935	0.0652	0.1134	0.0582	0.0547	0.1398	0.0517
BRISQUE	▼	38.92	34.86	33.48	36.33	33.71	35.44	33.42	34.74	41.74	33.78	34.10	29.87	32.95
SpEED-QA	▼	15.26	15.83	24.14	18.61	17.49	15.16	20.00	14.00	18.46	14.16	12.92	29.54	10.77
fom	▲	-	2	0	0	0	0	0.6	0.4	0.6	3	3.6	0.6	4.2

Table 5.4: *Quantitative comparison on the SOTS (Indoor) dataset. Best and second best results are shown in blue and red colors respectively. A figure of merit (fom) decides the final score as number of (0.6×Best+0.4×Second Best)/Total Metrics. TV-Error is* 10^7.

Outdoor

Measure	Behaviour	Cycle-Dehaze [67]	MAMF [65]	MS-PPD [159]	Proposed
SSIM	▲	0.7850	0.7502	0.8119	0.8941
PSNR	▲	12.93	17.81	17.23	20.57
SpEED-QA	▼	10.64	21.50	14.84	10.17

Indoor

Measure	Behaviour	Cycle-Dehaze [67]	MAMF [65]	MS-PPD [159]	Proposed
SSIM	▲	0.7748	0.7269	0.7687	0.8598
PSNR	▲	17.18	17.16	16.67	19.68
SpEED-QA	▼	16.86	24.36	18.97	10.77

Table 5.5: *Comparison with other existing methods on SOTS.*

methods. Unlike [10, 64, 67, 159], the proposed scheme does not suffer from color degradation. As shown in Figure 5.5(c), results obtained by using [60–62] still contain the hazy part and obscured edgy structures. Whereas, the result obtained by using the proposed scheme is free from such artifacts.

The primary reason behind such improvement may be the use of perceptual loss [8] and the introduced LoG difference as the cost functions. Especially, the LoG loss, which may have improved the efficiency of the proposed model by considering the scale-space of the objects from the initial epoch. The proposed method has also been tested on the *benchmark images* provided by the Fattal *et al.* [17] and results are tabulated in the Table 5.2.

Evaluation on real-world dataset. The proposed model has been evaluated on several real-world hazy images, as shown in Figure 5.6. It can be observed

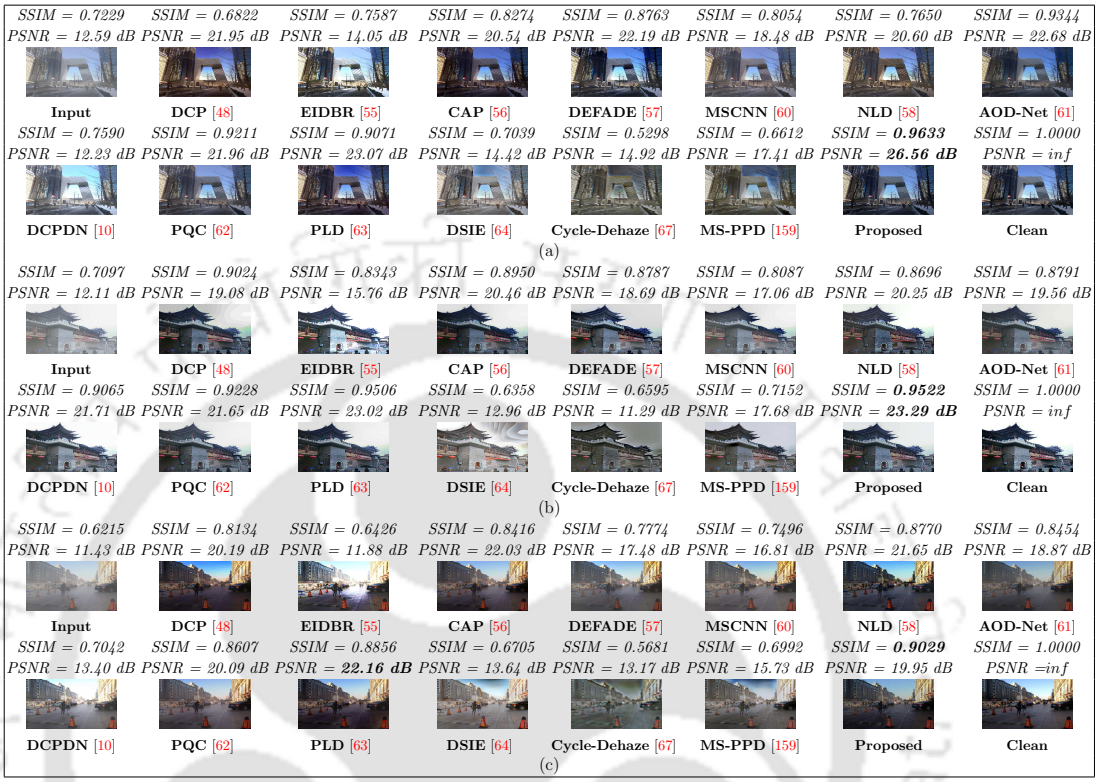


Figure 5.5: *Subjective evaluation of the proposed method with existing schemes in terms of SSIM and PSNR(dB) on SOTS (Outdoor) images.*

Figure 5.6: *Subjective comparison of the proposed model with the existing methods on the real-world hazy images.*

	DCP [48]	EIDBR [55]	CAP [56]	DEFADE [57]	MSCNN [60]	NLD [58]	AOD-Net [61]	DCPDN [10]	PQC [62]	PLD [63]	DSIE [64]	Proposed
Platform	MATLAB [160]	MATLAB [160]	MATLAB [160]	MATLAB [160]	MATLAB [160]	MATLAB [160]	Pycaffe [161]	Torch [156]	Keras [162]	MATLAB [160]	Torch [156]	Torch [156]
Time	16.37	2.64	0.78	34.84	1.71	5.05	0.48	0.13†	29	1.68	6.10‡	0.05

Table 5.6: *Average running time (in seconds) on the test set SOTS (Indoor).* † *Tested with images of size* 512×512. ‡ *On CPU.*

Figure 5.7: *Comparison with the existing schemes on a synthetic hazy image (Indoor).*

that the earlier existing approaches such as [48], zhu tend to under dehaze the given images whereas schemes such as [55, 57, 58] have produced the dehazed images with oversaturated tones. It may be because these methods have used a hand-crafted feature such as dark channel prior, to estimate the haze distribution in the images. As a result, the models may not have generalized well on a variety of hazy images. Recent deep learning based approaches such as [60–63] have been successful compared to the previous models. However, such methods have failed to address the perceptual quality of the dehazed images. The proposed scheme has produced visually appealing results compared to other existing methods. More results are shown in Figures 5.9 5.10, 5.11.

Run-time comparison and failure case. The runtime comparison of the proposed scheme with existing methods has been shown in Table 5.6. It can be observed that the proposed model takes about ~ 0.05 seconds to test an image

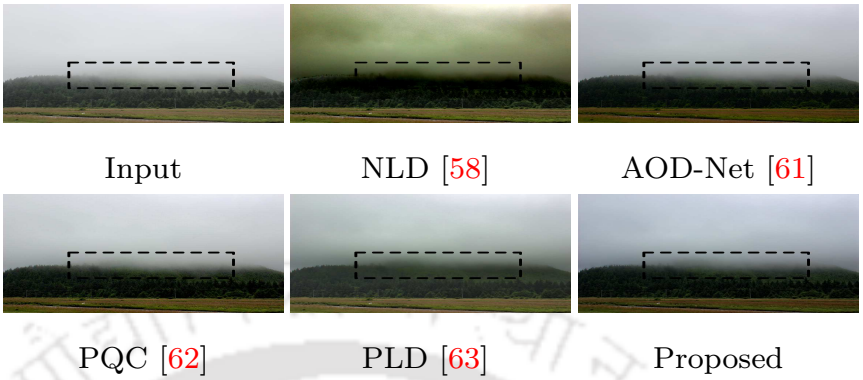


Figure 5.8: ***Failure case****. The proposed model does not perform well on the images with dense haze.*

with an average size of 620 × 460. The proposed method fails to address the images with dense haze, as shown in Figure 5.8. However, the perceptual quality of the dehazed image recovered by using the proposed scheme is better than the same by using the existing methods [58, 61–63].

5.3 Summary

This chapter presents an end-to-end deep learning-based approach for the single image haze removal problem. The proposed scheme is built upon the conditional GAN framework and directly estimates the de-hazed image. We have shown the better preservation of the edgy structures in the LoGs of the hazy images, which inspired us to consider the LoG difference as a cost function. The generalization of the proposed model has been verified by using three benchmark test sets, namely: SOTS (Indoor and Outdoor), Fattal *et al.* [17] and, real-world hazy images. Despite the fact that the proposed model fails to address the images with dense haze, it has been evaluated using 15 image quality assessment metrics, and extensive comparison with existing methods proves its primacy.

As of now, in the three contributory chapters, we have shown the deep learning-based methods for different image restoration tasks. In the next and final contributory chapter, we will show how a GAN-based model can be used for a video restoration task.

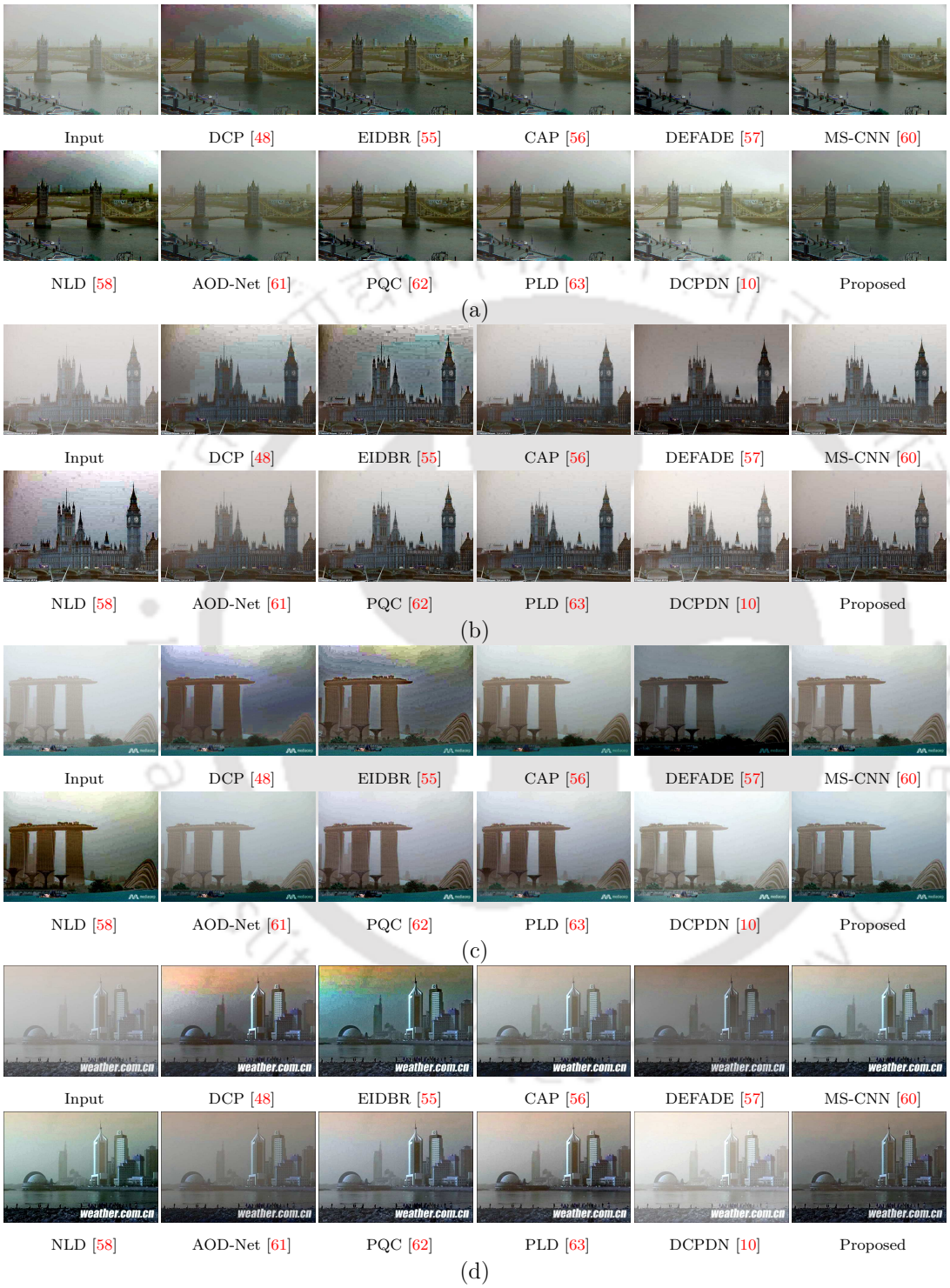


Figure 5.9: *Qualitative comparison of the proposed model with existing schemes on real-world hazy images.*

Figure 5.10: *Qualitative comparison of the proposed model with existing schemes on real-world hazy images.*

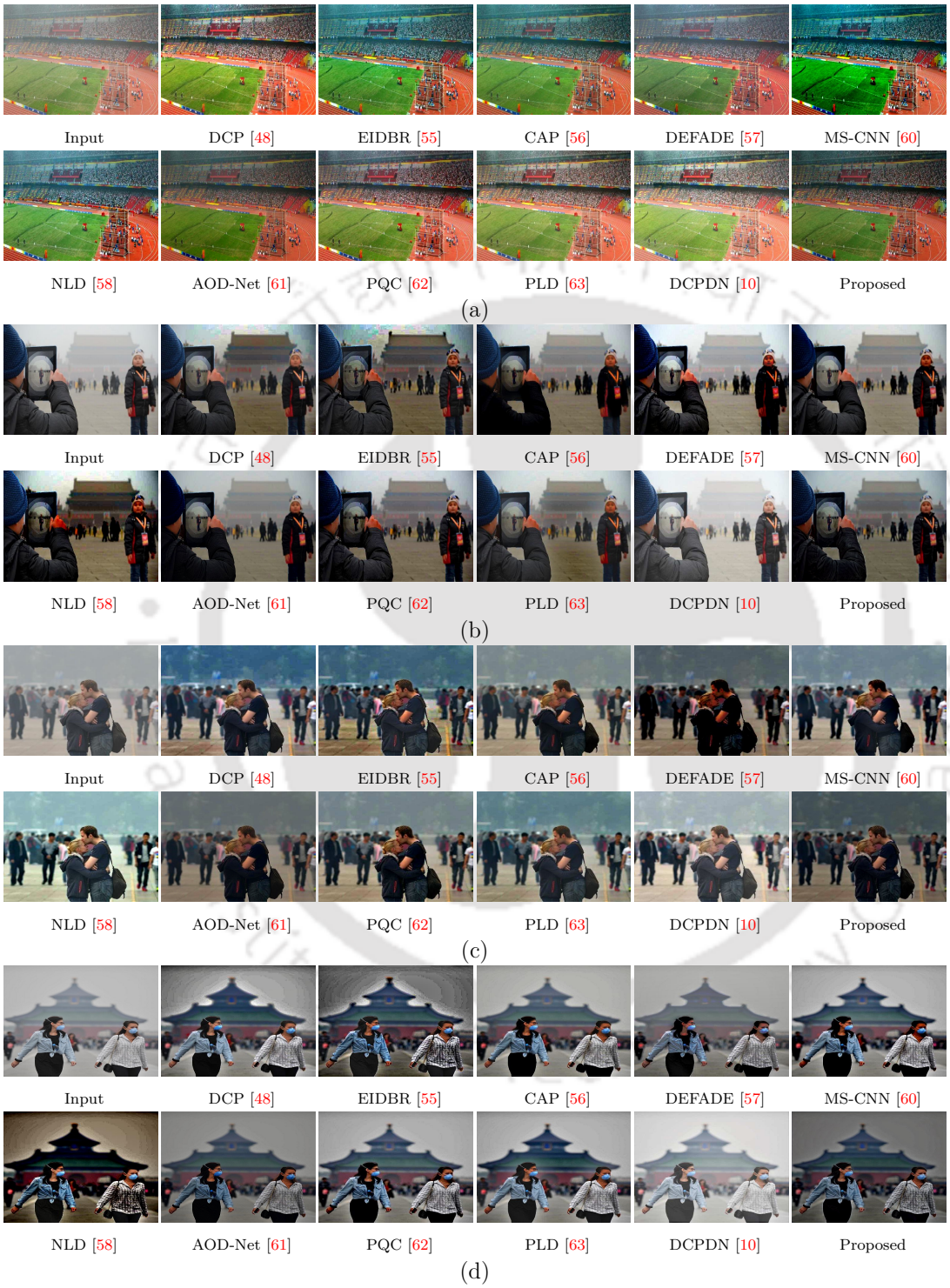


Figure 5.11: *Qualitative comparison of the proposed model with existing schemes on real-world hazy images.*

Chapter 6

Frame-Recurrent Multi-Contextual Adversarial Network for Video De-Raining

In the last three contributory chapters, we have proposed deep learning-based image restoration models. In this chapter, we proposed a deep video de-raining model. Video restoration has an additional complexity of preserving the temporal consistency across the frames in addition to spatial enhancement. Following are the points extending the limitations of existing approaches from Chapter 1, in detail:

- It is observed that a recent method proposed by TCL [4] suffers from the out-of-shape unrealistic transformation of the objects and severe motion blur in the de-rained videos as shown in Figure 6.1. Such methods fail when the displacement of the objects in the frame is large.
- It has also been observed that a few of the existing schemes such as Fast-DerainNet [11] suffer from high visual distortion, such as the visibility of rain-streaks, in de-rained frames with heavy rain-streaks as shown in Figure 6.1.
- It is also evident from the visual inspection that existing methods suffer from the color-saturation problem in the de-rained frames. It may be due to the estimation of de-rained images in the highly correlated RGB color space as shown in Figure 6.2.

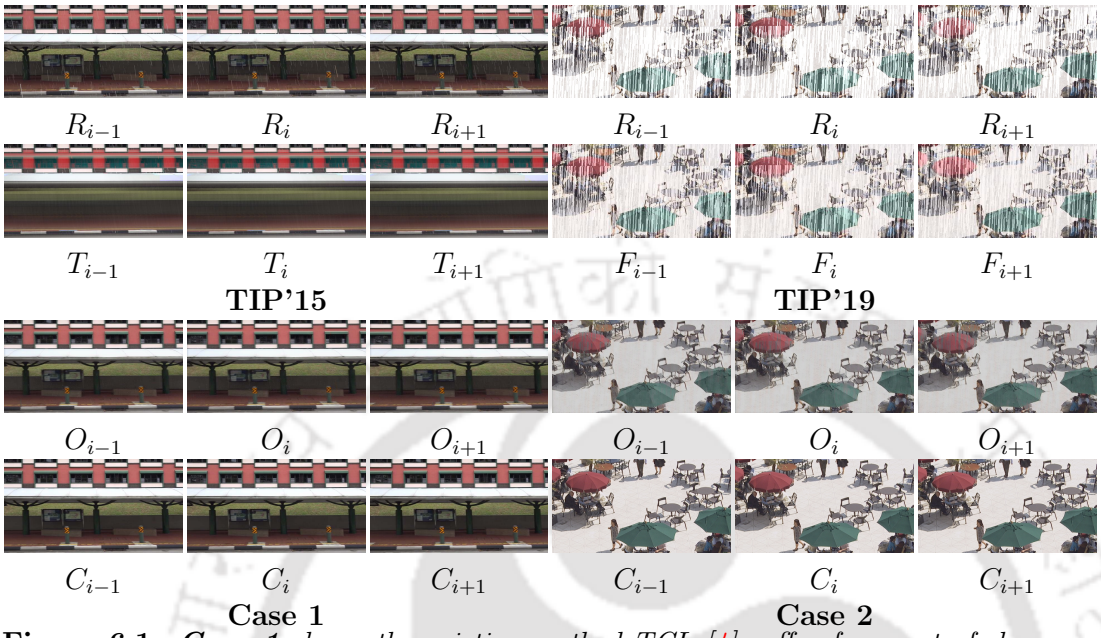


Figure 6.1: ***Case 1*** *shows the existing method TCL [4] suffer from out-of-shape reconstruction and heavy motion blur.* ***Case 2*** *shows the existing method FastDerain [11] suffer from incomplete rain-removal. R, O and C denote the rainy, our result and predicted clean results. T, and F denote TCL [4] and FastDerain [11] results. $i-1$, i and $i+1$ denote three consecutive frames in a video.* `Please zoom the figure for better comparative view.`

- It has been generally observed that the utilization of previously predicted frame to estimate the current frame, boosts the performance of the network in case of the videos in deep CNN's. However, if not carefully crafted, the imprints of previous frames can be seen in the current frame, as shown in Figure 6.2.
- It has also been a major concern to retain the objects that align with the rain-streaks in the frame. A recent method SPAC-CNN [14] suffer from such degradation as shown in Figure 6.3, where the poles bounded in the white-boxes, disappeared in the SPAC-CNN results but retained by the proposed model. It has been observed in various other images which are portrayed in the later sections of this chapter.
- High-frequency loss and heavy distortions are among other observations in the existing methods, as shown in Figure 6.3.

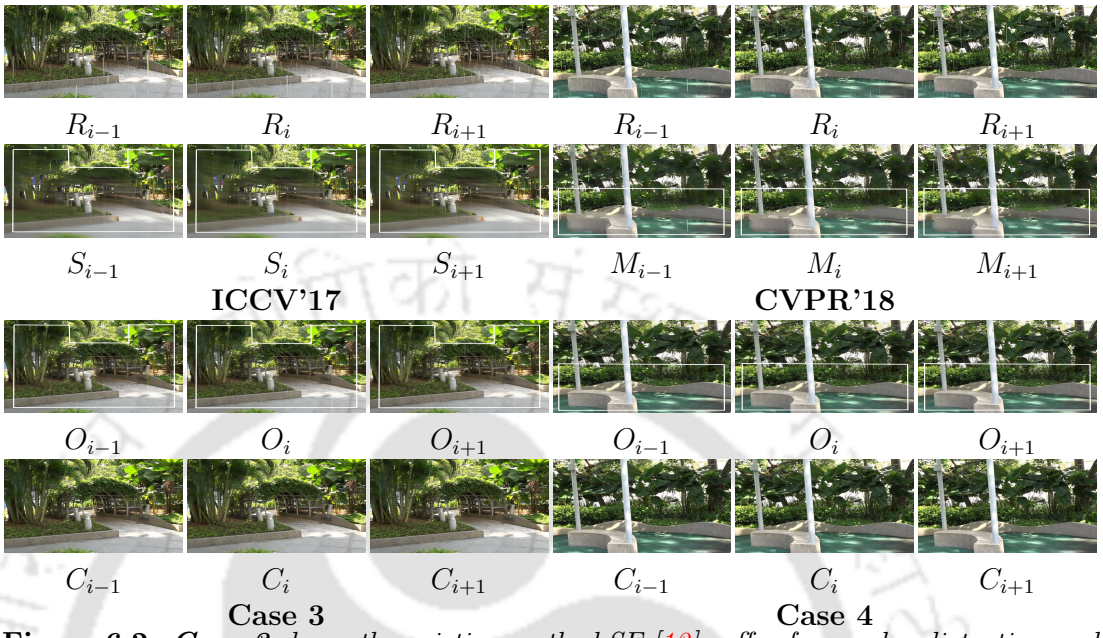


Figure 6.2: ***Case 3*** *shows the existing method SE [12] suffer from color distortion and motion blur.* ***Case 4*** *shows the existing method MSCSC [13] suffer from the artifacts of previous frame. R, O and C denote the rainy, our result and predicted clean results. S, and M denote SE [12] and MSCSC [13] results. $i-1$, i and $i+1$ denote three consecutive frames in a video.* `Please zoom the figure for better view.`

- The existing approaches, due to the separate modules for spatial and temporal enhancement, may require more computational resources.

To summarize, the heavy motion blur, color distortion, object disappearance, imprints from previous frames, and visible rain-streaks are major shortcomings of the existing works. Based on the above observations, the main goal of this work is to propose a unified deep model for handling spatial as well as temporal consistencies inherently and overcome the shortcomings mentioned above. To achieve these goals, following contributions have been made in this chapter:

1. A frame-recurrent conditional GAN [50] based framework has been proposed for the problem of video rain-streak removal, where the generator model directly estimates the de-rained frame by utilizing the previous de-rained frame and its rain-map. The proposed generator model consists of an Encoder-Decoder [7] based frame recurrent model which may help in pre-

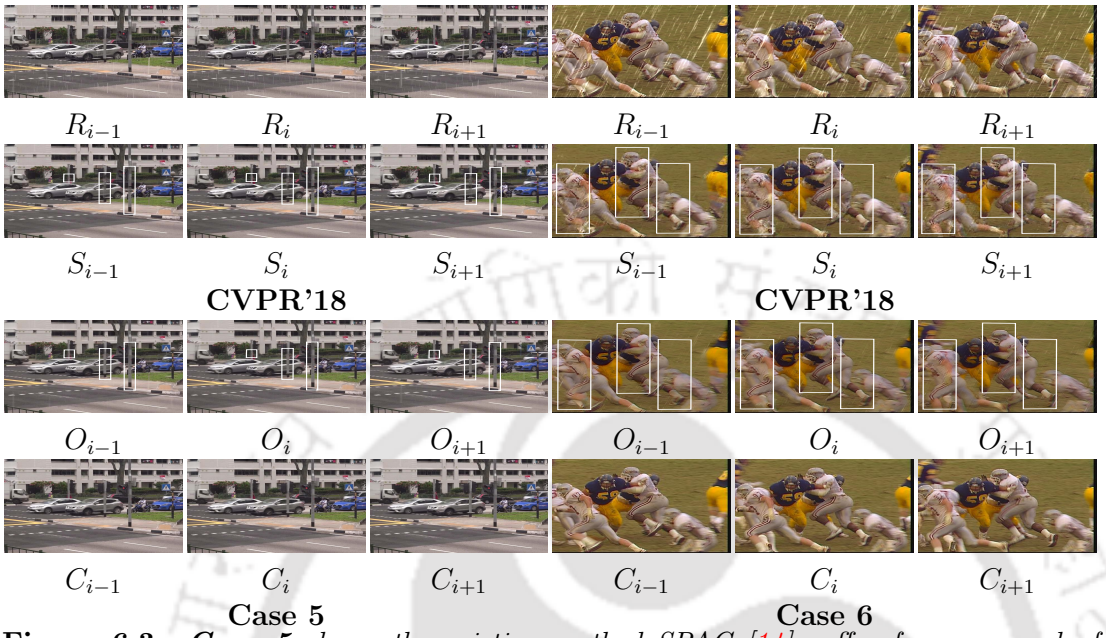


Figure 6.3: ***Case 5*** *shows the existing method SPAC [14] suffer from removal of objects that align with rain streaks.* ***Case 6*** *shows the existing method SPAC [14] suffer from the blurriness. R, O and C denote the rainy, our result and predicted clean results. S denote SPAC [14] results. $i-1$, i and $i+1$ denote three consecutive frames in a video.* `Please zoom the figure for better comparative view.`

serving the temporal consistency of the de-noised videos [163]. The detailed demonstration has been given in Section 6.1.

2. We also propose a 3D Convolution-based Multi-Contextual adversary for the generator model that learns the multi-contextual features from the previously estimated consecutive de-rained frames and helps the generator to estimate the high-resolution de-rained frames.

3. To avoid the loss of high-frequency details in the de-rained frames, we have incorporated the Perceptual loss [8] function in addition to traditional Euclidean distance for the optimization the proposed model. Also, instead of the conventional entropy loss function, we propose to use the Euclidean distance between multi-contextual features of the de-rained and clean frames as the adversarial loss from the proposed discriminator model for the video de-raining task.

4. Following [164], we slowly introduce the perceptual and adversarial loss to the proposed generator that increases exponentially based on the current iteration. It may help the proposed model to avoid the sudden deviations in the learned weights whenever a new loss is introduced. The experimental details re given in Section 6.2

5. A comprehensive comparison of the proposed method with 10 state-of-the-art methods on 11 test-sets using 14 image quality metrics are presented to justify the efficacy of the proposed work. We conclude the performance of the methods based on a proposed figure of merit (`fom`). Section 6.3 presents the quantitative and qualitative results.

6.1 Proposed Methodology

This section first presents the brief extension of GAN in video de-raining followed by the architecture of the proposed model, as shown in Figures 6.4, 6.5, and the cost functions incorporated for the video de-raining task.

6.1.1 Extending GAN for Video De-Raining

The proposed method is built upon the conditional GAN framework [165], which consists of two main sub-modules, namely (a) Generator, and (b) Discriminator, as ϕ_G, ϕ_D respectively in our case. While ϕ_G aims to de-rain the given rainy video, ϕ_D learns to distinguish between the real (actual clean) and fake (de-rained) videos. The proposed ϕ_G is trained from its adversary ϕ_D until a Nash Equilibrium [166] is achieved by playing a 2-player mini-max game based on the following equation

$$\min_{\phi_G} \max_{\phi_D} \mathcal{L}_{\text{GAN}} \tag{6.1}$$

,where ϕ_G, and ϕ_D learn by stochastically descending, and ascending their parameters, respectively. $\mathcal{L}_{\text{GAN}}$ can be written as

$$\begin{aligned} \mathcal{L}_{\text{GAN}} = \mathbb{E}_{\mathbf{V}_{\text{rainy}} \sim \mathbf{P}_{data}(\mathbf{V}_{\text{rainy}})} log(1 - \phi_D(\mathbf{V}_{\text{rainy}}, \phi_G(\mathbf{V}_{\text{rainy}}))) \\ + \mathbb{E}_{\mathbf{V}_{\text{rainy}} \sim \mathbf{P}_{data}(\mathbf{V}_{\text{rainy}}, \mathbf{V}_{\text{clean}})} log(\phi_D(\mathbf{V}_{\text{rainy}}, \mathbf{V}_{\text{clean}})) \end{aligned} \tag{6.2}$$

,where $\mathbf{V}_{\text{rainy}}$, $\mathbf{V}_{\text{clean}}$ are rainy and clean videos, respectively.

As described above, the proposed model comprises of two main sub-modules, Generator (ϕ_G) and Discriminator (ϕ_D). The regimes of operations of ϕ_G and ϕ_D are described as follows:

6.1.2 Network Architecture

It can be observed from Chapter 4 and Lian *et al.* [167] that the pseudo-periodic nature of the rain-streaks can be well described in YCbCr color space. By converting the RGB rainy image into YCbCr color space, the chrominance channels are smoothened, and the majority of the rain-streaks are present in the Y channel only. Therefore, we have processed only on the Y channel of the rainy frames and estimates the Y channel of de-rained frames. The chrominance channels can be re-used to convert the produced result into RGB color-space.

To formally define the problem, let $\mathcal{V}_{\text{Train}}$ be the training set consists of n rainy and clean video pairs ($\mathcal{P}$), where m^{th} pair is denoted as $\mathcal{P}(\mathcal{V}^m_{\text{rainy}}, \mathcal{V}^m_{\text{clean}})$. The m^{th} rainy video from $\mathcal{V}_{\text{Train}}$ is denoted as $\mathcal{V}^m_{\text{rainy}} = \{\mathbf{f}^m_{r,1}, \mathbf{f}^m_{r,2}, \mathbf{f}^m_{r,3},...,\mathbf{f}^m_{r,k}\}$, where $\mathbf{f}^m_{r,i}$ denotes i^{th} rainy frame. Similarly, $\mathcal{V}^m_{\text{clean}} = \{\mathbf{f}^m_{c,1}, \mathbf{f}^m_{c,2}, \mathbf{f}^m_{c,3},...,\mathbf{f}^m_{c,k}\}$, where $\mathbf{f}^m_{c,i}$ denotes i^{th} clean frame. Let $\hat{\mathbf{f}}^m_{c,i}$ denotes the i^{th} frame of m^{th} predicted clean video $\hat{\mathcal{V}}^m_{\text{clean}}$. The estimated rain map $\hat{\mathbf{r}}^m_{map,i}$ from the i^{th} frame of m^{th} predicted clean video can be formulated as

$$\hat{\mathbf{r}}^m_{map,i} = \mathbf{f}^m_{r,i} - \hat{\mathbf{f}}^m_{c,i} \tag{6.3}$$

Let $\mathcal{S}^*_i$, $\mathcal{O}^*_i$ denotes the input and output to/from the proposed generator ϕ_G corresponding to i^{th} frame of the m^{th} video, such that

$$\mathcal{S}^*_i = \hat{\mathbf{f}}^m_{c,i-1} \odot \hat{\mathbf{r}}^m_{map,i-1} \odot \mathbf{f}^m_{r,i-1} \odot \mathbf{f}^m_{r,i} \odot \mathbf{f}^m_{r,i+1} \tag{6.4}$$

,where $\odot$ denotes channel-wise concatenation, and

$$\mathcal{O}^*_i = \hat{\mathbf{f}}^m_{c,i} \odot \hat{\mathbf{r}}^m_{map,i} \tag{6.5}$$

The main objective of incorporating the previously estimated de-rained frames and the rain-streak map is to provide an estimated prior for rain-removal of current and subsequent frames. Whereas, the addition of the next frame as input for current frame de-raining may provide the temporal details and help

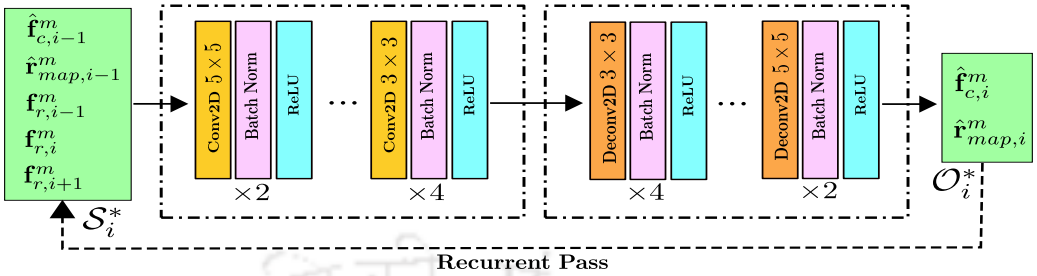


Figure 6.4: *An overview of the architecture of the proposed generator model ϕ_G for video rain-streak removal.*

the proposed model to inherently learn the optical flow between the frames and generate the de-rained video artifacts-free.

Generator (ϕ_G) : The proposed generator model ϕ_G, as shown in Figure 6.4, takes $\mathcal{S}^*_i$ as input the estimates $\mathcal{O}^*_i$ as output for the i^{th} current frame of the rainy video. The proposed ϕ_G consists of an encoder-decoder framework [7], where the encoder part comprises of six two-dimensional convolutional layers with 32 kernels each. While the first two layers have kernel size of 5×5, remaining layers have kernels of size 3×3, each with the stride of 1. The purpose of having different window-sized kernels is to make the proposed model learn the multi-contextual features. Each convolution layer is followed by the BN [134] and the ReLU activation function. The decoder part of the proposed ϕ_G consists of six deconvolutional layers[1]. Each deconvolution layer has 32 kernels except the last, which has 1 filter. While the first-four layers of the decoder part have kernels of size 3×3, the remaining last-two have a window of size 5×5, each of stride 1. It has been observed that U-Net [7] like architecture, has been beneficial in many of the image de-noising and reconstruction tasks [9]. However, to adopt such a model in the video-domain, we have incorporated the frame-recurrent methodology where previously estimated de-rained frames can be used to predict the current de-rained frame.

Discriminator (ϕ_D) : The proposed discriminator model ϕ_D consists of 4 Multi-Contextual Blocks (MCB), as shown in Figure 6.5. The proposed ϕ_D takes either three estimated consecutive de-rained frames ($\hat{\mathbf{f}}^m_{c,i-2} \odot \hat{\mathbf{f}}^m_{c,i-1} \odot \hat{\mathbf{f}}^m_{c,i}$) or three

[1]Transpose Convolution is also known as Deconvolution.

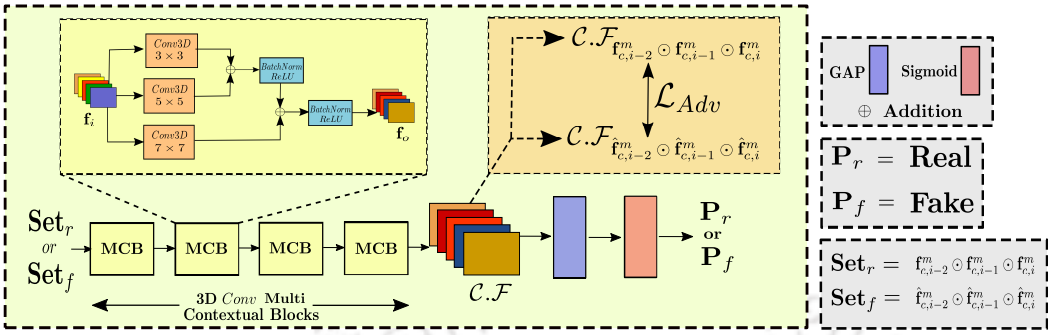


Figure 6.5: *An overview of the architecture of the proposed multi-contextual discriminator ϕ_D.*

clean frames ($\mathbf{f}^m_{c,i-2} \odot \mathbf{f}^m_{c,i-1} \odot \mathbf{f}^m_{c,i}$) as an input and classify them into real or fake derained frames, where $\odot$ denotes channel-wise concatenation. To formally define the proposed MCB block, let $\mathcal{X}$ be either $\{\hat{\mathbf{f}}^m_{c,i-2} \odot \hat{\mathbf{f}}^m_{c,i-1} \odot \hat{\mathbf{f}}^m_{c,i}\}$ or $\{\mathbf{f}^m_{c,i-2} \odot \mathbf{f}^m_{c,i-1} \odot \mathbf{f}^m_{c,i}\}$. Then the output ($\mathcal{Y}$) of the first MCB block in the proposed ϕ_D can be written, with $\oplus$ as addition, as

$$\mathcal{Y} = \mathcal{R}(\text{BN}(\mathcal{R}(\text{BN}(\mathcal{C}_{3D,3}(\mathcal{X}) \oplus \mathcal{C}_{3D,5}(\mathcal{X}))) \oplus \mathcal{C}_{3D,7}(\mathcal{X}))) \tag{6.6}$$

,where $\mathcal{R}$, BN denote ReLU activation function and BN [134], respectively and $\mathcal{C}_{3D,w}$ denote 3D convolution layer with 4 kernels of size $w \times w$. Iteratively, $\mathcal{C}.\mathcal{F}$ can be written as

$$\mathcal{C}.\mathcal{F}(\mathcal{X}) = \text{MCB}_L(\text{MCB}_{L-1}...(\mathcal{X})) \tag{6.7}$$

,for $L = 4$. The purpose of having 3D convolution in the proposed ϕ_D is to learn the temporal consistency among the consecutive three frames $i-2, i-1, i$. The computed multi-contextual feature space $\mathcal{C}.\mathcal{F}$ is then given as input to the Global Average Pooling (GAP) layer, followed by the sigmoid layer. While the proposed ϕ_D is optimized using real ($\mathbf{P}_r$) and fake ($\mathbf{P}_f$) probabilities, the proposed ϕ_G is trained by using the Euclidean distance between $\mathcal{C}.\mathcal{F}(\hat{\mathbf{f}}^m_{c,i-2} \odot \hat{\mathbf{f}}^m_{c,i-1} \odot \hat{\mathbf{f}}^m_{c,i})$ and $\mathcal{C}.\mathcal{F}(\mathbf{f}^m_{c,i-2} \odot \mathbf{f}^m_{c,i-1} \odot \mathbf{f}^m_{c,i})$.

6.1.3 Cost Function

By following the conventions made in the Section 6.1.2, the per-pixel loss ($\mathcal{L}_{MSE}$) between the i^{th} frames of m^{th} de-rained and clean videos, denoted by $\hat{\mathbf{f}}^m_{c,i}$, $\mathbf{f}^m_{c,i}$

respectively, can be written as

$$\mathcal{L}_{MSE} = ||\mathbf{f}_{c,i}^{m} - \hat{\mathbf{f}}_{c,i}^{m}||_2^2 \tag{6.8}$$

However, it becomes necessary to consider the high-frequency nature of the rain-streaks in the frames. The use of only $\mathcal{L}_{MSE}$ loss to optimize the proposed model may incur the loss of high-frequency details of the frames while removing the rain-streaks. Therefore, to ensure the preservation of such details, we have adopted the Perceptual loss function proposed by Johnson *et al.* [168] based on the following equation

$$\mathcal{L}_{Percep} = \sum_{i=1}^{2} ||\mathcal{VGG}_i(\mathbf{f}_{c,i}^{m}) - \mathcal{VGG}_i(\hat{\mathbf{f}}_{c,i}^{m})||_2^2 \tag{6.9}$$

, where $\mathcal{VGG}$ denotes the pre-trained VGG-16 [5] model. The first few layers of the pre-trained VGG-16 network are known to preserve the high-frequency details [9], therefore, we have used the features extracted from the first two layers of the pre-trained VGG-16 model to calculate the $\mathcal{L}_{Percep}$[1].

The adversarial training proposed by Goodfellow *et al.* [165] has been proved to be significantly effective in various image de-noising tasks (see Chapter 4, work 2). However, instead of using entropy loss, we have utilized the L_2 distance between the features of set of clean and de-rained frames, estimated by the proposed ϕ_D as a cost function to optimize ϕ_G. The advantage lies in two-fold : (a) In case of de-noising, the Euclidean distance may be more useful in distinguishing real and fake de-noised images compare to the traditional probabilities, and (b) it may help the proposed ϕ_G in learning the essential features that should be retained to make the de-rained frame look realistic. Therefore, the adversarial loss $\mathcal{L}_{Adv}$ can be written as

$$\mathcal{L}_{Adv} = ||\mathcal{C}.\mathcal{F}(\mathbf{f}_{c,i-2}^{m} \odot \mathbf{f}_{c,i-1}^{m} \odot \mathbf{f}_{c,i}^{m}) - \mathcal{C}.\mathcal{F}(\hat{\mathbf{f}}_{c,i-2}^{m} \odot \hat{\mathbf{f}}_{c,i-1}^{m} \odot \hat{\mathbf{f}}_{c,i}^{m})||_2^2 \tag{6.10}$$

Therefore, the combined loss function used to train the proposed ϕ_G can be written as

$$\mathcal{L}_G = \mathcal{L}_{MSE} + \alpha.\mathcal{L}_{Percep} + \beta.\mathcal{L}_{Adv} \tag{6.11}$$

[1]We have used the chrominance channels while calculating $\mathcal{L}_{Percep}$ and make sure the proper flow of gradients.

,where α and β are weight constants that non-linearly increases based on the current epoch(e) and total epochs(E) based on the following equation as

$$\alpha/\beta = \kappa.\frac{\omega + d * e^2}{E} \tag{6.12}$$

,where κ, ω and d are fixed constants. Finally, the objective function ($\mathcal{OF}$) of the proposed ϕ_G can be written as

$$\mathcal{OF}_{\phi_G} = arg \min_{\phi_G} \max_{\phi_D} \quad \beta.\mathcal{L}_{adv}(\phi_G;\phi_D) + \mathcal{L}_{MSE} + \alpha.\mathcal{L}_{Percep} \tag{6.13}$$

6.2 Experiments & Training Details

This section first presents the details of training and testing datasets followed by the training parameters. Later, we also describe the evaluation metrics used for fair comparison of proposed model with existing schemes.

6.2.1 Dataset

The dataset used for this work has been described in Section 2.6.

6.2.2 Training Parameters

The proposed network is trained on a Nvidia Tesla P100 16 GB GPU using the PyTorch [169] framework for about 500 epochs with batch size of 1. We have experimentally set the values of κ, ω, d and E as $1e-4$, 1, 0.99 and 10^5, respectively. We have used Adam optimization [142] with standard values of parameters β_1, β_2, and weight decays of ϕ_G and ϕ_D as 0.5, 0.999, 5×10^5, and 0, respectively. The learning rates of both ϕ_G and ϕ_D are initialized with 1e-4 and experimentally varied with epochs.

6.2.3 Evaluation Metrics

We have compared the proposed model with existing approaches using following 14 image quality metrics, namely: *Structural Similarity Index* (SSIM) [15], *Peak Signal to Noise Ratio* (PSNR), *Visual Information Fidelity* (VIF) [116], *Mean Squared Error* (MSE), *Learned Perceptual Image Patch Similarity* (LPIPS)

Behaviour	SSIM	PSNR	VIF	MSE	LPIPS	UQI	MS-SSIM	NIQE	PIQE	FSIM	Haar PSI	GMSD	BRISQUE	TV-Error
LB	✗	✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓	✓
HB	✓	✓	✓	✗	✗	✓	✓	✗	✗	✓	✓	✗	✗	✗

Table 6.1: *Image quality metrics behavior.*

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.8272	0.9622	0.9051	0.9054	0.7093	0.8635	0.8482	0.8828	0.8702	0.8010	0.9124	0.9239	0.9260
PSNR	24.89	35.80	30.53	31.13	23.95	26.74	27.58	29.37	28.62	26.50	30.91	31.92	31.81
VIF	0.6650	NA	0.5959	0.5462	0.2986	0.5573	0.5652	0.5644	0.5582	0.5536	0.6568	0.6607	0.6616
MSE	245.8	NA	67.30	61.91	284.7	156.8	123.8	94.98	99.52	104.8	62.75	62.86	61.29
LPIPS	0.2658	NA	0.1050	0.1014	0.3184	0.2032	0.2095	0.1394	0.2103	0.2684	0.1100	0.0982	0.0943
UQI	0.9877	NA	0.9970	0.9981	0.9834	0.9936	0.9946	0.9962	0.9936	0.9867	0.9967	0.9928	0.9956
MS-SSIM	0.8171	NA	0.9407	0.9434	0.7735	0.8888	0.8846	0.9238	0.8943	0.8173	0.9411	0.9455	0.9466
NIQE	5.209	NA	3.352	3.172	4.393	3.813	4.311	3.672	3.528	4.012	3.342	3.761	3.652
PIQE	35.80	NA	30.22	33.32	39.37	34.55	32.87	32.87	31.32	36.99	31.74	27.84	27.56
FSIM	0.8675	NA	0.9413	0.9437	0.8173	0.9060	0.9055	0.9284	0.9001	0.8636	0.9449	0.9500	0.9509
Haar PSI	0.5271	NA	0.7460	0.7783	0.4809	0.6381	0.6550	0.7299	0.6527	0.5358	0.7555	0.7861	0.7840
GMSD	0.1990	NA	0.0832	0.0760	0.1889	0.1333	0.1391	0.0956	0.1279	0.1672	0.0877	0.0757	0.0729
BRISQUE	31.40	NA	25.91	28.82	28.33	25.91	28.97	26.11	25.06	30.03	25.17	22.75	25.66
TV-Error	1.499	NA	1.162	1.051	1.194	1.354	1.293	1.139	1.182	1.197	1.232	1.160	1.157

Table 6.2: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the* **Test Set Light**. *Best and second best results are shown in red, blue colors, respectively.*

[117], *Universal-Image-Quality Index* (UQI) [115], *Multi-scale Structural Similarity Index* (MS-SSIM) [114], *Naturalness Image Quality Evaluator* (NIQE) [123], *Perception based Image Quality Evaluator* (PIQE) [124], *The Feature Similarity Index* (FSIM) Index [119], *Haar Wavelet-based Perceptual Similarity Index* (Haar PSI), *Gradient Magnitude Similarity Deviation* (GMSD) [122], *Blind/Referenceless Image Spatial Quality Evaluator* (BRISQUE) [126], and *Total Variation* (TV) error. The description of incorporated evaluation metrics can be found in Section 2.5. For fair comparison, we have decided a figure of merit (`fom`) based on the performances on all adopted test sets as, `fom` = {`0.6 * No. of Best + 0.4 * No. of Second Best`}/{`Total Metrics`}, where "`No. of Best`" and "`No. of Second Best`" denote the count of best and second best entries among all metrics for a particular test set. A slight high priority has been given to the best entry over the second best.

The behaviour of each adopted evaluation metric is described in Table 6.1 by using "HB" & "LB" ,where LB denotes "Lower value is better", and HB denotes "Higher value is better".

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.4983	0.8239	0.7463	0.5929	0.4766	0.5638	0.5316	0.7007	0.4955	0.5300	0.6331	0.8185	0.7983
PSNR	15.89	27.72	23.60	20.98	16.28	18.26	18.19	23.06	17.28	16.80	20.15	24.35	23.96
VIF	0.4027	NA	0.3959	0.2946	0.2428	0.3603	0.3657	0.3740	0.3243	0.3482	0.4000	0.3969	0.3769
MSE	3131	NA	804.5	1315	3009	2382	2185	626.2	2355	2964	2146	370.9	440.3
LPIPS	0.5008	NA	0.2692	0.3679	0.4989	0.4373	0.4502	0.3102	0.4284	0.4374	0.3240	0.2163	0.2320
UQI	0.8693	NA	0.9645	0.9321	0.8720	0.9065	0.8954	0.9507	0.8996	0.8993	0.9028	0.9660	0.9576
MS-SSIM	0.5177	NA	0.7698	0.6454	0.5310	0.5765	0.5879	0.7532	0.5690	0.5724	0.6781	0.8394	0.8159
NIQE	10.79	NA	3.698	4.631	9.100	8.119	8.469	4.328	6.720	5.982	5.487	3.756	3.806
PIQE	57.89	NA	31.92	35.39	51.53	48.11	47.13	34.47	37.98	38.16	42.85	26.45	27.47
FSIM	0.6785	NA	0.8326	0.7695	0.6765	0.7179	0.7226	0.8250	0.7375	0.7381	0.7452	0.8754	0.8614
Haar PSI	0.2874	NA	0.5076	0.4416	0.3012	0.3520	0.3585	0.5099	0.4376	0.4581	0.4022	0.5843	0.5548
GMSD	0.2865	NA	0.1626	0.2029	0.2685	0.2535	0.2523	0.1682	0.2282	0.2339	0.2310	0.1223	0.1329
BRISQUE	40.58	NA	30.64	34.51	36.53	36.47	39.14	32.27	35.01	38.91	35.53	25.47	27.40
TV-Error	2.498	NA	1.429	1.976	2.369	3.164	2.264	1.428	2.078	2.103	1.956	0.9569	0.9358

Table 6.3: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the* **Test Set Heavy**. *Best and second best results are shown in red, blue colors, respectively.*

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.9361	NA	0.9208	0.9542	0.8865	0.9031	0.9483	0.9427	0.9356	0.8971	0.9384	0.9589	0.9564
PSNR	32.98	NA	30.42	34.74	31.56	30.57	30.16	33.27	33.96	29.56	33.28	34.49	34.54
VIF	0.7482	NA	0.5944	0.6792	0.5579	0.6335	0.6951	0.6718	0.7501	0.6891	0.7482	0.7141	0.7253
MSE	40.21	NA	76.17	26.90	61.42	38.92	69.56	36.71	29.50	100.5	40.21	33.35	32.54
LPIPS	0.0832	NA	0.778	0.0362	0.0861	0.2059	0.1018	0.0434	0.0458	0.1622	0.0564	0.0356	0.0396
UQI	0.9982	NA	0.9960	0.9995	0.9976	0.9884	0.9956	0.9993	0.9980	0.9663	0.9982	0.9971	0.9984
MS-SSIM	0.9663	NA	0.9495	0.9821	0.9386	0.9290	0.9711	0.9787	0.9724	0.9358	0.9619	0.9831	0.9807
NIQE	5.071	NA	4.090	4.354	4.250	4.834	4.562	4.267	4.891	4.981	5.071	4.463	4.700
PIQE	46.04	NA	37.46	41.87	46.53	45.79	43.71	40.33	42.66	45.44	46.04	41.04	42.38
FSIM	0.9566	NA	0.9352	0.9651	0.9268	0.9163	0.9629	0.9571	0.9662	0.9356	0.9566	0.9670	0.9671
Haar PSI	0.8305	NA	0.7756	0.8872	0.7776	0.6790	0.8472	0.8658	0.8792	0.7940	0.8305	0.8674	0.8684
GMSD	0.0704	NA	0.0763	0.0355	0.0785	0.1048	0.0501	0.0471	0.0372	0.0772	0.0704	0.0436	0.0419
BRISQUE	27.37	NA	26.66	30.09	30.25	25.65	26.59	27.57	28.02	26.71	27.37	24.62	25.52
TV-Error	0.6826	NA	0.6153	0.6124	0.5929	0.7748	0.6323	0.6190	0.5387	0.5628	0.6337	0.6332	0.6409

Table 6.4: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the* **Test Set 1**. *Best and second best results are shown in red, blue colors, respectively.*

6.3 Results

6.3.1 Baseline Configurations

We have also compared our proposed method with its following baseline configurations.

- **Proposed** : The proposed model is trained with $\mathcal{L}_{MSE} + \alpha.\mathcal{L}_{Percep}$ loss.
- **Temporal** : The proposed model is trained with $\mathcal{L}_{MSE} + \alpha.\mathcal{L}_{Percep} + \beta.\mathcal{L}_{Adv}$ loss.

6.3.2 Quantitative Results

In this sub-section, we first compare the proposed model with existing methods, followed by the baseline configurations, quantitatively. The quantitative comparison of the proposed model with state-of-the-art video de-raining methods on **Test Set Light** is shown in Table 6.2. **Test Set Light** consists of videos with light-density synthetic rain streaks. It can be observed that the proposed model (**Temporal**) optimized using $\mathcal{L}_G$ loss has achieved a significant improvement of $\sim$ 9.17%, $\sim$ 15.33% in terms of SSIM and PSNR, respectively, over the recent FastDerain [11] method. Also, a significant reduction of $\sim$ 55.04% in the LPIPS metric over FastDerain [11] method has been observed. It favours our case in terms of the perceptual quality of the de-rained frame. It can also be observed that the proposed model has surpassed the existing single image rain-streak removal methods in terms of almost all adopted evaluation metrics. The proposed model has also surpassed the SPAC-CNN [14] framework by $\sim$ 2.27% in SSIM, $\sim$ 21.12% in VIF, and $\sim$ 2.18% in PSNR. Similarly, over J4RNet [76], a significant improvement of $\sim$ 2.30% in SSIM, and $\sim$ 4.19% in PSNR can be observed. However, the results obtained by the DualFlow [77] method on both **Test Set Light** and **Test Set Heavy** are significantly better than the proposed model.

The **Test Set Heavy** consists of the rainy videos with heavy density rain-streaks. The detailed comparison of the proposed model with existing schemes on **Test Set Heavy** dataset is shown in Table 6.3. It can be observed that the proposed model **Temporal** has shown a remarkable improvement of $\sim$ 50.16% in SSIM, $\sim$ 31.72% in PSNR, and $\sim$ 55% in NIQE over the existing video de-raining method FastDerain [11]. It can also be observed that the proposed model has outperformed the SPAC-CNN [14] method with a significant rise of $\sim$ 34.64% in SSIM, $\sim$ 14.20% in PSNR, and $\sim$ 52.64% in TV-Error. The TV-Error value describes the amount of noise present in an image, which is the de-rained image in this case. Following the trend on light-density rain-streaks, in this case too, the proposed model has outperformed the single image de-raining methods. The single image-based methods do not take temporal data such as previous or next frames into considerations, thus may suffer from overall video consistency artifacts. The proposed model has also outperformed the multi-scale CNN

based video de-raining method [13] by $\sim$ 67.49% in SSIM, $\sim$ 47.17% in PSNR, $\sim$ 53.49% in LPIPS, and $\sim$ 60.49% in TV-Error. However, on **Test Set 1**, a comparable performance has been observed between the proposed model **Proposed** and SPAC-CNN [14], as shown in Table 6.4. In case of **Test Set Light**, the proposed method **Temporal** has outperformed its baselines.

As described in Section 6.2.1, test-sets provided in SPAC-CNN [14], the *group a's* consist of videos shot with a panning and unstable camera, and *group b's* are shot using a fast-moving camera (with speed ranging between 20 to 30 kmph). The quantitative comparison of the proposed scheme with existing methods on test-set a_1 is shown in Table. 6.5. Note that the proposed model has shown a noticeable improvement of $\sim$ 0.99% in terms of SSIM over J4RNet [76] and JORDER [51]. Whereas, a remarkable improvement of $\sim$ 41.84% in SSIM and $\sim$ 27.13% in PSNR can be observed over the recent method MS-CSC [13]. In the case of videos, especially, the amount of information that can be extracted by the HVS directly shows the performance of the de-noising models. A significant rise of $\sim$ 97.69% in VIF value can be observed by the proposed model over the existing method MS-CSC [13]. Similar to a_1, a minor improvement of $\sim$ 2.72% in SSIM and $\sim$ 0.28% in PSNR has been observed over SPAC-CNN [14] in the case of a_2, as shown in Table. 6.6. In the rest of the evaluation metric cases, SPAC-CNN [14] has shown the best performances, whereas the proposed model has shown the second-best. We have also observed the remarkable performance by the proposed model over J4RNet [76] by $\sim$ 5.02% in SSIM and $\sim$ 14.36% in PSNR.

A few more similar statistics can also be seen over MS-CSC [13] by $\sim$ 30.69%, and $\sim$ 25.17%, and over FastDerain [11] by $\sim$ 4.60%, $\sim$ 5.20% in SSIM, and PSNR, respectively.

With such results on panning and unstable rainy videos in a_1 and a_2, it still cannot be concluded that the proposed model may not perform well on such inputs. In support, a clear dominance of the proposed model over all existing methods on almost every adopted evaluation metric can be observed in Tables. 6.7, and 6.8.

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.9086	NA	0.9319	0.9224	0.6564	0.8850	0.9188	0.9153	0.9110	0.8582	0.9319	0.9412	0.9311
PSNR	29.58	NA	30.04	30.04	23.51	26.07	29.44	29.25	28.82	26.23	30.63	30.05	29.89
VIF	0.7221	NA	0.6224	0.6319	0.3076	0.5382	0.6951	0.6258	0.5811	0.4711	0.6652	0.6300	0.6081
MSE	70.04	NA	64.52	63.12	293.1	160.9	69.56	70.20	79.29	203.7	55.01	71.51	78.31
LPIPS	0.0868	NA	0.0403	0.0423	0.2381	0.1537	0.0550	0.0547	0.0453	0.2154	0.413	0.0420	0.502
UQI	0.9969	NA	0.9963	0.9992	0.9832	0.9910	0.9956	0.9978	0.9971	0.9899	0.9980	0.9982	0.9976
MS-SSIM	0.9636	NA	0.9771	0.9805	0.7974	0.9598	0.9696	0.9574	0.9781	0.8496	0.9760	0.9817	0.9768
NIQE	2.483	NA	2.077	1.833	2.242	2.361	4.562	2.250	2.569	2.212	1.9816	2.250	2.360
PIQE	27.73	NA	23.40	23.05	31.25	26.31	43.71	24.77	25.77	33.30	25.66	19.97	20.37
FSIM	0.9733	NA	0.9711	0.9850	0.8644	0.9609	0.9629	0.9666	9612	0.9054	0.9727	0.9729	0.9675
Haar PSI	0.8183	NA	0.8331	0.8846	0.5303	0.7755	0.8472	0.8243	8331	0.6349	0.8485	0.8440	0.8260
GMSD	0.0984	NA	0.0582	0.0363	0.1864	0.0877	0.0501	0.0725	0592	0.1323	0.0670	0.0542	0.0603
BRISQUE	13.96	NA	16.61	20.24	26.15	16.74	26.59	16.85	20.95	18.48	21.62	18.84	17.52
TV-Error	2.421	NA	2.251	2.074	1.747	2.529	2.251	2.151	2.186	1.640	2.346	2.065	2.025

Table 6.5: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the a_1 test set. Best and second best results are shown in red, blue colors, respectively.*

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.9246	NA	0.9081	0.9284	0.7297	0.9050	0.9117	0.9031	0.9016	0.8902	0.9262	0.9537	0.9339
PSNR	29.20	NA	27.22	31.04	24.87	26.60	29.59	28.98	27.11	26.68	29.85	31.13	30.61
VIF	0.6275	NA	0.5256	0.5924	0.2766	0.5594	0.5766	0.5360	0.5581	0.4612	0.5880	0.5872	0.5781
MSE	76.82	NA	123.3	50.30	204.6	142.34	62.97	92.26	72.99	202.2	67.46	59.99	61.23
LPIPS	0.0838	NA	0.0764	0.0424	0.1901	0.0982	0.0639	0.0654	0.0540	0.3587	0.0579	0.0458	0.0458
UQI	0.9984	NA	0.9952	0.9994	0.9962	0.9931	0.9991	0.9988	0.9989	0.9963	0.9989	0.9996	0.9991
MS-SSIM	0.9700	NA	0.9567	0.9823	0.8535	0.9593	0.9735	0.9627	0.9733	0.7401	0.9717	0.9802	0.9790
NIQE	3.456	NA	2.572	2.325	3.113	2.750	3.424	2.758	3.267	2.899	2.989	2.659	3.093
PIQE	41.10	NA	34.53	38.56	37.80	38.12	38.98	36.10	39.70	39.39	38.81	35.31	36.77
FSIM	0.9616	NA	0.9504	0.9740	0.8655	0.9513	0.9647	0.9540	0.9613	0.9281	0.9645	0.9746	0.9683
Haar PSI	0.8048	NA	0.7526	0.8710	0.5472	0.7671	0.8219	0.7760	0.8059	0.7028	0.8068	0.8230	0.8050
GMSD	0.0849	NA	0.0864	0.0421	0.1668	0.0952	0.0704	0.0841	0.0739	0.1201	0.0746	0.0642	0.0602
BRISQUE	34.92	NA	26.89	32.39	24.46	29.52	28.61	25.89	30.70	29.91	30.74	28.63	28.06
TV-Error	1.722	NA	1.520	1.438	1.313	1.655	1.618	1.460	1.707	0.980	1.653	1.428	1.479

Table 6.6: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the a_2 test set. Best and second best results are shown in red, blue colors, respectively.*

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.8926	NA	0.9111	0.9118	0.6078	0.8546	0.9054	0.8972	0.8991	0.8781	0.9130	0.9349	0.9218
PSNR	28.64	NA	28.58	30.02	24.52	25.65	30.34	29.60	28.27	27.03	29.98	31.31	30.54
VIF	0.6651	NA	0.5744	0.6115	0.2402	0.4888	0.5943	0.5703	0.6161	0.5022	0.6125	0.6195	0.5927
MSE	80.84	NA	90.24	62.06	223.9	169.2	59.79	87.47	78.18	200.3	66.97	53.28	61.71
LPIPS	0.1189	NA	0.0652	0.0559	0.2871	0.1903	0.0657	0.0671	0.0748	0.3415	0.0673	0.0493	0.0603
UQI	0.9963	NA	0.9936	0.9988	0.9899	0.9887	0.9982	0.9984	0.9974	0.9899	0.9975	0.9989	0.9979
MS-SSIM	0.9464	NA	0.9562	0.9776	0.7844	0.9326	0.9648	0.9617	0.9594	0.7472	0.9594	0.9783	0.9677
NIQE	2.721	NA	2.196	1.955	2.314	2.392	2.183	2.125	2.213	2.147	1.982	1.894	2.281
PIQE	26.77	NA	22.95	21.09	34.24	25.66	24.88	23.03	26.10	30.61	25.44	19.59	20.59
FSIM	0.9617	NA	0.9613	0.9836	0.8378	0.9458	0.9688	0.9635	0.9629	0.9134	0.9673	0.9756	0.9701
Haar PSI	0.7721	NA	0.7748	0.8755	0.4953	0.7238	0.8191	0.8063	0.7982	0.7005	0.8002	0.8498	0.8280
GMSD	0.1029	NA	0.0753	0.0348	0.1824	0.0972	0.0688	0.0659	0.0831	0.1298	0.0799	0.0513	0.0592
BRISQUE	24.46	NA	13.72	21.05	27.16	15.45	13.58	15.66	18.10	25.22	6.870	5.037	2.723
TV-Error	2.373	NA	2.157	1.942	1.495	2.419	2.153	2.088	2.423	1.627	2.287	2.020	2.019

Table 6.7: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the a_3 test set. Best and second best results are shown in red, blue colors, respectively.*

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.9352	NA	0.9600	0.9607	0.8488	0.9514	0.9657	0.9686	0.9580	0.9026	0.9608	0.9776	0.9732
PSNR	32.29	NA	33.88	34.75	28.87	31.37	35.04	35.23	32.71	30.57	34.61	38.25	37.04
VIF	0.7931	NA	0.7136	0.6934	0.4290	0.7048	0.7658	0.7607	0.7163	0.6620	0.7755	0.8001	0.7831
MSE	35.85	NA	26.64	20.98	79.87	47.56	18.51	18.34	30.32	69.92	22.24	11.41	13.53
LPIPS	0.0993	NA	0.0270	0.0249	0.1049	0.0648	0.0279	0.0206	0.0311	0.1821	0.0344	0.0146	0.0168
UQI	0.9985	NA	0.9987	0.9995	0.9975	0.9970	0.9994	0.9975	0.9992	0.9970	0.9992	0.9997	0.9995
MS-SSIM	0.9741	NA	0.9863	0.9905	0.9450	0.9823	0.9909	0.9913	0.9887	0.8853	0.9867	0.9933	0.9922
NIQE	3.136	NA	2.951	2.941	3.113	3.194	2.898	2.963	3.102	2.280	2.875	2.905	2.918
PIQE	48.50	NA	47.84	51.51	54.73	50.23	49.25	50.38	50.14	55.49	48.73	45.60	44.58
FSIM	0.9745	NA	0.9799	0.9863	0.9314	0.9775	0.9861	0.9875	0.9711	0.9247	0.9833	0.9913	0.9902
Haar PSI	0.8396	NA	0.8757	0.9070	0.7076	0.8718	0.9123	0.9133	0.8888	0.7813	0.8907	0.9418	0.9328
GMSD	0.0848	NA	0.0461	0.0339	0.1094	0.0567	0.0415	0.0348	0.0501	0.0927	0.0543	0.0348	0.0276
BRISQUE	30.55	NA	21.31	21.50	28.23	14.80	15.06	18.19	12.96	28.33	25.34	19.89	29.96
TV-Error	1.366	NA	1.244	1.186	1.102	1.318	1.260	1.221	1.273	1.056	1.299	1.221	1.219

Table 6.8: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the a_4 test set. Best and second best results are shown in red, blue colors, respectively.*

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.8966	NA	0.9438	0.9368	0.7361	0.9275	0.9093	0.9169	0.9056	0.8823	0.9322	0.9460	0.9353
PSNR	29.98	NA	31.92	30.98	24.13	28.71	30.10	28.88	29.18	27.97	31.01	31.48	30.78
VIF	0.6693	NA	0.6219	0.5790	0.2855	0.5906	0.5569	0.5492	0.5065	0.4922	0.6667	0.6221	0.5842
MSE	65.44	NA	43.18	47.11	247.5	88.14	64.31	96.86	80.90	78.12	37.34	47.91	56.83
LPIPS	0.1741	NA	0.0526	0.0474	0.2467	0.0880	0.1170	0.0845	0.0873	0.3050	0.0699	0.0591	0.0790
UQI	0.9975	NA	0.9982	0.9991	0.9832	0.9939	0.9979	0.9973	0.9984	0.9953	0.9989	0.9989	0.9985
MS-SSIM	0.9466	NA	0.9719	0.9781	0.7577	0.9648	0.9532	0.9501	0.9589	0.7467	0.9741	0.9784	0.9695
NIQE	3.145	NA	2.178	2.391	2.568	2.495	2.666	2.317	2.250	2.399	2.230	2.545	2.896
PIQE	36.31	NA	30.71	35.11	40.40	34.45	31.39	31.77	27.95	35.23	33.66	29.47	28.77
FSIM	0.9619	NA	0.9730	0.9769	0.8302	0.9654	0.9589	0.9511	0.9568	0.9332	0.9767	0.9745	0.9669
Haar PSI	0.7696	NA	0.8305	0.8510	0.4753	0.8069	0.7873	0.7536	0.7521	0.7349	0.8440	0.8311	0.8063
GMSD	0.1013	NA	0.0611	0.0510	0.2023	0.0713	0.0769	0.0812	0.0818	0.0918	0.0617	0.0560	0.0612
BRISQUE	34.74	NA	24.26	33.24	25.15	28.87	29.02	28.54	31.91	32.51	33.93	27.74	25.46
TV-Error	1.498	NA	1.303	1.190	1.048	1.404	1.336	1.260	1.341	0.945	1.290	1.227	1.223

Table 6.9: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the b_1 test set. Best and second best results are shown in red, blue colors, respectively.*

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.8875	NA	0.9528	0.9591	0.8441	0.9223	0.9399	0.9466	0.9391	0.9026	0.9501	0.9671	0.9578
PSNR	30.25	NA	32.88	34.17	27.01	29.05	32.19	31.57	31.56	30.57	33.31	35.67	34.67
VIF	0.7057	NA	0.6622	0.6435	0.4185	0.5910	0.5569	0.6321	0.5935	0.5571	0.7051	0.7075	0.6823
MSE	56.01	NA	33.71	22.87	125.4	82.11	37.57	51.71	40.91	66.36	30.85	19.67	22.93
LPIPS	0.2011	NA	0.0355	0.0294	0.1412	0.1271	0.0765	0.0475	0.0560	0.1786	0.0752	0.0293	0.0458
UQI	0.9980	NA	0.9987	0.9993	0.9928	0.9957	0.9989	0.9988	0.9992	0.9976	0.9991	0.9995	0.9992
MS-SSIM	0.9357	NA	0.9836	0.9878	0.8980	0.9629	0.9757	0.9769	0.9766	0.9253	0.9763	0.9883	0.9846
NIQE	3.739	NA	2.608	2.611	3.060	2.794	2.959	2.571	2.566	3.128	2.626	2.614	3.012
PIQE	51.78	NA	40.88	42.49	42.76	42.25	42.88	41.11	35.99	46.20	45.93	41.70	41.17
FSIM	0.9587	NA	0.9772	0.9841	0.8966	0.9642	0.9695	0.9642	0.9672	0.9365	0.9770	0.9863	0.9815
Haar PSI	0.7459	NA	0.8583	0.8977	0.6177	0.7628	0.8333	0.8266	0.8222	0.7225	0.8505	0.9089	0.8898
GMSD	0.1296	NA	0.0462	0.0335	0.1459	0.0955	0.0672	0.0585	0.0601	0.1205	0.0658	0.0321	0.0382
BRISQUE	35.39	NA	28.49	30.46	30.62	25.92	25.28	27.55	30.62	25.06	25.82	29.08	29.32
TV-Error	1.254	NA	1.061	0.988	0.991	1.225	1.080	1.025	1.059	1.065	1.069	1.030	1.040

Table 6.10: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the b_2 test set. Best and second best results are shown in red, blue colors, respectively.*

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.9289	NA	0.9508	0.9467	0.7601	0.9398	0.9290	0.9282	0.9267	0.8926	0.9436	0.9628	0.9544
PSNR	31.56	NA	32.37	33.24	25.16	30.37	30.51	29.03	30.87	28.97	32.58	33.86	33.16
VIF	0.7877	NA	0.6959	0.6474	0.3222	0.6935	0.6391	0.6348	0.7583	0.6956	0.7511	0.7310	0.6917
MSE	38.93	NA	44.91	27.98	186.5	63.42	58.73	101.9	42.89	60.67	20.99	28.56	33.91
LPIPS	0.1492	NA	0.0511	0.0344	0.2420	0.1005	0.0975	0.0754	0.0840	0.3054	0.0533	0.0445	0.0535
UQI	0.9980	NA	0.9971	0.9989	0.9860	0.9950	0.9970	0.9959	0.9970	0.9867	0.9990	0.9990	0.9985
MS-SSIM	0.9679	NA	0.9718	0.9835	0.8241	0.9730	0.9604	0.9501	0.9630	0.7910	0.9861	0.9837	0.9788
NIQE	3.251	NA	3.226	3.250	3.140	3.364	3.225	3.231	3.234	3.481	3.180	3.208	3.290
PIQE	52.19	NA	50.66	54.91	59.02	52.81	50.66	52.05	52.29	58.73	53.52	48.34	46.94
FSIM	0.9749	NA	0.9697	0.9789	0.8647	0.9638	0.9623	0.9487	0.9598	0.9582	0.9799	0.9803	0.9734
Haar PSI	0.8356	NA	0.8376	0.8770	0.5333	0.8197	0.8207	0.7561	0.8166	0.8117	0.9039	0.8778	0.8613
GMSD	0.0861	NA	0.0653	0.0447	0.1809	0.0839	0.0749	0.0871	0.0831	0.0798	0.0523	0.0518	0.0570
BRISQUE	26.11	NA	24.27	29.98	34.96	27.20	27.68	27.48	36.26	28.99	25.34	22.89	22.83
TV-Error	1.198	NA	1.040	0.971	0.858	1.145	1.063	1.013	0.858	0.664	1.080	1.004	1.007

Table 6.11: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the b_3 test set. Best and second best results are shown in red, blue colors, respectively.*

The quantitative results obtained on the test-sets a_3 and a_4 are shown in Tables. 6.7, and 6.8, respectively. From Table. 6.7, it can be observed that the proposed model **Temporal** has shown a significant improvement of $\sim$ 1.18 in SSIM over FastDerain [11]. Whereas, the baseline **Proposed** has shown a remarkable rise of $\sim$ 3.25% in SSIM and $\sim$ 3.19% in PSNR, respectively, over the recent FastDerain [11] method. There is also a vast improvement of $\sim$ 51.55% in SSIM and $\sim$ 24.55% in PSNR over the recent method MS-CSC [13]. The existing method SPAC-CNN [14] which was better on a_1 and a_2, has been outperformed by the proposed model with a significant rise of $\sim$ 1.09% in SSIM and $\sim$ 1.7% in PSNR. It can also be observed that the proposed model has a clear supremacy over J4RNet [76] . Similarly, from Table. 6.8, it can be observed that the proposed method and its baseline configuration have outperformed almost all existing state-of-the-art methods on all evaluation metrics. So far it has been observed from tabular results on a_1, a_2 test-sets that the single image de-raining methods suffer with poor visual quality when the input frames are from unstable videos. To support this statement, a similar trend has been noticed in the case of a_3 and a_4 too. The quantitative comparison of the proposed model with existing schemes on the test-sets b_1, b_2, b_3, and b_4 are shown in the Tables. 6.9, 6.10, 6.11, and 6.12, respectively. To conclude a fair comparison, we have proposed a figure of merit (`fom`), and the results are shown in Table. 6.13.

Based on the proposed `fom`, it can be observed that the proposed model has outperformed the existing state-of-the-art methods for video rain-streak removal.

Metric	Rainy	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
SSIM	0.8914	NA	0.9426	0.9451	0.7581	0.9285	0.9000	0.9210	0.9129	0.8827	0.9380	0.9533	0.9475
PSNR	29.01	NA	32.11	33.36	25.32	29.77	29.91	30.96	30.51	28.99	31.91	34.67	34.02
VIF	0.7308	NA	0.6739	0.6264	0.3705	0.6512	0.5935	0.6144	0.5634	0.5636	0.7258	0.7075	0.6877
MSE	75.50	NA	42.70	25.29	189.4	70.33	62.61	57.84	59.98	123.9	43.21	23.80	26.63
LPIPS	0.2469	NA	0.0765	0.0378	0.2676	0.1390	0.1909	0.1165	0.0870	0.2278	0.1175	0.0675	0.0771
UQI	0.9970	NA	0.9981	0.9993	0.9879	0.9963	0.9977	0.9981	0.9985	0.9960	0.9984	0.9993	0.9990
MS-SSIM	0.9385	NA	0.9733	0.9823	0.8151	0.9674	0.9487	0.9593	0.9616	0.8741	0.9702	0.9825	0.9795
NIQE	4.249	NA	3.608	3.388	3.469	3.631	3.765	3.468	3.109	3.330	3.108	3.662	3.857
PIQE	45.07	NA	44.76	50.48	48.23	46.22	42.96	45.28	43.69	45.18	43.69	41.85	40.66
FSIM	0.9577	NA	0.9709	0.9770	0.8675	0.9618	0.9565	0.9578	0.9622	0.9376	0.9711	0.9804	0.9779
Haar PSI	0.7464	NA	0.8786	0.8739	0.5436	0.7965	0.7795	0.7947	0.7957	0.7459	0.8290	0.8865	0.8786
GMSD	0.1160	NA	0.0613	0.0441	0.1786	0.0805	0.0868	0.0765	0.0756	0.0899	0.0733	0.0458	0.0474
BRISQUE	31.78	NA	20.18	27.48	30.94	21.33	21.89	21.20	33.44	33.16	19.64	21.07	21.60
TV-Error	1.298	NA	1.111	1.023	1.029	1.217	1.157	1.093	1.151	1.167	1.190	1.084	1.076

Table 6.12: *Quantitative comparison of the proposed model with existing schemes using the incorporated evaluation metrics on the b_4 test set. Best and second best results are shown in red, blue colors, respectively.*

Test Set	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
Light	-	0.0285	0.1571	0	0	0	0.0285	0.0285	0	0.0285	0.2857	0.3571
Heavy	-	0.0714	0	0	0	0	0	0	0	0.0428	0.5285	0.2714
1	NA	0.0857	0.2714	0.0285	0	0	0.0571	0.1285	0.0285	0.0428	0.2	0.1571
a_1	NA	0.0857	0.2714	0.0285	0.0285	0.0714	0	0	0.0428	0.1428	0.2428	0.0571
a_2	NA	0.0428	0.3571	0.0714	0	0	0.0285	0	0.0428	0.0285	0.4	0.0857
a_3	NA	0	0.2142	0.0428	0	0.0285	0	0.0285	0.0285	0.0285	0.5	0.1285
a_4	NA	0	0.0571	0.0285	0.0285	0	0	0.0428	0.0857	0.0285	0.4142	0.4428
b_1	NA	0.1857	0.2428	0.0571	0	0	0	0.0428	0.0428	0.2	0.2285	0.0285
b_2	NA	0.0285	0.2714	0.0285	0	0.0285	0.0285	0.0857	0.0428	0.0285	0.7142	0.0285
b_3	NA	0	0.2	0.0714	0	0	0	0.0714	0.0428	0.2142	0.3571	0.1142
b_4	NA	0.0571	0.2285	0.0285	0	0	0	0.0285	0	0.1285	0.4142	0.1857

Table 6.13: *Quantitative comparison of the proposed model with existing methods in terms of a figure of merit (fom) = `0.6 * No. of Best + 0.4 * No. of Second Best/Total Metrics`. Best and second best values are in red & blue colors.*

We have also compared the proposed scheme based on the run-time (in seconds) parameter with existing approaches, as shown in Table. 6.14. It can be observed that the proposed model takes a minimal amount of time, which is ∼1.5 seconds per frame, for estimating the rain-free videos when compared to other existing methods. For a fair run-time evaluation, the results mentioned in the Table. 6.14 are from the experiments that have been conducted on a 12 GB GPU system on the **Test Set Light**.

6.3.3 Qualitative Results

Before presenting the subjective evaluation, we recall the major limitations of the existing schemes, such as color distortions, removal of objects that align with the

Methods	DualFlow	J4RNet	SPAC-CNN	MS-CSC	DetailNet	FastDerain	DIP	TCL	SE	JORDER	Proposed	Temporal
-	CVPR'19	CVPR'18	CVPR'18	CVPR'18	CVPR'17	TIP'19	CVPR'17	TIP'15	ICCV'17	CVPR'17	-	-
Framework	-	Caffe	Matlab	Matlab	Matlab	Matlab	Matlab	Matlab	Matlab	Caffe	Pytorch	Pytorch
For 9 videos in (s)	-	3821	7804	9226	811.4	252.1	523.6	1.72×10^5	1.9×10^5	203.6	**147.8**	153.2
Avg. SSIM	-	0.9051	0.9054	0.7093	0.8635	0.8482	0.8828	0.8702	0.8010	0.9124	0.9239	0.9260
Per frame in (s)	-	39.80	81.29	96.10	8.45	2.62	5.45	1791.66	1979.16	2.12	**1.53**	1.59

Table 6.14: *Run-time comparison of the proposed model with existing schemes over the* **Test Set Light**.

rain-streaks, massive motion blur, etc. Qualitative results of the proposed model are shown in Figures 6.6, 6.7, 6.8, and 6.9. The subjective results showcased in Figure 6.6 are based on the three consecutive frames from a real-world rainy video. It can be observed from Figure 6.6 that the proposed model does not suffer from any such artifacts. While the results obtained by using MS-CSC [13] suffer from heavy reconstruction artifacts such as high-frequency imprints from the previous frame, J4RNet [76] consists of blurry artifacts due to the rapid motion change between the frames, as shown in yellow bounding boxes. SPAC-CNN [14] has been one of the most competitive methods, as shown in previous subsections. However, the known method suffers from the blocky artifacts in certain regions, which are most affected by the sudden change in camera trajectory. The detailed justification is given in Section. 6.5. Even though the results obtained by using a single image de-raining method DDN [16] look promising, it can be observed from other figures, namely Figures 6.7, 6.8 and 6.9 that it still consists of rain-streaks in the de-rained frames with color distortions at certain regions. FastDerain [11], which is best among the existing video de-raining methods in terms of run-time computation as shown in Table. 6.14, still consists of rain-streaks and poor visual quality in the de-rained frames when compared to the proposed method, as shown in Figure 6.6. DIP [75] method also suffers from the poor visual quality of the de-rained frames, as shown in yellow bounding boxes.

Figure 6.6: *Qualitative comparison of the proposed model with existing schemes on a real-world rainy video frames.* $\mathbf{f}^m_{r,i-1}$, $\mathbf{f}^m_{r,i}$, *and* $\mathbf{f}^m_{r,i+1}$ *are three consecutive rainy-frames. Please magnify the figure for better details shown in yellow boxes.*

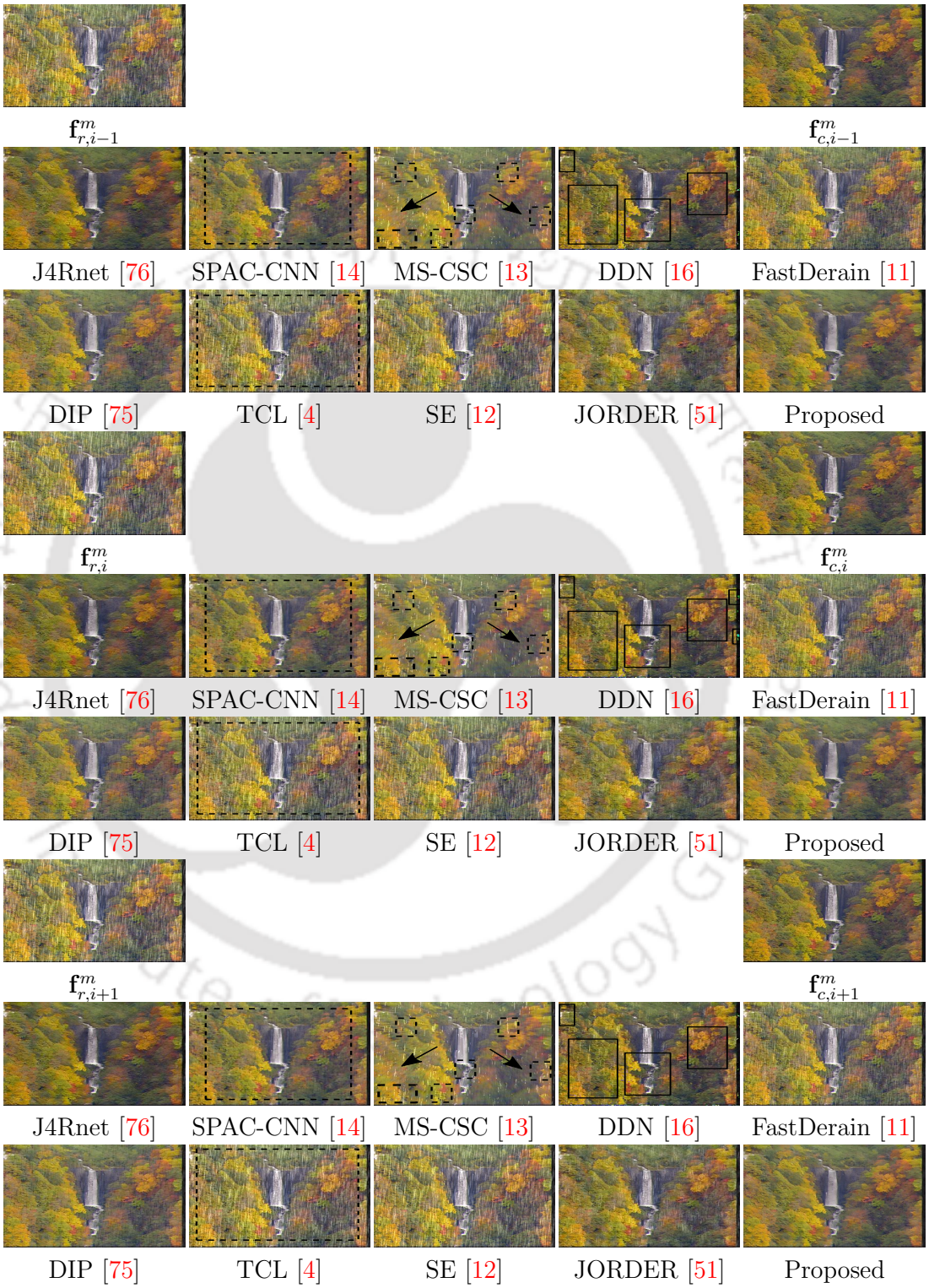


Figure 6.7: *Qualitative comparison of the proposed model with existing schemes on a synthetic rainy video frames.* $\mathbf{f}^m_{r,i-1}$, $\mathbf{f}^m_{r,i}$, $\mathbf{f}^m_{r,i+1}$ *depicts the three consecutive rainy frames. Please magnify the figure for better details shown in black boxes.*

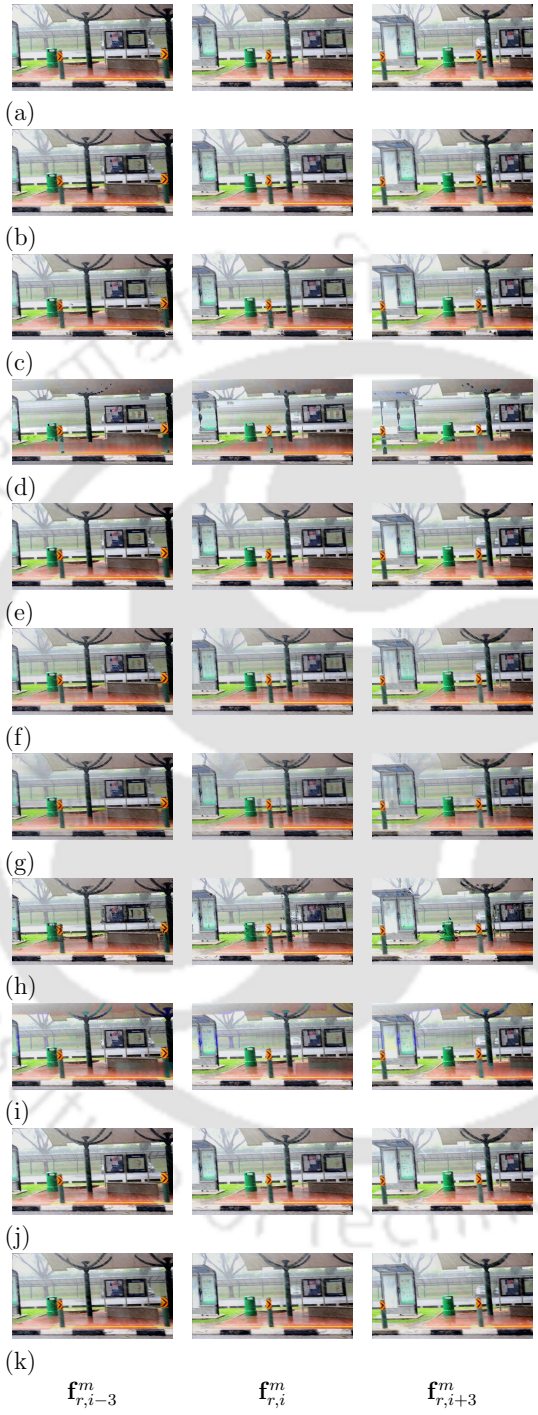


Figure 6.8: *Qualitative comparison of the proposed method with existing schemes on real-world rainy video. (a) Rainy frames, (b) J4RNet, (c) SPAC-CNN, (d) MSCSC, (e) DDN, (f) FastDerain, (g) DIP, (h) TCL, (i) SE, (j) JORDER, (k) Proposed.* $\mathbf{f}^m_{r,i-3}, \mathbf{f}^m_{r,i}, \mathbf{f}^m_{r,i+3}$ *denote frame sequences. Please magnify the figure for better details.*

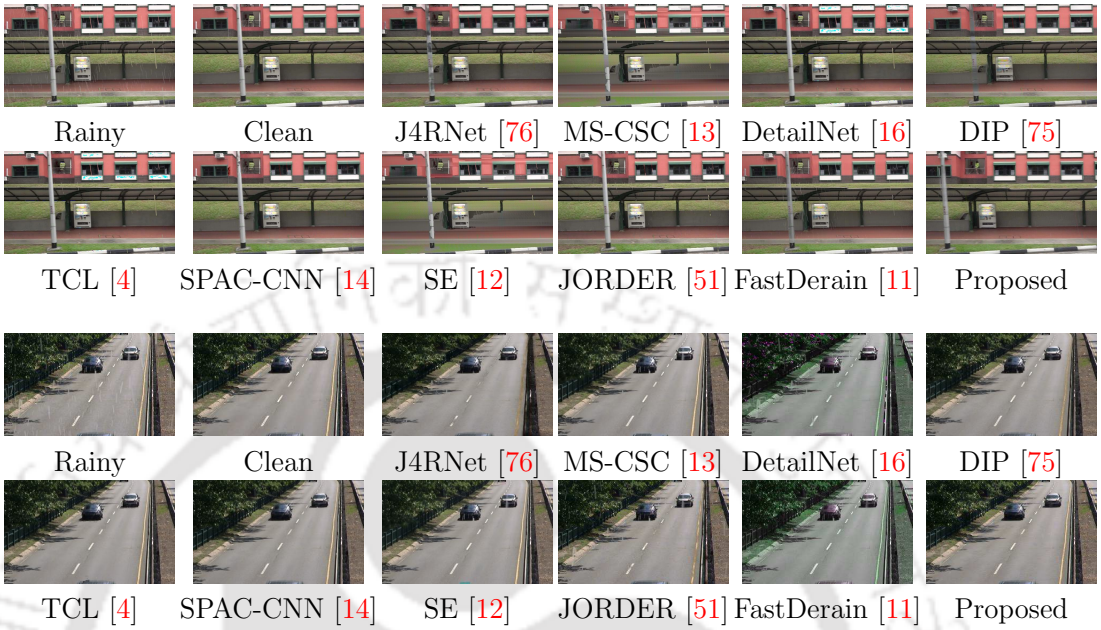


Figure 6.9: *Qualitative comparison of the proposed model with existing schemes on a synthetic rainy video frames. Please magnify the figure for better details*

While the artifacts gradually increase in the subsequent de-rained frames estimated by the existing scheme TCL [4], the method SE [12] still consists of rain-streaks in the generated de-rained frames. Single image de-raining method JORDER [51] does not suffer much from visual artifacts such as color distortion, whereas it can be observed from subsequent Figures 6.7, 6.8, and 6.9 that it has been unsuccessful in removing rain-streaks with dense rain-drops. Figure 6.7 depicts the visual comparison of the proposed model with existing schemes on a synthetic rainy video. It can be observed that the existing methods SPAC-CNN [14], FastDerain [11], DIP [75], TCL [4], SE [12], and JORDER [51] still consists of rain-streaks in the de-rained frames when compared to the proposed model and ground truth. It can also be observed that the existing image de-raining method DDN [16] suffers from visible rain-streaks and color distorted patches. Interestingly, while the results obtained by using J4RNet [76] suffer from color-saturation, the de-rained frames generated by using MS-CSC [13] consists of (a) thick rain-streaks, and (b) an unwanted in-out motion blur artifact around the moving objects in the frames, which is only *Waterfall*. The direction of the motion blur is shown by using the black arrows in the de-rained frames of

Figure 6.10: *Failure-case on the rainy frames from a video.* ***Top*** *row shows the rainy,* ***Bottom*** *row shows the de-rained frames.*

MS-CSC [13]. Figure 6.8 presents the visual comparison of the proposed model with existing schemes on real-world rainy frames with temporal width 3. It can be observed from Figure 6.8 (c) that the existing method SPAC-CNN [14] suffer from the visual artifacts such as object disappearance, which is a *Car* in the given example (bounded by a black box). Besides, it also suffers from color distortion. Similarly methods FastDerain [11], DIP [75], and TCL [4] also suffer from reconstruction error in the de-rained frames. Figure 6.9 depicts the visual comparison on another synthetic rainy video frame. While the proposed model is successful in removing the rain-streaks from the videos, we have observed that it fails to eradicate the snowy-rain from the frames, as shown in Figure 6.10. This may be because the proposed model has not seen snowy videos when training. We also present a detailed ablation study with a variety of baseline configurations to show the effect of each module in the proposed work.

6.4 Ablation Study

We present the following baselines in addition to those mentioned above and performed the ablation study to demonstrate the significance of each module using the adopted test-sets Light, Heavy, a_1, and b_1:

1. **G-M-RGB** : The proposed generator (ϕ_G) model is trained using $\mathcal{L}_{MSE}$ only, with RGB input/output color-space.

2. **G-M** : The proposed generator (ϕ_G) model is trained using $\mathcal{L}_{MSE}$ only.

3. **G-EP** : The proposed generator (ϕ_G) model is trained using $\alpha.\mathcal{L}_{Percep}$ only.

4. **G-M-FP** : The proposed generator (ϕ_G) model is trained using $\mathcal{L}_{MSE} + \lambda\mathcal{L}_{Percep}$ only, where λ is a fixed constant with value of 1.

5. **G-M-FP-EA** : The proposed model is trained using $\mathcal{L}_{MSE} + \lambda\mathcal{L}_{Percep} + \beta\mathcal{L}_{Adv}$ loss.

6. **G-M-FP-FA** : The proposed model is trained using $\mathcal{L}_{MSE} + \lambda\mathcal{L}_{Percep} + \gamma\mathcal{L}_{Adv}$ loss, where γ is a fixed constant with value of 6.6×10^{-3}.

7. **G-M-EP-FA** : The proposed model is trained using $\mathcal{L}_{MSE} + \alpha\mathcal{L}_{Percep} + \gamma\mathcal{L}_{Adv}$ loss.

8. **G-M-EP-EA-N** : Instead of proposed adversarial loss $\mathcal{L}_{Adv}$, we perform an ablation with conventional entropy-based adversarial loss, denoting as $\mathcal{L}_{Adv}^{C.E}$, and train the proposed model with $\mathcal{L}_{MSE} + \alpha\mathcal{L}_{Percep} + \beta\mathcal{L}_{Adv}^{C.E}$ loss.

9. **G-M-EP-EA-D** : It denotes, the temporal baseline where, the proposed discriminator model is replaced by the discriminator proposed in Zhang *et al.* [9], and trained using $\mathcal{L}_G$ loss.

6.4.1 Quantitative Results

The quantitative comparison of the proposed scheme with baselines is shown in Tables. 6.15, 6.16, 6.17, and 6.18. Note, we have adopted the fixed-constants λ, γ for perceptual and adversarial loss, respectively, for the aforementioned baselines following Zhang *et al.* [9].

Metric	G-M-RGB	G-M	G-EP	G-M-FP	G-M-FP-EA	G-M-FP-FA	G-M-EP-FA	G-M-EP-EA-N	G-M-EP-EA-D	Proposed	Temporal
SSIM	0.8826	0.9064	0.9258	0.9202	0.9214	0.9258	0.9195	0.9120	0.9179	0.9239	0.9260
PSNR	24.17	29.89	31.28	31.15	30.53	31.35	31.52	30.57	30.49	31.92	31.81
VIF	0.4505	0.5980	0.6523	0.6398	0.6445	0.6467	0.6401	0.6471	0.5150	0.6607	0.6616
MSE	366.7	114.5	80.39	85.12	84.25	66.56	79.76	110.1	111.2	62.86	61.92
LPIPS	0.1212	0.1027	0.0922	0.1001	0.0984	0.0991	0.1012	0.1024	0.1002	0.0982	0.0943
UQI	0.9794	0.9904	0.9942	0.9941	0.9938	0.9951	0.9950	0.9927	0.9911	0.9928	0.9956
MS-SSIM	0.8955	0.9216	0.9452	0.9389	0.9395	0.9406	0.9359	0.9364	0.9417	0.9455	0.9466
NIQE	3.698	3.781	3.657	3.703	3.784	3.696	3.802	3.904	3.698	3.761	3.652
PIQE	28.80	26.82	27.97	27.40	26.49	26.88	27.21	27.78	28.27	27.84	27.56
FSIM	0.9095	0.9361	0.9519	0.9465	0.9470	0.9527	0.9465	0.9414	0.9424	0.9500	0.9509
Haar PSI	0.6408	0.7400	0.7974	0.7754	0.7809	0.7984	0.7663	0.7559	0.7734	0.7861	0.7840
GMSD	0.1163	0.0830	0.0677	0.0796	0.0750	0.0699	0.0814	0.0851	0.0733	0.0757	0.0729
BRISQUE	23.05	23.18	24.75	24.96	23.88	23.18	26.28	26.26	24.87	22.75	25.66
TV-Error	1.259	1.211	1.187	1.193	1.182	1.179	1.189	1.203	1.182	1.160	1.157

Table 6.15: *Quantitative comparison of the proposed scheme with different baselines on the* ***Test Set Light****.*

Metric	G-M-RGB	G-M	G-EP	G-M-FP	G-M-FP-EA	G-M-FP-FA	G-M-EP-FA	G-M-EP-EA-N	G-M-EP-EA-D	Proposed	Temporal
SSIM	0.7662	0.7808	0.8127	0.8082	0.8092	0.8076	0.7965	0.8012	0.7962	0.8185	0.7983
PSNR	22.41	23.34	23.95	24.15	24.30	24.05	22.80	23.48	23.53	24.35	23.96
VIF	0.3586	0.3604	0.3862	0.3795	0.3812	0.3776	0.3711	0.3748	0.3729	0.3969	0.3769
MSE	518.7	414.9	428.2	499.5	383.9	521.6	687.5	555.3	512.8	370.9	440.3
LPIPS	0.2676	0.2563	0.2203	0.2341	0.2327	0.2367	24.28	0.2386	0.2358	0.2163	0.2320
UQI	0.9492	0.9572	0.9652	0.9557	0.9557	0.9512	0.9352	0.9522	0.9548	0.9660	0.9576
MS-SSIM	0.7957	0.7995	0.8306	0.8264	0.8311	0.8249	0.8177	0.8201	0.8169	0.8394	0.8159
NIQE	3.898	3.807	4.462	4.149	4.026	4.265	4.023	3.728	4.391	3.756	3.806
PIQE	31.04	28.66	28.62	26.01	27.26	26.63	24.53	26.63	27.31	26.45	27.47
FSIM	0.8441	0.7808	0.8717	0.8713	0.8694	0.8716	0.8717	0.8640	0.8643	0.8754	0.8614
Haar PSI	0.5092	0.5208	0.5805	0.5771	0.5792	0.5753	0.5662	0.5622	0.5606	0.5843	0.5548
GMSD	0.1499	0.1423	0.1234	0.1261	0.1294	0.1269	0.1268	0.1298	0.1313	0.1223	0.1329
BRISQUE	31.73	26.74	26.02	23.98	24.84	23.23	24.19	25.71	24.29	25.47	27.40
TV-Error	0.9989	0.9838	0.9422	0.9527	0.9562	0.9569	0.9732	0.9702	0.9716	0.9569	0.9358

Table 6.16: *Quantitative comparison of the proposed scheme with different baselines on the* ***Test Set Heavy****.*

Metric	G-M-RGB	G-M	G-EP	G-M-FP	G-M-FP-EA	G-M-FP-FA	G-M-EP-FA	G-M-EP-EA-N	G-M-EP-EA-D	Proposed	Temporal
SSIM	0.8562	0.9102	0.9243	0.939	0.9363	0.9305	0.9282	0.9259	0.9263	0.9412	0.9311
PSNR	18.84	27.26	28.38	29.15	28.12	29.81	29.11	28.42	29.2	30.05	29.89
VIF	0.5483	0.5671	0.5922	0.6105	0.6033	0.6007	0.6021	0.6063	0.5958	0.63	0.6081
MSE	880	134.7	110.9	94.47	115.9	79.84	95.31	116.9	101.2	71.51	78.31
LPIPS	0.1962	0.0749	0.0622	0.0585	0.0592	0.0596	0.0634	0.0644	0.0624	0.042	0.0502
UQI	0.9797	0.9952	0.9961	0.9957	0.9956	0.9977	0.9966	0.9969	0.9961	0.9982	0.9976
MS-SSIM	0.8869	0.9416	0.9457	0.9674	0.9688	0.9629	0.9563	0.9527	0.9501	0.9817	0.9768
NIQE	2.505	2.5067	2.783	2.237	2.361	2.412	2.505	2.343	2.411	2.25	2.36
PIQE	21.56	19.93	22.33	22.33	18.82	20.38	21.23	20.28	20.06	19.97	20.37
FSIM	0.8851	0.9503	0.9647	0.9735	0.9676	0.9701	0.9649	0.9588	0.959	0.9729	0.9675
Haar PSI	0.5559	0.766	0.8152	0.8474	0.8348	0.8233	0.8187	0.7926	0.8118	0.844	0.826
GMSD	0.1421	0.0783	0.0629	0.0532	0.0557	0.0555	0.0627	0.0675	0.0655	0.0542	0.0603
BRISQUE	21.88	18.51	18.6	17.95	15.53	17.83	17.94	19.63	13.9	18.84	17.52
TV-Error	2.481	2.337	2.289	2.195	2.189	2.206	2.197	2.177	2.246	2.065	2.025

Table 6.17: *Quantitative comparison of the proposed scheme with different baselines on the* ***Test Set*** a_1*.*

Metric	G-M-RGB	G-M	G-EP	G-M-FP	G-M-FP-EA	G-M-FP-FA	G-M-EP-FA	G-M-EP-EA-N	G-M-EP-EA-D	Proposed	Temporal
SSIM	0.8273	0.9205	0.9113	0.93	0.9331	0.934	0.9314	0.9322	0.931	0.946	0.9353
PSNR	13.02	28.98	28.13	29.35	28.5	30.71	29.58	29.44	29.62	31.48	30.78
VIF	0.4459	0.5731	0.5772	0.5903	0.5926	0.5897	0.5974	0.6017	0.5931	0.6221	0.5842
MSE	3904	90.05	106.3	80.92	96.26	58.12	75.61	80.71	80.15	47.91	56.83
LPIPS	0.3691	0.0927	0.0899	0.0864	0.0838	0.0821	0.0839	0.081	0.0682	0.0591	0.079
UQI	0.9135	0.9972	0.9967	0.9968	0.9967	0.9983	0.9983	0.9979	0.9973	0.9989	0.9985
MS-SSIM	0.846	0.9522	0.9542	0.9617	0.9638	0.9659	0.9647	0.9655	0.9595	0.9784	0.9695
NIQE	2.314	2.458	2.898	2.354	2.506	2.642	2.779	2.259	2.568	2.545	2.896
PIQE	30.15	27.56	30.85	28.69	26.79	28.82	29.33	31.44	29.83	29.47	28.77
FSIM	0.8853	0.9533	0.953	0.9646	0.9632	0.9677	0.9641	0.9576	0.9555	0.9745	0.9669
Haar PSI	0.5053	0.75	0.7601	0.8024	0.8009	0.8049	0.7991	0.7318	0.7653	0.8311	0.8063
GMSD	0.14	0.0754	0.0787	0.0672	0.0659	0.0621	0.0671	0.0726	0.0728	0.056	0.0612
BRISQUE	26.76	26.22	26.17	23.76	25.68	25.49	25.76	27.35	28.1	27.74	25.46
TV-Error	1.451	1.303	1.297	1.243	1.238	1.228	1.227	1.231	1.229	1.227	1.223

Table 6.18: *Quantitative comparison of the proposed scheme with different baselines on the* ***Test Set*** $b1$*.*

6.4.2 Qualitative comparison

6.4.2.1 Improvement from the perspective of input color-space

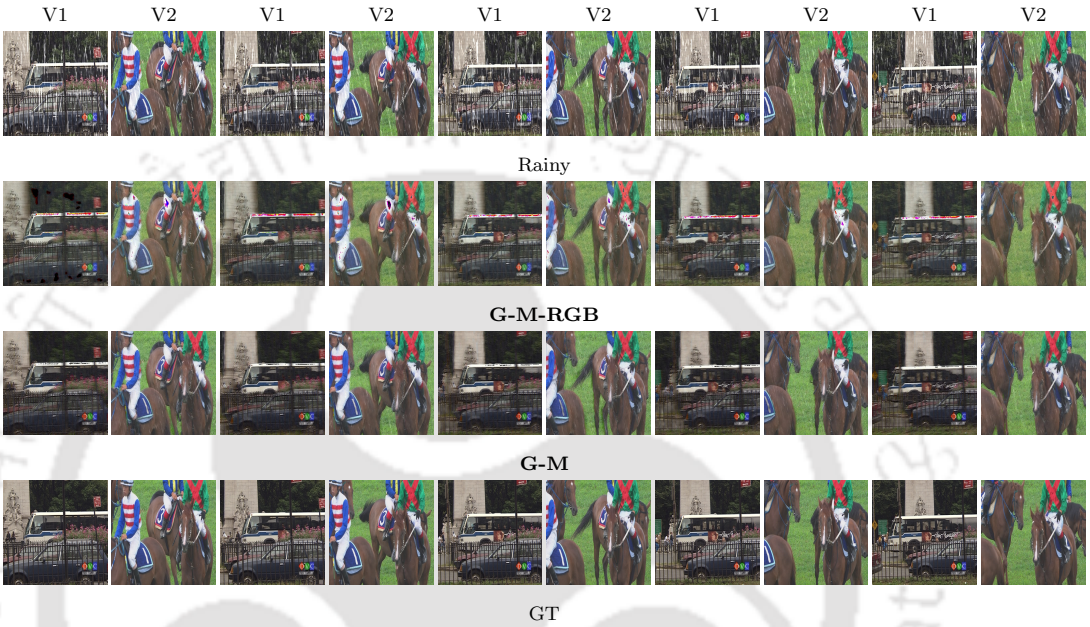


Figure 6.11: *Results to show the comparison between **G-M** and **G-M-RGB** configs. V# denote the video number.*

The rain-streaks in an image/frame exhibits pseudo-periodic nature. Due to the cancellation property of the YCBCR color-space in the case of rain-streaks [107], most of the rain only exists in the Y channel. Therefore, it is preferable to adopt YCBCR as input color-space which has a smaller solution space, compared to 3-channel RGB space. Quantitatively, it can be observed from the Tables 6.15, 6.16, 6.17, and 6.18 that the configuration **G-M** has outperformed **G-M-RGB** in most of the evaluation metrics with a significant margin. Visually, it may be observed from the Figure 6.11 that **G-M-RGB** suffer from visual artifacts.

6.4.2.2 Improvement from the perspective of model architecture

To show the efficacy of the proposed multi-contextual discriminator (ϕ_D), we performed an ablation (**G-M-EP-EA-D**) where the proposed ϕ_D is replaced with a discriminator used in Zhang *et al.* [9]. It can be observed from Figure 6.12

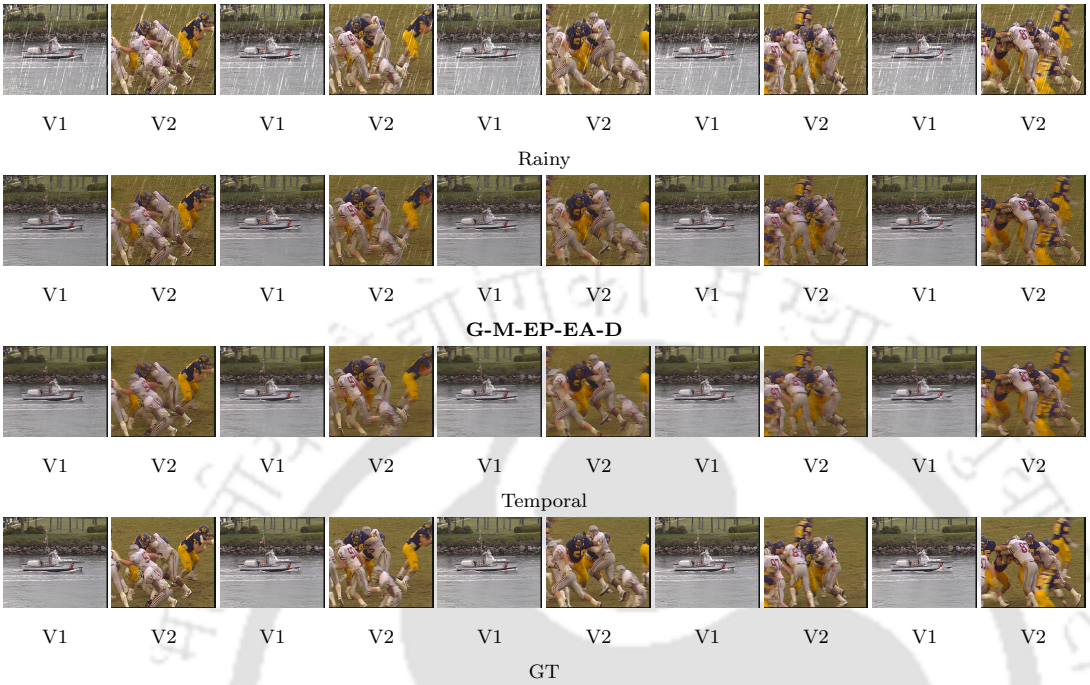


Figure 6.12: *To show the comparison between Temporal and **G-M-EP-EA-D** configs. V# denote the video no.*

that the results obtained by using **G-M-EP-EA-D** contains the visible rain-streaks when compared to the proposed method. It is also evident from the Tables 6.15, 6.16, 6.17, and 6.18 that the temporal solution that is 3D-conv based multi-contextual network, has outperformed the configuration **G-M-EP-EA-D** in most of the evaluation metrics. It may be because the discriminator proposed in [9] (1) use 2D-conv with fixed contextual size that may not learn the temporal consistency well compared to 3D-conv with multi-contextual, (2) allows down-upsampling of the features that may incur loss of high-frequency details.

6.4.2.3 Exponential Perceptual Loss + MSE vs MSE

It is clear from the quantitative results that the inclusion of perceptual loss over conventional MSE has proven to be beneficial, especially in terms of SSIM and PSNR. It can be observed from the Figure 6.13 that the results obtained by using only MSE loss still contains the visible rain-streaks compared to the proposed baseline.

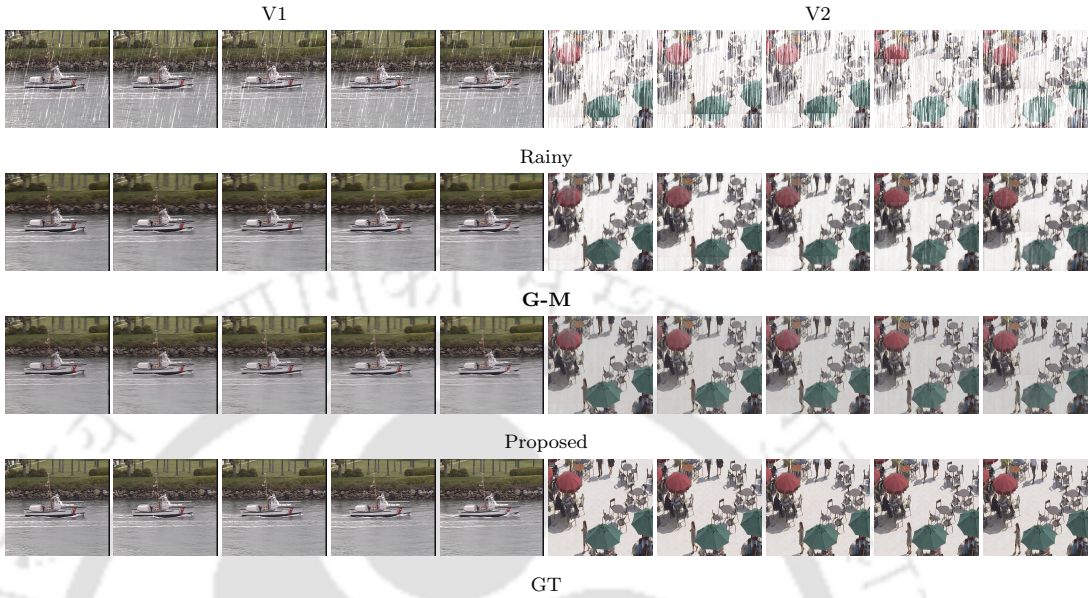


Figure 6.13: *Sample results to show the comparison between Proposed and* ***G-M*** *configurations. V# denote the video number. Please magnify the figure to see the visible rain-streaks in* ***G-M****.* `Quantitative results are given in Tables 6.15, 6.16, 6.17, and 6.18 of this chapter.`

6.4.2.4 Exponential Adversarial Loss vs Fixed Constant Adversarial Loss

It can be observed from the quantitative results that introducing the exponentially increasing loss constant compared to fixed, has proven to be beneficial, especially on the test-sets a_1, b_1. It can also be observed from the Figures 6.14, and 6.15 that results obtained using fixed constant losses suffer from visible rain-streaks. It is based on the intuition that the learned weights should deviate much (avoid large variance) when introduced a new penalty.

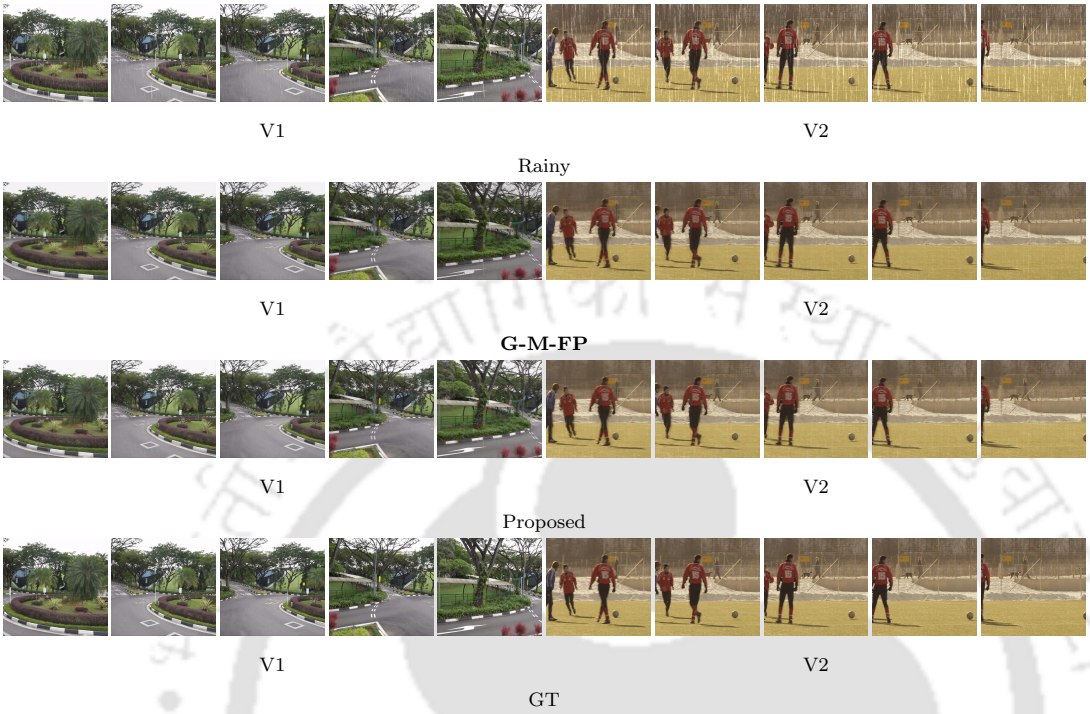


Figure 6.14: *Sample results to show the comparison between Proposed and* ***G-M-FP*** *configurations. V# denote the video number. Please magnify the figure to see the visible rain-streaks in* ***G-M-FP****.*

6.4.2.5 MSE-based Adversarial Loss vs Entropy-based Adversarial Loss

To show the efficacy of the proposed MSE-based Adversarial loss used in discriminator ϕ_D, we performed an ablation (**G-M-EP-EA-N**) where the proposed MSE-based Adversarial loss is replaced with conventional Entropy-based Adversarial loss for training. It can be observed from Figure 6.17 that the results obtained by using **G-M-EP-EA-N** contain visible rain-streaks and reconstruction errors compared to the model trained using the proposed loss function. It is also evident from Tables 6.15, 6.16, 6.17, and 6.18 that the temporal solution, which uses the proposed MSE-based Adversarial loss, outperforms the configuration **G-M-EP-EA-N** in most of the evaluation metrics. It may be because the proposed MSE-based Adversarial loss can learn more useful distinguishing characteristics between the real and fake de-rained images compared to conventional entropy loss, thereby driving the generator ϕ_G to learn the essential features that

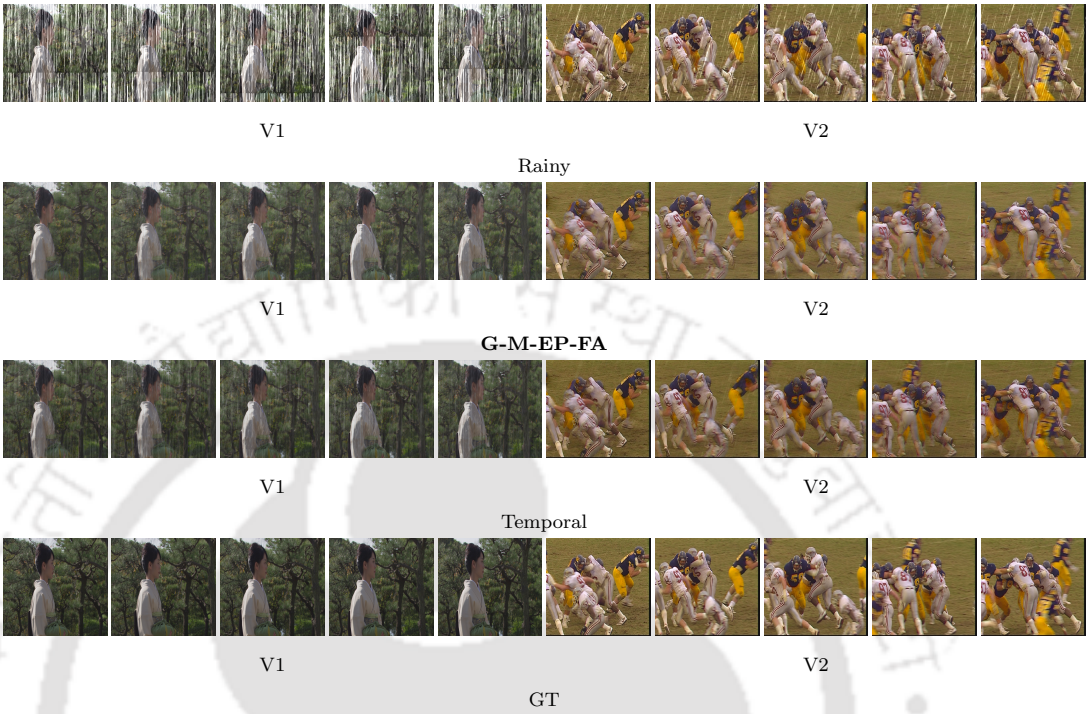


Figure 6.15: *Sample results to show the comparison between Temporal and* ***G-M-EP-FA*** *configurations. V# denote the video number. Please magnify the figure to see the visible rain-streaks in* ***G-M-EP-FA****.*

should be retained to make the de-rained image look realistic.

6.5 Justification

The three main aspects that govern the success or failure of any image/video noise removal algorithm are: (a) Color space of the input, (b) Model architecture, and (c) Cost functions. A majority of the shortcomings mentioned earlier in this chapter can be overcome by carefully drafting the above-defined aspects. The proposed model has outperformed the existing schemes, as shown in Section. 6.3 with minimal artifacts. To justify, the following assertions can be considered: (a) Unlike single image de-hazing (see Chapter 5) where the noise exponentially varies with the depth of the pixel in an image, the rain-streak noise exhibit pseudo-periodic characteristics. Therefore, YCbCr may be the right choice of color space for input when compared to a highly correlated RGB color space in the case of

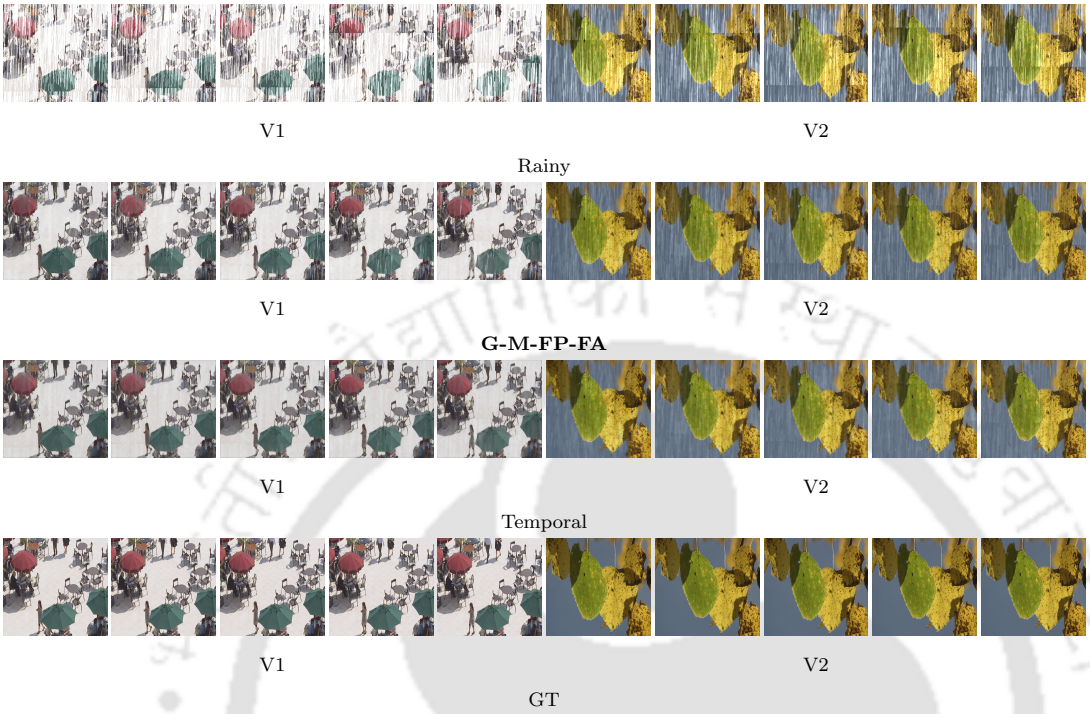


Figure 6.16: *Sample results to show the comparison between Temporal and* ***G-M-FP-FA*** *configurations. V# denote the video number.* `Quantitative results are given in Tables 6.15, 6.16, 6.17, and 6.18 of this chapter.`

image/video de-raining. This may help in avoiding the color distortions in the de-rained frames. (b) A majority of the video noise removal methods that are based on the deep learning framework separately consider the objectives of spatial and temporal enhancement. However, in this work, we attempt to unified these objectives and entirely rely on the proposed model for inherently estimating the optical flow followed by the de-rained frames. We thus present a light-weight deep CNN for video de-raining which is not only a resource favoured, but also overcome the heavy motion blur due to rapid change in motion between the frames because of inherently estimating the optical flow and its frame-recurrent nature. (c) While the encoder-decoder model might have improved the spatial resolution of the de-rained frame, the incorporated frame-recurrent methodology and temporal loss from the adversary may have further enhanced the performance of the proposed model by eliminating the problem of imprints from the previous frames and object disappearance. This may be due to the good choice of temporal

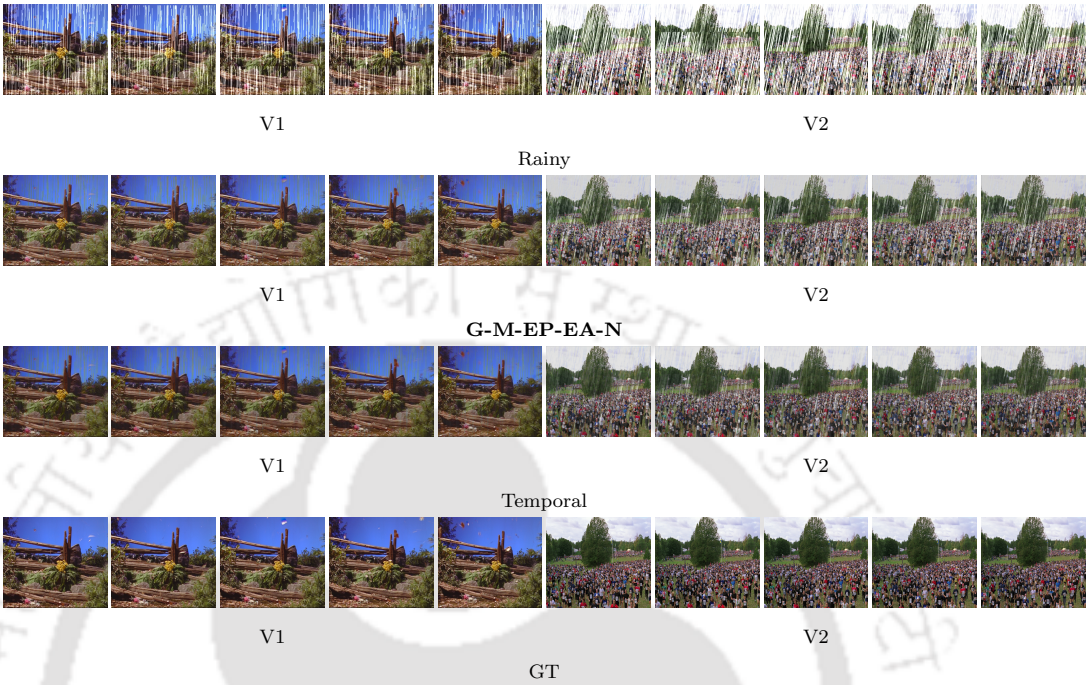


Figure 6.17: *To show the comparison between Temporal and* ***G-M-EP-EA-N*** *configs. V# denote the video no.*

width in the input, which is 3 in our case. However, the impact of increasing or decreasing the temporal width on the performance of the network may be taken as a future scope of this work.

6.6 Summary

In this contributory chapter, we have presented a light-weight unified deep learning-based frame-recurrent method for the video rain-streak removal task, which is built upon the Conditional GAN framework. The proposed generator method takes a previously estimated de-rained frame and rain-streak map to predict the current rain-free frame from a rainy video. Whereas the adversary is a multi-contextual 3D convolution-based CNN that classifies the set of de-rained frames into real or fake. In addition to the traditional $\mathcal{L}_2$ loss, we have also adopted the perceptual cost function for the optimization of the proposed model. Instead of traditional entropy loss from the adversary, we attempt to use the Euclidean

distance between the feature maps returned by the adversary to optimize the generator model for the video de-raining. To prove the efficacy of the proposed method, we have given an extensive comparison with ten state-of-the-art methods for video and image de-raining using fourteen image quality metrics on eleven test-sets. We have also shown the applicability of the proposed model on real-world rainy videos. In terms of computation, we have observed that the proposed model takes a minimal amount of time, which is $\sim$1.5 seconds per frame, for estimating the rain-free videos when compared to other existing methods.

The next chapter concludes the thesis by briefly summarizing the work presented in the thesis and discussing the future research works.

Chapter 7

Conclusion and Future Works

The main objective of this dissertation is to propose image and video restoration algorithms to obtain noise-free images and videos without compromising the visual quality. Two major tasks have been achieved in this research work: firstly, analyzing the noise characteristics in a noisy image or video, and secondly, devise deeper models to remove such noise based on the noise characteristics. In this chapter, we have summarized the major contributions of this thesis and highlighted some future scope of the research.

7.1 Summary of the Contributions

In next subsection, we have presented the summary of contributions.

7.1.1 Exploiting Efficient Spatial Upscaling for Single Image De-Raining

In the first contributory chapter, a learning-based approach has been presented to avoid over-coloring and white-dot artifacts in the de-rained images, which is empowered with efficient sub-pixel upscaling and adversarial training. The proposed approach utilizes the luminance channel of the rainy images only to bypass the visual artifacts due to the correlated RGB domain. It has been shown that the usage of efficient sub-pixel upscaling is beneficial over traditional deconvolution in the case of single image de-raining.

7.1.2 Exploiting Transformed Domain Features for Single Image De-Raining

The second contribution introduces the transformed domain coefficients of the rain-streaks in deep learning. In the first part of the second contribution, an uncorrelated transformed domain has been exploited by processing the DFT coefficients using a deep CNN. The proposed approach takes DFT coefficients of the rainy image as input and outputs the same of the de-rained image. Whereas, in the second part of the second contributory chapter, a correlated transformed domain has been exploited in terms of DWT coefficients for the same task. It has been shown that a significant improvement can be achieved if correlated transformed domain cues are given as input to deep CNN in addition to the spatial domain features.

7.1.3 A Probe Towards Scale-Space Invariant Conditional GAN for Image De-Hazing

The third contribution uncovers the aspect of scale-space invariance in the deep CNN for single image de-hazing by utilizing the LoG of the images. The LoG preserves a variety of edgy structures which can be utilized to remove the halo artifacts in the de-hazed images. The proposed model incorporates the Euclidean difference between the LoG features of de-hazed and clean ground truth images as a supervised cost function to optimize the conditional GAN-based framework.

7.1.4 Frame-Recurrent Multi-Contextual Adversarial Network for Video De-Raining

Lastly, in the final contribution, a unified multi-contextual deep CNN has been proposed for the task of video de-raining. It has been experimentally shown that the proposed multi-contextual 3D convolution-based design has been highly beneficial for efficient video de-raining. The method is further empowered with adversarial and perceptual cost functions.

7.2 Future works

The present study of this dissertation can be extended further in several directions as listed below:

- The proposed works in chapters 3, 4, and 5 can be extended to the respective video restoration. Particularly, it may be interesting to see how learning-based methods perform when presented with transformed domain coefficients of temporally connected noisy frames in the case of video de-noising.
- The proposed work in chapter 5 can be re-engineered to accommodate the scale-space invariance in the respective architecture instead of utilizing it as a supervised cost function.
- The presented approach in the last contributory chapter can be further extended to solve other video restoration tasks, such as video de-snowing and inpainting.
- Also, one may extend the presented ideas to image or video de-noising in a completely different domain, such as underwater or satellite optical image and video restoration using deep learning techniques.

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

References

[1] M. Bojarski, D. D. Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, X. Zhang, J. Zhao, and K. Zieba, "End to end learning for self-driving cars," *CoRR*, vol. abs/1604.07316, 2016. [Pg.xxi], [Pg.1], [Pg.2]

[2] H. Machiraju and V. N. Balasubramanian, "A little fog for a large turn," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, 2020. [Pg.xxi], [Pg.1], [Pg.2]

[3] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [Pg.xxi], [Pg.xxiv], [Pg.2], [Pg.8], [Pg.12], [Pg.29], [Pg.36], [Pg.45], [Pg.46], [Pg.48], [Pg.49], [Pg.50], [Pg.51], [Pg.52], [Pg.53], [Pg.54], [Pg.67], [Pg.68], [Pg.69], [Pg.71], [Pg.72], [Pg.80], [Pg.81], [Pg.82], [Pg.83], [Pg.84], [Pg.89]

[4] J. Kim, J. Sim, and C. Kim, "Video deraining and desnowing using temporal correlation and low-rank matrix completion," *IEEE Transactions on Image Processing*, vol. 24, no. 9, pp. 2658–2670, 2015. [Pg.xxi], [Pg.xxv], [Pg.11], [Pg.13], [Pg.14], [Pg.37], [Pg.101], [Pg.102], [Pg.120], [Pg.121], [Pg.123], [Pg.124]

[5] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning*

Representations, 2015. [Pg.xxii], [Pg.19], [Pg.25], [Pg.26], [Pg.29], [Pg.32], [Pg.80], [Pg.89], [Pg.109]

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [Pg.xxii], [Pg.7], [Pg.19], [Pg.26], [Pg.40], [Pg.53], [Pg.54]

[7] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *CoRR*, vol. abs/1505.04597, 2015. [Pg.xxii], [Pg.8], [Pg.10], [Pg.19], [Pg.27], [Pg.40], [Pg.50], [Pg.87], [Pg.103], [Pg.107]

[8] J. Johnson, A. Alahi, and F. Li, "Perceptual losses for real-time style transfer and super-resolution," *CoRR*, vol. abs/1603.08155, 2016. [Pg.xxii], [Pg.8], [Pg.28], [Pg.29], [Pg.40], [Pg.53], [Pg.80], [Pg.86], [Pg.89], [Pg.94], [Pg.104]

[9] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2019. [Pg.xxii], [Pg.xxiv], [Pg.8], [Pg.12], [Pg.40], [Pg.41], [Pg.44], [Pg.45], [Pg.46], [Pg.47], [Pg.48], [Pg.49], [Pg.50], [Pg.51], [Pg.52], [Pg.53], [Pg.54], [Pg.80], [Pg.81], [Pg.82], [Pg.83], [Pg.107], [Pg.109], [Pg.125], [Pg.127], [Pg.128]

[10] H. Zhang and V. M. Patel, "Densely connected pyramid dehazing network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [Pg.xxv], [Pg.10], [Pg.13], [Pg.37], [Pg.85], [Pg.87], [Pg.88], [Pg.90], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.98], [Pg.99], [Pg.100]

[11] T. Jiang, T. Huang, X. Zhao, L. Deng, and Y. Wang, "FastDeRain: A novel video rain streak removal method using directional gradient priors," *IEEE Transactions on Image Processing*, vol. 28, no. 4, pp. 2089–2102, 2019. [Pg.xxv], [Pg.11], [Pg.13], [Pg.101], [Pg.102], [Pg.113], [Pg.114], [Pg.117], [Pg.119], [Pg.120], [Pg.121], [Pg.123], [Pg.124]

[12] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu, "Should we encode rain streaks in video as deterministic or stochastic?" in *The IEEE International Conference on Computer Vision (ICCV)*, 2017. [Pg.xxvi], [Pg.11], [Pg.103], [Pg.120], [Pg.121], [Pg.123]

[13] M. Li, Q. Xie, Q. Zhao, W. Wei, S. Gu, J. Tao, and D. Meng, "Video rain streak removal by multiscale convolutional sparse coding," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6644–6653. [Pg.xxvi], [Pg.11], [Pg.14], [Pg.103], [Pg.114], [Pg.117], [Pg.119], [Pg.120], [Pg.121], [Pg.123], [Pg.124]

[14] J. Chen, C.-H. Tan, J. Hou, L.-P. Chau, and H. Li, "Robust video content alignment and compensation for rain removal in a cnn framework," 2018. [Pg.xxvi], [Pg.2], [Pg.11], [Pg.14], [Pg.37], [Pg.102], [Pg.104], [Pg.113], [Pg.114], [Pg.117], [Pg.119], [Pg.120], [Pg.121], [Pg.123], [Pg.124]

[15] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004. [Pg.xxix], [Pg.30], [Pg.47], [Pg.67], [Pg.68], [Pg.73], [Pg.74], [Pg.82], [Pg.91], [Pg.110]

[16] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1715–1723. [Pg.xxix], [Pg.7], [Pg.45], [Pg.46], [Pg.48], [Pg.49], [Pg.50], [Pg.51], [Pg.52], [Pg.54], [Pg.67], [Pg.68], [Pg.73], [Pg.74], [Pg.75], [Pg.79], [Pg.80], [Pg.82], [Pg.83], [Pg.84], [Pg.119], [Pg.120], [Pg.121], [Pg.123]

[17] R. Fattal, "Dehazing using color-lines," *ACM Transactions on Graphics*, vol. 34, no. 1, pp. 13:1–13:14, Dec. 2014. [Pg.xxx], [Pg.37], [Pg.91], [Pg.93], [Pg.94], [Pg.97]

[18] H. Zhang, V. Sindagi, and V. M. Patel, "Joint transmission map estimation and dehazing using deep networks," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 7, pp. 1975–1986, 2020. [Pg.2], [Pg.13], [Pg.85]

REFERENCES

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, 2012, p. 1097–1105. [Pg.5], [Pg.26], [Pg.32]

[20] D. Liu, B. Wen, X. Liu, Z. Wang, and T. S. Huang, "When image denoising meets high-level vision tasks: A deep learning approach," in *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, 2018, p. 842–848. [Pg.5]

[21] P. K. Sharma, I. Bisht, and A. Sur, "Wavelength-based attributed deep neural network for underwater image restoration," *ACM Transactions on Multimedia Computing, Communications, and Applications*, jan 2022. [Pg.5], [Pg.6]

[22] S. Ahmed, U. Kamal, and M. K. Hasan, "DFR-TSD: A deep learning based framework for robust traffic sign detection under challenging weather conditions," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2021. [Pg.5]

[23] M. Hassaballah, M. A. Kenk, K. Muhammad, and S. Minaee, "Vehicle detection and tracking in adverse weather using a deep learning framework," *IEEE Transactions on Intelligent Transportation Systems*, pp. 1–13, 2020. [Pg.5]

[24] A. Valada, J. Vertens, A. Dhall, and W. Burgard, "AdapNet: Adaptive semantic segmentation in adverse environmental conditions," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017, pp. 4644–4651. [Pg.5]

[25] A. Gopan and A. H. Muhammed, "Dehazing and road feature extraction from satellite images," in *2019 1st International Conference on Innovations in Information and Communication Technology (ICIICT)*, 2019, pp. 1–4. [Pg.5]

[26] X. Chen, Y. Li, L. Dai, and C. Kong, "Hybrid high-resolution learning for single remote sensing satellite image dehazing," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2021. [Pg.5]

[27] A. Bhat, A. Tyagi, A. Verdhan, and V. Verma, "Fast under water image enhancement for real time applications," in *2021 6th International Conference for Convergence in Technology (I2CT)*, 2021, pp. 1–8. [Pg.6]

[28] J. Zhang, L. Zhu, L. Xu, and Q. Xie, "Research on the correlation between image enhancement and underwater object detection," in *2020 Chinese Automation Congress (CAC)*, 2020, pp. 5928–5933. [Pg.6]

[29] H. D. Bhoir, N. M. Dongre, and R. R. Gulwani, "Visibility enhancement for remote surveillance system," in *2016 International Conference on Inventive Computation Technologies (ICICT)*, vol. 3, 2016, pp. 1–4. [Pg.6]

[30] S. Gu, D. Meng, W. Zuo, and L. Zhang, "Joint convolutional analysis and synthesis sparse representation for single image layer separation," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1717–1725. [Pg.6], [Pg.45], [Pg.46], [Pg.48], [Pg.49], [Pg.50], [Pg.51], [Pg.52], [Pg.80], [Pg.82], [Pg.83]

[31] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 2838–2847. [Pg.6]

[32] H. Zhang and V. M. Patel, "Convolutional sparse and low-rank coding-based rain streak removal," in *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2017, pp. 1259–1267. [Pg.6]

[33] S. Yu, W. Ou, X. You, Y. Mou, X. Jiang, and Y. Tang, "Single image rain streaks removal based on self-learning and structured sparse representation," in *2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP)*, July 2015, pp. 215–219. [Pg.6]

[34] D. Y. Chen, C. C. Chen, and L. W. Kang, "Visual depth guided color image rain streaks removal using sparse coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 8, pp. 1430–1455, Aug 2014. [Pg.6]

[35] L. W. Kang, C. W. Lin, and Y. H. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, April 2012. [Pg.6]

[36] D. A. Huang, L. W. Kang, Y. C. F. Wang, and C. W. Lin, "Self-learning based image decomposition with applications to single image denoising," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 83–93, Jan 2014. [Pg.6]

[37] K. Park, S. Yu, and J. Jeong, "A contrast restoration method for effective single image rain removal algorithm," in *2018 International Workshop on Advanced Image Technology (IWAIT)*, 2018, pp. 1–4. [Pg.6]

[38] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 3397–3405. [Pg.6], [Pg.68]

[39] Y. Chang, L. Yan, and S. Zhong, "Transformed low-rank model for line pattern noise removal," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 1735–1743. [Pg.7]

[40] C. H. Yeh, P. H. Liu, C. E. Yu, and C. Y. Lin, "Single image rain removal based on part-based model," in *2015 IEEE International Conference on Consumer Electronics - Taiwan*, 2015, pp. 462–463. [Pg.7]

[41] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, p. 679–698, Jun. 1986. [Pg.7]

[42] Y. Wang, C. Chen, S. Zhu, and B. Zeng, "A framework of single-image deraining method based on analysis of rain characteristics," in *2016 IEEE*

International Conference on Image Processing (ICIP), 2016, pp. 4087–4091. [Pg.7]

[43] L. Zhu, C. W. Fu, D. Lischinski, and P. A. Heng, "Joint bi-layer optimization for single-image rain streak removal," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2545–2553. [Pg.7], [Pg.68]

[44] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Single image rain streak decomposition using layer priors," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3874–3885, Aug 2017. [Pg.7], [Pg.68]

[45] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, "Clearing the skies: A deep network architecture for single-image rain removal," *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, June 2017. [Pg.7], [Pg.45], [Pg.46], [Pg.48], [Pg.49], [Pg.50], [Pg.51], [Pg.52], [Pg.68], [Pg.80], [Pg.82], [Pg.83]

[46] L. Shen, Z. Yue, Q. Chen, F. Feng, and J. Ma, "Deep joint rain and haze removal from single images," *CoRR*, vol. abs/1801.06769, 2018. [Pg.8]

[47] A. Haar, "Zur theorie der orthogonalen funktionensysteme," *Mathematische Annalen*, vol. 69, no. 3, pp. 331–371, Sep 1910. [Pg.8]

[48] K. He, J. Sun, and X. Tang, "Single image haze removal using dark channel prior," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 12, pp. 2341–2353, Dec 2011. [Pg.8], [Pg.9], [Pg.13], [Pg.75], [Pg.85], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.98], [Pg.99], [Pg.100]

[49] Q. Chen, X. Yi, B. Ni, Z. Shen, and X. Yang, "Rain removal via residual generation cascading," in *2017 IEEE Visual Communications and Image Processing (VCIP)*, 2017, pp. 1–4. [Pg.8]

[50] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, 2014, pp. 2672–2680. [Pg.8], [Pg.28], [Pg.78], [Pg.103]

[51] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 1685–1694. [Pg.8], [Pg.67], [Pg.68], [Pg.84], [Pg.114], [Pg.120], [Pg.121], [Pg.123]

[52] G. Huang, Z. Liu, and K. Q. Weinberger, "Densely connected convolutional networks," *CoRR*, vol. abs/1608.06993, 2016. [Pg.8]

[53] J. Yu, C. Xiao, and D. Li, "Physics-based fast single image fog removal," in *IEEE 10th International Conference on Signal Processing Proceedings*, 2010, pp. 1048–1052. [Pg.9]

[54] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 6, pp. 1397–1409, Jun. 2013. [Pg.9]

[55] G. Meng, Y. Wang, J. Duan, S. Xiang, and C. Pan, "Efficient image dehazing with boundary constraint and contextual regularization," in *The IEEE International Conference on Computer Vision (ICCV)*, 2013. [Pg.9], [Pg.13], [Pg.85], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.98], [Pg.99], [Pg.100]

[56] Q. Zhu, J. Mai, and L. Shao, "A fast single image haze removal algorithm using color attenuation prior," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3522–3533, Nov 2015. [Pg.9], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.98], [Pg.99], [Pg.100]

[57] L. K. Choi, J. You, and A. C. Bovik, "Referenceless prediction of perceptual fog density and perceptual image defogging," *IEEE Transactions on Image Processing*, vol. 24, no. 11, pp. 3888–3901, Nov 2015. [Pg.9], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.98], [Pg.99], [Pg.100]

[58] D. Berman, T. treibitz, and S. Avidan, "Non-local image dehazing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [Pg.9], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.97], [Pg.98], [Pg.99], [Pg.100]

[59] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, no. 4, pp. 541–551, Dec 1989. [Pg.10]

[60] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M.-H. Yang, "Single image dehazing via multi-scale convolutional neural networks," in *European Conference on Computer Vision*, 2016. [Pg.10], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.98], [Pg.99], [Pg.100]

[61] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2736–2744. [Pg.10], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.97], [Pg.98], [Pg.99], [Pg.100]

[62] S. Santra, R. Mondal, and B. Chanda, "Learning a patch quality comparator for single image dehazing," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4598–4607, Sep. 2018. [Pg.10], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.97], [Pg.98], [Pg.99], [Pg.100]

[63] D. Yang and J. Sun, "Proximal dehaze-net: A prior learning-based deep network for single image dehazing," in *Computer Vision – ECCV 2018*, 2018, pp. 729–746. [Pg.10], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96], [Pg.97], [Pg.98], [Pg.99], [Pg.100]

[64] T. Guo, X. Li, V. Cherukuri, and V. Monga, "Dense scene information estimation network for dehazing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2019. [Pg.10], [Pg.92], [Pg.93], [Pg.94], [Pg.95], [Pg.96]

[65] Y. Cho, J. Jeong, and A. Kim, "Model-assisted multiband fusion for single image enhancement and applications to robot vision," *IEEE Robotics and Automation Letters*, vol. 3, pp. 2822–2829, 2018. [Pg.10], [Pg.93], [Pg.94]

[66] D. Han, J. Kim, and J. Kim, "Deep pyramidal residual networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 6307–6315. [Pg.10]

[67] D. Engin, A. Genç, and H. K. Ekenel, "Cycle-dehaze: Enhanced cycle-gan for single image dehazing," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2018. [Pg.10], [Pg.93], [Pg.94], [Pg.95], [Pg.96]

[68] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017. [Pg.10]

[69] K. Garg and S. K. Nayar, "Detection and removal of rain from videos," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, vol. 1, 2004, pp. I–I. [Pg.10]

[70] K. Garg and S. K. Nayar, "When does a camera see rain?" in *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, vol. 2, 2005, pp. 1067–1074 Vol. 2. [Pg.10]

[71] X. Zhang, H. Li, Y. Qi, W. K. Leow, and T. K. Ng, "Rain removal in video by combining temporal and chromatic properties," in *2006 IEEE International Conference on Multimedia and Expo*, 2006, pp. 461–464. [Pg.11]

[72] J. Vis, P. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *International Journal of Computer Vision*, vol. 86, 01 2010. [Pg.11]

[73] J. Chen and L.-P. Chau, "A rain pixel recovery algorithm for videos with highly dynamic scenes," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 23, 11 2013. [Pg.11]

[74] W. Ren, J. Tian, Z. Han, A. Chan, and Y. Tang, "Video desnowing and deraining based on matrix decomposition," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [Pg.11]

[75] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discrimina-

tively intrinsic priors," 2017. [Pg.11], [Pg.119], [Pg.120], [Pg.121], [Pg.123], [Pg.124]

[76] J. Liu, W. Yang, S. Yang, and Z. Guo, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3233–3242. [Pg.11], [Pg.37], [Pg.113], [Pg.114], [Pg.117], [Pg.119], [Pg.120], [Pg.121], [Pg.123]

[77] W. Yang, J. Liu, and J. Feng, "Frame-consistent recurrent video deraining with dual-level flow," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019, pp. 1661–1670. [Pg.11], [Pg.37], [Pg.113]

[78] J. Liu, W. Yang, S. Yang, and Z. Guo, "D3R-Net: Dynamic routing residue recurrent network for video rain removal," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 699–712, 2019. [Pg.12]

[79] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Pg.19]

[80] Y. Katznelson, "An introduction to harmonic analysis," 1976. [Pg.20]

[81] H. Pratt, B. Williams, F. Coenen, and Y. Zheng, "FCNN: Fourier convolutional neural networks," in *Machine Learning and Knowledge Discovery in Databases*, 2017, pp. 786–798. [Pg.20]

[82] C. Davis, "The norm of the schur product operation," *Numer. Math.*, vol. 4, no. 1, pp. 343–344, Dec. 1962. [Pg.20]

[83] A. V. Oppenheim and R. W. Schafer, *Discrete-time signal processing*. Prentice Hall, 1989. [Pg.20], [Pg.62]

[84] W. Dunham, *Euler: The Master of Us All*. Mathematical Association of America, 1999. [Pg.20]

[85] T. Lindeberg, *Scale-Space Theory in Computer Vision*. Kluwer Academic Publishers, 1994. [Pg.23], [Pg.86]

[86] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, Nov 2004. [Pg.23]

[87] R. Mackowiak, L. Ardizzone, U. Kothe, and C. Rother, "Generative classifiers as a basis for trustworthy image classification," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 2971–2981. [Pg.23]

[88] K. Kahatapitiya and M. S. Ryoo, "Coarse-fine networks for temporal activity detection in videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 8385–8394. [Pg.23]

[89] G. Feng, Z. Hu, L. Zhang, and H. Lu, "Encoder fusion network with co-attention embedding for referring image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 506–15 515. [Pg.23]

[90] R. Gao and K. Grauman, "VisualVoice: Audio-visual speech separation with cross-modal consistency," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 15 495–15 505. [Pg.23]

[91] J. Lee, S.-W. Chung, S. Kim, H.-G. Kang, and K. Sohn, "Looking into your speech: Learning cross-modal affinity for audio-visual speech separation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 1336–1345. [Pg.23]

[92] Y. He, D. Yang, H. Roth, C. Zhao, and D. Xu, "Dints: Differentiable neural network topology search for 3d medical image segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 5841–5850. [Pg.23]

[93] S. Reiss, C. Seibold, A. Freytag, E. Rodner, and R. Stiefelhagen, "Every annotation counts: Multi-label deep supervision for medical image segmen-

tation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 9532–9542. [Pg.23]

[94] W. Ji, S. Yu, J. Wu, K. Ma, C. Bian, Q. Bi, J. Li, H. Liu, L. Cheng, and Y. Zheng, "Learning calibrated medical image segmentation via multi-rater agreement modeling," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 12 341–12 351. [Pg.23]

[95] H. Wu, Y. Gao, X. Guo, Z. Al-Halah, S. Rennie, K. Grauman, and R. Feris, "Fashion IQ: A new dataset towards retrieving images by natural language feedback," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 11 307–11 317. [Pg.23]

[96] X. Wang, X. Shu, Z. Zhang, B. Jiang, Y. Wang, Y. Tian, and F. Wu, "Towards more flexible and accurate object tracking with natural language: Algorithms and benchmark," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021, pp. 13 763–13 773. [Pg.23]

[97] Q. Liu, L. Chen, Y. Yuan, and H. Wu, "History reuse and bag-of-words loss for long summary generation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–1, 2021. [Pg.23]

[98] M. Yan, C. Chen, J. Du, X. Peng, J. T. Zhou, and Z. Zeng, "Memory-assistant collaborative language understanding for artificial intelligence of things," *IEEE Transactions on Industrial Informatics*, pp. 1–1, 2021. [Pg.23]

[99] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. S. Bernstein, A. C. Berg, and F. Li, "Imagenet large scale visual recognition challenge," *CoRR*, vol. abs/1409.0575, 2014. [Pg.25], [Pg.26]

[100] F. Foroughi, J. Wang, A. Nemati, Z. Chen, and H. Pei, "MapSegNet: A fully automated model based on the encoder-decoder architecture for indoor map segmentation," *IEEE Access*, vol. 9, pp. 101 530–101 542, 2021. [Pg.27]

[101] Y. Benkhoui, T. El-Korchi, and R. Ludwig, "Automatic crack segmentation in pavements using a dilated encoder-decoder network," in *2021 4th International Conference on Information and Computer Technologies (ICICT)*, 2021, pp. 88–92. [Pg.27]

[102] I. A. Kazerouni, G. Dooly, and D. Toal, "Ghost-UNet: An asymmetric encoder-decoder architecture for semantic segmentation from scratch," *IEEE Access*, vol. 9, pp. 97 457–97 465, 2021. [Pg.27]

[103] H. Wang, J. Dong, B. Cheng, and J. Feng, "PVRED: A position-velocity recurrent encoder-decoder for human motion prediction," *IEEE Transactions on Image Processing*, vol. 30, pp. 6096–6106, 2021. [Pg.27]

[104] M. Arif and A. Mahalanobis, "Infra-red target recognition using realistic training images generated by modifying latent features of an encoder-decoder network," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2021. [Pg.27]

[105] S. Gao, J. Zhu, and H. Xi, "Attention-based encoder-decoder network for single image dehazing," in *2021 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2021, pp. 1–6. [Pg.27]

[106] S. D. Da Cruz, B. Taetz, T. Stifter, and D. Stricker, "Illumination normalization by partially impossible encoder-decoder cost function," in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2021, pp. 1458–1467. [Pg.27]

[107] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, March 2017. [Pg.29], [Pg.80], [Pg.89], [Pg.127]

[108] M. S. M. Sajjadi, B. Scholkopf, and M. Hirsch, "EnhanceNet: Single image super-resolution through automated texture synthesis," in *The IEEE International Conference on Computer Vision (ICCV)*, 2017. [Pg.29], [Pg.89]

[109] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *CoRR*, vol. abs/1701.05957, 2017. [Pg.29], [Pg.89]

[110] Q. Yang, P. Yan, Y. Zhang, H. Yu, Y. Shi, X. Mou, M. K. Kalra, Y. Zhang, L. Sun, and G. Wang, "Low-dose CT image denoising using a generative adversarial network with wasserstein distance and perceptual loss," *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1348–1357, June 2018. [Pg.29], [Pg.89]

[111] C. Wang, C. Xu, C. Wang, and D. Tao, "Perceptual adversarial networks for image-to-image transformation," *IEEE Transactions on Image Processing*, vol. 27, no. 8, pp. 4066–4079, Aug 2018. [Pg.29], [Pg.89]

[112] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *The European Conference on Computer Vision (ECCV)*, 2018. [Pg.29], [Pg.89]

[113] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. [Pg.29], [Pg.89]

[114] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems Computers, 2003*, vol. 2, 2003, pp. 1398–1402 Vol.2. [Pg.31], [Pg.48], [Pg.82], [Pg.91], [Pg.111]

[115] Zhou Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, 2002. [Pg.31], [Pg.47], [Pg.82], [Pg.91], [Pg.111]

[116] H. Sheikh and A. Bovik, "Image information and visual quality," *IEEE Transactions on Image Processing*, vol. 15, no. 2, pp. 430–444, 2006. [Pg.32], [Pg.48], [Pg.82], [Pg.91], [Pg.110]

[117] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *CVPR*, 2018. [Pg.32], [Pg.91], [Pg.111]

[118] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <0.5mb model size," *arXiv:1602.07360*, 2016. [Pg.32]

[119] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Transactions on Image Processing*, vol. 20, no. 8, pp. 2378–2386, 2011. [Pg.32], [Pg.91], [Pg.111]

[120] G. Sharma, W. Wu, and E. N. Dalal, "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations," *Color Research and Application*, vol. 30, no. 1, pp. 21–30, 2005. [Pg.33], [Pg.91]

[121] R. Reisenhofer, S. Bosse, G. Kutyniok, and T. Wiegand, "A haar wavelet-based perceptual similarity index for image quality assessment," *Signal Processing: Image Communication*, vol. 61, pp. 33–43, 2018. [Pg.33], [Pg.91]

[122] W. Xue, L. Zhang, X. Mou, and A. C. Bovik, "Gradient magnitude similarity deviation: A highly efficient perceptual image quality index," *IEEE Transactions on Image Processing*, vol. 23, no. 2, pp. 684–695, 2014. [Pg.34], [Pg.91], [Pg.111]

[123] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer." *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013. [Pg.35], [Pg.91], [Pg.111]

[124] V. N, P. D, M. Chandrasekhar, S. Channappayya, and S. Medasani, "Blind image quality evaluation using perception based features," in *2015 Twenty First National Conference on Communications (NCC)*, 2015, pp. 1–6. [Pg.35], [Pg.111]

[125] V. N, P. D, M. C. Bh, S. S. Channappayya, and S. S. Medasani, "Blind image quality evaluation using perception based features," in *2015 Twenty*

First National Conference on Communications (NCC), 2015, pp. 1–6. [Pg.35]

[126] A. Mittal, A. K. Moorthy, and A. C. Bovik, "No-reference image quality assessment in the spatial domain," *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, 2012. [Pg.36], [Pg.92], [Pg.111]

[127] M. A. Saad, A. C. Bovik, and C. Charrier, "Blind image quality assessment: A natural scene statistics approach in the dct domain," *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3339–3352, Aug 2012. [Pg.36], [Pg.91]

[128] C. O. Ancuti, C. Ancuti, R. Timofte, and C. D. Vleeschouwer, "O-HAZE: A dehazing benchmark with real hazy and haze-free outdoor images," in *2018 IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2018, Salt Lake City, UT, USA, June 18-22, 2018*, 2018, pp. 754–762. [Pg.37], [Pg.90]

[129] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, "Benchmarking single-image dehazing and beyond," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 492–505, 2019. [Pg.37], [Pg.91]

[130] W. Shi, J. Caballero, F. Huszar, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. [Pg.40], [Pg.41], [Pg.54]

[131] A. Odena, V. Dumoulin, and C. Olah, "Deconvolution and checkerboard artifacts," *Distill*, 2016. [Pg.41]

[132] X. Mao, C. Shen, and Y.-B. Yang, "Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 2802–2810. [Pg.42]

[133] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 4, pp. 640–651, April 2017. [Pg.42]

[134] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *CoRR*, vol. abs/1502.03167, 2015. [Pg.42], [Pg.43], [Pg.78], [Pg.87], [Pg.107], [Pg.108]

[135] X. Lu, Y. Guo, N. Liu, L. Wan, and T. Fang, "Non-convex joint bilateral guided depth upsampling," *Multimedia Tools and Applications*, vol. 77, no. 12, pp. 15 521–15 544, Jun 2018. [Pg.43]

[136] X. Lu, W. Wang, C. Ma, J. Shen, L. Shao, and F. Porikli, "See more, know more: Unsupervised video object segmentation with co-attention siamese networks," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [Pg.43]

[137] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Transactions on Graphics*, vol. 26, no. 3, p. 96–es, Jul. 2007. [Pg.43]

[138] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. [Pg.45], [Pg.69]

[139] T. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, pp. 2999–3007. [Pg.45]

[140] X. Lu, C. Ma, B. Ni, X. Yang, I. Reid, and M.-H. Yang, "Deep regression tracking with shrinkage loss," in *Computer Vision – ECCV 2018*, 2018, pp. 369–386. [Pg.45]

[141] M. Abadi and A. Agarwal, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Pg.48], [Pg.67]

[142] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, 3rd International Conference for Learning Representations, San Diego, 2015. [Pg.48], [Pg.82], [Pg.91], [Pg.110]

[143] J.-H. Lee, M. Heo, K.-R. Kim, and C.-S. Kim, "Single-image depth estimation based on fourier domain analysis," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [Pg.58]

[144] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *Storage and Retrieval Methods and Applications for Multimedia 2004*, vol. 5307, 2003, pp. 472–480. [Pg.67]

[145] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, May 2011. [Pg.67]

[146] L. Shen, Z. Yue, Q. Chen, F. Feng, and J. Ma, "Deep joint rain and haze removal from a single image," in *2018 24th International Conference on Pattern Recognition (ICPR)*, 2018, pp. 2821–2826. [Pg.75]

[147] C. K. Chui, *An Introduction to Wavelets*. Academic Press Professional, Inc., 1992. [Pg.75]

[148] N. Lian, V. Zagorodnov, and Y. Tan, "Edge-preserving image denoising via optimal color space projection," *IEEE Transactions on Image Processing*, vol. 15, no. 9, Sep. 2006. [Pg.75], [Pg.76]

[149] E. Orhan and X. Pitkow, "Skip connections eliminate singularities," in *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*, 2018. [Pg.88]

[150] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1026–1034. [Pg.88]

[151] J. Chen, J. Chen, H. Chao, and M. Yang, "Image blind denoising with generative adversarial network based noise modeling," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [Pg.89]

[152] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 105–114. [Pg.89]

[153] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin, "Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2018, pp. 814–81 409. [Pg.89]

[154] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas, "Deblurgan: Blind motion deblurring using conditional adversarial networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 8183–8192. [Pg.89]

[155] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 2472–2481. [Pg.90]

[156] R. Collobert, K. Kavukcuoglu, and C. Farabet, "Torch7: A matlab-like environment for machine learning," in *BigLearn, NIPS Workshop*, 2011. [Pg.91], [Pg.96]

[157] C. G. Bampis, P. Gupta, R. Soundararajan, and A. C. Bovik, "SpEED-QA: Spatial efficient entropic differencing for image and video quality," *IEEE Signal Processing Letters*, vol. 24, no. 9, pp. 1333–1337, Sep. 2017. [Pg.91]

[158] https://www.tensorflow.org/api_docs/python/tf/image/total_variation. [Pg.91]

[159] H. Zhang, V. Sindagi, and V. M. Patel, "Multi-scale single image dehazing using perceptual pyramid deep network," in *2018 IEEE/CVF Conference*

on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, pp. 1015–101 509. [Pg.93], [Pg.94], [Pg.95], [Pg.96]

[160] MATLAB. The MathWorks Inc. [Pg.96]

[161] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014. [Pg.96]

[162] F. Chollet *et al.*, "Keras," 2015. [Pg.96]

[163] M. S. M. Sajjadi, R. Vemulapalli, and M. Brown, "Frame-Recurrent Video Super-Resolution," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. [Pg.104]

[164] N. Divakar and R. V. Babu, "Image denoising via cnns: An adversarial approach," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017, pp. 1076–1083. [Pg.105]

[165] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014. [Pg.105], [Pg.109]

[166] J. F. Nash, "Equilibrium points in n-person games," *Proceedings of the National Academy of Sciences*, vol. 36, no. 1, pp. 48–49, 1950. [Pg.105]

[167] Nai-Xiang Lian, V. Zagorodnov, and Yap-Peng Tan, "Edge-preserving image denoising via optimal color space projection," *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2575–2587, 2006. [Pg.106]

[168] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, 2016. [Pg.109]

[169] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017. [Pg.110]

भारतीय प्रौद्योगिकी संस्थान गुवाहाटी
Indian Institute of Technology Guwahati

Appendix: A

Related to thesis

- **Journals**

1. P. K. Sharma, S. Ghosh, and A. Sur. "High-quality Frame Recurrent Video Deraining with Multi-contextual Adversarial Network". *ACM Transactions on Multimedia Computing, Communications, and Applications,* vol. 17, no.2, (may. 2021).
2. P. K. Sharma, S. Basavaraju, and A. Sur. "High-resolution image deraining using conditional GAN with sub-pixel upscaling". *Multimedia Tools and Applications,* (Jan. 2021), pp. 1075-1094. issn: 1573-7721..
3. P. K. Sharma, S. Basavaraju, and A. Sur. "Deep learning-based image deraining using discrete Fourier transformation". *The Visual Computer,* (Sept. 2020). issn: 1432- 2315.

- **Conferences**

1. P. Sharma, P. Jain, and A. Sur. "Scale-aware Conditional Generative Adversarial Network for Image Dehazing". *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV),* Mar. 2020, USA.
2. P. K. Sharma, P. Jain, and A. Sur. "Dual-Domain Single Image De-Raining Using Conditional Generative Adversarial Network". *IEEE*

International Conference on Image Processing (ICIP), 2019, pp. 2796-2800.

Others

- **Published**

1. P. K. Sharma, A. Abraham, and V. N. Rajendiran. "A Generalized Framework for Deep Neural Network Compression," in *IEEE Transactions on Multimedia,* Dec. 2021.
2. P. K. Sharma, I. Bisht, and A. Sur. "Wavelength-based Attributed Deep Neural Network for Underwater Image Restoration". *ACM Transactions on Multimedia Computing, Communications, and Applications,* Jan. 2022.
3. S. Basavaraju, P. K. Sharma, and A. Sur. "Memorability based image to image translation". In *Twelfth International Conference on Machine Vision (ICMV 2019),* Vol. 11433. International Society for Optics and Photonics. SPIE, 2020, pp. 395-401.
4. B. Singh, P. K. Sharma, R. Saxena, A. Sur, and P. Mitra, "A New Steganalysis Method Using Densely Connected ConvNets". In *Pattern Recognition and Machine Intelligence,* 2019, pp. 277-285. isbn: 978-3-030-34869-4.
5. A. Ignatov, J. Patel, R. Timofte, ..., P. K. Sharma, and A. Sur, "AIM 2019 Challenge on Bokeh Effect Synthesis: Methods and Results". In *IEEE/CVF International Conference on Computer Vision Workshop (ICCVW),* 2019, pp. 3591- 3598.
6. S. Gu, R. Timofte, R. Zhang, .. , P. K. Sharma, .., G. Ozbulak. "NTIRE 2019 Challenge on Image Colorization: Report", In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops,* June 2019.

- **Under review**

1. R. Jain, P. K. Sharma, S. Gaj, A. Sur, and P. Ghosh, "Knee Osteoarthritis Severity Prediction using an Attentive Multi-Scale Deep Convolutional Neural Network". Under review in *Springer journal of Multimedia Tools and Applications (2021)*.

2. B. Singh, P. K. Sharma, S. Anil, A. Sur, and P. Mitra, "StegGAN: Hiding Image within Image using Conditional Generative Adversarial Network". Under review *In Springer journal of Multimedia Tools and Applications (2021)*.