IMPLEMENTATION AND ANALYSIS OF A SIMPLE DEEP LEARNING TECHNIQUE FOR SINGLE IMAGE DE-RAINING

A PROJECT REPORT

Submitted by

BL.EN.U4ECE17106	M. Krishna Chaitanya
BL.EN.U4ECE17138	N. Vishwa Tej
BL.EN.U4ECE17144	P. Vignesh

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

ELECTRONICS & COMMUNICATION

ENGINEERING

AMRITA SCHOOL OF ENGINEERING, BENGALURU

AMRITA VISHWA VIDYAPEETHAM

BENGALURU 560 035

MAY - 2021

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF ENGINEERING, BENGALURU, 560035

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"IMPLEMENTATION AND ANALYSIS OF A SIMPLE DEEP LEARNING TECHNIQUE FOR SINGLE IMAGE DE-RAINING"** submitted by

BL.EN.U4ECE17106	M. Krishna. Chaitanya
BL.EN.U4ECE17138	N. Vishwa Tej
BL.EN.U4ECE17144	P. Vignesh

In partial fulfilment of the requirements for the award **Degree Bachelor of Technology** in "**Electronics & Communications Engineering**" is a bonafide record of the work carried out under our guidance and supervision at Amrita School of Engineering, Bengaluru.

Mrs. Latha (Project Supervisor)
Assistant Professor (S.G.)
Department of ECE
Amrita School of Engineering
Bangalore

Dr. Navin Kumar
Professor and Chairman
Department of ECE
Amrita School of Engineering
Bangalore

This project report was evaluated by us on ……

|Mrs Bhavana V Mr Satish Kumar Dr Sreeja K Mrs Vinodini M

ACKNOWLEDGEMENT

First and foremost, we would like to express our sincere gratitude to our beloved **Amma**, **Mata Amritanandamayi Devi** for her blessings.

We would like to express our earnest gratefulness to our guide, **Mrs. Latha**, Assistant Professor (S.G.), Department of Electronics and Communication Engineering, Amrita Vishwa Vidyapeetham, Bengaluru whose valuable guidance, keen interest and encouragement at various stages through our learning period ensuring that we worked sincerely and efficiently.

We are very thankful to **Dr. Navin Kumar**, Chairman, Department of Electronics and Communication Engineering, for his encouragement and support that he extended throughout the course of this project.

We would also like to extend our deepest gratitude to **Mr. Sriram Devanathan**, Principal, Amrita School of Engineering, Bengaluru. We are greatly thankful to the entire management team as well, for giving us the wonderful opportunity and the resources needed to undertake this project, and the help extended in finishing it within the time allotted.

Finally, we are extremely thankful to our parents and friends, who were sources of constant motivation and support.

ABSTRACT

In this work, we intend to remove imperfections caused by rain on images that are either affected by real rain or synthetic rain streaks. Such images can negatively affect multiple computer vision tasks and hamper image quality in various security systems, autonomous driving or object detection. To achieve superiority over these effects, we implement a Convolution Neural Network (CNN) architecture that is trained on data that consist of both synthetic and real rain images to improve image quality by eliminating rain streaks.

The training data consists of image pairs - each pair having a ground truth and a rain affected image. ClearCNN, the name of our CNN model, consists of feature blocks that help in extracting necessary streak details and map residue information that will be removed using a subtraction function in the next block of the model. The results are evaluated and summarized using multiple performance metrics that include Peak Signal-To-Noise Ratio (PSNR), Structural Similarity Index (SSIM), BRISQUE (Blind Referenceless Image Spatial Quality Evaluator) and NIQE (Naturalness Image Quality Evaluator). There have been numerous approaches in which various algorithms and methods were used to implement image de-raining. Comparison of our model with the state-of-the-art models clearly prove that ClearCNN achieves comparable de-rained outputs (PSNR - 36.11dB & SSIM - 0.97, Dataset - 100L) both when trained on synthetic and real rain image datasets. It is noteworthy that ClearCNN uses a structure with reduced complexity.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF SYMBOLS

1. **f** - represents the matrix data of our original image
2. **g** - represents the matrix data
3. of our degraded image in question
4. **m** - represents the numbers of rows of pixels of the images
5. **i** represents the index of that row
6. **n** - represents the number of columns of pixels of the image
7. **j** represents the index of that column
8. MAX_f- is the maximum signal value that exists in our original "known to be good image
9. σ_x **-** Luminance
10. μ_x **-** Contrast
11. L_c - Confidence guided loss
12. L_{Er} - Per Pixel Euclidean Loss
13. $\phi_{x,y}^{k}(sD, C)$ - Downsampling Operation
14. f_{out}^{k} - Output general features
15. **R** - Residue or rain Streaks
16. **Y** - Input rainy Image
17. **X** - Detained Image

LIST OF FIGURES

ABBREVIATIONS

1.	**CNN**	-	Convolutional Neural Networks
2.	**PSNR**	-	Peak Signal-to-Noise Ratio
3.	**SSIM**	-	Structural Similarity Index Metric
4.	**SPANet**	-	SPatial Attentive Network
5.	**MPID**	-	Multi-Purpose Image de-raining
6.	**UMRL**	-	Uncertainty Guided Multi-scale Residual
7.	**DRDNet**	-	Detailed Recovery Image de-raining Network
8.	**SE**	-	Squeeze and Excite
9.	**SDCAB**	-	Structure Detail Context Aggregation Block
10.	**SDE**	-	Spatially Adaptive de-raining Module
11.	**CW**	-	Channel Wise de-raining Scheme
12.	**ResNet**	-	Residual Network
13.	**MSGMFFNet**	-	Multi-scale Loss Guided Multiple FeatureFusion De-raining Network
14.	**MSPFN**	-	Multi Scale Progressive Fusion Network
15.	**PDIP**	-	Principal Direction of Image Path
16.	**SVCC**	-	Sensitivity of Variance of Coloured Channels
17.	**SMReNE**	-	
18.	**ResGuideNet**	-	Residue Guided Network
19.	**RFFF**	-	Residual Project Feature Fusion
20.	**PRN**	-	Progressive Residual Network
21.	**PreNet**	-	Progressive Recurrent Network
22.	**LSTM**	-	Long Short Term Memory Network
23.	**CGAN**	-	Conditional Generative Adversarial Network
24.	**OUCD**	-	Over and Under-complete Deep Network
25.	**SemiDerainGAN**	-	Semi De-rained Generative Adversarial Network
26.	**SSRML**	-	Semi Supervised Rain Streak Learner
27.	**RCDNet**	-	Rain Convolutional Dictionary Network
28.	**SNet**	-	Streak Network
29.	**VNet**	-	Vapour Network
30.	**BRISQUE**	-	Blind Referenceless Image Spatial Quality

		Evaluator
31. **NIQE**	-	Naturalness Image Quality Evaluator
32. **PIQE**	-	Perception based Image Quality Evaluator
33. **ReLU**	-	Rectified Linear Unit

1

INTRODUCTION

1.1 Image De-raining

Images taken during the rain often suffer from visual degradation. Image de-raining is a technique that is used to overcome such harsh effects on images to produce clean, crisp output. Factors such as scattering and refraction can also negatively contribute to image quality. Image de-raining thus has numerous advantages and can be used confidently in the many fields that include surveillance, autonomous vehicles and many such use cases that involve crucial imaging. Building and orchestrating competent algorithms for the same is necessary.

1.2 Neural Networks

Neural Networks are the information Processing systems which are constructed and implemented to model the human brain. This network is usually implemented by using electronic components or is simulated in software on a digital computer. These tasks are very difficult for traditional computers; hence high-speed digital computers are required to simulate the neural networks.

1.2.1 Convolutional Neural Networks (CNN)

Convolutional Neural Network typically called convnet is a class of deep neural networks that is applied in the field of visual imagery. CNNs are also considered to be regularized versions of multi-layer perceptrons. But instead of being fully connected like how perceptrons are, CNNs take advantage of the pattern in the data used to assemble them in increasing complexity.

The first layer is called the convolutional layer. Each neuron in the convolutional layer only processes the information from a small part of the visual field. Input features are taken batch-wise, like a filter. The network understands the images in parts and can compute these operations multiple times to complete the full image processing. It is a go to method for image processing. Propagation is unidirectional where CNN contains one or more convolutional layers followed by pooling. The output of the convolutional layer goes to a fully connected neural network for classifying the images.

A CNN layer has a lot of hyper parameters that include the filter / kernel size that are defined by width and height, number of input and output channels, padding, stride and dilation, which are optimized to achieve maximum efficiency and the best results in the resources available.

Biological processes are considered to be the inspiration behind CNNs [37]. The connectivity patterns in the neurons closely replicate the animal visual cortex.

CNNs use relatively less complexity as compared to other image classification algorithms. This means that the network learns to optimize the filters through automated learning as compared to other traditional algorithms.

In the field of image processing, the reason for not using fully connected networks is because of its complexity. In Fully connected networks, each neuron is connected to every other neuron in a particular layer. This might be an efficient method for images with smaller dimensions but as the size of image increases the network becomes more complex and time consuming. This is the reason for using CNN as the most preferred architecture in this field.

1.3 Related Study

The past decade has seen momentous development in the field of image de-raining. Each solution proposed, attempts to carefully construct the physical characteristics of rain streaks. Pyramid structures were used by Kui Jiand et. al. and Xueyang Fu et. al. to simplify the learning problem at each neural network

level [1][12], along with the usage of residual and recurrent networks. Dong Dong Chen et. al. used a different approach by fusing features from multiple levels, by using a gated subnetwork [2]. Tianyu Wang et. al. attempts to condense pre existing datasets into one single group of 29.5k images. SPANet [4] is introduced to remove rain streaks in a local to global discipline.

Siyuan Li et. al. derived the MPID - Multi-purpose image de-raining benchmark, which contains data images affected by different volumes and types of rain streaks [3]. These image datasets contain both synthetic and real rain image datasets. Rajeev Yasarla et. al. introduces UMRL - Uncertainty guided Multi-scale Residual Learning [5] that attempts to gather rain content at different scales to estimate final output. Yanyan Wei et. al. introduces an approach to obtain negative rain streaks using a dual path residual dense block. Sen Deng et. al. proposes DRDNet - Detail Recovery image De-raining Network, where rain removal and detail recovery is done in consecutive steps, rather than simultaneously de-raining and preserving details in a closely-knit network [6].

The two parallel subnetworks are the SE (Squeeze and Excite) Network to eliminate rain streaks, and the SDCAB (Structure Detail Context Aggregation Block) to amplify the lost details to reduce image degradations. Yingjun Du et. al. takes into account the effect spatial location and colour channels have on rain intensity. Inside the Autoencoder model used, an SDE (Spatially adaptive de-raining module) is used to estimate a rain information mapping for each image in the given dataset. This is followed by Channel Wise (CW) de-raining scheme to address the problem of colour channels having an effect over the intensity of rain streaks on the affected image [7].

Xueyang Fu et. al. introduces a new Deep Learning Architecture for de-raining that is built on the fundamentals of Deep Convolutional Neural Networks (Deep CNNs) [8]. This network is inspired by ResNet - simplifies the learning curve by deliberately bringing changes to the mapping. Background Interference is removed using priori-domain knowledge. This model has benefitted both high and

low level vision imaging problems. DerainNet is used to implement Deep CNNs in another approach used by the same author [9].

This approach is said to have improved rain removal and faster computation. Huiyan Fu et. al. has worked on a network named Multi-scale Loss Guided Multiple feature fusion de-raining network (MSGMFFNet). Pre-processing steps include gamma correction and contrast enhancement. A rain streak attention map is generated, and the L1 and edge loss functions are incorporated [10]. The Multi-Scale Progressive Fusion Network (MSPFN) designed by Kui Jiang et. al. extracts rain information and deep features from input images scaled to different sizes.

Recurrent Network blocks are also used to capture global texture [11]. Yinglong Wang et. al. pursues a thought where images are separated into low and high frequency parts, with most rain components existing in the high frequency parts. In addition to this PDIP (Principal direction of an Image Path) and SVCC (Sensitivity of Variance of Coloured Channels) are used to enhance image output quality [12]. Atmospheric scattering and veiling effects of rain streaks are studied and removed in the work done by Ruoteng et. al. using a model called SMReNE, which is an adaptation of parallel sub networks [23]. H Zhang et. al. adapts an approach wherein rain streaks are classified according to their densities, before they are eliminated [14]. Zhi Weng et. al. attempts to build a single and a separable network, namely ResGuideNet, which collects residue from shallow blocks and guides deeper blocks. They are also supervised at every block [15]. An hourglass model was proposed by Xiang Cheng et. al. which up-samples and down-samples images at different scales and excavates features. The reason behind this being the number of features derived are much higher than when extracted from a single scale.

To produce realistic results, an RFFF (Residual project Feature Fusion) was introduced [24]. Dongwei Ren et. al. proposed PRN and PreNet, wherein intra-stage recursive computation is developed, along with reduction in network

parameters while maintaining performance. This was also followed by a technique of taking stage-wise results and feeding the original rainy image as input into each ResNet. Another adaptation of Recurrent Networks and LSTMs were proposed which concentrate both on streak-removal and the background layer [17, 18]. Yupei Zheng et. al. also develops residual frameworks which remove rain streaks in a coarse-to-fine fashion, i.e. the heavy rain is removed in the coarse level and the light rain is removed in the fine level.

It also studies the changes and improvement in de-raining at each level and compares it with the original rainy image to achieve comparable results [19]. H Zhang et. al. proposes a CGAN (Conditional Generative Adversarial Network) where a discriminator is used as a reference to orchestrate images that are free from climate-based degeneration. This generator acts as a mapping function to render an input rainy sample into a de-rained output sample [20]. Rajeev Yasarla et. al. develops a model called OUCD - Over and Under-complete Deep network for image de-raining. This network extracts local features as well as global features, especially the explicit extraction of local features makes the de-raining process more efficient. Cong Wang et. al. refers to the pyramid architecture and hence the model contains two modules, the capture and extraction from different scales. SE and skip connections are implemented to enhance relation between channels and to improve information transmission from low to high level [22]. Wenhan Yang et. al. attempts summarizing all the state-of-the-art models at the time based on the method of streak removal and results [25]. Yanyan et. al. uses a data driven approach to beat the problem of over-de-rained and under-de-rained images. A semi-supervised network, SemiDerainGAN is used to implement the same.

The proposed model, SSRML, can learn rain streak information from both synthetic and real world images [26]. Hong Wang et. al. deploys the proximal gradient descent technique to design an iterative algorithm containing simple operations to solve the model. RCDNet - Rain Convolutional Dictionary Network is developed to encode rain streak structure [27]. Yinglong et. al. proposes an encoder-decoder CNN called SNet to learn the transmission map of rain streaks. This work also considers the vapur factor which is brought in by the rain, and

hence VNet is proposed to predict the map for vapour in a multi-scale pyramid structure. ANet is an encoder CNN that is used to estimate atmospheric light. The networks are joint-trained to estimate the transmission map for rain image restoration [28].

1.4 Problem Statement

Single Image de-raining is an extremely challenging problem and has grabbed much attention in the recent past. Previous approaches have attempted to address this problem but these are mainly evaluated either using synthetic or real dataset. This Project studies previous approaches and tries to build a computationally efficient model for single image deraining for both synthetic and real datasets.

1.5 Motivation

Numerous single image de-raining algorithms have been recently proposed. However, these algorithms are very complex and require a huge amount of computational cost. Moreover, these models are mostly implemented using synthetic datasets. It is absolutely necessary to train the model on real datasets for better accuracy. Therefore, it is required to build a model which is less complex and which is implemented on both synthetic and real datasets. Our evaluation and analysis will allow us to better identify the strengths and limitations of our algorithm and comprehensively help us identify the use cases for future research purposes.

1.6 Objective

The main objective of this work is to study and understand the design and working of various approaches used for implementing image de-raining. We did an extensive literature survey on some of the best architectures and tried to build a model which can be implemented at a lesser computational cost. We have implemented this model in an unbiased environment, which enables us to derive the best performance metrics (PSNR, SSIM, BRISQUE, etc.), thereby resulting in a sharp output image for an input rainy, degraded image.

1.7 Methodology

The Proposed model is an end-to-end implementation of CNN to satisfy the objective of our project. These are the following stages /blocks of the proposed Model:

1. Pre-Processing
2. Feature Block-1
3. Feature Block-2
4. Residual Block

In the pre-processing stage the input rainy image is resized to (128,128). We intend to pass the pre-processed images through two feature blocks. Each of these feature blocks contains 3 convolutional layers which adds up to a total of 6 hidden layers along with one input and one output convolutional layers. The layers in each of these feature blocks contain a different set of hyper-parameters. The hyper-Parameters that are considered in these feature blocks are as follows:

- No of Filters
- Kernel Size
- Stride Length
- Activation Function
- No of Epochs
- Loss Function
- Batch Size and Batch Normalization

The notable differences between the two feature blocks are the no of filters that are being taken. To maximize feature Extraction and to make the model more reliable, the first feature block contains 128 filters and the second feature block contains 64 filters. The last block in the model is called the residual block. The output images from the second feature block are passed to this residual block. This output image is then subtracted from the original input image and the

noise/residue gets stored in this block. The subtracted image is the final de-rained image.

1.8 Expected Result

The output contains de-rained images of the Proposed model. The full reference (PSNR, SSIM) and the no-reference (BRISQUE, NIQE) metrics of the model are evaluated and noted in tabular form.

1.9 Contents of the rest of the report

The rest of this paper is outlined as follows:

Chapter 2: Provides Background and methodologies of the relevant literature which are state of the art Image de-raining methods.

Chapter 3: It is the Design and Analysis section which discusses the design of the implemented model along with the analysis of each block.

Chapter 4: It provides the results and performance of the implemented system.

Chapter 5: Conclusion and future scope are discussed in this section.

2

RECENT AND RELEVANT WORK

This Chapter discusses the various methodologies and approaches related to single image de-raining followed by various authors in different publications. It also discusses the challenges faced by them in each work.

R. Yasarla et al. [5] proposes a model which is called Uncertainty guided multiscale residual learning (UMRL) which focuses on the location information of the rain and attempts to address the issue by learning the rain content at different scales and using them to predict the final output. The proposed model UMRL generates the rain streak content at each location of image along with the uncertainty map which gives the information about rain streak content at each location. A cycle spinning technique is also introduced in both training and testing to improve final de-raining performance. In Cycle spinning, the data is first shifted by some amount and then it is denoised, the denoised data is then unshifted and then finally the unshifted data are averaged to obtain the final result. Confidence guided loss of the model is mentioned in Equation (1) and (2).

$$L_c = \sum_{i \in \{\times 1, \times 2, \times 4\}} \left(\sum_j \sum_k log(c_{i_{jk}}) \right) \quad (1)$$

$$L_c = L_1 - \lambda_1 L_c \quad (2)$$

For the given input image, K. Jiang [12] generates Gaussian pyramid images which is the representation of an image in which information of the image is explicitly available and there is no need to compute it. These Gaussian pyramids

are generated using Gaussian kernels. The reason for generating Gaussian pyramid images is to down sample the images to ½ and ¼ times the original size.

This is done to uncover the correlations of rain streaks in an image at different scales. The network first takes as input the pyramid images and extracts the shallow features through multiple initial convolution layers. Based on these input features, the Coarse Fusion Module is activated to exploit repetition of rain streaks under the same scale. The outputs of CFM will go through the Fine Fusion Module to refine the co-related information in CFM. A Channel attention unit (CAU) is introduced to enhance the discriminative learning ability of the network through focusing on most informative specific scale knowledge. The outputs of CFM are concatenated with outputs of FFM and are sent to Reconstruction Module(RM) where the iterative sampling and fusion of rain information across different pyramid layers are implemented to get residual rain image.

H. Zhang et al. [14] focuses on density of rain (heavy, medium, light) and rain streak removal. In the previous approaches, they used basically a single network for image de-raining which often over-de-rain or under-de-rain the image mainly following two steps which are Rain density classification and Rain streak removal. In the first Step the rain density information (heavy, medium, light) is classified from the input image using residual aware classifier. In the second step the rain streak removal is performed using a multi-stream dense network. These two steps are then fused for the final image de-raining. Along with this, a large dataset with 12,000 images of rain density labels is also developed. The datasets used for the model are Discriminative sparse coding based method, Gaussian mixture model based method CNN method, Joint Rain Detection and Removal (JORDER) method, Deep detailed Network method, Joint Bi-layer Optimization.For the above model, de-raining network loss is mentioned in equation (3) and (4).

$$L_{Er} + L_{C} \tag{3}$$

$$L_{Er} + L_{Ed} + \lambda_F L_F \quad (4)$$

L_{EF}represents the per-pixel Euclidean loss function to reconstruct the de-rained image and L_F is the feature based loss for the de-rained image. Ren et al. [17] proposes two de-raining networks called PRN and PreNet by which better and simpler networks can work well in removing rain streaks and provide a suitable basis to future studies on image de-raining. By taking advantage of intra-stage recursive computation, PRNr and PReNet are also suggested to reduce network parameters while maintaining state-of-the- art de-raining performance. Using PRN and PReNet, the de-raining performance can be further improved by taking both stage-wise result and original rainy image as input to each ResNet, and our progressive networks can be readily trained with single negative SSIM or MSE loss. Extensive experiments show that our baseline networks are computationally very efficient, and perform favorably against state-of-the-arts on both synthetic and real rainy images.

The same author here [18] follows a deep CNN model for single image de-raining. Instead of following standard CNN models which concentrate on only rain-streak removal and neglects the background layer of image, it concentrates on both the things. In the first step a Single Recurrent network SRN is developed which recursively unfolds a shallow network, and propagates deep features across multiple stages. Two SRN's are used via LSTM which removes rain streaks and produces a clearer background.

In the last step a Bilateral Recurrent network is used which acts as an interplay between two SRNs and also propagates deep features for rain streak removal and clear background image. The hyper-parameters used are PSNR, SSIM and it is tested on both synthetic and real datasets and achieved better results than state of art models.

H. Zhang et al. [20] develops a model called ID-CGAN (Conditional Adversarial Generator Matrix) where a learned discriminator network is used as a guidance to synthesize images free from weather-based degradations. The use of discriminator for classifying between real/fake samples provides additional feedback, enabling the generator to produce results that are visually similar to the ground-truth clean samples.

The generator acts as a mapping function to translate an input rainy image to de-rained image such that it fools the discriminator, which is trained to distinguish rainy images from images without rain refined perceptual loss to serve as an additional loss function to aid the proposed network in generating visually pleasing outputs.

R. Yasarla et al. [21] develops a model called OUCD (Over and Under complete deep network for image de-raining problem. Previous approaches have considered image de-raining as a simple encoder-decoder problem and have concentrated more on global features of image. The rain streaks in the rainy images are of different densities, directions and thickness and cannot be completely removed by considering only global features.

$$L_{MSE} = \left|\hat{x} - x\right|^2 \tag{5}$$

The standard `2 loss calculated between the predicted $\hat{x}$ and the ground truth x is explained in the above equation (5).

The major contributions of this paper are to develop a model called OUCD for image de-raining problem which extracts local features or deep features as well as global features. Over complete model extracts local features of rain streaks explicitly which makes de-raining process more efficient. In order to not avoid completely the global features an under complete model is developed. Both these models go hand in hand and make the de-raining process more efficient.

R. Li et al. [23] has a primary goal to visually remove the rain streaks and veiling effect caused by scattering and transmission of rain streaks. They have developed a model called SMReNET (Scale aware MultiSca5le Recurrent Neural Network) where they parallel sub-networks for image de-raining. The problems faced by previous approaches is that they have assumed that the distribution of rain-streaks are sparse, but in case of heavy or moderate rain, this assumption becomes invalid and they have neglected the problems caused by overlapping of rain streaks with different densities. To address these issues they have used an end to end dense network for feature extraction which improved the quality of de-rained images. They have used Parallel Sub networks where they have considered different scales of rain streaks in each network which allowed them to break a giant task into smaller sub tasks. They were able to remove the veiling effect caused by atmospheric scattering.

X. Chen et al. [24] develops a model called Multi-Scale Hourglass Hierarchical Fusion (MHHF) for image de-raining. Previous approaches have basically concentrated on either model-driven and data driven methods. Model driven methods basically consider de-raining as an optimization problem which results in a limited approach. Data driven models which are basically deep learning models have shown good results but still there is a scope of improvement. To address these issues, they have constructed a Multi scale Hourglass model which up-sample or down-sample the image at various scales and extracts the features. This helps in extracting many features rather than restricting it to one scale.

$$\phi^{k}_{x,y}(sD, C) = \frac{\psi^{k}_{x,y}(sD)}{\psi^{k}_{x,y}(C) + \epsilon'} \tag{6}$$

The implementation of the down-sampling operation $\phi^{k}_{x,y}(sD, C)$ can be formulated as above Equation (6), where $\psi^{k}_{x,y}(sD)$ is the average pooling, ϵ is a small number to avoid division by zero

A Hierarchical distilled model is constructed which recalibrates the features at each block and neglects the unnecessary or useless features. In the last step, a residual project Feature Fusion (RFFF) which aggregates all the feature maps and produces realistic results rather than simply adding all the features. The hyperparameters used are PSNR and SSIM and compared with all the datasets and compared on all the state-of-the-art models and achieved better results.

W. Yang et al. [25] discusses about two categories of de-raining approaches i.e. model-based and data-driven.

In model-based methods, an optimization framework is employed for de-raining. In these types of methods, the presence of rain accumulation is ignored and focuses mainly on rain streak detection. Another important aspect of model-based methods is sparse coding. Typically, the input vectors are represented as a sparse linear combination of basic vectors. The collection of these basic vectors is called a dictionary which is used to reconstruct the rain streaks in the image.

Another approach is called Data Driven where a deep CNN or a generative adversarial network is developed. Some of the drawbacks of this paper include fusing physical models and real-rain images. The evaluation methodology also needs further development.

The main objective in the work done by C. Wang et al. [29] is to increase the visual effect of the image. A model JDNet is proposed for semantic segmentation and detection of rain streaks. An input rainy image passes through a convolution layer and then followed by an activation function to transform the channel dimension from image to feature. These transformed features are input into several joint units to extract information from rain streaks.

Finally, the rain streaks are obtained by a convolution layer followed by an activation function to convert the features to image. Two modules namely Self aggregation module and Self attention module are introduced to learn features of the image with different scales and feature aggregation respectively. For each spatial location a Self-calibrated convolution is applied to build long range spatial dependencies.

W. Yang et al. [35] focuses on the problem of single image rain removal even in the presence of dense rain streaks and rain accumulation. For this a new model and an architecture is introduced which constructs a binary map which indicates rain streak region and includes various shapes, directions and sizes of overlapping rain streaks. A contextual dilated network is introduced which generates features which are invariant of rain streaks. A recurrent process which progressively removes rain streaks is utilized to handle various shapes, directions and densities of rain streaks. Output general features are represented in equations (7) and (8).

$$f_{in}^{1} = max(0, W_{input} * f_{input} + b_{input}) \quad (7)$$

$$F = f_{out}^{k} \quad (8)$$

In the first step a binary streak map is constructed in which ‘1’ represents presence of rain-streaks and ‘0’ otherwise. In the second step a deep network is built to remove the rain streaks jointly. The automatically detected rain streak regions provide useful information to constrain the rain removal, and enable the network to perform an adaptive operation on rain and non-rain regions, preserving richer details. In the third step a contextualized dilated network is constructed with features which are invariant of rain-streaks.

To demonstrate the superiority of their rain accumulation removal method, they compared the proposed method with other state-of-the-art dehazing methods. As one can observe, the result of Nonlocal has a color shift. Many other models tend to produce darker results and retain some accumulation. Comparatively, their rain

accumulation removal method successfully removes most accumulation and lights up dark details in the images.

They evaluated the effect of over-detection and under-detection of rains on the final performance. They use dilation and erosion operations to dilate and erode the detected rain mask. The numbers after Exp and Shrink denote the changed percentage of areas after dilation and erosion operations, respectively. As one can observe, over-detection and under-detection lead to a performance drop, which indicates that the detected rain mask indeed plays a guided role in rain streak removal of JORDER network.

3

DESIGN AND IMPLEMENTATION

3.1 Network architecture and specifications

This section contains details of hyper-parameters that are necessary to construct any neural network. These are derived by trial-and-error, by individually implementing a set of values and comparing the results with other sets. The values that give the best results (accuracy) are chosen to be the final specifications of the network.

3.1.1 Model Specifications

Two feature blocks are used with **128** and **64** filters respectively with kernel size **3 x 3**. Each feature block contains **3 hidden layers** so the **total number of layers** used are **8**, including the input and output layer. Activation Function used in the model is **ReLU** (Rectified Linear Unit).

3.1.1.1 Filter

Normal neural networks consist of dense layers where all the input data is sent to each neuron on the first layer, each neuron then performs a dot product of the input data and the neuron's weights to produce a single number as output. The outputs of these neurons are then concatenated (joined together) and sent to all of the neurons on the next layer and so the process goes on until the output neuron(s) are reached.

Whereas, in a CNN the weights (in the convolutional layers) are a small matrix (often 3x3) which is dot produced with each pixel to produce a new pixel thus acting as image filters. The new images produced by each neuron/filter in a layer are then combined and then passed as the inputs to every neuron in the next layer and so on until the end of the network is reached (there is often a

single dense layer at the end to turn the image output of the final convolutional layer into the numerical class prediction we are typically training it to produce). Therefore, A filter is a 3x3 matrix of weights that is slid over an image in order to produce a filtered output

3.1.1.2 Kernel Size

Deep neural networks, more concretely convolutional neural networks (CNN), are basically a stack of layers which are defined by the action of a number of filters on the input. Those filters are usually called kernels.
For example, the kernels in the convolutional layer are the convolutional filters. Actually, no convolution is performed, but a cross-correlation. The kernel size here refers to the width x height of the filter mask.

3.1.1.3 Activation function

An activation function in a neural network defines how the weighted sum of the input is transformed into an output from a node or nodes in a layer of the network.

Sometimes the activation function is called a "*transfer function*." If the output range of the activation function is limited, then it may be called a "*squashing function*." Many activation functions are nonlinear and may be referred to as the "*non-linearity*" in the layer or the network design.

The choice of activation function has a large impact on the capability and performance of the neural network, and different activation functions may be used in different parts of the model.

Technically, the activation function is used within or after the internal processing of each node in the network, although networks are designed to use the same activation function for all nodes in a layer

3.1.1.4 Stride Length

The given input image is represented in the form of a 2 dimensional matrix. The features from the input image are extracted using filters or feature detectors which are also represented in the form of 2 dimensional matrix. Stride length is the length or the number of pixels by which the given filter is slided or moved across the input image matrix. When the stride is 1 then we move the filters to 1 pixel at a time. When the stride is 2 then we move the filters to 2 pixels at a time and so on.

3.1.1.5 Loss Function

It is the function which describes the amount of loss or error that occurred in the neural network. It basically tells the robustness of neural networks. The Loss is used to calculate the gradients and gradients are used to update the weights of the Neural Net. This is how a Neural Net is trained.

- **Mean Squared Error**
  MSE loss is used for regression tasks. As the name suggests, this loss is calculated by taking the mean of squared differences between actual (target) and predicted values.

3.1.1.6 Epochs

Epoch is basically no of times the algorithm will work through the entire training dataset. Deep learning often deals with copious amounts of data. This data is

broken down into smaller chunks (called batches) and fed to the neural networks one-by-one. The batch size is the total number of training examples in a batch. Gradient descent is an iterative process and updating the parameters through backpropagation in a single pass (or one epoch) is not enough. One epoch is when the entire dataset is passed forward and backward through the neural network once. In order to generalize the model, we use more than one epoch in the majority of the deep learning models.

The more the number of epochs, the more the parameters are adjusted thus resulting in a better performing model. However, too many epochs might lead to overfitting. If a model is overfitted, it does well in the train data and performs poorly on the test data.

3.1.7 Batch Size

Batch Size is the number of samples passed through while training the dataset. The batch size should always be less than the number of samples because it requires less memory. Since the network is trained using fewer samples, the overall training procedure requires less memory. That's especially important if the whole dataset is not fitted in the machine's memory. Typically networks train faster with mini-batches. That's because the weights are updated after each propagation.

3.2 Performance Metrics

Performance metrics are the metrics/parameters based on which the output image i.e. the de-rained image is being evaluated. There are basically two types of Performance Metrics:

- Full Reference Quality Metrics
- No-reference Quality Metrics

3.2.1 Full Reference Quality Metrics

Full reference metrics are the parameters which depend on original/clean input images for the evaluation. The quality or performance of images in this type of metric are evaluated by comparing clean or input image and output or de-rained image. It is the measure of fidelity. There are three performance metrics that have been considered in our model:

3.2.1.1 Peak Signal to Noise Ratio (PSNR)

The term peak signal-to-noise ratio (PSNR) is an expression for the ratio between the maximum possible value (power) of a signal and the power of distorting noise that affects the quality of its representation. Because many signals have a very wide dynamic range, (ratio between the largest and smallest possible values of a changeable quantity) the PSNR is usually expressed in terms of the logarithmic decibel scale.

Image enhancement or improving the visual quality of a digital image can depend on many factors. For this reason, it is necessary to establish quantitative/empirical measures to compare the effects of image enhancement algorithms on image quality. Using the same set of test images, different image enhancement algorithms can be compared systematically to identify whether a particular algorithm produces better results. The metric under investigation is the peak-signal-to-noise ratio. If it can be shown that an algorithm or set of algorithms can enhance a degraded known image to more closely resemble the original, then it can be more accurately concluded that it is a better algorithm. The same is represented in Figure 3.1. PSNR is defined in eq (9)

$$PSNR = 20\, log_{10}(\frac{MAX_f}{\sqrt{MSE}}) \tag{9}$$

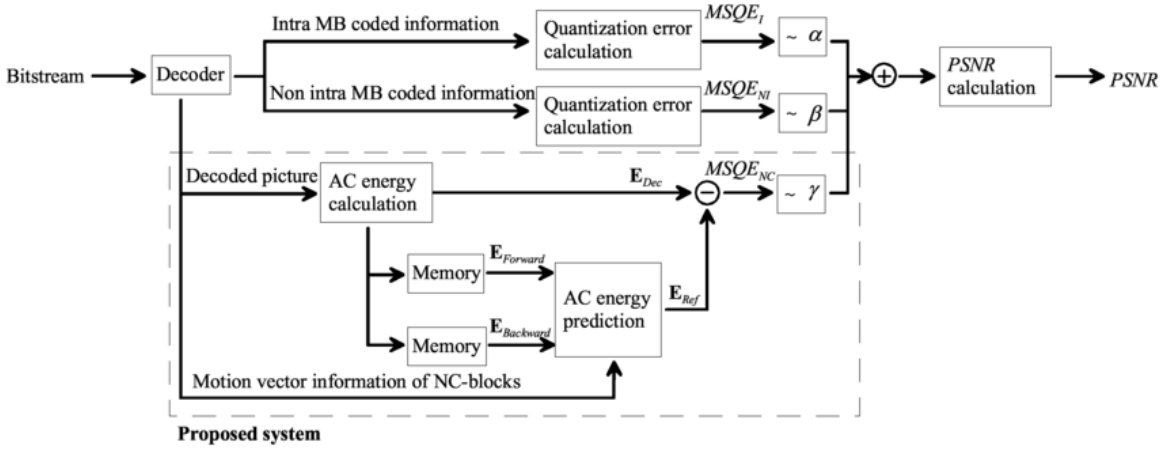

Figure 3.1 - Block Diagram for PSNR [36]

3.2.1.2 Mean Squared error(MSE)

The mean squared error (MSE) tells you how close a regression line is to a set of points. It does this by taking the distances from the points to the regression line (these distances are the "errors") and squaring them. The squaring is necessary to remove any negative signs. It also gives more weight to larger differences. It's called the mean squared error as you're finding the average of a set of errors. The lower the MSE, the better the forecast. MSE (Mean Squared Error) is defined in eq. (10)

$$MSE = \frac{1}{m.n} \sum_{0}^{m-1} \sum_{0}^{n-1} |f(i,j) - g(i,j)|^2 \quad (10)$$

Legend -

f - represents the matrix data of our original image
g - represents the matrix data of our degraded image in question
m - represents the numbers of rows of pixels of the images and i represents the index of that row
n - represents the number of columns of pixels of the image and j represents the index of that column

MAX_f - is the maximum signal value that exists in our original "known to be good" image

3.2.1.3 Structural Similarity Index (SSIM)

Structural Similarity (SSIM) Index in Figure 3.2 is an image quality metric. SSIM index is computed for the image with respect to the reference image. The reference image usually needs to be of perfect quality. This quantitative measure considers three parameters namely luminance, contrast and structural information between the two images to compute the SSIM value. Luminance is measured by averaging over all pixel values. It is denoted by μ_x. Contrast is measured by taking the standard deviation (square root of variance) of all pixel values given in equation (11). It is denoted by σ, and is represented by the following:

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i-1}^{N}(x_i - \mu_x)^2\right) \quad (11)$$

Structure is measured by dividing the input signal with its standard deviation. So, the result from equation (11) and (12) has unit standard deviation which allows for more robust comparison.

$$\frac{(x-\mu_x)}{\sigma_x} \quad (12)$$

Legend:
μ_x - Contrast
σ_x - Luminance

Overall, SSIM can be expressed in eq (13)

$$SSIM(x,y) = (2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + \quad (13)$$

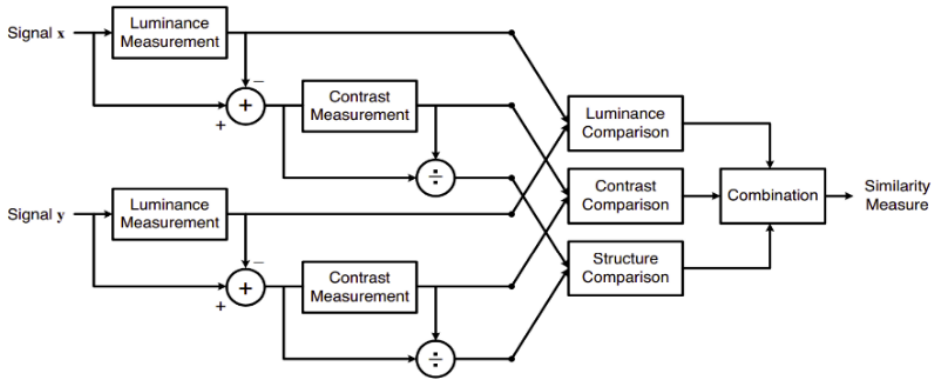

Figure 3.2 - Block Diagram for SSIM [36]

3.2.2 No Reference Quality Metrics

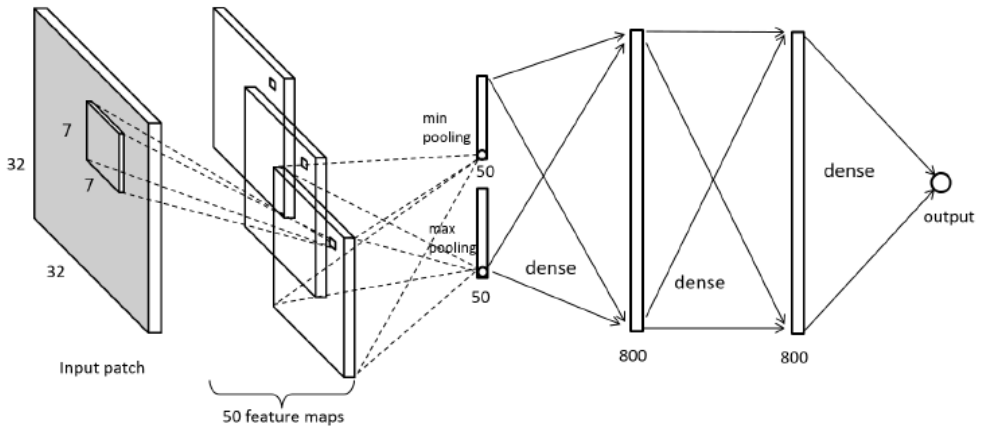

Figure 3.3 Block Diagram for No reference Image Quality Metrics [36]

The most commonly used objective image quality metric is the peak signal to noise ratio (PSNR). However, PSNR does not correlate well with human perception in some cases. Recently, a number of other objective quality metrics have been developed, which consider the human visual system (HVS). Since human observers do not require original images to assess the quality of degraded images, efforts have been made to develop no-reference (NR) metrics that also do not require original images. No Reference Quality metrics in Figure 3.3 does not

depend on other images or which does not have any reference image for the quality assessment. It is the measure of quality. There are two metrics that are considered in our model.

3.2.2.1 Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE)

Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [36] is a natural scene statistic (NSS)-based distortion-generic blind/no-reference (NR) image quality assessment (IQA) model which operates in the spatial domain. It does not compute distortion specific features such as ringing, blur or blocking, but instead uses scene statistics of locally normalized luminance coefficients to quantify possible losses of 'naturalness' in the image due to the presence of distortions, thereby leading to a holistic measure of quality.

The underlying features used derive from the empirical distribution of locally normalized luminances and products of locally normalized luminances under a spatial natural scene statistic model. No transformation to another coordinate frame (DCT, wavelet, etc.) is required, distinguishing it from prior no reference IQA approaches. Despite its simplicity, we are able to show that BRISQUE is statistically better than the full-reference peak signal-to-noise ratio (PSNR) and the structural similarity index (SSIM) and highly competitive to all present-day distortion-generic NR IQA algorithms. BRISQUE has very low computational complexity, making it well suited for real time applications. BRISQUE features may be used for distortion-identification as well.

3.2.2.2 Naturalness Image Quality Evaluator (NIQE)

Natural Image Quality Evaluator (NIQE) blind image quality assessment (IQA) is a completely blind image quality analyzer that only makes use of measurable deviations from statistical regularities observed in natural images, without training

on human-rated distorted images, and, indeed without any exposure to distorted images. However, all current state-of-the-art general purpose no reference (NR) IQA algorithms require knowledge about anticipated distortions in the form of training examples and corresponding human opinion scores.

It is based on the construction of a quality aware collection of statistical features based on a simple and successful space domain natural scene statistic (NSS) model. These features are derived from a corpus of natural, undistorted images. Experimental results show that the new index delivers performance comparable to top performing NR IQA models that require training on large databases of human opinions of distorted images [36].

3.2.2.3 Entropy

Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image. Entropy is defined as the below equation (14)

$$sum(p.* log_2 p) \quad (14)$$

where ***p*** contains the normalized histogram, counts returned from imhist.

By default, entropy uses two bins for logical arrays and 256 bins for uint8, uint16, or double arrays. entropy converts any class other than logical to uint8 for the histogram count calculation so that the pixel values are discrete and directly correspond to a bin value.

3.3 Methodology

As we can see from the design of the architecture, we intend to pass the pre-processed images through two feature blocks. Each of these feature blocks contain 5 convolutional layers with a different set of hyperparameters. One of the major differences between these blocks is the filter size. The first block has 128

filters with a kernel size 3x3 and the second block contains 64 filters with the same kernel size.

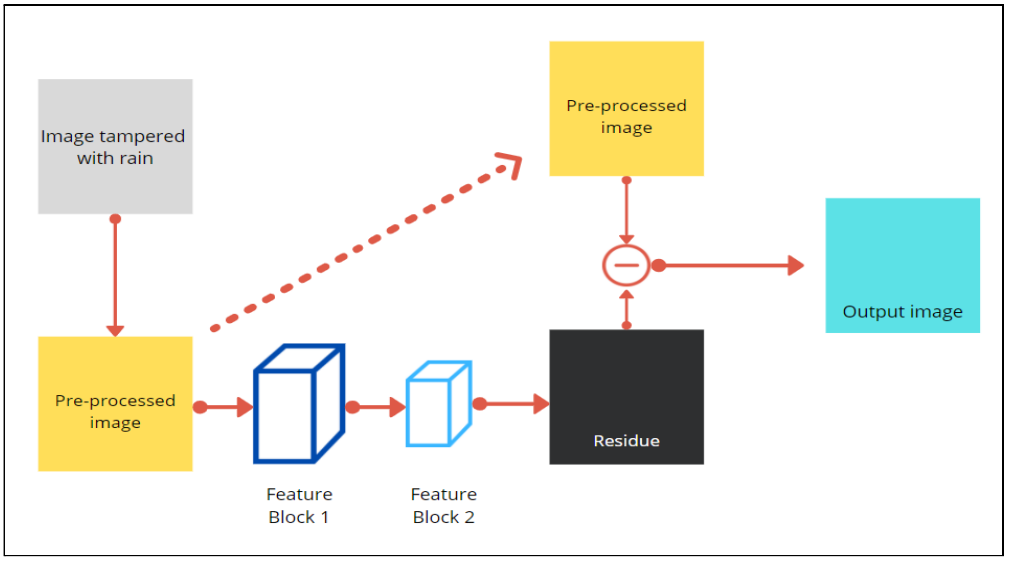

Figure 3.4 - Block Diagram of Proposed Model

The first block in Figure 3.4 tries to get the coarse details in the image, i.e., edges One thing that makes neural networks so interesting is that every subset of layers can be thought of as a neural network itself. So, after the first layer transforms its input, the second-through-last layers can be thought of as a network of their own.

So, in an optimized network, the goal of the first layer is to transform the input so that the later layers can classify it, as refined as possible. This means turning the input into something that's easier to work with. Hence, edges. For any image, the edges are always brighter and when the image passes through the filter it detects the frequency change which in turn lets it detect the edge of an image.

Whenever the image goes through these layers, the dimension of the image reduces so padding is required to maintain the dimension of the image. In the second block, there are 64 filters. These filters are used to get more contextual details in the image. In this case, since rain streaks are common in the images,

those are considered as finer details. Therefore, the second block captures the rain streaks in the image.

We experimented with varying combinations of feature blocks and filter sizes by keeping layers & kernel sizes as constant and found this combination giving best results. Generally, when a max pooling layer is used, it down-samples the images after it passes through every layer. So, to prevent the resolution loss from the image, we need to increase the filter size for every immediate feature block. Increasing the filter size helps us increase the block capacity but since we haven't used any max pooling layer, we do not necessarily have to increase the filter size in the next feature block.

After the image passes through both the blocks, rain streaks are accumulated as residue and this residue is subtracted from the original pre-processed image. Thus, giving us the de-rained image in equation (15).

$$X = Y - R \tag{15}$$

Legend:

R - Residue or rain streaks
Y - Input Rainy image
X - De-rained Image

The output images are then post processed for further refinement and finally, different performance metrics were used to find the quality of the de-rained images.

3.4 Challenges Faced

We used an open source software such as Google Colab which provided us a limited 12 gb ram which was not sufficient to train huge datasets, due to which training data was reduced. Due to the limited computational capacity, the

complexity of the model had to be simple. A complex architecture with more layers and feature blocks could have given us better results, but we had to make a tradeoff without compromising much on the results.

Although we faced many challenges, we achieved better results than most of the state-of-the-art-models. Our simple architecture stands out when compared to these models, since they were quite complex and difficult to implement.

4

RESULTS AND ANALYSIS

This section discusses the results that we have achieved from the proposed model ClearCNN.

4.1 Software and Tools

Google Colab

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education. More technically, Colab is a hosted Jupyter notebook service that requires no setup to use, while providing free access to computing resources including GPUs.

Tensorflow

TensorFlow is a free and open-source software library for machine learning. It can be used across a range of tasks but has a particular focus on training and inference of deep neural networks. TensorFlow is a symbolic math library based on dataflow and differentiable programming.

Keras

Keras is an open-source software library that provides a Python interface for artificial neural networks. Keras acts as an interface for the TensorFlow library. Up until version 2.3 Keras supported multiple backends, including TensorFlow, Microsoft Cognitive Toolkit, R, Theano, and PlaidML.

Matlab

Matlab is a programming platform designed specifically for engineers and scientists to analyze and design systems and products that transform our world. The heart of MATLAB is the MATLAB language, a matrix-based language allowing the most natural expression of computational mathematics.

4.2 Results

A total of four datasets are evaluated in our experiments. Table 4.1 shows the results of our model compared to other state-of-the-art models used for rain removal. To quantify and compare, a total of five evaluation metrics are used: Peak Signal-to-Noise Ratio (PSNR), Structural Similarity Index Metric (SSIM), BRISQUE (Blind Referenceless Image Spatial Quality Evaluator), NIQE (Naturalness Image Quality Evaluator) and Entropy.

The Rain12 dataset includes 12 synthesized rain images with a particular type of rain streak, which can be considered as light or moderate rain. Our model was able to perform well on this dataset when it comes to streak removal. In Table 4.1 we observe that the performance of our model surpasses almost all of the state-of-the-art models. It also comes very close to surpassing the performance of the JORDER framework.

The Rain100H dataset consists of 100 image pairs of clean and rainy images. This is a synthetic dataset, specific to heavy streak injection or impurities. Visual results estimate the quality of work to be comparable with all state-of-the-art models. For Example, In terms of PSNR, ClearCNN performs better than UMRL and DerainNet by 4.247dB and 6.397dB respectively. It also surpasses the MSPFN framework by 1.02dB.

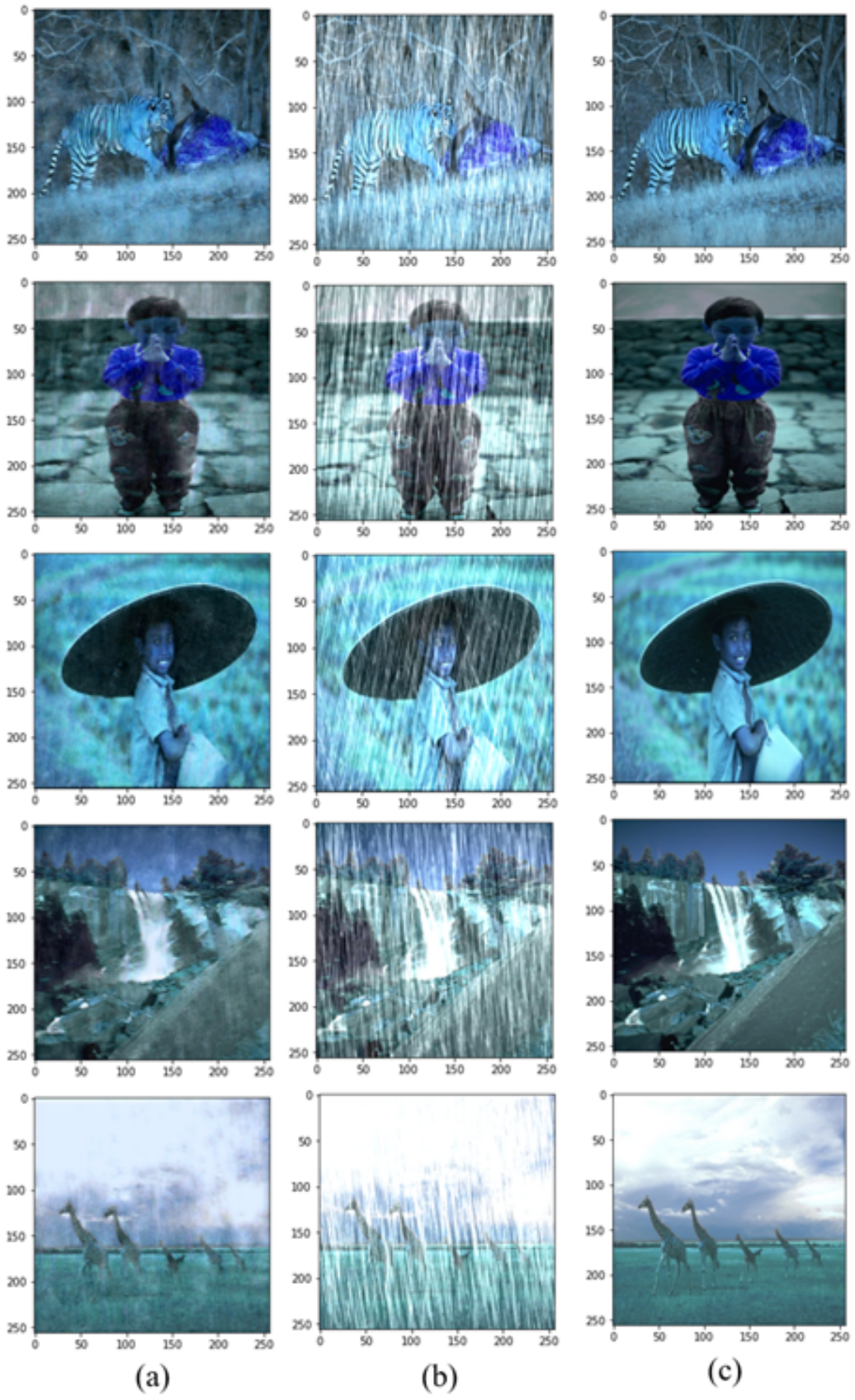

Fig. 4.1 - Rain 100H - (a) ground truth images, (b) rain affected images, (c) de-rained images

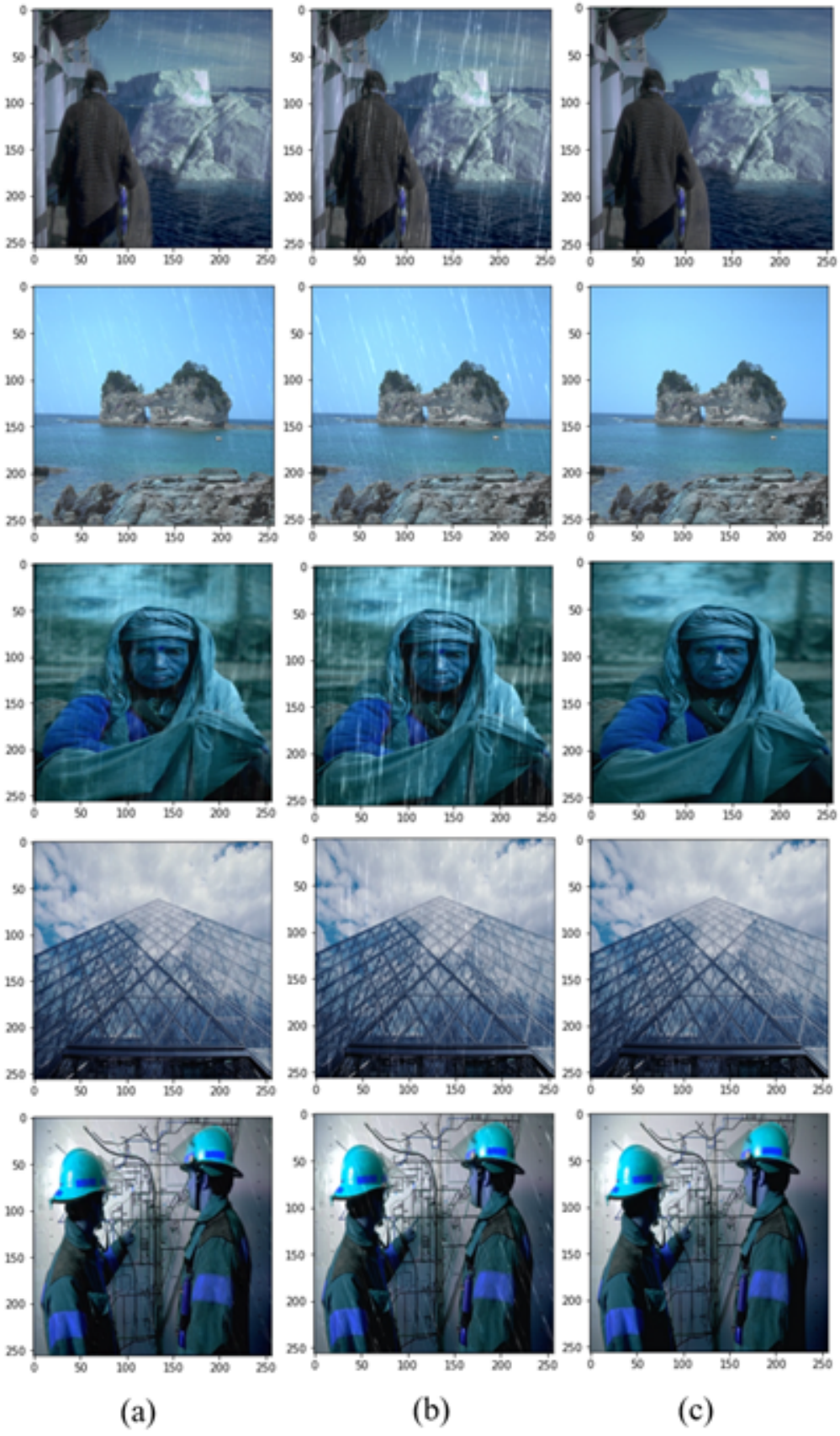

Fig. 4.2 - Rain 100L - (a) de-rained images, (b) rain affected images, (c) clean images

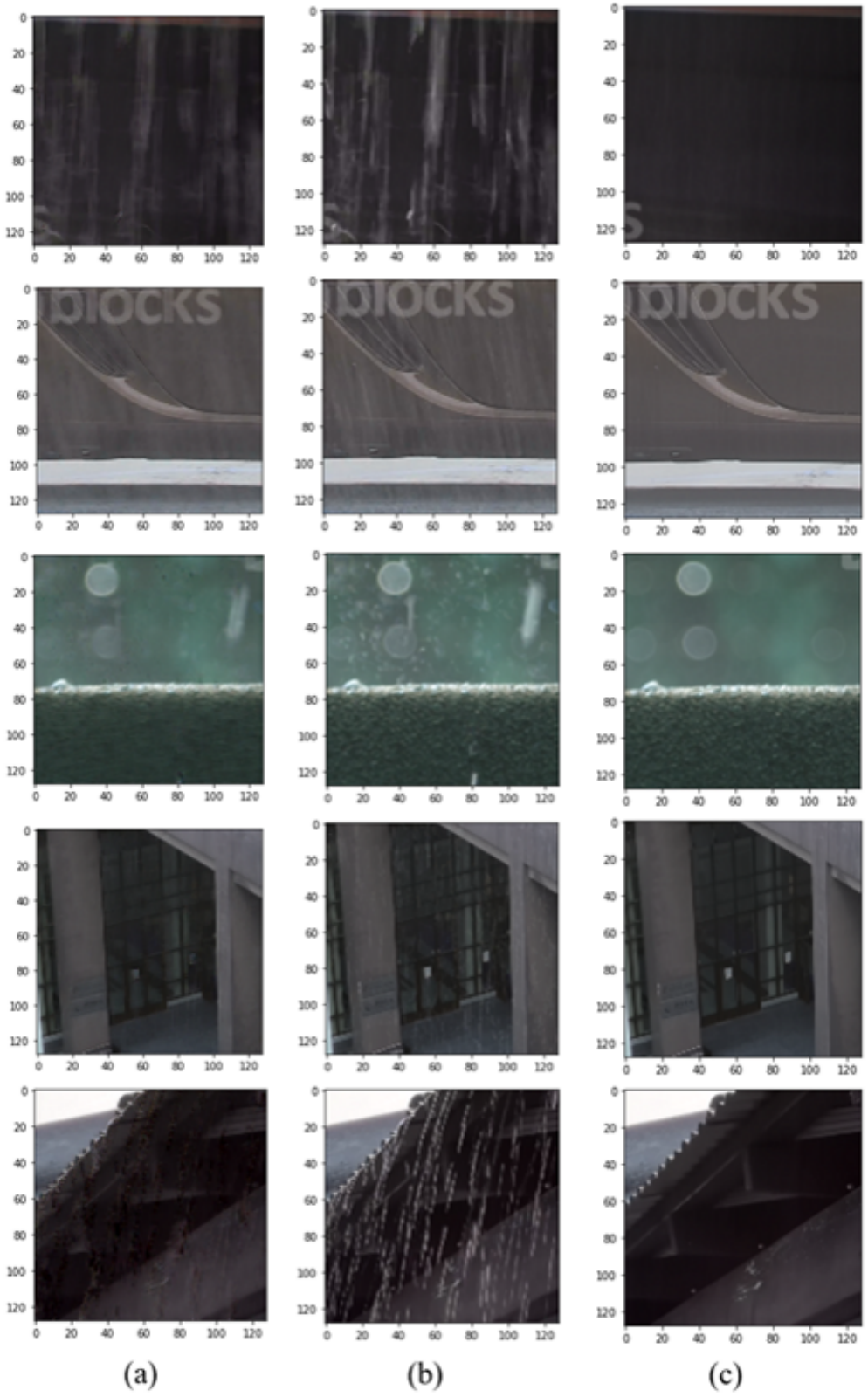

Fig. 4.3 - Real Rain and Rain-free images - (a) ground truth images, (b) rain affected images, (c) de-rained images

The Rain100L dataset is a replication of the Rain100H, but the synthetically injected streaks are far less in intensity when compared. Again, visual confirmation suggests the comparability of ClearCNN with most of the state-of-the-art models. Quantitatively - in terms of PSNR, it surpasses models like DerainNet and SEMI by 9.436dB and 7.796dB respectively. In terms of SSIM it surpasses the JORDER framework by 0.51 as seen in Table 4.1.

Table 4.1 - PSNR and SSIM values for different SOTA models in comparison with ClearCNN [35, 12]

Model / Dataset	**Rain100H**	**Rain100L**	**Rain12**
JORDER	23.45 / 0.749	**36.11 / 0.97**	**36.02 / 0.96**
MSPFN	**28.66 / 0.860**	32.4 / 0.933	-
DerainNet	14.92 / 0.592	27.03 / 0.884	-
RESCAN	26.36 / 0.786	29.80 / 0.881	-
DIDMDN	17.35 / 0.524	25.23 / 0.741	-
UMRL	26.01 / 0.832	29.18 / 0.923	-
SEMI	16.56 / 0.486	25.03 / 0.842	-
PreNet	26.77 / 0.858	32.44 / 0.950	-
ID	14.02 / 0.5239	23.13 / 0.70	27.21 / 0.75
DSC	15.66 / 0.5444	24.16 / 0.87	30.02 / 0.87
LP	14.26 / 0.4225	29.11 / 0.88	32.02 / 0.91
CNN	-	23.70 / 0.81	26.65 / 0.78
SRCNN	-	32.63 / 0.94	34.41 / 0.94
ClearCNN	24.356 / 0.8027	33.427 / 0.9692	32.411 / 0.95

Color Codes

	Results that have been surpassed by our model
	Best results
	Results of our model
	Models that show better results than ours

The Real Rain and Rainfree - 1000 is a set of 1000 image pairs that are images that are naturally rain-affected. Considering how real rain and synthetic streaks are treated differently by multiple models, we intend to train ours with both real rain and synthetic data to make it more robust. Figure 4.3 shows visual representation of de-raining. Quantitatively, the model exhibited a PSNR of 32.0169dB and an SSIM of 0.944.

Table 4.2 - PSNR and SSIM values for the real rain dataset

Model / Dataset	Real rain and Rainfree 1000
ClearNN	32.0169 / 0.9444

No Reference Image Quality metrics are used to evaluate the overall picture quality. The metrics used as a part of our work are BRISQUE and NIQE. These do not compute distortion specific features such as ringing and blurring, they rather use scene statistics of locally normalized luminance coefficients to quantify probable losses of naturalness. BRISQUE can be also used for distortion identification, and requires low computational complexity as compared to Full Reference Image Quality Metrics, i.e., PSNR and SSIM. Such metrics use spatial information and hence require no conversion. Table 4.3 shows the values for No Reference Metrics of all the datasets used.

Table 4.3 - No reference image quality metrics of ClearCNN

Dataset / Parameters	**Rain100L**	**Rain100H**	**Rain and Rain-free images**
BRISQUE	40.11913333	40.13932	54.40472
NIQE	5.333166667	4.3256	6.17436
Entropy	6.214916667	6.035	5.48342

5

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In this Project, we proposed a model which is called ClearCNN which is an end to end implementation of CNN by studying and understanding the design and working of various approaches used for single image de-raining. The basic modules like (Feature Block 1, Feature Block 2, Residual Block) were designed to satisfy the objective. The Proposed model ClearCNN was trained on different datasets both real-world and synthetic datasets and has been evaluated on two different performance metrics i.e. PSNR and SSIM. These Parameters were compared to the state-of-the-art models and were observed to be better than those models with lesser computational cost and execution time.

5.2 Future Scope

Image de-raining is the most important application under complex weather conditions. Many Practical computer vision tasks like object detection, security systems etc. are severely affected under adverse weather conditions. Therefore, Image de-raining is required to remove the artifacts that affect the Processing of these systems.

Due to the fact that rain is closely related to other weather conditions [28] like haze, mist and snow, a multi task model can be implemented by accumulating multiple sample sources collected in bad weather. A number of security-based applications can also be implemented using this concept, which can effectively reduce crime rates in adverse weather conditions.

SOCIAL RELEVANCE

Digital image processing has always had multiple use cases in the real world. Notably, it has been widely deployed such as small target detection and tracking, missile guidance, vehicle navigation, wide area surveillance, and automatic/aided target recognition. Image de-raining can be an additive measure that can enhance surveillance even during adverse weather conditions. The simplified architecture of our model means results are available almost instantaneously, as soon as the distorted image is fed into the model. Since time is considered a very crucial resource in the defense forces, the model can help avert potential damage and loss.

Driver assistance technologies in today's motor vehicles are already helping in saving lives and avoiding accidents. Some areas in which the machine helps the driver avert such mishaps include - warnings when drivers of other vehicles are backing up, automatic brake application when the vehicle in front slows down or stops. These technologies use a combination of hardware and software in order to produce the best conversion rate and results. Likewise, Image de-raining through our model can be implemented here as an addition to already used software to assist the driver during harsh weather conditions during rain, usually during which visibility is a major factor.

REFERENCES

[1] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image de-raining," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 1794–1807, Jun. 2020, doi: 10.1109/tnnls.2019.2926481.

[2] D. Chen *et al.*, "Gated Context Aggregation Network for Image Dehazing and de-raining," *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2019, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/wacv.2019.00151.

[3] S. Li *et al.*, "Single Image de-raining: A Comprehensive Benchmark Analysis," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2019.00396.

[4] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. H. Lau, "Spatial Attentive Single-Image de-raining With a High Quality Real Rain Dataset," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2019.01255.

[5] R. Yasarla and V. M. Patel, "Uncertainty Guided Multi-Scale Residual Learning-Using a Cycle Spinning CNN for Single Image De-Raining," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2019.00860.

[6] Y. Wei, Z. Zhang, H. Zhang, R. Hong, and M. Wang, "A Coarse-to-Fine Multi-stream Hybrid de-raining Network for Single Image de-raining," *2019 IEEE International Conference on Data Mining (ICDM)*, Nov. 2019, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/icdm.2019.00073.

[7] S. Deng *et al.*, "Detail-recovery Image de-raining via Context Aggregation Networks," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr42600.2020.01457.

[8] Y. Du, J. Xu, X. Zhen, M.-M. Cheng, and L. Shao, "Conditional Variational Image de-raining," *IEEE Transactions on Image Processing*, vol. 29, pp. 6288–6301, 2020, doi: 10.1109/tip.2020.2990606.

[9] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley,

"Removing Rain from Single Images via a Deep Detail Network," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2017.186.

[10] 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, Jun. 2017, doi: 10.1109/tip.2017.2691802.

[11] H. Fu, Y. Zhang, and H. Ma, "See clearly on rainy days: Hybrid multiscale loss guided multi-feature fusion network for single image rain removal," *Computational Visual Media*, Mar. 2021, doi: 10.1007/s41095-021-0210-3.

[12] K. Jiang *et al.*, "Multi-Scale Progressive Fusion Network for Single Image de-raining," *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr42600.2020.00837.

[13] Y. Wang, C. Chen, S. Zhu, and B. Zeng, "A framework of single-image de-raining method based on analysis of rain characteristics," *2016 IEEE International Conference on Image Processing (ICIP)*, Sep. 2016, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/icip.2016.7533128.

[14] H. Zhang and V. M. Patel, "Density-Aware Single Image De-raining Using a Multi-stream Dense Network," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2018.00079.

[15] Z. Fan, H. Wu, X. Fu, Y. Huang, and X. Ding, "Residual-Guide Network for Single Image de-raining," *Proceedings of the 26th ACM international conference on Multimedia*, Oct. 2018, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1145/3240508.3240694.

[16] T. Chen and C. Fu, "Single-image-based Rain Detection and Removal via CNN," *Journal of Physics: Conference Series*, vol. 1004, p. 012007, Apr. 2018, doi: 10.1088/1742-6596/1004/1/012007.

[17] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive Image de-raining Networks: A Better and Simpler Baseline," *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2019, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2019.00406.

[18] D. Ren, W. Shang, P. Zhu, Q. Hu, D. Meng, and W. Zuo, "Single Image de-raining Using Bilateral Recurrent Network," *IEEE Transactions on Image Processing*, vol. 29, pp. 6852–6863, 2020, doi:

10.1109/tip.2020.2994443.

[19] Y. Zheng, X. Yu, M. Liu, and S. Zhang, “Single-Image de-raining via Recurrent Residual Multiscale Networks,” *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2020, doi: 10.1109/tnnls.2020.3041752.

[20] 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*, vol. 30, no. 11, pp. 3943–3956, Nov. 2020, doi: 10.1109/tcsvt.2019.2920407.

[21] R. Yasarla, J. M. J. Valanarasu, and V. M. Patel, “Exploring Overcomplete Representations for Single Image de-raining Using CNNs,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 2, pp. 229–239, Feb. 2021, doi: 10.1109/jstsp.2020.3039393.

[22] C. Wang, Y. Wu, Y. Cai, G. Yao, Z. Su, and H. Wang, “Single image de-raining via deep pyramid network with spatial contextual information aggregation,” *Applied Intelligence*, vol. 50, no. 5, pp. 1437–1447, Jan. 2020, doi: 10.1007/s10489-019-01567-5.

[23] R. Li, L.-F. Cheong, and R. T. Tan, “Single Image de-raining using Scale-Aware Multi-Stage Recurrent Network,” *arXiv.org*, Dec. 19, 2017. https://arxiv.org/abs/1712.06830.

[24] X. Chen, Y. Huang, and L. Xu, “Multi-Scale Hourglass Hierarchical Fusion Network for Single Image de-raining,” *arXiv.org*, Apr. 25, 2021. https://arxiv.org/abs/2104.12100v1.

[25] W. Yang, R. T. Tan, S. Wang, Y. Fang, and J. Liu, “Single Image de-raining: From Model-Based to Data-Driven and Beyond,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020, doi: 10.1109/tpami.2020.2995190.

[26] Y. Wei *et al.*, “Semi-DerainGAN: A New Semi-supervised Single Image de-raining Network,” *arXiv.org*, Jan. 23, 2020. https://arxiv.org/abs/2001.08388v3.

[27] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A Model-Driven Deep Neural Network for Single Image Rain Removal,” *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr42600.2020.00317.

[28] Y. Wang, Y. Song, C. Ma, and B. Zeng, “Rethinking Image de-raining via Rain Streaks and Vapors,” *Computer Vision – ECCV 2020*, pp. 367–382, 2020, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-58520-4_22.

[29] C. Wang, Y. Wu, Z. Su, and J. Chen, “Joint Self-Attention and Scale-Aggregation for Self-Calibrated de-raining Network,”

Proceedings of the 28th ACM International Conference on Multimedia, Oct. 2020, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1145/3394171.3413559.

[30] Q. Guo *et al.*, "EfficientDeRain: Learning Pixel-wise Dilation Filtering for High-Efficiency Single-Image de-raining," *arXiv.org*, Sep. 19, 2020. https://arxiv.org/abs/2009.09238v1.

[31] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu, "Automatic Single-Image-Based Rain Streaks Removal via Image Decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, Apr. 2012, doi: 10.1109/tip.2011.2179057.

[32] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain Streak Removal Using Layer Priors," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/cvpr.2016.299.

[33] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent Squeeze-and-Excitation Context Aggregation Net for Single Image de-raining," *Computer Vision – ECCV 2018*, pp. 262–277, 2018, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-01234-2_16.

[34] X. Zhang, H. Li, Y. Qi, W. Leow, and T. Ng, "Rain Removal in Video by Combining Temporal and Chromatic Properties," *2006 IEEE International Conference on Multimedia and Expo*, Jul. 2006, Accessed: May 19, 2021. [Online]. Available: http://dx.doi.org/10.1109/icme.2006.262572.

[35] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu, "Joint Rain Detection and Removal from a Single Image with Contextualized Deep Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 42, no. 6, pp. 1377–1393, Jun. 2020, doi: 10.1109/tpami.2019.2895793.

[36] A. Ichigaya, Y. Nishida, and E. Nakasu, "Non reference Method for Estimating PSNR of MPEG-2 Coded Video by Using DCT Coefficients and Picture Energy," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 6, pp. 817–826, Jun. 2008, doi: 10.1109/tcsvt.2008.920658.

[37] L. O. Chua, *CNN: A Paradigm for Complexity*. World Scientific, 1998.