Single Image De-raining Using Convolutional Neural Network

by

PINAK GAJERA
202111063

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

Dhirubhai Ambani Institute of Information and Communication Technology

June 2023

Declaration

I hereby declare that

i) The thesis comprises my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) Due acknowledgment has been made in the text to all the reference material used.

Pinak Gajera

Certificate

This is to certify that the thesis work entitled **Single Image De-raining Using Convolutional Neural Network** has been carried out by **Pinak Gajera** for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

Dr. Rajib Lochan Das
Thesis Supervisor

Dr. Srimanta Mandal
Thesis Co-Supervisor

Acknowledgments

First of all, I would like to thank Almighty God for giving me the opportunity and guidance to achieve my goal and to be successful in the journey of pursuing M.Tech. There are many instances where without divine help, I wouldn't be able to achieve the desired outcome.

I would like to appreciate the opportunity to complete my studies successfully and granting me M.Tech degree with Software System Specialization by Dhirubhai Ambani Institute of Information and Communication Technology (DAIICT).

Words cannot express my gratitude to my research supervisor and my mentor Dr. Srimanta Mandal, for his invaluable guidance and dedicated involvement in every part of my work. The way he conveyed the detailed information required for this research work left a strong impression on me. Also, his humbleness and focused vision steered me in the right direction. I strongly feel blessed to have him as a mentor and his help during the learning process.

At last, I want to thank my parents for their constant support and guidance which enables me to be part of this endeavour.

Contents

Abstract

Rain streaks vary in size, quantity, and direction, making removing them from individual images difficult. Recent advancements in deep learning, especially those using CNN-based techniques, have shown promising results in addressing this issue. However, the requirement for additional consideration of the rain streaks location information in the image is a significant drawback of these methods. Methods based on deep learning have proven to be quite effective in handling synthetic and real-world rainy images. These methods use convolutional neural networks (CNNs) to their full potential to learn the correspondence between rainy and rain-free images. We typically use an encoder-decoder architecture where the encoder pulls features from the rainy image and then creates the rain-free image using the learned features. These algorithms can efficiently learn the complicated correlations between rain streaks and ground truths by training on large-scale datasets that combine images with and without rain. End-to-end methods aim to train a single model that converts the rainy image into its rain-free counterpart without explicitly decomposing it into the rain and the background components. Additionally, researching end-to-end approaches offers a fascinating way of improving the de-raining algorithm's efficiency. More effective and efficient techniques for removing rain streaks from single images will probably be developed when this research study continues to be investigated.

Index Terms: *Rain streaks, image de-raining, contextual information, residual map, synthetic and real-world rainy image*

List of Principal Symbols and Acronyms

RN	Residual Network
CN	Confidence Network
CA	Channel Attention
PA	Pixel Attention
MAE	Mean Absolute Error
MSE	Mean Square Error
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index

List of Tables

List of Figures

CHAPTER 1
Introduction

Rain streaks pose a significant challenge to the visual quality of photos and videos captured in rainy conditions. They introduce various visible degradations, such as blurring, contrast loss, and distortion, which greatly affect the sharpness and clarity of the captured images. Removing these rain streaks and restoring the original image is a complex task due to the diverse characteristics of rain streaks, including variations in size, direction, and density[23]. The complexity further increases when the rain streaks align with the structure and orientation of objects in the image. In such cases, accurately separating the rain streaks while preserving the underlying structure becomes challenging. One of the primary objectives in the de-raining process is to retain important image details while avoiding introducing artifacts. Developing a universal method that effectively addresses these challenges and achieves high-quality rain removal remains a significant research goal in this field.

De-raining techniques that enhance image quality have a significant impact on various computer vision applications, including object identification and recognition in intelligent vehicles. When it is raining, rain streaks can obstruct the view, making it challenging to detect objects on the road. By effectively removing rain streaks from captured images, the performance of computer vision algorithms can be improved, leading to safer driving conditions. Similarly, outdoor monitoring systems, such as surveillance cameras, can greatly benefit from image de-raining. Removing rain streaks from surveillance footage captured in wet conditions improves visibility and enables more accurate analysis and detection of objects or events of interest. This is particularly crucial to ensure effective security and surveillance in outdoor environments. To enhance visual perception for both human observers and computer vision systems, researchers are actively developing single image de-raining techniques. These techniques aim to generate precise, high-quality rain-free images from their original rainy counterparts[9]. By mitigating the adverse effects of rain streaks, these techniques improve image clarity and enable more robust and reliable computer vision applications.

The additive model, which is the basis for existing image de-raining techniques, assumes that the rainy image (x) is the result of superimposing a clean image (y) and a rain component(i.e. residual map) (r). i.e,

$$y = x + r \tag{1.1}$$

The primary goal of image de-raining is to generate a rain-free image "x" from an observed rainy image "y." Traditionally, this is achieved by estimating the residual map "r," which represents the rain streak component present in the observed image. The residual map captures the specific characteristics and patterns of rain streaks in the image. By subtracting the predicted residual map from the observed image, the de-rained image can be obtained, effectively removing the rain streaks and restoring image clarity[23]. In recent years, deep learning-based techniques have emerged as powerful approaches for single image de-raining. These techniques leverage the capabilities of deep neural networks to directly predict the de-rained image from the noisy observation. The methodology employed in this study differs from many existing deep learning-based approaches, highlighting novel strategies and advancements in the field.

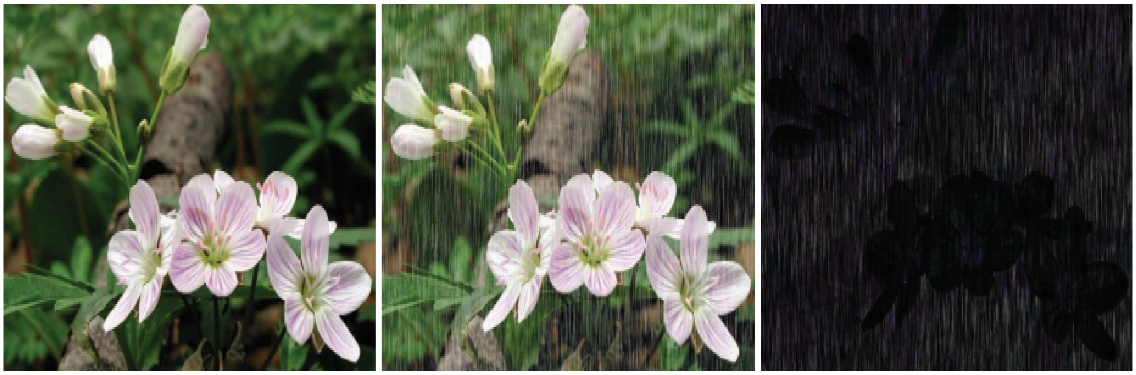

Figure 1.1: An example of clean image Y, rainy image X, rain streaks Y - X. (from left to right)

The proposed approach follows a two-step process instead of directly estimating the de-rained image. It initially focuses on computing the rain streak component, denoted as the residual map "r," from the observed image "y." The method aims to accurately capture the characteristics and variations of rain streaks by independently estimating this rain streak component. Subsequently, the estimated residual map is utilized to derive the de-rained image by removing the rain streak component from the observed image. This approach acknowledges the significance of explicitly modeling and separating rain streaks from the underlying structure of the image. By doing so, it strives to enhance the precision and quality of the de-rained results[23].

$$x = y - r \tag{1.2}$$

To summarize, this research introduces a novel approach that departs from conventional deep learning-based methods by initially estimating the rain streak component (residual map) and subsequently using it to estimate the de-rained image. Unlike previous approaches that directly estimate the de-rained image from the noisy observation, this method focuses on explicitly representing the rain streaks and their impact on the observed image. By adopting this strategy, the research aims to enhance the accuracy and quality of the de-raining process.

1.1 Objective

The objectives of the thesis can be summarized as follows:

1. Developing an end-to-end architecture that can handle single image deraining using a convolutional neural network(CNN).

2. Developing an architecture that can handle the problems provided by various rain content scales, and then applying that architecture to estimate the final de-rained image.

3. Creating an architecture that can manage the varying density levels visible in rainy images, even when a dataset is provided.

4. Introducing a module to the architecture to deal with the problem of color distortion and ensure that the de-rained images with accurate colors and structure are preserved.

1.2 Contribution

The contributions of the thesis can be summarized as follows:

- We introduce an end-to-end architecture that can remove the rain streaks contained with different rain streak densities like light, medium, and heavy.
- We compare the existing state-of-the-art methods of single image de-raining and try to understand the various methods that can remove the rain streaks and analyze the results of it.
- We trained the model with the dataset that contains three different level densities, light, medium, and heavy, containing around 12,000 images.
- Additionally, we try to improve the results of this architecture with a cycle spinning technique that can increase the objective metrics such as PSNR and SSIM.

  Overall, the thesis advances the state-of-the-art and offers insightful information for further study in the field by contributing an end-to-end design, appropriate management of various density levels, and usage in practice.

1.3 Organization of Thesis

Chapter 2 provides an overview of existing techniques for image dehazing, including classical methods and deep learning-based methods.

Chapter 3 presents the first proposed method - RCLU; the encoder-decoder architecture of UNet and the Uncertainty Guided Multi-Scale Residual Learning (UMRL) method. Additionally, we incorporate Residual and Confidence Networks (RN & CN) as part of the process & its loss function.

Chapter 4, the limitations of the proposed method 1 - RCLU are addressed, and a new approach SID-U-CNN for removing rain streaks, called RainRemovalBlock, is introduced. The RainRemovalBlock is implemented in conjunction with the encoder-decoder architecture of Unet, offering a novel solution to overcome the limitations previously identified. The loss function is also mentioned.

Chapter 5 presents the results of the experiments along with the datasets conducted in the study. This chapter compares the obtained results with existing state-of-the-art methods.

Chapter 6 concludes the thesis, summarizing and contributions and it outlines potential areas for future research and development.

CHAPTER 2

Literature Survey

Recent advancements in the field of single image de-raining have shown significant progress. Siyuan et al. proposed a comprehensive analysis that considers different types of rain, including rain streaks, raindrops, and rain with mist. They conducted extensive comparisons, examining various techniques and datasets, to evaluate the effectiveness of different approaches[3]. Their analysis provides valuable insights into the advancements and limitations of current methods in single image de-raining. Notably, a particular study [8] achieved state-of-the-art performance by effectively distinguishing rain streaks using Gaussian mixture models. However, there is still room for improvement in preserving small details and textures, as some smoothing effects were observed in the de-rained images, particularly in the background regions.

Li et al. [20] proposed an alternative strategy that specifically targeted the challenge of heavy rain and rain streak formation. Their research aimed to address the difficulties associated with images containing a high density of rain streaks. By developing specialized algorithms, they were able to achieve improved results in rain removal, even in severe rain conditions. Another noteworthy contribution in this field was made by Fu et al. [5], who introduced an end-to-end deep learning architecture for rain removal. Their approach utilized a deep detail network, which effectively reduced the mapping range from the input to the output. By leveraging deep learning techniques, their method demonstrated promising results in eliminating rain artifacts and enhancing the visual quality of de-rained images. These advancements in end-to-end deep learning algorithms highlight the progress made in the field of single image de-raining, addressing various challenges such as different rain categories and heavy rain scenarios. Ongoing research in this area holds great potential for further improving the effectiveness and practicality of single image de-raining techniques, leading to clearer and visually appealing rain-free images.

There are a number of different methods that have been proposed for image deraining. These can be generally described below one by one:

2.1 UMRL : Uncertainty Guided Multi-Scale Residual Learning-using a Cycle Spinning CNN

The proposed Uncertainty guided Multi-scale Residual Learning (UMRL) network is designed to estimate the clean image from a corresponding rainy image. It utilizes a multi-scale approach to capture rain content at different levels and incorporate it into the de-raining process [21]. Additionally, a novel method is introduced to guide the network's weight learning based on the confidence measure of the estimates. To further enhance de-raining performance, a unique training and testing approach inspired by cycle spinning is employed [21]. The UMRL network consists of two main components, namely the Rain Network (RN) and Confidence Network (CN). The outputs of these networks are fed into subsequent layers to guide the estimation of the clean image. The rain streak component, also known as the residual map, is first estimated and subtracted from the rainy image to obtain the de-rained image. In this process, a confidence score, represented as c, is computed to indicate the level of uncertainty associated with estimating the residual map. The confidence score measures the network's confidence in the calculated residual value for each pixel [21]. The architecture of the UMRL network is illustrated in Figure 2.1, showcasing the flow of information and integrating the RN, CN, and subsequent layers to achieve the de-raining task.

The cycle spinning technique is applied to generate shifted images by cyclically shifting an input image of size m × n in p-row and q-column steps. These shifted images are then processed by the UMRL network to obtain de-rained results during testing. To obtain the final de-rained image, the inverse shift operation is performed on the shifted images, followed by de-raining and averaging of the results [21]. The use of cycle spinning is not limited to the UMRL network and can be beneficial for improving the performance of any CNN-based de-raining technique.

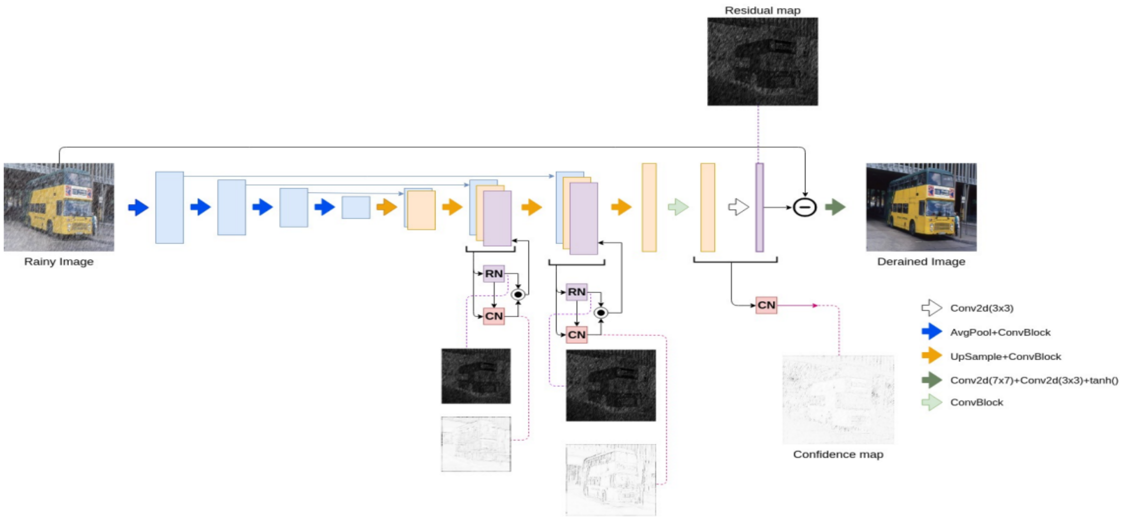

Figure 2.1: An overview of the UMRL network[21].

2.2 DIDMDN : Density-aware Multi-stream Dense Network

A novel approach called DID-MDN is proposed for simultaneous rain density estimation and de-raining. This approach utilizes a multi-stream densely connected convolutional neural network to effectively remove rain streaks based on the estimated rain-density label[23]. By incorporating information about the rain density, the network is able to accurately handle rain streaks of different scales and shapes. Using a multi-stream densely connected de-raining network allows for better characterization of rain streaks by leveraging features from multiple scales. The suggested approach offers a promising solution for addressing rain density estimation and de-raining challenges.

The proposed DID-MDN architecture consists of two key components: the residual-aware rain-density classifier and the multi-stream densely connected de-raining network. The role of the residual-aware rain-density classifier is to estimate the level of rain present in a given wet image. On the other hand, the multi-stream densely connected de-raining network is designed to effectively remove rain streaks from rainy images, considering the guidance provided by the estimated rain-density information[23, 4]. The overall network architecture of the proposed DID-MDN technique is illustrated in Figure 2.2. This architecture combines rain-density estimation and de-raining processes to achieve improved rain removal results.

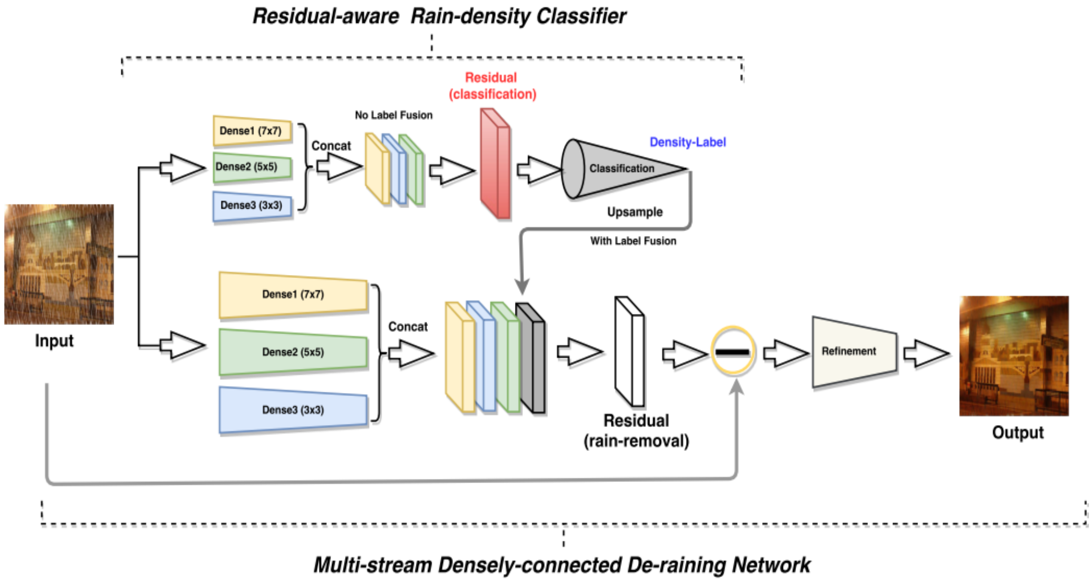

Figure 2.2: An overview of the DID-MDN method[23].

2.3 DRT: De-raining Recursive Transformer

In recent times, transformer-based deep learning models have proven to be highly effective for various vision tasks. However, these models often have a large number of parameters and can be challenging to train. This poses a problem for low-level computer vision tasks that involve dense-prediction, such as rain streak removal, as they typically require devices with limited memory and processing capabilities. To address this issue, the authors propose a novel approach called de-raining a recursive transformer (DRT) that utilizes a recursive local window-based self-attention structure with residual connections. This approach combines the advantages of transformers while keeping the computational resource requirements low, making it suitable for constrained devices. The DRT model offers a promising solution for efficiently removing rain streaks in images while considering resource limitations[9].

The DRT (de-raining a recursive transformer) model consists of three stages: patch embedding (stage f1), deep feature extraction (stage f2), and image reconstruction (stage f3). The process is visually depicted in Figure 2.3. Initially, the rainy image is passed through a convolutional layer and then divided into patches, which are stacked depth-wise in the patch embedding step (f1). The next stage involves the utilization of multiple recursive transformer blocks (RTBs) for performing deep feature extraction. The parameter "N" represents the total number of RTBs em-

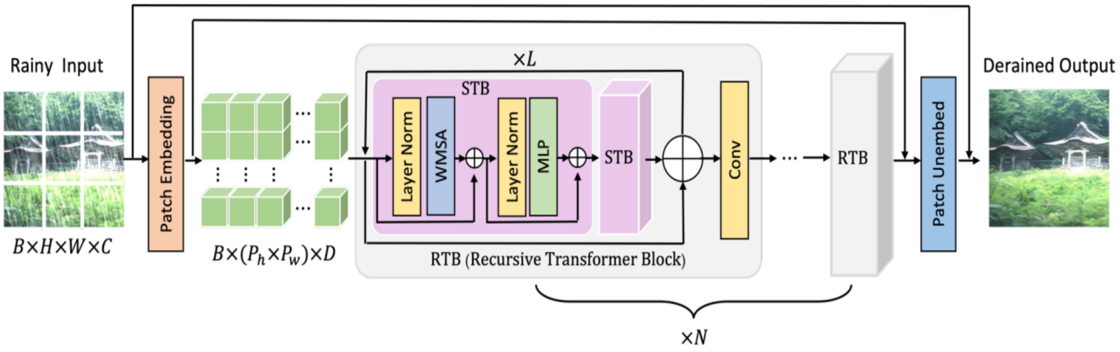

Figure 2.3: The De-raining Recursive Transformer architecture. RTB stands for recursive transformer block, and STB stands for Swin Transformer block. N refers to the number of RTBs, and L refers to the number of recursive calls[9].

ployed in the model. To ensure that the RTBs don't solely focus on rain streak detection, a residual connection is incorporated at the end of the process, where the input is added to the output of the deep feature extraction stage. This helps in preserving important details and preventing the network from overemphasizing rain streaks. Subsequently, the image restoration stage processes the deep features, reversing the operations performed in the previous stage. To remove the rain streak features extracted by the network, another residual connection is introduced between the input and output of the network. The proposed DRT model draws inspiration from various studies [2, 11, 19] and combines the advantages of recursive transformers with the inclusion of residual connections. This architecture enables effective rain streak removal while considering the constraints of computational resources and memory.

2.4 Sync2Real Transfer Learning using Gaussian Processes

The researchers propose a semi-supervised learning approach utilizing the Gaussian Process, which enables the network to learn the task of de-raining by leveraging synthetic datasets while effectively generalizing to unlabeled real-world images. Existing literature has provided limited insights into training image-processing networks using real-world data. The scarcity of fully-labeled real-world image de-raining datasets poses a challenge, leading current algorithms to rely heavily on synthetic data, resulting in subpar performance when applied to real-world images[22].

The proposed approach involves semi-supervised learning (SSL) using a Gaussian process (GP) and iterative training with both labeled and unlabeled data. In the labeled learning phase, the model is trained using labeled data by minimizing the mean squared error between the predictions and the actual data. Additionally, the inputs from the labeled dataset are projected onto a latent space, which is modeled using GP. In the unlabeled training phase, pseudo ground truth (pseudo-GT) is created for the unlabeled inputs based on the GP model from the labeled training phase. The intermediate latent space for the unlabeled data is supervised by this pseudo-GT. The pseudo-GT concept assumes that unlabeled images can be described as a weighted mixture of characteristics from labeled data when projected onto the latent space, with the weights determined by a kernel function. By minimizing the variance between the network weights and the unlabeled data domain, the network weights automatically adjust to the characteristics of the unlabeled data[22].

CHAPTER 3

Residual & Confidence Learning with Unet in Single Image De-raining (RCLU)

The skip-connected encoder-decoder network-based U-Net architecture[12] has proven very efficient in various image processing tasks. This architecture focuses heavily on the decoder network to provide a segmentation map appropriate for the size of the input image. On the other hand, the encoder network is in charge of preserving the high-level characteristics and features of the input image. The availability of skip connections, which allow for exact localization and improve information flow between the encoder and decoder networks, is a significant component of the U-Net architecture[12]. The decoder network can access the high-level data collected by the encoder network due to these skip connections, which create direct connections between essential encoder and decoder layers.

The U-Net architecture's skip connections allow the decoder network to take full advantage of the in-depth knowledge acquired by the encoder network. Using this approach, the network can better collect fine-grained structures and critical visual features throughout the decoding process. Overall, the U-Net architecture's skip-connected encoder-decoder network[12] promotes effective information propagation and allows the network to capture both low-level and high-level aspects of the input image. This architecture's capacity to efficiently localize and retain crucial image features while producing precise segmentation outputs has led to its wide use in various computer vision tasks, including image de-raining.

A pooling layer is put in after each convolutional layer in the encoder network of the U-Net architecture. These pooling layers' role is to decrease the feature maps that the preceding convolutional layers have produced. The spatial dimensions of the original input image are gradually reduced due to the down-sampling process. The feature maps are divided into non-overlapping regions by the pooling layer, which then chooses the best-expected value for each region. Average or maximum pooling are two standard processes used in this selection process. The pooling layers significantly lower the quantity of information and spatial resolution in the feature representations by downsampling the feature maps.

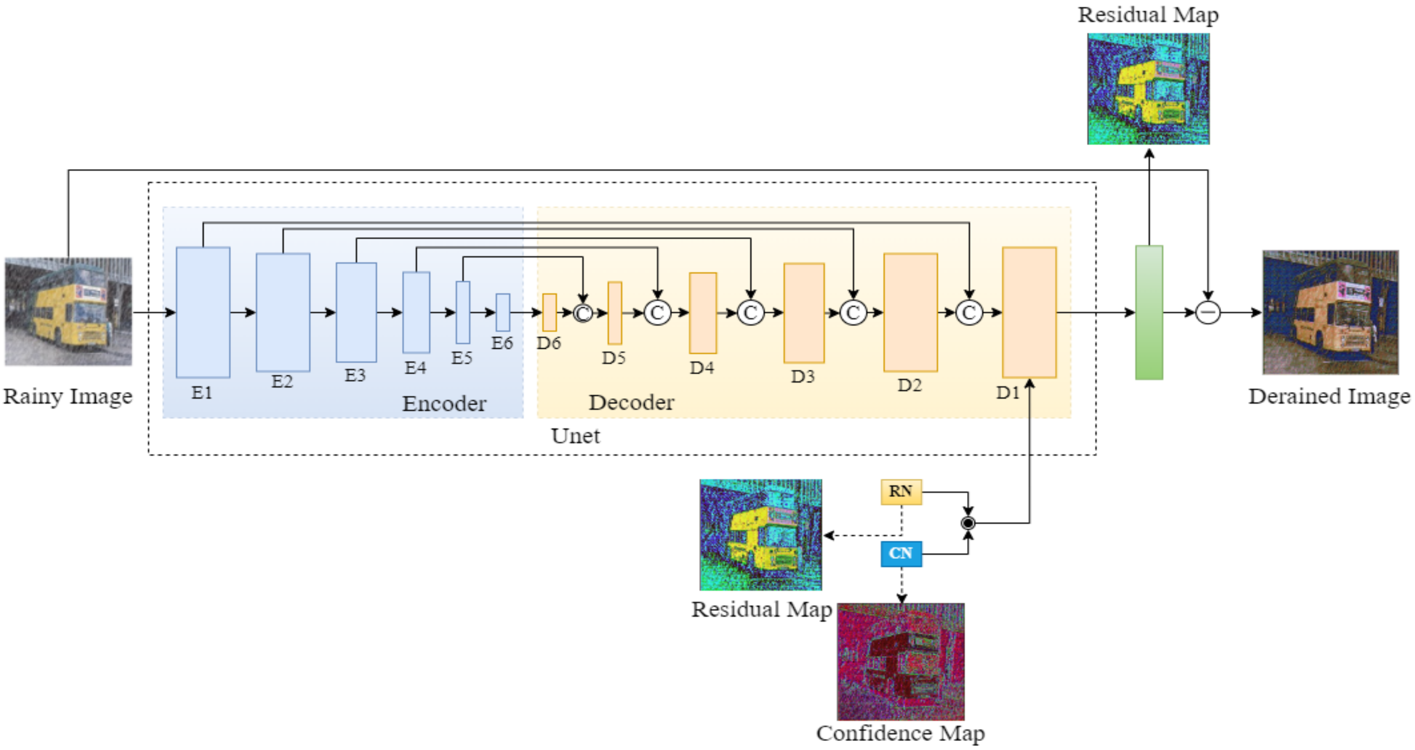

Figure 3.1: The architecture of the RCLU.

The U-Net architecture[12] gradually compresses the feature maps over many iterations of the down-sampling process, producing a highly condensed and abstracted version of the original image. This network-efficient calculation and parameter sharing are made possible by the compressed feature representation, which also captures the input image's high-level details and overall context. The pooling layers' down-sampling ensures that the U-Net architecture can still capture and simulate intricate patterns and structures in the input data even as the spatial dimensions are decreased. In the decoder network for accurate localization and segmentation tasks, for example, this enables the network to extract increasingly abstract and invariant information important for later processing stages. Overall, the U-Net encoder network's placement of pooling layers after each convolutional layer allows for the steady decrease of spatial dimensions and the extraction of high-level features, creating a representation of the input image that is both effective and efficient.

The deconvolutional layers that collectively make up the U-Net decoder network is essential for up-scaling the feature maps to the size of the input image. Since it facilitates recovering the spatial information lost during the down-sampling process carried out by the pooling layers in the encoder network, this up-sampling approach is crucial for correct localization. The U-Net decoder ensures that detailed spatial information is retrieved by utilizing deconvolutional layers, enabling accurate localization of objects and fine-grained features.

Skip connections are used to establish a connection between the encoder and decoder networks. Direct communication between the encoder and the appropriate layers in the decoder is made possible by such connections. The U-Net architecture may take advantage of high-level and low-level characteristics from various levels of abstraction by adding skip connections. This improves the network's ability to carry out precise localization and segmentation tasks by successfully capturing global context and local information.

In conclusion, the U-Net architecture[12] is a powerful tool for image processing, especially for tasks requiring semantic segmentation. The network can achieve exact localization while simultaneously gathering contextual data and fine-grained characteristics due to combining an encoder-decoder network and skip connections. This architecture is a solid option for applications requiring precise segmentation and in-depth image analysis because of its capacity to leverage skip connections and restore spatial information through deconvolutional layers.

In our alternate methodology, the rain streak component, the residual map (r), is estimated and used in two steps. After performing this estimation as the first step, the de-rained image (x) is calculated by subtracting the estimated residual map from the observed image (y)[23]. We also provide the idea of a confidence score (c)[21], which is a map that represents the degree of uncertainty in the estimation of the residual map. The confidence score (c), which expresses how confident the network is in the accuracy of the residual values at each pixel, is valuable information. It provides information about how uncertain the anticipated rain streak component is. In our method, we do more than estimate the residual map and confidence score independently. Instead, we thoroughly combine the two pieces of data and use them as inputs for future network layers.

We provide a communication channel that efficiently spreads location-specific rain information throughout the network by logically integrating the residual map and confidence score. Using this strategy, the network is guaranteed to take advantage of the spatial context and make wise judgments about the presence and features of rain streaks. As a result, we can compute both the revised residual map and the uncertainty map. Overall, our approach combines the calculation of the residual map, evaluation of confidence scores, and application of these two pieces of knowledge within the network's design. This method makes good use of location-based rain information. It takes the network's confidence in the estimating process into account to produce results that are more reliable and accurate when it comes to de-raining.

3.1 Residual and Confidence Map Networks

A key component of our method is the Residual Network (RN)[21, 10], which uses feature maps to estimate the residual map, which is the difference between the observed image and the de-rained image. Convblock(64,32), Convblock(32,32), and Convblock(32,3), a series of convolutional layers that were specifically created to capture the underlying patterns and features of the image, make up the RN.

We use the Confidence map Network (CN)[21], which accepts the estimated residual map and the associated feature maps as input, to evaluate the confidence of the residual values at each pixel. Convblock(67,16), Convblock(16,16), and Convblock(16,3) convolutional layers make up the CN and are used to encode the confidence measure and retrieve relevant information. The confidence map and the residual map are combined using element-wise multiplication to create an improved representation, which is then up-sampled and fed into the network's subsequent layers.

The uncertainty-guided multi-scale residual learning (UMRL)[21] last layer's feature maps and output residual map r are fed into the CN for the confidence map estimation. This enables the network to consider the confidence information and the learned high-level characteristics, ensuring a thorough comprehension of the image structure and rain streaks. The de-rained image is then created by subtracting the observed image's y component from the estimated residual map's r component.

By successfully merging the residual and confidence information, we want to improve the precision of rain streak estimation and provide visually pleasing de-rained images by utilizing the RN and CN[21] inside our framework.

3.2 Loss Function

The UMRL loss has been used in this method. This loss consists of L1 or MAE loss along with perceptual loss.

3.2.1 MAE or L1 Loss

L1 loss, often known as the mean absolute error (MAE), is a standard loss function in image processing. It calculates the average absolute difference between each pixel of the expected and actual images. By penalizing significant changes in pixel values, L1 loss preserves image structure and details by concentrating on the size of errors. To produce images that closely resemble the ground truth regarding pixel intensity, L1 loss must be minimal.

$$L_{l1} = \|(c \odot \hat{x}) - (c \odot x)\|_1 \tag{3.1}$$

where c is the confidence map along with its x's pixel of the image.

3.2.2 Perceptual Loss

The perceptual loss is feature-based loss, and in our case, extracted features from layer relu1_2 of pre-trained network VGG-16. Let F(.) denote the features obtained using the VGG16 model[18]; then the perceptual loss is defined as follows

$$L_{perc} = \frac{1}{NHW} \sum_i ||F(\hat{x_1})^i - F(x_1)^i||_2^2 \tag{3.2}$$

where N is the number of channels of F(.), H is the height and W is the width of feature maps[18].

The total loss function is defined as

$$L = \lambda_1 L_{l1} + \lambda_2 L_{perc} \tag{3.3}$$

In our experimental setup, we have set values λ_1, λ_2 are 1, 0.5 respectively.

3.3 Limitation of RCLU

We conducted additional evaluations on the encoder-decoder networks based on the Unet [12] architecture in order to enhance the performance of rain removal and incorporate the UMRL (Uncertainty Guided Multi-Scale Residual Learning) loss [21]. We also utilized the UMRL loss to investigate the refinement networks derived from Unet and Trident.

To address the limitations observed in previous experiments, particularly the unsatisfactory results, we modified the loss function. Initially, we introduced the L1 loss of the confidence map and the perceptual loss as the primary loss function. However, due to the underwhelming outcomes, we decided to include the SSIM loss (Structural Similarity Index) as an additional component in the loss function. This enhancement aimed to improve the preservation of structural elements and enhance the visual quality of the de-rained images.

Recognizing the need for further improvements in the loss function to tackle ongoing challenges, we introduced the L2 loss (Mean Squared Error) as a secondary component. This addition enhanced overall performance by minimizing discrepancies between the predicted de-rained images and the corresponding ground truth images. Through these adjustments to the loss function, we aimed to empower the networks to capture better the desired properties of rain removal, such as effectively eliminating rain streaks while preserving important image details. These modifications were crucial in addressing the limitations and achieving significant advancements in the quality of rain removal, bringing us closer to surpassing the current state-of-the-art techniques.

CHAPTER 4

Single Image De-raining with Unet using CNN (SID-U-CNN)

4.1 Encoder-Decoder of UNet

The U-Net architecture[12, 15] has been extensively used to research single image de-raining and has produced encouraging results. In U-Net architecture, single image de-raining is accomplished by combining decoder and encoder networks.

The encoder network in U-Net collects the high-level characteristics and contextual information from the input image. It typically includes several convolutional layers, followed by layers that are maximize pooling. The max-pooling layers down-sample the feature maps, reducing their spatial dimensions, while the convolutional layers execute feature extraction by applying filters to the input image. This down-sampling helps in preserving the image's overall composition and information.

On the other hand, the U-Net decoder network requires reconstructing the de-rained image using the encoder's extracted characteristics. Deconvolutional layers, often transposing convolutional layers, gradually up-sample the feature map's spatial dimensions. The spatial features lost during the down-sampling process in the encoder are recovered with the aid of the deconvolutional layers. Furthermore, U-Net incorporates skip connections to create simple connections between equivalent levels in the encoder and decoder. These skip connections give the decoder access to the encoder's low-level and fine-grained data, enabling accurate localization and protecting crucial details in the de-rained image.

The model can accurately capture the global context and local features of the input image by merging the encoder and decoder networks in the U-Net architecture. This is essential for the de-raining operation since it enables the network to comprehend the patterns of the rain streaks and produce excellent de-rained images. The research community has embraced the U-Net architecture widely and has proven successful in single image de-raining applications.

Implementing the encoder-decoder architecture of the Unet[12, 6] along with a rain removal model, has demonstrated significant improvements in the field of single image de-raining. The architecture includes a decoder and an encoder that cooperate to complete the task of removing rain. The encoder and decoder use six layers of up- and down-sampling, respectively.

4.2 The Novel Method of Rain-Streak Removal with Encoder-Decoder of Unet

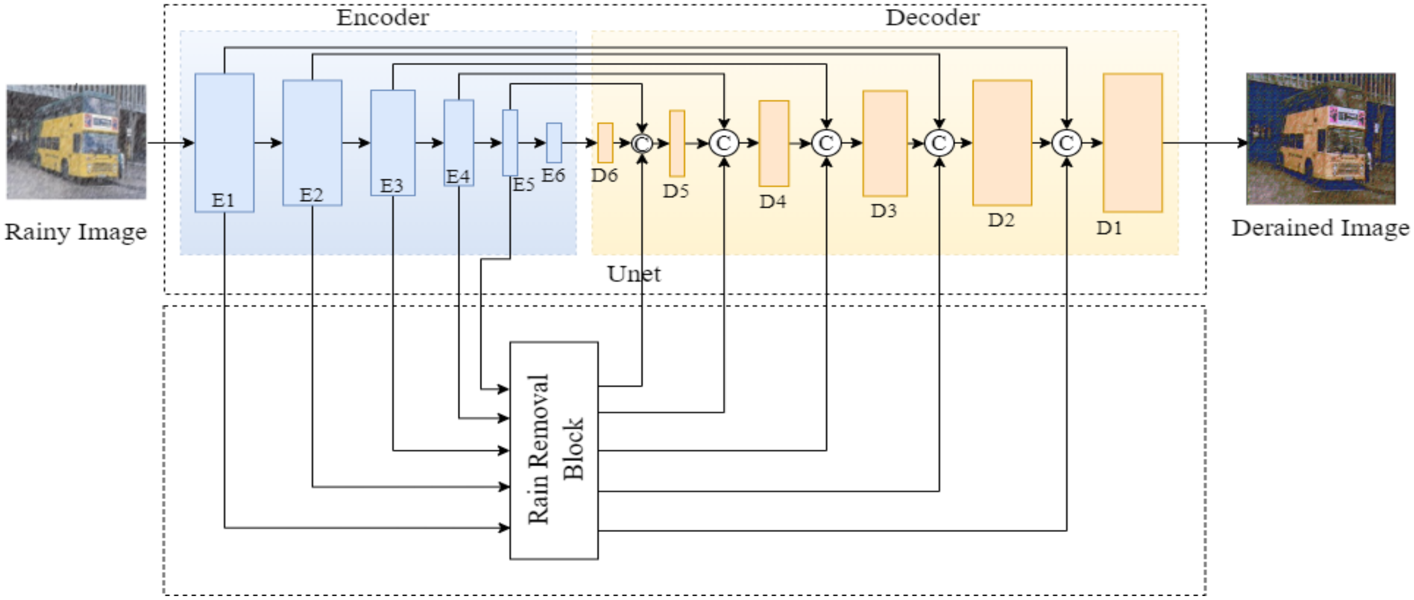

Figure 4.1: The architecture of the SID-U-CNN.

The novel method for removing rain streaks incorporates the Rain Removal Block, which consists of multiple convolutional layers with Rectified Linear Unit (ReLU)

activation functions, is mentioned in figure 4.1. The Rain Removal Block plays a crucial role in eliminating rain and includes residual blocks that capture intricate rain-related features. These blocks are composed of a series of convolutional layers and ReLU activations. Notably, the residual blocks employ skip connections, allowing information to directly flow from the input to the output of the block. This approach significantly enhances the network's capability to effectively remove rain streaks while accurately preserving important image features.
The innovative approach for rain-streak removal incorporates a novel method that effectively eliminates rain streaks. This method utilizes a specialized block, referred to as the Rain Removal Block, which comprises multiple convolutional layers with Rectified Linear Unit (ReLU) activation functions. The Rain Removal Block plays a crucial role in the process by capturing intricate rain-related features through the incorporation of residual blocks. These blocks, consisting of a sequence of convolutional layers and ReLU activations, allow for the direct flow of information from the input to the output of the block through skip connections. This unique methodology significantly enhances the network's ability to remove rain streaks while accurately preserving essential image features.

Essential parts of the single image de-raining architecture created expressly to improve rain removal performance are the RainRemovalBlock and ResidualBlock. These blocks are developed as modules within the PyTorch framework to predict and eliminate rain streaks from images accurately. We execute this by using convolutional layers and skip connections.

As an innovative essential building block for rain removal, the RainRemovalBlock makes extracting features connected to rain easier which is illustrated in figure 4.2. There are three primary phases to it. First, two convolutional layers process the input tensor representing the three-channel image. The network can gather pertinent information about rain streaks due to these layers' execution of spatial filtering operations with a kernel size of 3 and padding of 1. Then, non-linearity is introduced using Rectified Linear Unit (ReLU) activation functions, enabling the model to learn complicated patterns and improve its expressive capability.

To improve the rain removal procedure even more, the RainRemovalBlock additionally integrates ResidualBlocks. These blocks include convolutional layers that extract and manipulate rain-related features at various levels of abstraction. The

ResidualBlocks permit the flow of information from previous layers to the last layers by applying skip connections, maintaining critical image characteristics, and facilitating the precise elimination of rain streaks.

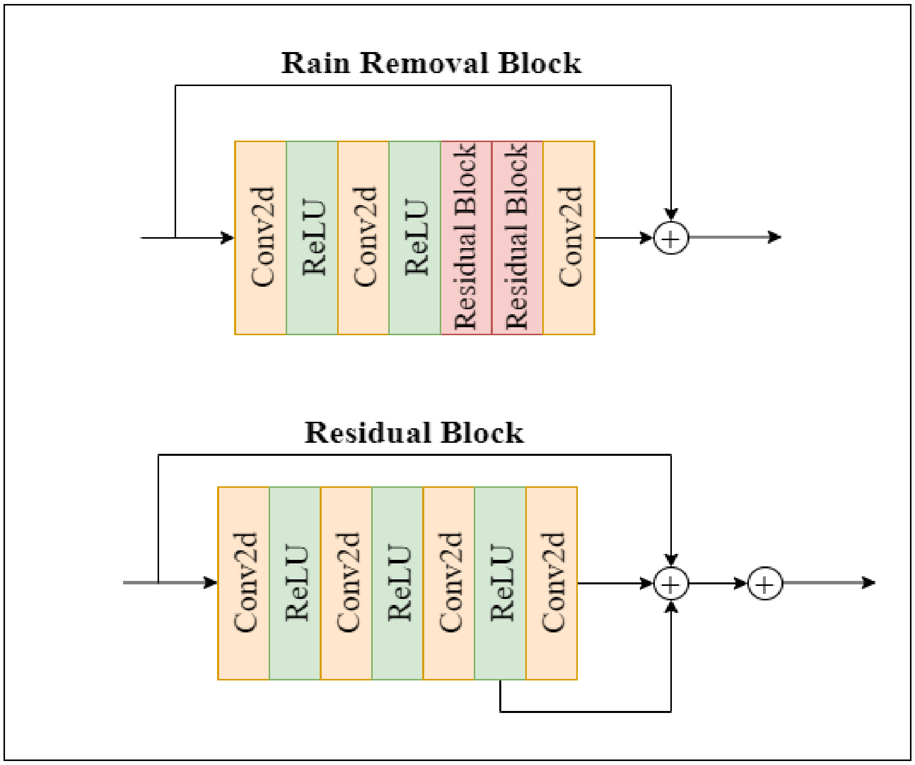

Figure 4.2: The architecture of the Rain Removal Block.

The novel approach to rain streak removal uses the final convolutional layer after the ResidualBlocks to create the output tensor representing the estimated residual rain component. This component defines the difference between the input image and the de-rained image. The estimated residual rain component is added to the input tensor to produce the de-rained image. By combining the estimated residual rain component with the original image, this addition process creates an output tensor that symbolizes the rain-free image.

On the other hand, the ResidualBlock, a vital part of the novel approach to rain streak removal, captures and processes rain-related features in a residual way. It has four convolutional layers, each with a kernel size of three and one padding. The ReLU activation functions add non-linearity after each convolutional layer, much like the RainRemovalBlock. The addition operation creates the remaining connections within the block. By allowing data to move from earlier layers to the last layers, these connections enable the network to capture both low-level and high-level rain-related components efficiently. The residual rain component calculated by the ResidualBlock is represented in the final output tensor.

The model can predict and remove rain streaks from images by integrating the RainRemovalBlock and ResidualBlock into the Single Image de-raining architecture. This allows the model to use convolutional operations, skip connections, and residual learning. Using these blocks, the model can capture information linked to rain at several levels of abstraction, enabling precise rain removal while keeping crucial image elements.

4.3 Loss Function

In this section, we discuss the loss used to train our model.

4.3.1 MAE or L1 Loss

L1 loss, often known as the mean absolute error (MAE), is a standard loss function in image processing. It calculates the average absolute difference between each pixel of the expected and actual images. By penalizing significant changes in pixel values, L1 loss preserves image structure and details by concentrating on the size of errors. To produce images that closely resemble the ground truth regarding pixel intensity, L1 loss must be minimal.

$$L_{l1} = \|\hat{x} - x\|_1 \tag{4.1}$$

4.3.2 MSE or L2 Loss

L2 loss, also known as mean squared error (MSE), calculates the average squared difference between ground truth and predicted images. Minimizing the squared disparities penalizes significant faults and seeks to produce more even images.

$$L_{l2} = \|\hat{x} - x\|_2^2 \tag{4.2}$$

4.3.3 Perceptual Loss

The perceptual loss is feature-based loss, and in our case, extracted features from layer relu1_2 of pre-trained network VGG-16. Let F(.) denote the features obtained using the VGG16 model[18]; then the perceptual loss is defined as follows

$$L_{perc} = \frac{1}{NHW} \sum_i ||F(\hat{x_1})^i - F(x_1)^i||_2^2 \tag{4.3}$$

where N is the number of channels of F(.), H is the height and W is the width of feature maps[18].

The total loss function is defined as

$$L = \lambda_1 L_{l1} + \lambda_2 L_{l2} + \lambda_3 L_{perc} \tag{4.4}$$

In our experimental setup, we have set values $\lambda_1, \lambda_2, \lambda_3$ are 1, 1, 0.25 respectively. We utilized the Adam optimizer with a learning rate of 0.00005. We trained our model using a batch size of 1 and conducted training for a total of 90 epochs.

CHAPTER 5

Experiments and Results

5.1 Datasets

1. DID-MDN dataset 12k : A new dataset with 12,000 images was produced and designated "Train1". Each image in the dataset is labeled based on the amount of rain it depicts. Light, medium, and heavy are the three rain-density descriptors included in the dataset [23, 24]. There are roughly 4,000 images connected with each rain-density label, giving the dataset a balanced spread of rainfall levels. A new test set called "Test1" was also produced, consisting of 1,200 images in addition to the training dataset. Test 1 includes photos with different rain streak sizes and orientations, much like the training dataset. These photos are crucial for assessing how well de-raining algorithms function in various rain scenarios [23, 24]. Another testing set called "Test 2" was arbitrarily chosen from the synthetic dataset further to evaluate the generalization potential of the suggested technique. Test 2 assesses the algorithm's capability to handle various rain patterns and changes and comprises 1,000 images[23, 24]. A large variety of instances with different rain densities, sizes, and orientations are offered by the datasets Train1, Test1, and Test 2. They help researchers create reliable strategies to manage a range of raininess and generalize to unseen images by serving as valuable resources for training and testing de-raining algorithms.

2. Rain800 : The training set, "Rain800," consists of 700 images. The images also originate from the BSD500 training set and the UCID dataset[16]. The images in Rain800, however, have many more rain streaks than the test dataset. As a result, de-raining algorithms may understand the characteristics and variations of rain streaks in many settings and can be trained on a wide variety of cases. In contrast, the test dataset, "Test 100," comprises 100 im-

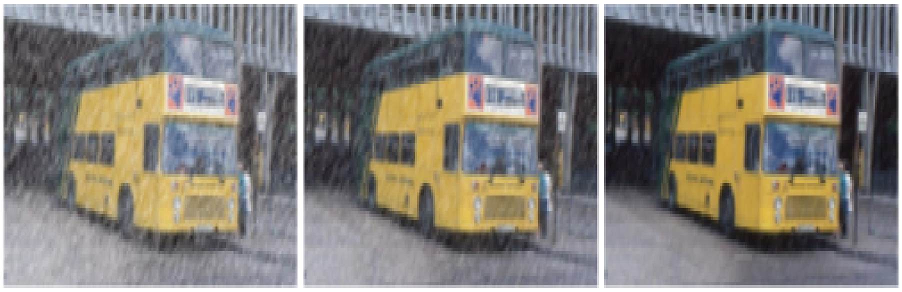

Figure 5.1: Samples synthetic images (Heavy, Medium, and Light) in three different conditions[23].

ages. These images were chosen from the BSD500 training set and the UCID dataset [17, 1]. These images have artificial rain streaks applied to them to make it look like it is raining. This dataset assesses how well de-raining algorithms perform on real-world images with synthetic rain streaks. The effectiveness and generalizability of the researchers' de-raining algorithms may be evaluated using both the Test 100 and Rain800 datasets. The test dataset contains real-world images with artificial rain streaks, which makes it easier to assess how well the algorithms perform under actual conditions. To improve the algorithm's capacity to handle various types and densities of rain streaks, the Rain800 training dataset offers a sizable and varied set of training examples. These datasets are essential for creating and assessing de-raining algorithms, enabling the performance evaluation of various approaches, and advancing single image de-raining research.

3. Rain 100L & Rain 100H There are 100 testing images and 1800 training images in the dataset used in this study. They used the BSD200 dataset[13], and background images were chosen. Light rain streaks pointed in one direction can be seen in each image in the dataset. A second dataset, Rain100L, was also produced; it has 1800 training and 100 test images. From BSD200[13], background images for Rain100L were selected. It is essential to remember that adding strong rain streaks to an image may help de-raining algorithms work better.

5.2 Experiments & Results

We thoroughly tested the Trident method utilizing several datasets, different subnetwork configurations, and various loss functions. We first used the Rain100L and Rain100H datasets to evaluate the effectiveness of our technique. But after realizing the necessity for a more extensive and varied dataset, we trained our model using the DID-MDN 12k dataset.

The DID-MDN 12k dataset includes a substantially more extensive collection of 12,000 images than the Rain100L and Rain100H datasets, which only have 100 images. This dataset includes Light, Medium, and Heavy variations across various rain streak densities. But to ensure the DID-MDN 12k dataset was compatible with our method's particular requirements and adjustments, we carried out crucial data cleaning operations before using it.

By merging these distinct datasets, we attempted to thoroughly assess the efficacy of the Trident approach, considering varying weather circumstances and image properties. Such thorough analyses enable us to evaluate our technique's resilience and generalization abilities under various circumstances, ultimately producing more trustworthy and precise answers.

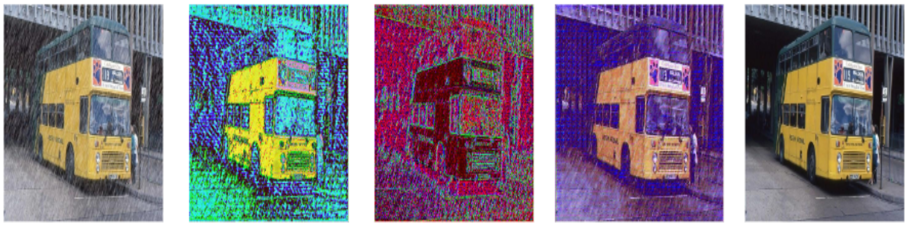

Figure 5.2: Results from RCLU. From left to right - Rainy Image, Residual Map, Confidence Map, RCLU, Ground Truth.

The results of our RCLU, which makes use of Unet's encoder-decoder architecture[12] and the UMRL (Uncertainty Guided Multi-Scale Residual Learning)[21] technique using RN (Residual Network) and CN (Confidence map Network) networks, are shown in Figure 6.2. A complete set of images, comprising the original input image, the predicted output image, as well as the other residual map and confidence map, is shown in Figure 6.2.

The estimated rain streak component is displayed on the residual map, showing the areas where rain has been observed. The confidence map, on the other hand, indicates the degree of confidence or uncertainty connected to the calculated residual values at each pixel. These supplementary maps offer insightful information on how the network evaluates rain streaks and its confidence in the predictions.

Figure 5.2 provides a comprehensive image of de-raining and the network's capacity to extract rain streak information from the input image by displaying the related images with the residual map and confidence map. This visual depiction makes it possible to evaluate our suggested method's performance and accuracy and conduct a thorough investigation of its efficacy.

Figure 5.3 visually compares results from our suggested approaches with those from currently used methods and the ground truth. Each image is accompanied by numerical image quality measurements, such as Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) scores.

Dataset	*Test-1*	*Test-2*
Rainy Image	*21.15 | 0.77*	19.31 | 0.77
JORDAR[3]	24.32 | 0.86	22.26 | 0.84
DDN[4]	27.33 | 0.90	25.63 | 0.88
DID-MDN[8]	27.95 | 0.91	26.08 | 0.90
UMRL[6]	29.77 | 0.92	26.67 | 0.92
RCLU	10.93 | 0.03	11.05 | 0.06
SID-U-CNN	22.15 | 0.78	19.98 | 0.77

Table 5.1: PSNR and SSIM comparison of our methods against state-of-art methods (PSNR - SSIM))

Additionally, Table 5.1 thoroughly compares our suggested methodologies and other state-of-the-art approaches. It systematically compares multiple metrics and performance indicators using diverse methods. Compared to the current techniques, the analysis shows that our proposed methods offer greater accuracy and performance. Despite these encouraging outcomes, we acknowledge the ongoing work to improve the performance of our methodologies further. We're still dedicated to enhancing our strategy, investigating fresh ideas, and implementing cutting-edge methods to produce more precise and trustworthy solutions. We want to make a significant impact and advance the field of image processing research by consistently pushing its limits.

To improve the results, we experimented with Unet's encoder-decoder network, using both channel attention[12] and pixel attention techniques[12]. By construct-

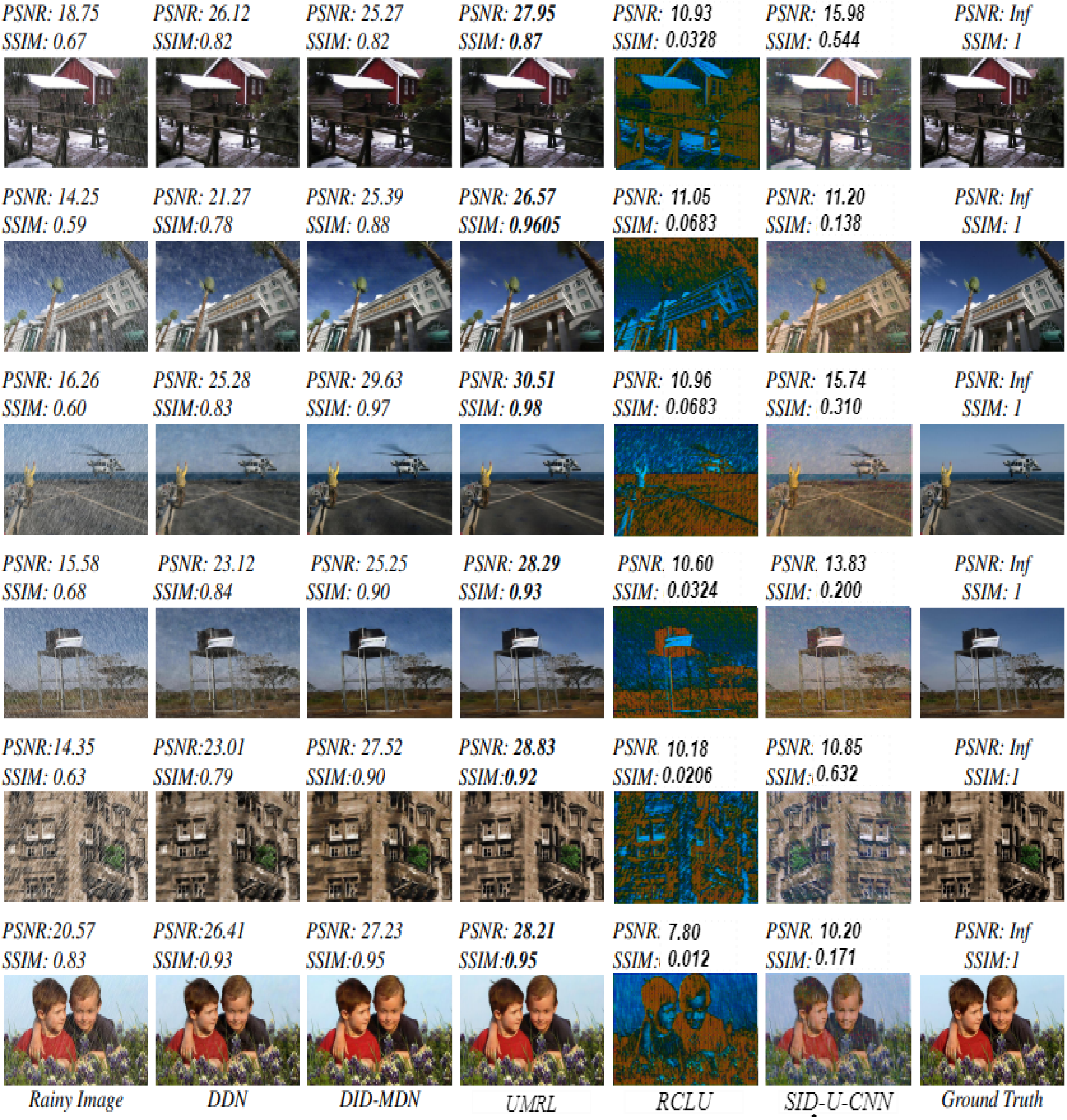

Figure 5.3: De-rained results on synthetic datasets Test-1 and Test-2[23] consisting of different rain levels (low, medium, and heavy) and different directions.

ing a channel-wise attention map based on global information, channel attention seeks to capture the dependencies between channels inside a feature map. By creating a spatial attention map utilizing local information, pixel attention, on the other hand, seeks to capture the dependencies between various spatial places inside a feature map.

We saw a few improvements after using this attentional strategies[7]. However, we looked at other strategies and changed the attention mechanism in the network design to a block that removes rain. Comparing this adjustment to the attention-based approach[14], the findings were more encouraging.

Despite these developments, our results still need to be improved to match the most recent performance in the field. Because of this, our present attention is on ways to improve the results of our model and extend its possibilities.

5.3 Ablation Studies

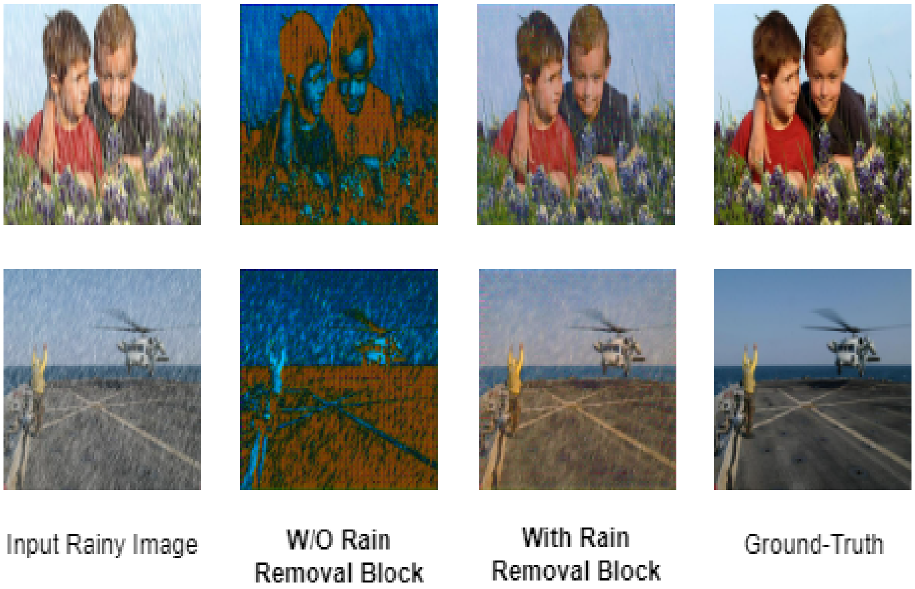

Figure 5.4: De-rained results on without and with Rain Removal Block using SID-U-CNN

The Rain Removal Block is crucial in removing rain streaks from the images. This is evident from the experimental results presented in Figure 5.4, which demonstrate the outcomes obtained without the Rain Removal Block. The inclusion of this block, along with the specific loss function, significantly contributes to the effective removal of rain streaks. Both the Rain Removal Block and the employed loss function are specifically designed to address the challenge of rain streak removal. These components are carefully designed and optimized to prioritize the removal of rain streaks and improve the overall quality of the de-rained images.

The loss function employed in our de-raining technique comprises three components: L1 loss, L2 loss, and perceptual loss. The primary objective of the technique is to remove rain streaks, for which the L1 loss is predominantly utilized. In our experiments, we systematically varied the values of the lambda parameters associated with the L1, L2, and perceptual losses. Through extensive experimentation, we identified the optimal lambda values that yielded the best results. Specifically, we determined that setting lambda values of 1 for both L1 and L2 losses and 0.25 for the perceptual loss resulted in superior de-raining performance. These lambda values strike an appropriate balance between emphasizing the removal of rain streaks (L1 loss) and preserving important image details (L2 loss and perceptual loss).

Chapter 6
Conclusion & Future Scope

The objective of our research is to develop an end-to-end deep learning approach that addresses the challenges associated with single image de-raining using a convolutional neural network (CNN). Our goal is to surpass the performance achieved by current state-of-the-art deep learning techniques in this field. While our method may not surpass the most recent results, it demonstrates excellent performance on images with light to medium levels of rain streaks. Our technique is trained on a comprehensive dataset of de-rain images, enabling it to effectively detect and remove rain streaks while preserving the essential structures and details of the objects in the images. To further improve the de-raining outcomes, we have explored alternative architectural methods, such as adding a dedicated rain removal block. This specialized block has proven to be highly effective in suppressing rain streaks while preserving the intrinsic features of the image. It has been specifically designed to address the unique requirements of rain removal tasks.

Despite these researches, we acknowledge that the field of single image de-raining still has growth opportunities. Our constant goal is to improve our method to surpass current state-of-the-art techniques in terms of performance. We aspire to contribute to the field of image processing and open the door for more effective and efficient de-raining solutions by continuously iterating and implementing current methodologies.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.

[2] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[3] S. L. et al. Single image deraining: A comprehensive benchmark analysis. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3833–3842, 2019.

[4] X. Fu, J. Huang, X. Ding, Y. Huang, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26:2944–2956, 2017.

[5] 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)*, page 1715–1723, July 2017.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] K. Jiang and et al. Multi-scale progressive fusion network for single image deraining. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8343–8352, Seattle, WA, USA, 2020.

[8] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, page 2736–2744, 2016.

[9] Y. Liang, S. Anwar, and Y. Liu. Drt: A lightweight single image deraining recursive transformer. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 588–597, 2022.

[10] J. Liu, H. Wu, Y. Xie, Y. Qu, and L. Ma. Trident dehazing network. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1732–1741, Seattle, WA, USA, 2020.

[11] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo. Swin transformer: Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021.

[12] S. Mandal, A. Dhedhi, and R. L. Das. Non-homogeneous dehazing of images by attention mechanism in deep framework. Mtech Thesis, DAIICT, 2022.

[13] D. Martin, C. Fowlkes, D. Tal, and J. Malik. A database of human-segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 2, page 416–423, 2001.

[14] G. Qian, R. T. Tan, W. Yang, J. Su, and J. Liu. Attentive generative adversarial network for raindrop removal from a single image. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[15] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, volume 9351, page 234–241, 2015.

[16] G. Schaefer and M. Stich. Ucid: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia*, 2003.

[17] G. Schaefer and M. Stich. Ucid: An uncompressed color image database. In *Storage and Retrieval Methods and Applications for Multimedia 2004*, volume 5307, page 472–480, 2003.

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *arXiv:1409.1556*, 2014.

[19] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, page 5998–6008, 2017.

[20] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, page 1357–1366, 2017.

[21] R. Yasarla and V. M. Patel. Uncertainty guided multi-scale residual learning using a cycle spinning cnn for single image de-raining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, page 8405–8414, 2019.

[22] R. Yasarla, V. A. Sindagi, and V. M. Patel. Syn2real transfer learning for image deraining using gaussian processes. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2723–2733, 2020.

[23] H. Zhang and V. M. Patel. Density-aware single image de-raining using a multi-stream dense network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, page 695–704, 2018.

[24] H. Zhang, V. Sindagi, and V. M. Patel. Image de-raining using a conditional generative adversarial network. 2019.