Using Deep Learning to improve the quality of rain-accumulated images

A project report submitted in partial fulfillment of the requirements for B-Tech Project

by

P Madhu Chandra
(2018IMT-049)

Under the Supervision of

Dr. Ajay Kumar

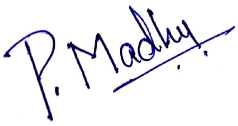
विश्वजीवनामृतं ज्ञानम्

ABV-INDIAN INSTITUTE OF INFORMATION TECHNOLOGY AND MANAGEMENT
GWALIOR-474 015

2021

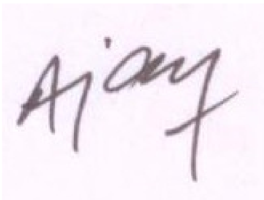
CANDIDATES DECLARATION

We hereby certify that the work, which is being presented in the report, entitled **Using Deep Learning to improve the quality of rain-accumulated images**, in partial fulfillment of the requirement for the award of the Degree of **Bachelor of Technology** and submitted to the institution is an authentic record of our own work carried out during the period *June 2021* to *october 2021* under the supervision of **Dr.Ajay Kumar**. We also cited the reference about the text(s)/figure(s)/table(s) from where they have been taken.



Date: Sunday 31 October, 2021 Signature of the Candidate

This is to certify that the above statement made by the candidates is correct to the best of my knowledge.



Date: Sunday 31 October, 2021 Signature of the Research Supervisor

DR. AJAY KUMAR

ABSTRACT

Computer vision applications in today's world have been extensively developed and employed in different areas such as traffic, security monitoring, object tracking, scene analysis, etc.Rain is one of the typical weather phenomena and excessive rain may affect these applications relying on computer vision.Because these large rainfalls can considerably reduce the quality of pictures shot in these heavy precipitations with air pollutants and rain streaks. It is therefore highly vital to improve the quality of photographs made by rain. The existing methods treats rain removal as a single stage problem and don't make use of the available contextual Information. Due to this rain images from heavy rains usually gets blurred. The major role of our model is to improve the quality of these rainy photos by eliminating the thick rain streaks that overlap each other in many phases using deep learning methods.

Convolutional neural network (CNN) with dilated convolutions have been used in our model to detect and learn about rain streak layers. We have used Recurrent neural network (RNN) to establish the relationship between different stages in rain removal where we will remove rain streak layers in single images and make use of the contextual information. Two data sets Synthesized Rain Image I (SRI-I) and Synthesized Rain Image II (SRI-II) have been proposed and used for training and testing purposes on our model.

KEYWORDS: Convolutional Neural Network (CNN), Recurrent Neural Network (RNN),Peak Signal to Noise Ratio (PSNR), Structural Similarity Index(SSIM)

ACKNOWLEDGEMENT

I am extremely thankful to **Dr. Ajay Kumar** for giving me the opportunity to carry out my research work under his guidance. He have supported me constantly throughout the research work, not only academic support but he have motivated me whenever needed and also increased my confidence which was very helpful for me to trust myself that I will be able to complete my research successfully. He have always answered all my doubts with patience and appreciated my thoughts and also explained me about the improvements which can be done to achieve better results. The research work could be completed successfully without any problem only because of his encouragement and support.

I would also like to thank my peer students and also the Institute for encouraging me at each step and helping me in all possible ways till I could successfully complete my research work.

Signature:

P. Madhu

Name: P Madhu Chandra
Roll. No: 2018IMT-049
Date: Saunday 31 October, 2021

DR. AJAY KUMAR

TABLE OF CONTENTS

List of Figures

Chapter 1

INTRODUCTION

Rain is one of the most prevalent weather conditions and occurrences in our everyday lives.High rainfall generally causes much in the atmosphere, such as haze or layers of mist.These atmospheric moods in multiple different directions with overlapping rain streaks typically obscure the backdrop image.Thus our visual ability typically deteriorates during these severe rains and we cannot perceive the backdrop pictures as they are blocked and blurred. .

For some days now, the use of apps for machine learning and profound learning has greatly enhanced.In the last decade, they were invented and exponentially expanded extremely fast and in our daily lives we can see them used, for example for mobile applications, chat bots, recommendation systems and much more.The significance of data has also increased, which has led to the rise in data driven approaches such as AI, computer vision, natural language processing and other areas.

Computer Vision has also been tremendously developed together with AI and is utilised for a variety of applications. Applications based on computer vision are utilised in many systems such as convenience shops, medicinal reasons such as diagnostics, animal and crop health tracking, automated drive vehicles such as Tesla.The applications are also employed for security objectives in numerous areas, such banking, trade, authentication, classification, identification cards, etc. Vision computer based applications. We can thus conclude that such computer-based apps are extremely commonly utilised in many sectors and ensure they function effectively. .

Likewise, when using these computer vision apps, they also tend to operate effectively under bad atmospheric conditions like precipitation or pollution and don't like our eyesight, which in such situations is generally fluffy.These strong precipitations influence the several visual systems of the computer.Thanks to this rain and air traffic these computer-based technologies cannot collect images effectively and to a certain extent limit the eyesight of such devices. Thus, in such severe rains, different computer-based viewing systems as security surveillance, traffic systems, object monitoring, scene analyses, personal identifying and event detection and autonomous driving vehicles generally fail.Therefore, rain removal in a number of computer-based application applications has become a crucial aspect to further improve their performance for later activities.

It is extremely vital that the quality of such rain collected pictures be enhanced to tackle such vision issues in various computer-based view applications. By improving the image quality, we can make sure that these computer-based systems operate correctly even under severe precipitation. To tackle this problem, we have built an example based on profound education techniques such as revolutionary neural networks and recurrent neural networks. This model removes one layer after another of the rain strips and produces the picture backdrop by eliminating the air plumes generated by rain stripes.

1.1 Rain Removal

Computer vision based systems used in outdoor are usually effected by weather conditions and also due to surroundings. During weather conditions like rainfall and smog the cameras used in these computer vision based systems wont be able to capture images and videos properly. Due to this, the vision of those usually degrades in such atmospheric conditions and it have adverse effects on industries and fields where these computer vision based applications are being used.

Removing of rain in images and videos has become an very important part in various applications. By removing rain in the captured images and videos we can ensure that the quality is improved.

(a) Figure 1.1: Image without Rain

(b) Figure 1.2: Image effected with Rain

Rain removal in videos and images has became a very important step in various fields. This removal process is usually classified into two approaches one approach is based on model and other approach is based on data. We can also call these approaches as Data oriented and Model oriented approaches. Traditional methods normally use model oriented techniques. Model oriented techniques usually depend on the proper utilization of available information like spatial information and properties of background and characteristics of rain streak etc,. In these model oriented techniques we will try to make full use of available information and try to solve it a like a optimization problem by designing solutions to remove rain.

In recent years with sudden surge of deep learning and availability of large datasets, Data oriented techniques have gained a huge interest in every field. In recent years most problems have been using this data oriented techniques and they have been able to establish good results using them. Similarly in our rain rain removal problem also we have employed these data driven techniques and used deep learning so that we can remove rain from images.

1.1.1 Rain removal in Videos

Rain removal in videos is very much different when compared to single images. Because in videos we will try to make use of the information available in whole video. we can differentiate between rain streaks,their directions and background by making use of the temporal information available in video. In videos, we will try to establish the relationship between sequences of frames and make use of that information while removing the rain from videos. As discussed above, rain removal methods in videos are also divided into model oriented techniques and data oriented techniques.

In the video based techniques most traditional methods use the information available in time domain, frequency domain and some methods also use low rank and sparse based techniques.In all of these above traditional methods they have established interdependence in either time or used the frequencies to distinguish between the background and rain streaks.Most of these methods worked well for normal rainfall conditions but during heavy rainfalls these methods either over-smoothed or made background blurry. We will discuss further about these models in below sections.

1.1.2 Rain removal in Images

Rain removal in single images is far more different from videos and is difficult also. In videos we will have temporal information and we will try to establish relationship between various frames to generate background and distinguish background from rain streaks. But in single images we don't have any such information available. Due to this it is very difficult to remove rain in single images and also research related to rain removal in single images has gathered more attention when compared to the latter one.

Rain removal in single image methods have also been divided like model oriented and data oriented techniques. Most of the traditional methods are model oriented because in early days there is a lot of difficulty to have large datasets and it is further more difficult to have datasets of rain images. So, most traditional methods are often model driven and they tried to used methods like dictionary learning,sparse representational and filter based methods. Most these methods have developed a model where they will remove rain streaks in single stage. But usually these rain streaks are of different directions and usually overlap on one another creating adverse effects on captured Images. Due to this during heavy rainfalls, most of these methods usually fail because heavy rainfalls have other effects like atmospheric vielings, smog etc.

Data oriented techniques have increased in recent years. Similarly in rain removal methods also there use has been increased.Another problem with data oriented methods is that availability of data. Because these require large datasets and it is very difficult to have dataset with rain and without rain. So, in these methods we have to make sure that available data is utilised properly. We will discuss more about these methods in later sections. We addressed deep learning, recurring neural networks, convolutionary neural networks and other issues that are employed in our model in the following part.

1.2 Why CNN and RNN ?

There are so many techniques available in deep learning, But among all of them we have only chosen CNN and RNN. Here I will discuss why we have preferred them compared to other techniques.

CNN is usually used for spatial data like images and it is very good for extracting features, whereas RNN is used for temporal or sequential data like videos. CNN networks usually take input of fixed size and also generate outputs of fixed size whereas RNN can take inputs of any length and similarly can generate outputs of any length. CNN is more commonly used for images and RNN is used for speech or text analysis.

Knowing all those differences described above we may think how we can use two different networks, which are used for two different purposes. But, we can use them because usually CNN is very good at labelling and classifying but it usually lacks at temporal information like sequence of data. We can overcome this drawback by using RNN at that point. So by combining CNN and RNN in our model we can use CNN to label and separate different rain streak layers, as we usually remove rain iteratively in several different steps we will make use of available contextual information by incorporating RNN to our model.

1.3 Motivation

This work is motivated by the following factors: Firstly, In today's world, Computer Vision System are of paramount importance in real life. With the rapid developments in technology, Computer vision field is developed vastly and these computer vision based applications are used in various works in our daily life.

Secondly, Rain is the one of the normal climate and weather phenomena occurring in our daily life, during heavy rains most of these computer vision based applications like security surveillance, Traffic systems and automated driver less car will likely to fail due to the atmospheric vielings caused by heavy rains.

Thirdly, attempts of solving the problem using different machine learning and deep learning techniques were started from previous decade. But it is still an active research problem in Computer vision. Most of the proposed solutions have treated this rain removal problem in single images as signal separation problem trying to separate background image from rain streaks.But very few research has been done using Deep learning and only a few attempts have been made so far.

1.4 Objectives

- To improve the quality of images affected by rain streaks.
- To develop a model which will remove rain streaks in several stages and make use of the available contextual information in later stages.
- To make comparison of the proposed model and state of art methods by using performance metrics used in previous papers.

1.5 Research work flow

The report will describe the workflow as follows, in accordance with the study objectives:

Step 1 : In order to develop a rain removal pattern as described in the objectives, the function of the neural Convolution and the neural recurrent network must be well understood. The history and a good knowledge of CNN and RNN are discussed in Chapter 2.

Step 2 : The chapter 2 also discusses various research works that have been done so far in rain removal in both videos and images. It also discusses about different model oriented and data oriented techniques used for rain removal till now. It also discusses in depth about the limitations of those models.

Step 3 : Developing and analyzing the deep learning model which can remove rain from images and make use of available contextual information with the help of convolutional neural network and recurrent neural network (Chapter 3).

Step 4 : Using these proposed algorithms to develop our model and evaluation of the model developed (chapter 4).

1.6 Conclusion

The problems of rain effected images and videos have been discussed in this chapter. This chapter discusses about the need and importance of rain removal in images and videos for computer vision based applications. A new model has been proposed by us using CNN and RNN to achieve better results which we will discuss in further chapters.

Chapter 2

LITERATURE REVIEW

This chapter primarily focuses on the existing rain removal methods in images and videos. It presents the required background knowledge, review of key related search, research gaps, problem formulation and conclusion.

2.1 Deep Learning

Deep Learning is a Machine Learning artificial intellectual and subfield that deals with algorithms inspired by human brain function and structure, dubbed artificial neural networks. We can attain outcomes that were not before achievable by means of profound learning. Deep education helps achieve the state-of-the-art accuracy, which can be more than human level performance, as stated in [13].Deep learning is the cornerstone to a number of current technologies, such as autonomous automobiles, photo detection etc.

Cognitive science and machine learning have changed their paradigms a lot over many years, but there was a big interest recently among technology giants such as Google, Facebook and Amazon, who invested in the creation of technology to learn deeply. The future of research and development is deep learning technology.

Deep learning represents the set of machine learning algorithms which

- Do the extraction of the task function using several cascade levels that comprise non-linear processing units. The current layer is entered from the previous layer output.
- Learning takes place in supervised as well as unattended ways.
- It consists of the use of neural networks that comprise horizontally cascaded layers.

2.2 Background : Convolutional Neural Networks

Deep learning is a deep neural network class used most frequently for the processing of visual images (CNN or ConvNet). You have numerous applications, such as natural language processing, image identification and classification, medical picture analysis, financial time series and recommendation systems. CNNs are regularised perceptrons with many layers. Typically, multilayer perceptrons indicate every neuron in a layer of the next layer has a full relationship with another.

This network's "complete accessibility" allows them to circumvent data. Applying any type of weight computation to the loss function requires usual regularisation methods. The CNNs use the hierarchy of data to adopt a new regularisation strategy; with smaller and simpler patterns, they build increasingly complex patterns. The level of connection and complication of CNNs is thus at the lower edge. Compared to other methods of image categorization, Very minimal pre-processing is used by CNNs. The web then finds that traditional algorithms have created filters. This freedom from past expertise and the effort made by people in the development of applications is a big advantage.

2.2.1 Components of Convolutional Neural Networks

The CNN or ConvNets neural networks are made up of neurons with changeable convictions and training weights. By neuron certain input data are collected, working with a dot product that assures non-linearity. CNN has been taught to minimise a loss function based on the last fully-connected layer performance. The inputs of CNN design are pictures as an evident assumption. CNN Architectures are built on three main layer formats: foundation layer, layer pooling, fully linked layer.

2.2.2 Convolutional layers

The convolutionary neural network calculates the discreet calculation of convolution in the convolutionary layer. Indeed, a Linear Transformation is a discreet convergence. The discrete convolution is sparse here, as a single output is only achieved by a restricted number of input units. Input parameters are repeated via the discrete convolution, implying various locations with the same weights. The discreet convolution applies the same filter at numerous picture places when an image is supplied as an input. The illustration below illustrates the visual process of convolution.

2.2.3 Pooling layers

The function map performance of convolutional layers is limited by documenting the accuracy of the features in the data. This implies that a separate map arises from small movements in the position of the component in the input image. It will happen as the input image is recropped,rotated, shifted, among other small modifications.

Sampling is considered a common approach to solving this problem from signal processing. A lower resolution representation of an input signal, including the large or relevant structural elements only, is produced without the fine information, which are not useful for the mission.Down sampling with convolutional layers can be done by adjusting the processing phase around the image. Using a pooling layer is a more stable and growing solution.After the convolution layer, a pooling layer is added. In particular, the inclusion of a pooling layer after a convolutionary layer after a non-linearity (e.g. ReLU) is a common technique used to order layers in a convolution neural network, which can be replicated one or more times in a given model.

The pooling layer works independently on each characteristic map to create a new collection of feature maps of the same number. The pooling requires the collection and implementation of a pooling process, almost like a filter for maps. The pooling method or filter size is smaller than the characteristic map size; in fact, the code is almost always 2 x 2 with a 2-pixel phase.In other words, the pooling layer also reduces the size of each characteristic map by a factor of 2, for example, every dimension is halved and reduced to 1 quarter the number of pixels or values on each characteristic map. For eg, a pooling map with a configuration of 6x 6 (36 pixels) results in a pooled output map of 3 x 3 (9 pixels).

The process of pooling is specified instead of known. Two normal apps

- Max-Pooling: In the pooling window, the largest valued element will be used.
- Average-pooling: The mean of all the items in the pool window would be used.

Pooling utilises a window sliding across the characteristic maps. For the measurement of the sample value of the components in the pooling window, such as Max-Pooling or Average-Pooling is used.

2.2.4 Fully connected layers

The neurons have complete connections in a completely connected layer with all previous layer activations, ensuring that every neuron in an earlier layer is linked to the next layer of neurons. The output data from the convolution and pooling layers represents the high-level picture data. These characteristics are utilised to categorise the picture in a classification issue in several groups by the fully connected layer.

2.2.5 Activation functions

The activation functions include non-linearity inside the network, which allows the neural network to approximate any function. Four typical functions for activation are Sigmoid, Tanh, ReLU, and Leaky ReLU.ReLU and Leaky ReLU are also utilised in the convolutionary layer.

- A sigmoid function is defined as

$$f(X) = \frac{1}{1+e^{-x}}$$

  The Sigmoid function output falls (0,1). The neural network utilising sigmoid suffers from the loss of the gradient issue if we utilise gradient-based training. In the front layers of the networks the gradient value declines considerably, making the previous layer sluggish.

- A Tanh function is defined as

$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

the output of tanh function falls into (-1,1).

- A Rectified Linear Unit is defined as

$$f(x) = max(0, x)$$

ReLU has lately become widely used and the learning process has been significantly shortened because to its circular non-saturating shape. ReLU is quicker than Sigmoid and Tanh processing levels. Nevertheless, when ReLU is ready to continue to provide the same value for each data, the ReLU units will die.

- Leaky ReLUs are one attempt at solving the death ReLU problem. The leaky ReLU does not have a tiny positive gradient for negative inputs, but rather makes the function zero when x¡ 0. The definition is It

$$f(x) = \begin{Bmatrix} 0.01 * x & x < 0 \\ 1 & x \geq 0 \end{Bmatrix}$$

2.3 Recurrent Neural Networks

RNNs are a network model which connects many neuron units in time. The information is disseminated in a similar fashion. Based upon the prior activity and inputs provided, the neuron develops.

The representation of the latent state is constant in contrast to the hidden Markov model.When RNN uses input information much later, the true difference occurs.In the Long Short Term Memory (LSTM) networks, this is highly crucial.

These networks may be used anywhere time series data are required. In translation, for example, speech to text and other disciplines such as image categorization and to learn about rain image contexts, to produce backdrop pictures.

2.4 LSTM

Long-Short term memory is an advanced approach used to eradicate the issue of the repetitive neural networks (RNNs) LSTM implements memory blocks in contrast to traditional plain RNN modules. A memory block includes a memory cell and a set of gates. For catching long-term connections, this becomes more efficient. In general, a repeating hidden layer function can be defined as follows at a time.The representation of LSTM unit can be in seen in the Figure 2.1. This figure has been taken from [25].

As f, o and i denote the forget, output and input gate, b term denote bias, w denote weight matrices, c marks a memory cell, is a non-linear element-significant sigmoid activation function, values in the range [0,1], tanh is a non-linear function unit, values in the range [-1,1].

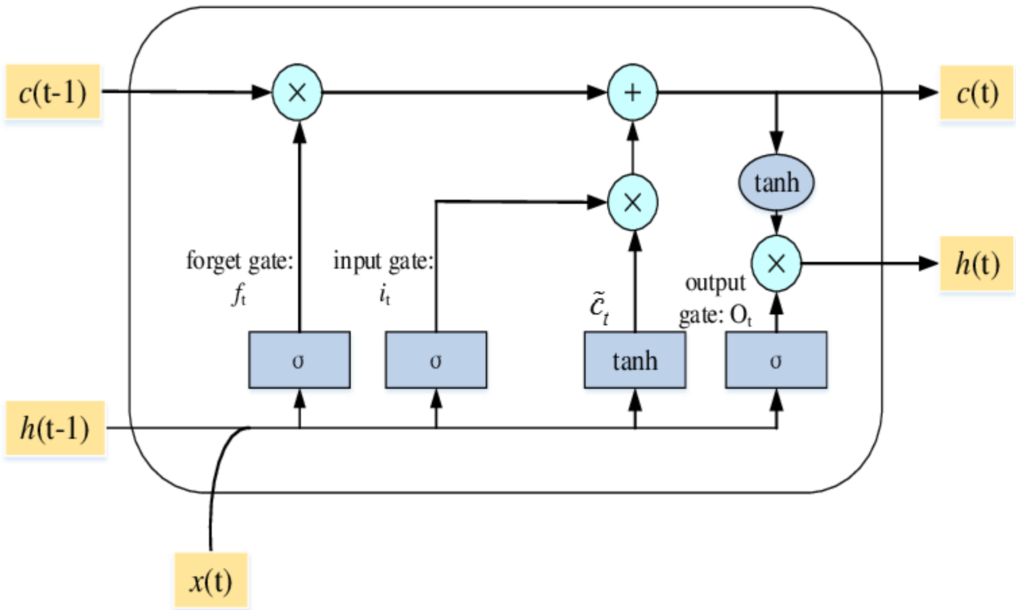


Figure 2.1: Long Short Term Memory Model

2.5 Review of key related research

Rain removal is one of the most important step for various computer vision based applications.In the last few years, several approaches have been discussed to separate those rain streaks from background. In this section, we will discuss about the key research which has been carried in this area and we will also discuss limitations of some of the existing methods.

Rain removal methods proposed till now can be broadly divided into two types like Model oriented techniques and Data oriented techniques depending upon the way we are designing techniques. In Model oriented techniques, we will try to make use of the available intrinsic properties of rain streaks and background.With the help of those properties we will design and optimise a model which can remove rain streaks.

With the recent advancements in fields of technology and computation power, Data oriented techniques have gained a lot of popularity in every field. Similarly ,in rain removal also data oriented techniques have gained popularity but they were limited by the fact that there is very less real data available in this field when compared to other. So, we have to make sure that available data is used properly. In Data oriented techniques, we will design specific network architectures and we will train these network with datasets which contain both rain and non rain data. These network architectures will be trained to learn complex rain streaks by optimising loss functions.

We can further divide our rain removal techniques into two different ways depending upon the type of input they are taking that is Images and Videos. We are dividing them separately because both methods work in different way when compared to other.Similarly in case of videos, we have information of background along with temporal information of various sequential which is very help and useful for rain removal.But in case of images, we don‘t have any such information available.Here we only have spatial information available.So, when compared to videos, rain removal in images is difficult and more challenging.

In [7] Garg and Nayar developed a model based technique which can remove the rain streaks from videos. Here they have used temporal and spatial properties within sequences of frames and compared the differences between frames with rain drops and without raindrops. In [28] Zhang proposed a model using K means clustering which will use the temporal properties to distinguish background scenes from rain in videos.In [18] Park proposed a method which used kalman filter to remove rain in videos.But this methods works well only in videos with still background.

In [2] Brewer proposed a method where he used the rain drops intensity to find the rain streak layers and he replaced those regions with the mean of pixels present in that region in consecutive frames. It worked well in videos with little rainfall or rain streaks because when we are using mean of pixels in consecutive frames. But in case of heavy rainfall this method wont work properly because in heavy rainfall if we can mean of pixels in consecutive frames hen it wont remove rain because every pixel usually have rain drops in heavy rainfall.

In [1] Bossu proposed a model using Gaussian mixture model to detect the rain streak layers in videos. In [12] Kim proposed a method using SVM to separate the images in sequence of frames from rain streak layers using spatial and temporal properties. After detecting these layers we will separate them from image frames of videos so that rain can be removed.Here, we have to make sure that our SVM is already trained with some data.

In [4] Chen used deep learning techniques like convulutional neural network for removal rain in videos.Similarly, In [15] Liu proposed a model called a dynamic routing residue recurrent network which uses recurrent network along with temporal information available in videos to remove rain streaks in videos.The methods which we have discussed above till now have been developed for videos.In below we will discuss about the methods developed for rain removal in single images.

Rain removal in single is far more complex and difficult when compared to videos because we don't have any temporal information available in single images.These single image based methods can also be divided depending upon the type of methods employed.Most traditional methods are model oriented which will try to make use of the chromatic and spatial properties available in images to design filters based techniques and other models. With recent surge in data and deep learning, data oriented techniques are also gaining interest but the problem with them is the less availability of datasets with rain and without rain which is required for training and testing purposes of these data oriented methods.

In [22] Xu proposed a model for removal of rain and snow in single images.Here he used a guided filter[8] which will try to identify rain using chromatic property after that it will filter that image by removing the rain streaks which are obtained previously.This model is further enhanced by Zhen in [29] where he used multiguided filter to identify rain streaks in images.Along with chromatic property, this multiguided filter also uses the brightness, frequencies, It will distinguish frequencies by using their values like high and low parts to identify different rain layers in images.

In [16] Luo proposed a method using discriminative sparse coding, here he has used a dictionary learning technique to estimate the rainstreaks in images and tried to generate the background by removing those estimated rainstreaks from images.In [21] Wang proposed a hierarchical model which employs hierarchical scheme with 3 layers to remove snow or rain streaks in single images.

In [5] Eigen proposed a method using convulutional Neural network for removing dirt and rain drops captured on images.However this methods usually fails in images containing heavy rainfalls and often gives blurry backgrounds of those heay rain images.In [19] Qian, proposed a model using attentive genarative network the model will separate by discriminating the rain regions and non region regions .

In [6] Fu proposed a Deep detail network (DDN) which will try to learn the relation between rainy and clean images by using nonlinear mapping function among them.Here for that learning purpose we have to rain that deep detail network with clean and rain images.

2.6 Research gaps

Our goal is to generate a data oriented deep learning model which can remove rain streaks from rain effected images and generate clean backgrounds without rain streaks.The research works stated in the above section gives us brief knowledge and clear view of different works done in field of rain removal in videos and images.But there are some limitations and research gaps in those works.Among them, according to [9] for removal of rain streak layers in a single Image,contextual and spatial information is very useful and important.But many discussed models won't even use this contextual information instead they image patching techniques.

Another drawback of above discussed models is that in heavy rains they will remove only rain streak layers leaving atmospheric vielings (smog, mist like white layer formed in heavy rains). Due to this the images are usually blurred and sometimes they are over smoothened and background is not clearly visible. Another drawback is that in most models during heavy rainfalls the rain streaks usually overlap with each other in different directions.In such cases, multiple stages are used to remove the rain streaks in images. As several different stages are used to remove rain in a single image, it is also important to note that the information we gained in previous stages can be used in further stages for removing rain. But most of the models don't use this information and treat each removal layer separately without having any correlation between them.

S.No	Paper Title	Author	Details and Limitations
1	Robust video content alignment and compensation for rain removal in a cnn framework. [4]	Chen J et al., (2018)	The work discusses about how to detect rain streak layers in videos shot from fast moving objects and videos during heavy rainfalls. They used superpixel segmentation to divide videos scenes along with convolutional neural networks to remove rain streaks from videos. This model has performed well for videos shot in heavy rainfall from fast moving cameras. The main drawback is that it cannot be used for rain removal in images.
2	Deep joint rain detection and removal from a single image. [23]	Wenhan Y et al., (2017)	The work attempts to separate rain streak layers by using binary maps which provide the location of rain streak regions consisting of rain and background. They removed rain streaks recurrently from those rain regions to obtain clean background. The drawback is that it don't establishes relation between recurrent stages instead it simply works like a cascade network where it will take input from previous outputs.
3	A hierarchical approach for rain or snow removing in a single color image. [21]	Yinglong W et al., (2017)	In this work a 3 layer hierarchical algorithm was proposed to remove rain streaks and snow from images. They detected regions with high frequency parts consisting of rain, snow and image details.Using Dictionary learning they decomposed those regions into rain free and snow free parts. The main drawback of this model is that it usually fails for images with heavy rainfall by blurring the background
4	Rain streak removal using layer priors. [14]	Michael S Brown et al., (2016)	The reasearch uses a method based on simple patch based priors for background and rain layers. These simple priors are based on Gaussian mixture models which can accommodate rains streaks of several directions to separate rain streaks from images.This work fares well compared to other model oriented techniques. The main drawback of this model is that for real images with heavy rainfall it smoothens the background.

Figure 2.2: Literature review of the key works done in the field of rain removal

2.7 Problem formulation

Now a days computer vision based applications are used everywhere. They are used for objection detection, security purposes, driverless cars, traffic surveillance, identification and authentication purposes, medical diagnosis and many more. So, ensuring that these applications work properly is of utmost importance. Similarly, most computer vision based applications used in outside are often effected by atmospheric phenomena like smog, smoke, fog and climate changes.So, there is great need for removing rain in rain affected images.

2.8 Mathematical Model Formulation

Most of the proposed solutions have used the follow- ing rain model (2.1) for rain removal where I represents our captured Image, R represents the rain streak layers formed on our captured Image due to rainfall and atmospheric vielings, B represents the background scene of Image without rain streaks and atmospheric vielings.

$$I = B + R \tag{2.1}$$

The problem with the above model is that it usually treats rain removal problem as signal separation or decomposition problem where it will try to separate the background image from the rain. But, in reality what happens is that these rain streak usually wont be seen as single rain layer.But they are streaks falling in different directions overlapping on one another due to which they will usually degrade the quality of captured images.So, inorder to overcome this problem, we wont remove rainstreak layers at a single stage but we will remove those rain streaks in several stages iteratively. At each stage we will remove a rain streak layer of a certain direction and after that we will use that enhanced image in further stages. To accomodate that we will make modification to base rain model, instead of using single rain layer R, we will use Ri where i will represent rain streaks occurring in different directions and shapes. Now, our enhanced mathematical model can be further represented as:

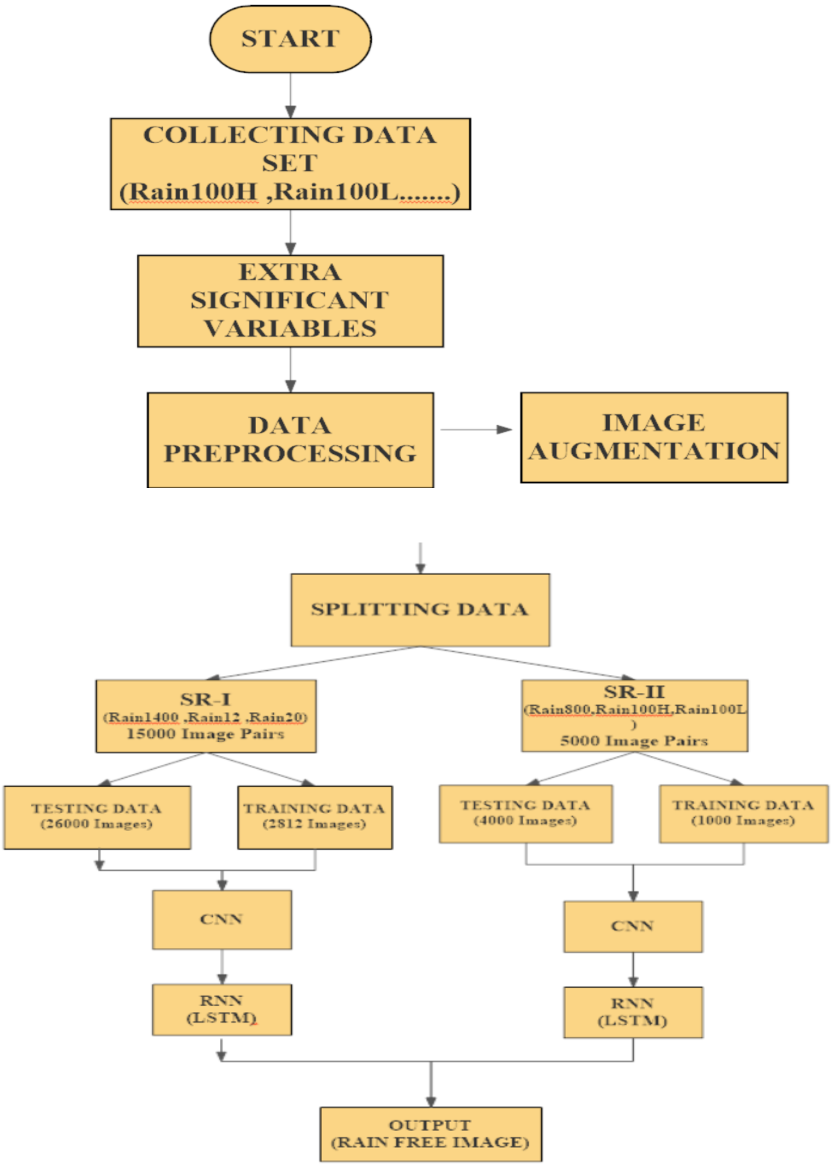
$$I = B + \sum_{i=1}^{n} \alpha_i R^i \tag{2.2}$$

In Eq.(2.2) Ri represents dierent rain streak layers where Ri stands for i-th rain streak layer, @i represents rain streak layer brightness and n represents the maximum number of such different rain streaks. The main objective is to develop a model which will establish relation between several rain removal stages and make use of the available contextual information.

2.9 Conclusion

We propose a deep learning model which can remove the rain streaks from the rain effected image and will give us background image without rain streaks. This model can be used for computer vision based applications which are used outside, by using this model we can enhance the vision of such applications by improving the quality of the images captured in rain.The model takes rain effected images and will remove rain streak layers from images iteratively one after another in several different stages.During rain removal process it also make use of the contextual information available from the previous stages in removal process. The proposed model does not require more resources compared to other models and will take less time for training compared to other previous models.

SYSTEM ARCHITECTURE



Chapter 3

METHODOLOGY

The chapter of methodology discuses about the proposed hypothesis of the research work in section 3.1 . The mechanism of the model, dataset preprocessing, model architecture are discussed in section 3.2. The Analytical validation of our model is discussed in section 3.3 .

3.1 Proposed Hypothesis

The proposal is to build a deep learning model which can remove rain streaks layers from rain effected images. Our model will learn features of rain effected images by using CNN. So, for that purpose what we will do is we incorporate our model with CNN and train them with rained images data. As we want to make proper utilisation of the available contextual we will use dilated convolutions which has larger receptive field when compared to normal convolutions. So, basically we will get more contextual information by incorporating them in our model.

We know that rain streaks will be of different directions and shapes depending upon the particular climatic and atmospheric condition. So, instead of removing rainfall in single stage we will try to remove them in several different stages one rain streak layer after another. In order to make proper utilization of the contextual information available in different steps we will make use of the recurrent neural network.The next section discusses the above steps in detail.

3.2 Mechanism/Algorithm

The process of building a deep learning model for rain removal in single images can be basically divided into three steps. They are:

- Dataset
- Model Architecture
- Training, Validation and Testing

3.2.1 Dataset

One of the main problem of rain removal is proper availability of datasets.For training and testing purposes we need to have data labeled separately i.e images with rain and without rain. But is hard to have such dataset because we cant capture a same image with and without rain at the same time. So, In order to overcome this problem we have used synthesized datasets where rain streaks will introduced in images. We will use those synthesized datasets with clean and rainy images for our model. In our model we will use some standard synthesized rain datasets like Rain100L, Rain100H, Rain12 and Rain800.

- Rain12 and Rain20 :
  The dataset Rain12 [6] consists of 12 image pairs.Similarly Rain20 dataset takes image from surroundings and injects rainstreaks in it. Each pair consists of clean image and its synthesized rain image.These synthesized images consist of only one type of rain streak.
- Rain100H and Rain100L:
  The Rain100H and Rain100L datasets are collected and synthesized by Yang in [23].These datasets also contain images with synthesized rains and without rains. The clean images used in these datasets were taken from BDS200 [17]. Rain 100L contains image pairs with rain streaks of only single direction whereas Rain100H contains image pairs with rain streaks of only five different directions.
- Rain800:
  Sindhagi et al synthesized this Rain800 dataset in [27]. This is one of difficult dataset present in this rain removal field. Sindhagi et al synthesized this dataset by capturing different pictures from surroundings and have injected rain streaks into those captured images. Rain800 dataset contains 1800 image pairs with clean and rain effected images respectively.

- Rain1400:
  Rain1400 dataset contains 14,000 image pairs with rain and without rain. Among them the dataset usually contains 1000 clean images taken from surroundings.To these clean images rain streaks of 14 different directions and have been injected separately. So, that we have a synthesized dataset of image pairs with both rained and cleaned images. We further divided the dataset into testing and training purposes.

```
final_train_data = []
final_target_train = []
for i in tqdm(range(train_x.shape[0])):
    final_train_data.append(train_x[i])
    final_train_data.append(rotate(train_x[i], angle=45, mode = 'wrap'))
    final_train_data.append(np.fliplr(train_x[i]))
    final_train_data.append(np.flipud(train_x[i]))
    for j in range(4):
        final_target_train.append(train_y[i])
```

Figure 3.1: Code snippet showing image augmentation

Using those above datasets we have prepared our two new datasets called Synthesized Rain Image I (SRI I) dataset and Synthesized Rain Image II (SRI II) dataset.For Synthesized rain image I(SRI I) dataset we have used Rain1400, Rain12 and Rain20 datasets. We have combined these datasets that is Rain1400, Rain12 and Rain20 and further divided them into training and testing purposes. Rain1400 contains 14,000 image pairs, among them we divided 26,000 images for training purpose and 2000 images for testing pairs.As Rain12 only contains 12 clean images with different rain streak layers we have used it for testing purpose. Similarly Rain20 which contains 800 image pairs of rained and cleaned images.So,our SRI I dataset contains 15,000 image pairs with both cleaned and rained images.

Similarly Synthesized Rain Image II (SRI II) dataset was created by using Rain800, Rain100H and Rain100L datasets. Rain800 dataset contains 1800 image pairs with both rained and cleaned images. In these each rained images are synthesized by injecting rain streaks of five different directions in each cleaned image.This will generate a complex rain streak and is most difficult when compared to others because in other datasets we synthesized images by using rain streak of only one direction.

Similarly we used image pairs from Rain100L and Rain100H datasets. We combined these three datasets and further classified it for training and testing purposes. These combined dataset Synthesized Rain Image II (SRI II) conatins 5000 image pairs with both cleaned and rained images.

SRI I and SRI II datasets have been used for our model which contains image pairs with rained and without rain images.In order to make proper use of the available datasets we have used image augmentation. Image augmentation helps in generating new images from the existing images so we don't have to capture them separately. In our model we have used following image augmentation techniques using pytorch :

- Shear range(Shears the original image): 0.2
- Zoom range(zooms the original image): 0.2
- Rotation range(image is subjected to rotation at angle of): 40 degrees.

We also allowed horizontal and vertical flips of images. In order to ensure equal augmentation we used a seed.

3.2.2 Model Architecture

The limitation of existing deep learning learning models for rain removal are either not making proper use of available contextual of information or removing rain streaks in a single stage. So in our deep learning model in order to make proper use of contextual information we used CNN with dilated convolutions. Our model architecture is represented in figure 3.2 and we discussed about it in below sections.

3.2.3 Base Model

Our model is developed by first incorporating a convolutional neural network(CNN). Our model is designed with depth of 8 layers and it will take images of size 64 X 64 X 3 as its input. In first layer our model will take image as an input and it will transform that input image into feature map. We will use 3 X 3 convolutions in first layer.

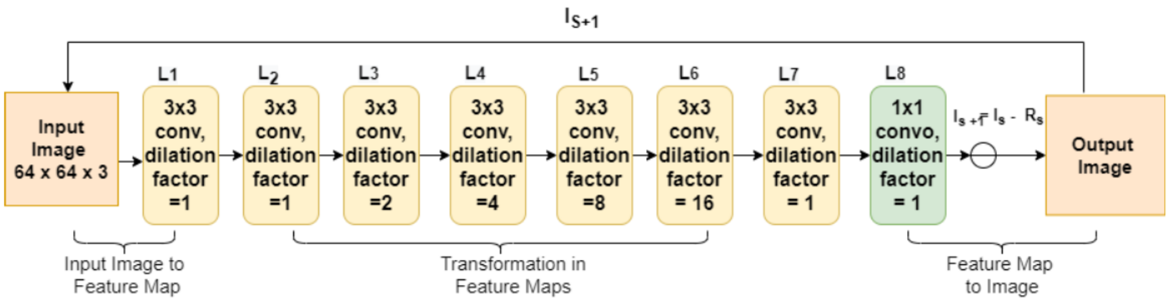


Figure 3.2: Model Architecture

3.2.4 Dilated Convolution

From second layer to seventh layer the features of those rain effected images are learned by our model and feature maps of those layers are transformed accordingly. While learning these features we want to make sure that our model learns more contextualised information.

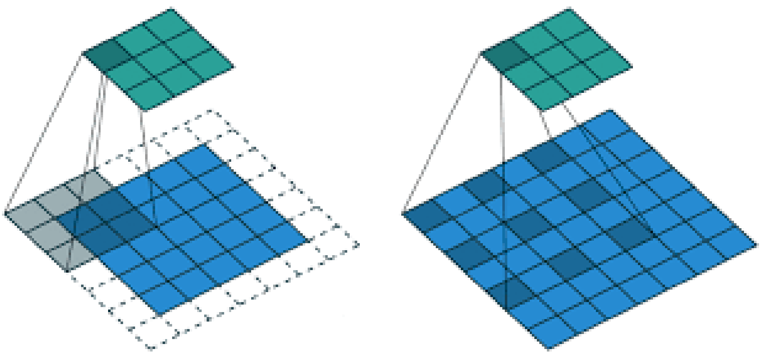


Figure 3.3: Standard Convolution vs Dilated Convolution

So, for that purpose instead of using standard convolution operations we will use dilated convolutions in our model as shown in figure 3.3. The advantage of dilated convolutions is that they will have larger receptive field when compared to standard convolutions. Another advantage is that they require less parameters which will further decrease computation power required for our model.

From Layer L2 to L7 we have used 3 X 3 convolution. Instead of using standard convolution we will use dilated convolutions here use to acquire larger receptive field. So, to incorporate that dilation operation we will use dilation factor of 1,2,4,8,.. from Layer L2 to Layer L6. As dilation factor increases receptive field also increases in every layer. For layer L7 and L8 we use dilation factor of 1. At last layer L8 we will use 1 X 1 convolution. After every layer we have pooling layer where we used average pooling. Similarly we have ReLu layer after every layer except last which will add non-linearity.

At last layer we will get rain streak and we will separate this rain streak from our image to get background without rainfall. We will compare that obtained background image with original background image and try to optimise our model by optimising this loss function.Here we want to separate rain streak layers one after another i.e we want to detect different rain streak layers.During feature extraction process in our model, each feature map is usually given some weight, instead of using those default values we will use squeeze and excitation network which will update the feature maps weights accordingly and help to detect different rain streaks. Thus, each rain streak layer will have different @i value as they updated by squeeze and excitation network in our model.

3.2.5 Recurrent Architecture

As we want to remove rain streaks in several stages one after another, for that purpose we will use recurrent neural network in our model. The recurrent architecture of our model is shown in figure 3.4. By using recurrent neural network we will give the output generated in previous stage as input to the current stage. At each stage we will use the information available in previous stages as input to current stage and remove separate rain streak layers. This process can be formulated where i stands for the stage number, Ri stands for i-th stage rain streak, Bi stands for i-th stage background image and Ii+1 stands for the output image which we got after the i-th stage.

$$I_1 = I, \tag{3.1}$$

$$R_i = f_c(I_i), \tag{3.2}$$

$$B_i = I_i - R_i, \tag{3.3}$$

$$I_{i+1} = I_i - R_i, \tag{3.4}$$

Here at the end of every stage we will get background image as output by removing rain streak layer.We will use that output as the input to the next stage in our recurrent network. Similarly, we will use this for every stage and these all stages will work together as we are giving the output of one stage as input to its next stage. Thus, these separate stages will make use of the available contextual information in future stages because rain streak layers will usually decrease after every stage . In order to make proper use of available previous information we will incorporate our recurrent neural network with different memory unit architectures like LSTM, MultiplicativeLSTM and GRU.

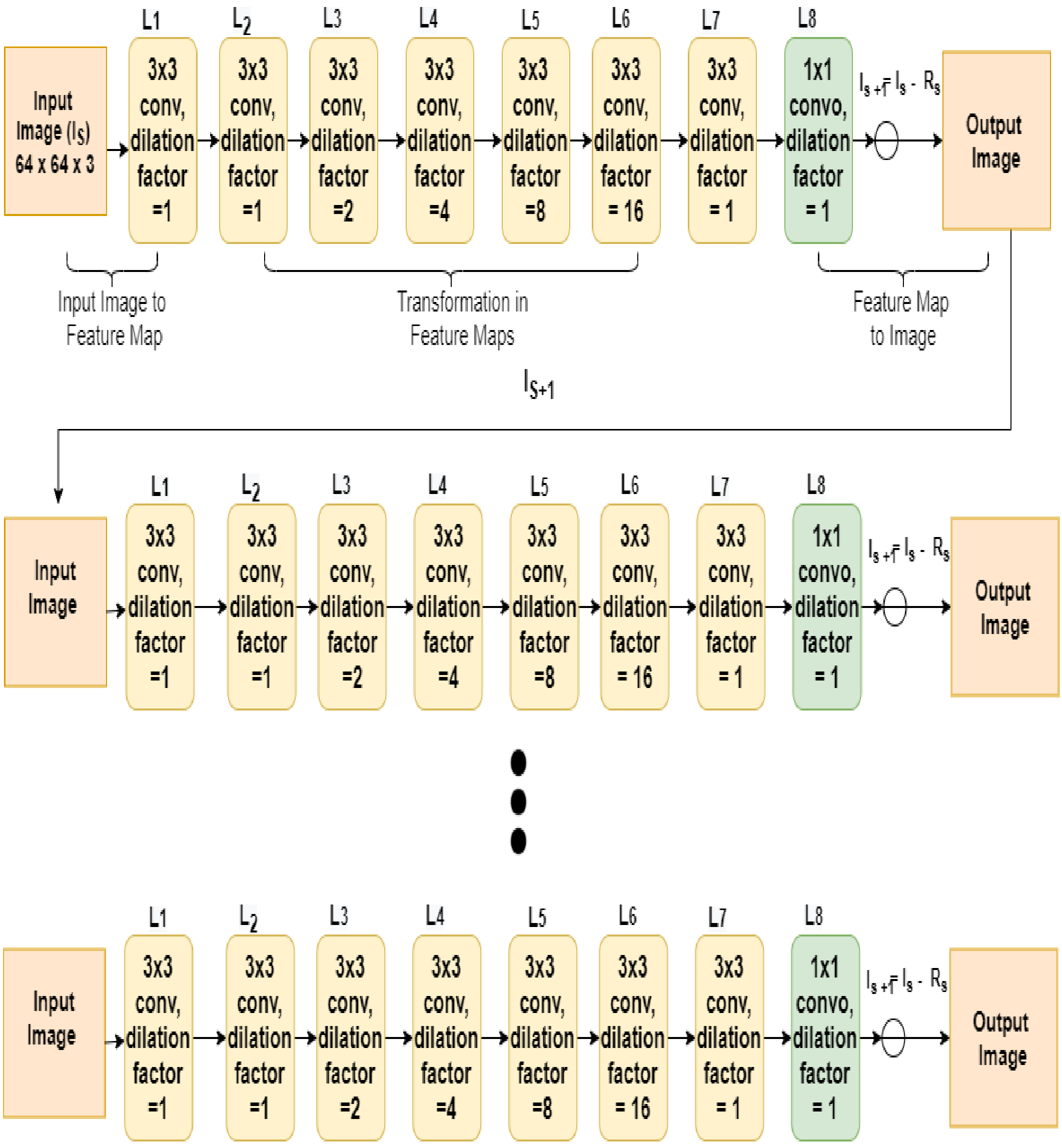


Figure 3.4: Recurrent Architecture

3.2.6 Training, Validation and Testing

This section concentrates on all the crucial factors that are necessary for training, validation and testing in the particular project.As train, valve and test, we have separated our data into three sections.For training purposes and validation we used 80 percent of our dataset with rain picture pairings, while the rest of the data is maintained for testing purposes. This section explains the loss function and assessment measures for which testing is performed.

3.2.7 Loss Function

The loss function used for our model is standard squared error function and described in the equation 3.5.

$$L(R_i,R) = \sum_{i=1}^{n} \left|\left| \sum_{k}^{n} R_i - R \right|\right|^2 \tag{3.5}$$

Here Ri represents all the rain streak layers produced at different stages after rain removal and R is original rain streaks.

3.2.8 Evaluation Metrics

Our Model has been evaluated by using two important evaluation metrics peak signal to noise ratio (PSNR) and structure similarity index (SSIM).
Peak signal to noise ratio (PSNR) is calculated by using formula 3.7.

$$MSE = \left|\left|O_i - O\right|\right|^2 \tag{3.6}$$

$$PSNR = 10 \log_{10}\left(\frac{1}{MSE}\right) \tag{3.7}$$

Here O is original background image and Oi is the output background image at i-th stage, MSE is cummulative squared error between original background image O and output background image at i-th stage Oi. We will take average psnr to model by calculating psnr across different stages.

Structure similarity index (SSIM) is calculated by using pytorch builtin ssim function. To this builtin ssim function we will provide original background image O and our output background image at i-th stage Oi. Similarly like psnr we will take average ssim to model by computing ssim at every stage.

$$SSIM = ssim(O_i - O) \tag{3.8}$$

3.3 Analytical Validation

The section of analytical validation discusses about how the above proposed model overcomes the limitations of existing rain removal models in single images. Main limitations of existing rain removal models are that they fail for images taken in heavy rainfall and makes image backgrounds smooth. In order to overcome that we remove rain in several stages rather than a single stage. Another limitation of existing multi stage rain removal models is that they don't make proper use of available contextual information in previous stages.

Recurrent neural networks have been used in our model to properly utilise available contextualinformation of preceding stages. By using recurrent neural networks we can establish relationship between previous outputs and current stages. Recurrent neural networks usually work very well for series data as they preserve information from step to step unlike CNN. Recurrent neural networks mainly work by the concept of hidden layers which preserve information of previous stages and gives output of new stages by using both preserved information of hidden layers and new input. We use the same concept here as we are removing rain in multiple stages these previous output information can be supplied to hidden layers which which will establish relation among them and make proper use of contextual information. Let si denote the current stage input image, hi denote the hidden state of current stage and hi-1 denote hidden state of previous stage. Then current state hi can be calculated by

$$h_i = \tanh\left(Q \circledast s_i + M \circledast h_{i\text{-}1} + b\right) \tag{3.9}$$

In 3.9 is convolution operation.Q and M are convolution kernels inside rnn cell. From 3.9 we can observe that hi uses the previous data hi-1 in its cell state and uses that for prediction of output background image of ith stage. Thus, RNN remembers the previous stage output images and uses contextual information which will help in in learning image features. These above methods theoretically overcome the limitations stated above. We further evaluate our model in next section by performing experiments and considering results of different rnn architectures LSTM and GRU.

3.4 Conclusion

The chapter of methodology mainly discusses about the dataset, the architecture of the proposed model and analytical validation of it . Firstly, it discusses about different synthetic available rain image datasets. After that, the architecture of the proposed model with convolutional layers and recurrent structure is discussed. Finally, the process of splitting dataset into training, testing, validation and also analytical validation of the proposed model is discussed. In next chapter we have discussed about experiments conducted on proposed model with different recurrent architectures and we will evaluate the results of those models.

Chapter 4

EXPERIMENTS AND RESULTS

4.1 TOOLS

- **Google Colab:**

  The system is built with the assistance of Google Colab.Google Colaboratory is a research effort created by Google to aid in the dissemination of machine learning teaching and research.It is a Jupyter notebook system that does not enable any setup and runs totally on the cloud.It also enables anybody to create deep learning applications using popular libraries such as PyTorch, TensorFlow, Keras, and OpenCV.Python 2.7 and 3.6 are supported.Colab provides GPU and is absolutely free.

- **GSuite:**

  We also utilised Gsuite for uploading and unzipping the datasets, which were about 37gb and 3gb in size, respectively.Many scholars and researchers like this option since it provides infinite storage.

- **Anaconda:**

  Anaconda is a great platform for doing data science.It is used for training and testing on a host for approximately 15 million users globally, allowing private data scientists to quickly access 2,000+ modules using various Python frameworks for data science.

- **NumPy:**

  NumPy is a library for the Python programming language.It accommodates massive multidimensional and matrices, as well as mathematical functions that may be applied to them.

- **Tensorflow:**

  TensorFlow is a free and open-source programming framework that may be used to do a variety of mathematical tasks.This is frequently used as a symbolic mathematics library for conducting AI tasks such as neural networks.Its scalable design enables rapid computer deployment across a wide range of platforms (CPUs, GPUs, TPUs).TensorFlow calculations are displayed as static data flow graphs.TensorFlow derives its name from the operation on multidimensional data frames known as tensors that these neural networks execute.During the June 2016 Google I/O conference, Jeff Dean mentioned 1,500 GitHub repositories where TensorFlow was listed, only five of them by Google.

- **Pandas:**

  For Python, pandas is a data handling and analysis program library. In particular, operations and data structures are defined for the handling of numerical tables and time series. The main features of Pandas are:

  - Data sets reorganization and pivoting.
  - DataFrame object for integrated indexing data manipulation.

- **Keras:**

  Keras is a Python library for open source neural networks. Above TensorFlow, you may utilise the Microsoft Cognitive Toolkit, R, Theano, or PlaidML systems.This is meant to enable for quick deep neural network testing; it is simple to use, extensible, and modular.

- **SciPy:**

  SciPy is a Python library used during computer technology science and technology. SciPy comprises optimization systems, special functions,interpolation, FFT, signal, imaging.

- **Python Image Library(PIL):**

  It's a free programming library of Python to open, modify and store various image file formats.

- **CUDA Toolkit:**

  A high-performance development environment for GPU-accelerated applications is included in the CUDA Toolkit.You may install, automate, and deploy your applications using the CUDA Toolkit on graphics card-accelerated embedded devices, mobile workstations, corporate data centres, cloud-based platforms, and HPC machines.GPU-accelerated libraries, debugging and application optimization tools, a lexer for C/C++, and a runtime library are all included in the toolkit.

4.2 Details of experiment

We have used pytorch backend for implementing deep learning architectures. We also used Scipy, Numpy, TensorboardX, pandas, dill, pillow. We used Google colaboratory (Colab) [3] which is open sourced ML playground which is already preconfigured. We used Google Colaboratory as a platform for developing models, it is an preconfigured ML playground. We trained all our models on Google ML engine, it is used to deploy after training at a bigger scale.For our experiments we used NVIDIA GTX 1060 GPU along with TPU.

The following GPUs made available in google cloud platform.Our model will take images of size 64×64 as input. We trained our model several times using several different combinations of hyper parameters, learning rates, batch sizes and different loss functions for building different architectures. We presented here best performed models in terms of accuracy. For finding out best optimizer, we conducted experiments with both SGD(Stochastic Gradient Descent), Adam algorithm and RMS prop. Among them we got adam algoritm as better option. We will perform our experiments on three different implementations of our model. At each implementation we have used different architectures of recurrent neural network. For evaluation purpose we can say that there are no specific metrics for comparing rained images and clean images because most of the time we have to make sure it with our naked eye as we are using synthesized rain dataset. So, for quality purposes like several other rain models we will also use peak signal to noise ratio (PSNR) and structure similarity index (SSIM) to measure the performance of our model.

4.3 Experiment Design

4.3.1 Experiment 1

We will first implement our model by taking traditional recurrent unit. We will give $64 \times 64 \times 3$ images as input to our model. We have taken mini batch size of 128 and have used adam optimiser. We have used leaky rectified linear unit (ReLU) as activation function in each convolution layer. We have taken a learning rate of 0.003. We performed this experiment by taking these values on our model with traditional recurrent unit and trained our model using the train dataset we have created by combining Rain12, Rain100H, Rain100L, Rain1800 and Rain1200. While training our model we have used traditional recurrent unit and passed the available output at each stage as input to next stages. For evaluating model we have used peak signal to noise ratio and structure similarity index. For our model with recurrent implementation we got PSNR and SSIM as 32.261 and 0.893 respectively.

4.3.2 Experiment 2

In this we have implemented our model by using two different implementations of architectures for recurrent neural network. We first implemented by using gated recurrent unit (GRU) architecture ,similarly we implemented another by using long short term memory (LSTM) and Multiplicative LSTM architecture of recurrent neural networks.In order to evaluate these models we had make sure that all three versions of our model are trained with same dataset.Here also we have taken images of size $64 \times 64 \times 3$ images as input to our model and used mini batch size of 128.we have used learning rate of 0.003 and adam optimiser. For convolutional layers we have used rectified linear unit (ReLU) as activation function. Compared to traditional recurrent architecture these implementations will take more time and more computation as these architectures have more parameters.For evaluating model we have used peak signal to noise ratio and structure similarity index. For our model with gated recurrent unit implementation architecture we got PSNR and SSIM as 33.582 and 0.918 respectively. Similarly, For our model with LSTM recurrent unit implementation architecture we got PSNR and SSIM as 33.375 and 0.923 respectively.Similarly, For model with multiplicative LSTM recurrent unit implementation architecture we got PSNR and SSIM as 33.645 and 0.922 respectively on SRI 1 dataset.

4.3.3 Experiment 3

Here in order to compare our models performance with standard methods like Joint rain detection and removal (JORDER) [24], Layer priors (LP) [14], deep detail network (DDN) [6], Progressive image deraining networks (PreNet) [20] . We have used our dataset on those models and evaluated them with our test dataset so that our models can be easily compared with those standard methods. For JORDER, with our dataset we got PSNR and SSIM as 33.615 and 0.919 respectively. For LP, with our dataset we got PSNR and SSIM as 33.213 and 0.911 respectively. For DDN, with our dataset we got PSNR and SSIM as 32.394 and 0.905 respectively. For PreNet, with our dataset we got PSNR and SSIM as 33.393 and 0.921 respectively.

4.4 Results

Our model takes rained images as input and generates images without rain streak. Below figures show the result of our model where the first image is the rained image which is given as input to our model.The second image is the output generated by our model that is after removing rain streaks from image. The third image is the original clear background image that is before injecting any synthesized rain streaks in that image.

4.5 Performance Metrics

From below figures by comparing the images generated by our model with the original background images. we can say that our model has worked well. In order to evaluate our models with other standard models we have used peak signal to noise ratio (PSNR) and structure similarity index (SSIM) as performance metrices. We have evaluated our model with two different datasets we have prepared that is Synthetic Rain Image 1 (SRI 1) and Synthetic Rain Image 2 (SRI 2) datasets. We have compared our results with other standard methods that are JORDER, LP, DDN, PreNet.

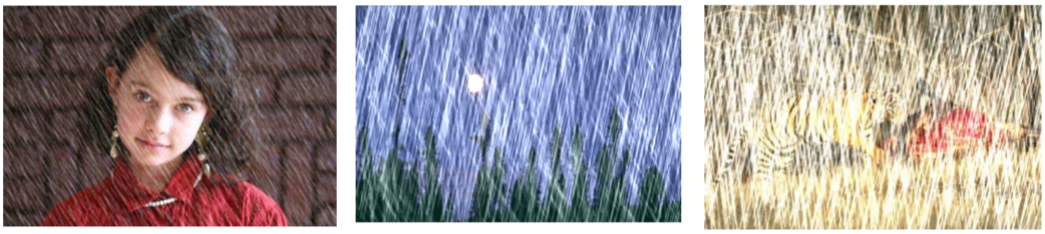


Figure 4.1: Rained Input Image

Figure 4.2: Predicted Image

Figure 4.3: Original Background

Model	SRI I		SRI II	
	PSNR	SSIM	PSNR	SSIM
DDN	32.394	0.905	26.017	0.864
LP	33.213	0.911	27.432	0.871
JORDER	**33.615**	0.919	**28.142**	0.881
PreNet	33.393	**0.921**	27.987	**0.898**
Model with RNN	32.261	0.893	26.014	0.861
Model with GRU	**33.582**	0.918	**28.053**	0.879
Model with LSTM	33.375	**0.923**	27.865	**0.902**
Model with Multiplicative LSTM	**33.645**	**0.922**	**28.217**	**0.898**

Table 4.1: Comparison Table

In above table we have compared our model with other standard models. we can see that our model fares well with other standard models and outperforms in some metrics also. Here we can see that performance of our model is decreased with SRI 2 dataset. That is due to the fact that we have prepared SRI 2 dataset using Rain800 dataset which contains very complex rain streaks with five different directions. Then also our model has worked well . Our Model with LSTM recurrent architecture has even outperformed the standard methods with SSIM values of 0.923 and 0.902 on SRI 1 and SRI 2 dataset respectively while PreNet has SSIM value of 0.921 and 0.898 which is less than our model.

Similarly our model with Multiplicative LSTM has PSNR value of 33.645 and 28.217 on SRI 1 and SRI 2 datasets which is better compared to JORDER using less computation power and less parameters than it. We can say that our model with recurrent architecture of GRU, LSTM and Multiplicative LSTM have performed better than our model with traditional RNN that is due to the fact that Gated Recurrent Units and Long Short Term Memory architectures usually have more parameters and use more computation power when compared to traditional recurrent architecture.

4.6 Conclusion

In the chapter of experiments and results, the various experiments conducted as a part of the thesis work to attain the final result are described. The results obtained in each of the experiments are then discussed. The main objectives of the research work are to develop a deep learning model using which can remove rain streak in images and enhance quality of images taken in heavy rainfall and to develop a model which can remove in multiple stages. The experiments have been designed by considering our objectives and we performed them by considering different variations of our recurrent architecture simple RNN, LSTM, GRU and multiplicative LSTM. Each experiment is performed by using different recurrent architecture.

For comparison of results we have taken same hyper parameters in each experiment and results of our experiments are shown in **??**.We have also performed experiments on standard methods and compared those results with our model.It has been observed that LSTM, GRU and multiplicative LSTM have performed better when compared to simple RNN and it is because LSTM, multiplicative LSTM and GRU more parameters when compared to simple RNN. These LSTM and GRU have also fared well with standard methods and have good performance metrics with less computation time and Multplicative LSTM has even outperformed standard methods in terms of psnr. It can be observed that the experiments are aligned with the initially set objectives. In the next section, the contributions of the research work and also the future work which can be done in this field are discussed in detail.

Chapter 5

DISCUSSIONS AND CONCLUSION

5.1 Contributions

This work attempts to a design a rain removal model for single images using deep learning. Here our model is able to remove rain streaks in several different stages and also used information available in previous stages by using the recurrent neural network. We have seen the image results generated by our model which have performed well when compared to original background images. Further we have trained our model and tested by using two different datasets SRI 1 and SRI 2, which we have created by combining different existing datasets. SRI 1 and SRI 2 consists of several different rain streak images with various types of rains streaks. Our model with Multiplicative LSTM achieves good results when compared with existing standard methods. Our model has achieved PSNR and SSIM value of 33.645 and 0.922 respectively for SRI 1 dataset, similarly it achived PSNR of and SSIM of 28.217 and 0.898 with SRI 2 dataset which is very well when compared with existing standard methods. The novelty of our work is that it generates rain background images by removing rain in several stages and when compared to other models our model not only takes output of previous stages as input to current stage but also establishes the relationship between these input by using recurrent architecture with memory units which improves performance of our model. Comparison tables have also been made to evaluate our model with state of art methods. It also achieves very good results when compared with standard models by using less computation time and less parameters.

5.2 Future Scope

Considerable space for change exists. The first thing is obvious: that we have used synthetic rain dataset for training and testing purposes. Compared to synthetic dataset if we have real world rain dataset it will be very useful and greatly increase the performance because real world rain dataset will have atmospheric vielings and effects which cant be replicated properly in synthesized datasets. Another thing is that deep learning and data driven techniques require high computation power and parameters considering that we want to use rain removal models in preprocessing phases its better to further optimise them. Model driven techniques takes less computation power but they usually fail in heavy rainfall and blurs background more. So, for efficiency and requirement we can further explore by incorporating both data driven and model driven techniques in future researches.

Bibliography

[1] Jérémie Bossu, Nicolas Hautière, and Jean-Philippe Tarel. Rain or snow detection in image sequences through use of a histogram of orientation of streaks. *International journal of computer vision*, 93(3):348–367, 2011.

[2] Nathan Brewer and Nianjun Liu. Using the shape characteristics of rain to identify and remove rain from video. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pages 451–458. Springer, 2008.

[3] Tiago Carneiro, Raul Victor Medeiros Da Nóbrega, Thiago Nepomuceno, Gui-Bin Bian, Victor Hugo C De Albuquerque, and Pedro Pedrosa Reboucas Filho. Performance analysis of google colaboratory as a tool for accelerating deep learning applications. *IEEE Access*, 6:61677–61685, 2018.

[4] Jie Chen, Cheen-Hau Tan, Junhui Hou, Lap-Pui Chau, and He Li. Robust video content alignment and compensation for rain removal in a cnn framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6286–6295, 2018.

[5] David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE international conference on computer vision*, pages 633–640, 2013.

[6] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3855–3863, 2017.

[7] Kshitiz Garg and Shree K Nayar. Detection and removal of rain from videos. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1. IEEE, 2004.

[8] Kaiming He, Jian Sun, and Xiaoou Tang. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1397–1409, 2012.

[9] De-An Huang, Li-Wei Kang, Min-Chun Yang, Chia-Wen Lin, and Yu-Chiang Frank Wang. Context-aware single image rain removal. In *2012 IEEE International Conference on Multimedia and Expo*, pages 164–169. IEEE, 2012.

[10] Goodfellow Ian, Pouget-Abadie Jean, Mirza Mehdi, Xu Bing, Warde-Farley David, Ozair Sherjil, Courville Aaron, and B Yoshua. Generative adversarial nets. *Advances in neural information processing systems*, 3, 2014.

[11] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196*, 2017.

[12] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing*, 24(9):2658–2670, 2015.

[13] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[14] Yu Li, Robby T Tan, Xiaojie Guo, Jiangbo Lu, and Michael S Brown. Rain streak removal using layer priors. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2736–2744, 2016.

[15] Jiaying Liu, Wenhan Yang, Shuai Yang, and Zongming Guo. D3r-net: Dynamic routing residue recurrent network for video rain removal. *IEEE Transactions on Image Processing*, 28(2):699–712, 2018.

[16] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3397–3405, 2015.

[17] David Martin, Charless Fowlkes, Doron Tal, and Jitendra 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, pages 416–423. IEEE, 2001.

[18] Wan-Joo Park and Kwae-Hi Lee. Rain removal using kalman filter in video. In *2008 International Conference on Smart Manufacturing Application*, pages 494–497. IEEE, 2008.

[19] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018.

[20] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: a better and simpler baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3937–3946, 2019.

[21] Yinglong Wang, Shuaicheng Liu, Chen Chen, and Bing Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE Transactions on Image Processing*, 26(8):3936–3950, 2017.

[22] Jing Xu, Wei Zhao, Peng Liu, and Xianglong Tang. Removing rain and snow in a single image using guided filter. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, volume 2, pages 304–307. IEEE, 2012.

[23] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017.

[24] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Shuicheng Yan, and Zongming Guo. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

[25] Xiaofeng Yuan, Lin Li, and Yalin Wang. Nonlinear dynamic soft sensor modeling with supervised long short-term memory network. *IEEE Transactions on Industrial Informatics*, 2019.

[26] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology*, 2019.

[27] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE International Conference on Multimedia and Expo*, pages 461–464. IEEE, 2006.

[28] Xianhui Zheng, Yinghao Liao, Wei Guo, Xueyang Fu, and Xinghao Ding. Single-image-based rain and snow removal using multi-guided filter. In *International Conference on Neural Information Processing*, pages 258–265. Springer, 2013.