

PROFESSIONAL TRAINING REPORT

at

**Sathyabama Institute of Science and Technology (Deemed to Be
University)**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Technology Degree in
Information Technology

By

**LOGANAYAGI.S
(38120046)**



**DEPARTMENT OF INFORMATION TECHNOLOGY SCHOOL OF
COMPUTING**

**SATHYABAMA INSTITUTE OF SCIENCE AND TECHNOLOGY JEPPIAAR NAGAR,
RAJIV GANDHI SALAI,
CHENNAI – 600119. TAMIL NADU**

January 2020



SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Accredited with "A" Grade by NAAC
Jeppiaar Nagar, Rajiv Gandhi Salai, Chennai - 600 119.
Phone: 044 - 2450 3150 / 51 / 52 / 54 / 55 Fax: 044 - 2450 2344
www.sathyabama.ac.in



DEPARTMENT OF INFORMATION TECHNOLOGY

BONAFIDE CERTIFICATE

This is to certify that this Professional Training Report

LOGANAYAGIS (38120046) who underwent the

professional training in "PYTHON" under our

supervision from AUGUST 2020 to DECEMBER 2020.

Internal Guide

Dr.R.SUBHASHINI,M.E.,Ph.D.,

Head of the Department

Dr. R. SUBHASHINI, M.E., Ph.D.,

Submitted for Viva voce Examination held on__

Internal Examiner

External Examiner

DECLARATION

I **LOGANAYAGI.S(38120046)** hereby declare that the Professional Training Report on **PYTHON** done by me under the guidance of **dr.SUBHASHINI M.E,ph.d.**, at Sathyabama Institute of Science and Technology is submitted in partial fulfillment of the requirements for the award of Bachelor of technology degree in Information Technology.

DATE:

PLACE:

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge our sincere thanks to Board of management of **SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey our thanks to **Dr. T. SASIKALA M.E., Ph.D., Dean, School of Computing** and **Dr. R. SUBHASHINI M.E., Ph.D., Head of the Department, Department of Information Technology** for providing us the necessary support and details at the right time during the progressive reviews.

I would like to express our sincere and deep sense of gratitude to our Project **dr.SUBHASHINI M.E,ph.d.,** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express our thanks to all Teaching and Non-teaching staff members of the Department of **INFORMATION TECHNOLOGY** who were helpful in many ways for the completion of the project.

ABSTRACT

In the modern computer era, rain streak removal from a video is an important topic in recent computer vision research. In this paper rain streaks removal is proposed using convolutional neural network. Video captured by outdoor surveillance equipment sometimes contain unexpected rain streaks , which brings difficulty in subsequent video processing task. The structure of the objects and orientation is similar with that of rain streaks and it is hard to remove simultaneously rain and preserve the structure. Hence an end to end deep network architecture is used for removing rain streaks from individual image based on deep convolution neural network proposed in this paper. Convolution neural network applies multiple filter to the raw input image to extract the high level features.cnn transform the original image layer by layer from the original pixel values to the final class score. For the further improvement of de-rained result , images stored in the image domain knowledge is considered by focusing on high frequency region as it remove the back ground interference and focuses the modal on the structure of rain in images . Performances analysis is done using structure similarity index method(SSIM) and peak signal noise ratio(PSNR) qualitative metrics is found the tool utilized to accomplish the objective is open cv2.

TABLE OF CONTENT

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iv
	LIST OF FIGURES	Vii
	LIST OF ABBREVIATION	viii
1	INTRODUCTION	1
2	LITERATURE REVIEW	3
	2.1 Literature Survey	3
3	EXPERIMENTAL WORK	9
	3.1 Work Flow	9
4	EXPERIMENTAL SETUP	10
	4.1 Rain Streaks Removal Method	12
	4.2 Image Processing	12
	4.3 Deep Learning	13
	4.4 RGB Color Models	14
	4.5 Low Pass Filtering	14
	4.6 Rain Pixel Detection	15
	4.7 Average Filtering	16
	4.8 Guided Filtering	16
	4.9 High Frequency Region	17
5	INTRODUCTION TO CNN	18
	5.1 Convolution Neural Network	18
	5.2 CNN Architecture	18
	5.2.1 Convolution Layer	18

CHAPTER NO	TITLE	PAGE NO
	5.2.2 Max Pooling Layer	19
	5.2.3 Fully Connected Layer	20
	5.3 RELU Layer	20
	5.4 Receptive Field	21
	5.5 Image Recognition	21
	5.6 Video Analysis	22
	5.7 Image Quality Metrics	22
	5.7.1PSNR	22
	5.7.2 SSIM	23
6	SOFTWARE CONFIGURATION	24
	6.1 Open cv2	24
	6.2 Python	25
	6.3 Tensor flow	25
	6.4 Keras	26
	6.5 Numpy	26
	6.6 Matplot	26
7	RESULT AND DISCUSSION	27
	7.1 Input Image	27
	7.2 Binary Image	28
	7.3 Rain Pixel Detection	29
	7.4 Mean Filtering	30
	7.5 Base Layer	31
	7.6 Detailed Layer	32
	7.7 De-rained Output	33
	7.8 SSIM	34
8	CONCLUSION	35
	REFERENCES	36

LIST OF FIGURES

1	Difference between rainy image and derained image	
7.1	Input rainy image	
7.2	Binary image	
7.3	Rain pixel detection	
7.4	Base layer	
7.5	High frequency region	
7.6	De- rained output	
7.7	SSIM	

LIST OF ABBREVIATION

DIP	Digital Image Processing
CNN	Convolution Neural Network
ReLU	Rectified Linear filter Unit
PSNR	Peak Signal to Noise Ratio
MSE	Mean Square Error
SSIM	Structural Similarity Index Method
OPEN CV	OPEN source Computer Vision
BN	Batch Normalization

CHAPTER 1

INTRODUCTION

Rain streaks impair visibility of an image and introduce undesirable interference that can severely affect the performance of computer vision algorithms. Rain streak removal can be formulated as a layer decomposition problem, with a rain streak layer superimposed on a background layer containing the true scene content. Effective methods for removing rain streaks are needed for a wide range of practical applications. However when an object's structure and orientation is similar with that of rain streaks, it is hard to simultaneously remove rain and preserve structure. To address this difficult problem, develop an end-to-end deep network architecture for removing rain from individual images. To date, many methods have been proposed for removing rain from images. These methods fall into two categories video-based methods and single-image based methods. The briefly review these approaches and then discuss the contributions of frame work.

Rainy videos captured by outdoor surveillance equipments may degenerate the performance of subsequent video processing tasks, like human detection person reidentification stereo correspondence object tracking and recognition and scene analysis. Thus, removing rain streaks from a video is an important issue and has attracted much attention in computer vision. A natural rainy video is separated into three layers of background scene, rain streaks and moving objects by the CNN. The rain streaks can be decomposed into diverse rain structures corresponding to different scales of rain appearance. All those

decompositions are attained through two scale filters convolved on sparse feature maps.

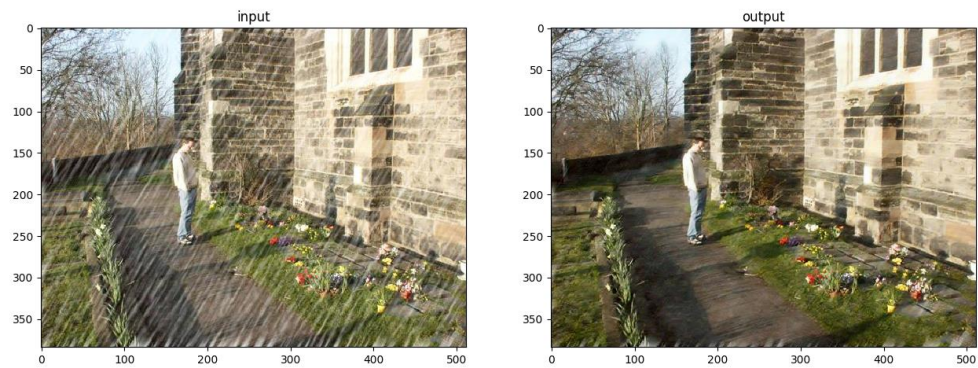


Figure 1.1 Difference between the rain image and derained image.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE SURVEY

2.1.1 “IMAGE DE-RAINING USING A CONDITIONAL GENERATIVE ADVERSARIAL NETWORK” *He Zhang, Vishwanath Sindagi, Vishal M. Patel*(Submitted on 21 Jan 2017 (v1), last revised 4 Feb 2017).

Severe weather conditions such as rain and snow adversely affect the visual quality of images captured under such conditions thus rendering them useless for further usage and sharing. In addition, such degraded images drastically affect performance of vision systems. Hence, it is important to solve the problem of single image de raining or de snowing. However, this is a difficult problem to solve due to its inherent ill-posed nature. Existing approaches attempt to introduce prior information to convert it into a well posed problem. In this paper, we investigate a new point of view in addressing the single image de raining problem. Instead of focusing only on deciding what is a good prior or a good framework to achieve good quantitative and qualitative performance, we also ensure that the de rained image itself does not degrade the performance of a given computer vision algorithm such as detection and classification. In other words, the de-rained result should be indistinguishable from its corresponding clear image to a given discriminator. This criterion can be directly incorporated into the optimization framework by using the recently introduced conditional generative adversarial networks (GANs). To minimize artifacts introduced by GANs and ensure better visual quality, a new refined loss function is introduced. Based on this, we propose a novel single image de-raining method called Image De raining Conditional General Adversarial Network (ID CGAN),

which considers quantitative, visual and also discriminative performance into the objective function. Experiments evaluated on synthetic images and real images show that the proposed method outperforms many recent state of the art single image de raining methods in terms of quantitative and visual performance.

2.1.2 “REMOVING RAIN FROM A SINGLE IMAGE VIA

DISCRIMINATIVE SPARSE CODING”*Yu Luo,YongXu,HuiJi. IEEE*

International Conference on Computer Vision 2015

DOI:10.1109/ICCV.2015.388

Rain streak removal is an important issue of the outdoor vision system and has been investigated extensively. In this paper, we propose a novel tensor optimization model for video rain streak removal by fully considering the discriminatively intrinsic characteristics of the rain streaks and the clean video. In specific, the rain streaks are group sparse and smooth along the rain streaks direction; the clean video is smooth along the perpendicular direction of the rain streaks and the temporal direction. For the rain streaks, we use $\ell_{2,1}$ norm to characterize the group sparsity and the unidirectional total variation (UTV) to promote the smoothness along the rain streaks direction. For the clean video, we use two UTV regularizers to enhance the smoothness along the perpendicular direction of the rain streaks and the temporal direction. To solve the proposed model we develop an efficient alternating direction method of multipliers (ADMM) algorithm. Experiments on synthetic and real data demonstrate the superiority of the proposed method over state-of-the-art methods both quantitatively and qualitatively.

2.1.3 “RAIN STREAK REMOVAL FOR SINGLE IMAGE VIA KERNEL GUIDED CNN”*Ye-Tao Wang, Xi-Le Zhao, Tai-Xiang Jiang, Liang-Jian Deng, Yi Chang, Ting-Zhu Huang*(Submitted on 26 Aug 2018 (v1), last revised 28 Aug 2018 (this version, v2))

Rain streak removal is an important issue and has recently been investigated extensively. Existing methods, especially the newly emerged deep learning methods, could remove the rain streaks well in many cases. However the essential factor in the generative procedure of the rain streaks, the motion blur, which leads to the line pattern appearances, were neglected by the deep learning rain streaks approaches and this resulted in over derain or under derain results. In this paper, we propose a novel rain streak removal approach using a kernel guided convolution neural network (KGCNN), achieving the state-of-the-art performance with simple network architectures. We first model the rain streak interference with its motion blur mechanism. Then, our framework starts with learning the motion blur kernel, which is determined by two factors including angle and length, by a plain neural network, denoted as parameter net, from a patch of the texture component. Then, after a dimensionality stretching operation, the learned motion blur kernel is stretched into a degradation map with the same spatial size as the rainy patch. The stretched degradation map together with the texture patch is subsequently input into a derainconvolution network, which is a typical ResNet architecture and trained to output the rain streaks with the guidance of the learned motion blur kernel. Experiments conducted on extensive synthetic and real data demonstrate the effectiveness of

the proposed method, which preserves the texture and the contrast while removing the rain streaks.

2.1.4 “SINGLE IMAGE DRAINING USING DEEP CONVOLUTIONAL NETWORKS” *Meihua Wang, Jiaming Mai, Ruichu Cai, Liang Yun, Hua Wan*
Published in Multimedia Tools and Applications 2018 DOI:10.1007/s11042-018-5825-8

A deep learning-based single image deraining algorithm is proposed in this work. Instead of modeling a rain layer as a linear function between the rain image and its clear version as previous works do, we directly formulate the clear image as the result of a non-linear mapping of derain image. We construct a coarse deraining convolutional network and a refinement convolutional network to learn this non-linear mapping function. The coarse deraining network is trained to detect the rain streaks with different directions, and restore a raw derained result. The refinement network aims at refining the result according to the raw derained image and the original rain image. By combining the two networks, we are able to well-restore the rain-free image. Experimental results demonstrate that the proposed deraining method can produce high-quality clear images from both synthetic and real-world rain images, outperforming the state-of-the-art methods qualitatively and quantitatively.

2.1.5 “FAST SINGLE IMAGE RAIN REMOVAL VIA A DEEP DECOMPOSITION-COMPOSITION NETWORK” *Siyuan LI, Wenqi Ren, Jiawan Zhang, Jinke Yu, Xiaojie Guo* 2018

Rain effect in images typically is annoying for many multimedia and computer vision tasks. For removing rain effect from a single image, deep learning

techniques have been attracting considerable attentions. This paper designs a novel multi-task leaning architecture in an end-to-end manner to reduce the mapping range from input to output and boost the performance. Concretely, a decomposition net is built to split rain images into clean background and rain layers. Different from previous architectures, our model consists of, besides a component representing the desired clean image, an extra component for the rain layer. During the training phase, we further employ a composition structure to reproduce the input by the separated clean image and rain information for improving the quality of decomposition. Experimental results on both synthetic and real images are conducted to reveal the high-quality recovery by our design, and show its superiority over other state-of-the-art methods. Furthermore, our design is also applicable to other layer decomposition tasks like dust removal. More importantly, our method only requires about 50ms, significantly faster than the competitors, to process a testing image in VGA resolution on a GTX 1080 GPU, making it attractive for practical use.

2.1.6”FAST SINGLE IMAGE RAIN REMOVAL VIA A DEEP DECOMPOSITION-COMPOSITION NETWORK”

Siyuan Li¹, Wenqi Ren², Jiawan Zhang¹, Jinke Yu³ and Xiaojie Guo¹

1 Tianjin University 2 IIE, Chinese Academy of Sciences 3 Dalian University of Technology

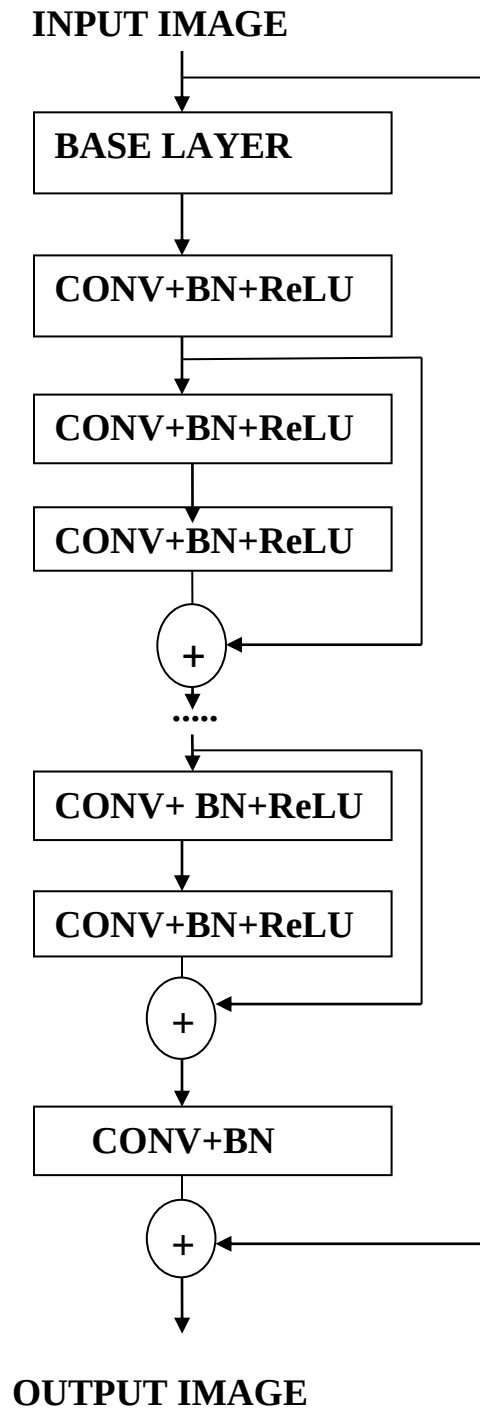
Rain effect in images typically is annoying for many multimedia and computer vision tasks. For removing rain effect from a single image, deep leaning techniques have been attracting considerable attentions. This paper designs a novel multi-task leaning architecture in an end-to-end manner to reduce the mapping range from input to output and boost the performance. Concretely, a decomposition net is built to split rain images into clean background and rain

layers. Different from previous architectures, our model consists of, besides a component representing the desired clean image, an extra component for the rain layer. During the training phase, we further employ a composition structure to reproduce the input by the separated clean image and rain information for improving the quality of decomposition. Experimental results on both synthetic and real images are conducted to reveal the high-quality recovery by our design, and show its superiority over other state-of-the-art methods. Furthermore, our design is also applicable to other layer decomposition tasks like dust removal. More importantly, our method only requires about 50ms, significantly faster than the competitors, to process a testing image in VGA resolution on a GTX 1080 GPU, making it attractive for practical use.

CHAPTER 3

EXPERIMENTAL WORK

3.1 WORK FLOW



CHAPTER 4

EXPERIMENTAL SETUP

4. RAIN STREAKS REMOVAL METHOD

Rain streaks impair visibility of an image and introduce undesirable interference that can severely affect the performance of computer vision algorithms. Rain streak removal can be formulated as a layer decomposition problem, with a rain streak layer superimposed on a background layer containing the true scene content. Effective methods for removing rain streaks are needed for a wide range of practical applications. However when an object's structure and orientation is similar with that of rain streaks, it is hard to simultaneously remove rain and preserve structure. To address this difficult problem, develop an end-to-end deep network architecture for removing rain from individual images. To date, many methods have been proposed for removing rain from images. These methods fall into two categories video-based methods and single-image based methods. The briefly review these approaches and then discuss the contributions of frame work.

Rainy videos captured by outdoor surveillance equipments may degenerate the performance of subsequent video processing tasks, like human detection person reidentification stereo correspondence object tracking and recognition and scene analysis. Thus, removing rain streaks from a video is an important issue and has attracted much attention in computer vision. A natural rainy video is separated into three layers of background scene, rain streaks and moving objects by the CNN. The rain streaks can be decomposed into diverse rain structures corresponding to different scales of rain appearance. All those

decompositions are attained through two scale filters convolved on sparse feature maps.

APPLICATION

- Image registration
- Event detection
- Object detection
- Tracking and Recognition
- Scene analysis and Classification
- Image indexing and Retrieval
- Image copy or near-duplicate detection

IMPORTANCE

Rain streak removal is an important issue and has recently been investigated extensively. Existing methods, especially the newly emerged deep learning methods, could remove the rain streaks well in many cases. The first step is to model the rain streak interference with its motion blur mechanism.

Because there exist so many objects which have similar linear edges with rainstreaks. However, in some cases, there is important application value to remove the rain from only one outdoor image is used to get more information. All the current approaches are based on removing rainstreaks from video.

4.1 IMAGE PROCESSING

Image processing is a method to convert an image into digital form and perform some operation on it, in order to get an enhanced image or to extract some useful information from it.

Image processing is a type of signal dispensation in which input is image, Like video frame or photograph and output may be image or characteristics associated with that image. Usually image processing system includes treating image as two dimensional signals while applying already set signal processing methods to them.

It is among rapidly growing technological today, with its application in various aspects of a business. Image processing forms core research area within engineering and computer science disciplines too.

DISADVANTAGE OF IMAGE PROCESSING

- Initial cost is high depending upon the system used.
- Once the system is damaged that image will be lost.

4.2 DEEP LEARNING

Computers today cannot only automatically classify photos, but they can also describe the various elements in pictures. This is done by the Deep Learning Network (CNN), which actually learns patterns that naturally occur in photos Image net is one of the biggest databases of labeled images to train the Convolution Neural Networks using GPU-accelerated Deep Learning

frameworks such as Caffe2, Chainer, Microsoft Cognitive Toolkit, MXNet, PaddlePaddle, TensorFlow, and inference optimizers such as TensorRT.

Deep Learning models, with their multi-level structures are very helpful in extracting complicated information from input images. Convolution Neural Networks are also able to drastically reduce computation time by taking advantage of GPU for computation which many networks fail to utilize.

ADVANTAGE

- Has best-in-class performance on problems that significantly outperforms other solutions in multiple domains. This includes speech, language, vision, playing games like Go etc. This isn't by a little bit, but by a significant amount.
- Reduces the need for feature engineering, one of the most time-consuming parts of machine learning practice.
- Is an architecture that can be adapted to new problems relatively easily e.g. Vision, time series, language etc., are using techniques like convolution neural networks, recurrent neural networks, long short-term memory etc.

4.3 RGB COLOR MODELS

The RGB color models is an additive color models in which red, green and blue light are added together in various ways to reproduce a broad array of colors. The name of the model comes from the initials of the three additive primary colors, red, green, and blue.

The main purpose of the RGB color model is for the sensing, representation and display of images in electronic systems, such as televisions and computers, though it has also been used in conventional photography. Before the electronic age, the RGB color model already had a solid theory behind it, based in human perception of colors.

RGB is a device dependent color model different devices detect or reproduce a given RGB value differently, since the color elements and their response to the individual R, G, B levels vary from manufacturer to manufacturer, or even in the same device over time. Thus an RGB value does not define the same color across devices without some kind of color management.

4.4 LOW PASS FILTERING

A low pass filter also called a blurring or smoothing filter averages changes in intensity. The simplest low pass filter just calculates the average of a pixel and all of immediate neighbors. So the low-pass filter affects the noise more than it original image.

Smoothing and blurring is one of the most important pre-processing steps in all of computer vision and image processing. By smoothing an image prior to applying techniques such as edge detection or thresholdingable to reduce the amount of high frequency content, such as noise and edges. While this may sound counter-intuitive, by reducing the detail in an image.

4.5 RAIN PIXEL DETECTION

The rain pixels will be assigned with higher values than their neighboring pixels. There often exists an edge-jump between rain streaks and other objects. An image patch that includes rain streaks will usually result in larger average absolute horizontal gradients than a patch without rain streaks. Rain removal

filters are applied on pixel after the chromatic constraints for rain detection. In a normalized image I , for each pixel $I(i, j)$, calculate 5 mean values I_k ($k = 1, 2, 3, 4, 5$) in the windows W_k with pixel $I(i, j)$ located in the center, top-left, top-right, below-left and below-right of the window. Rain removal filters are applied on pixel after the chromatic constraints for rain detection.

$$I(i, j) > \overline{I_k} = \frac{\sum_{\{M, N\} \in W_k} I(m, n)}{|W_k|}$$

4.6 AVERAGE FILTERING

The first blurring method to explore is averaging. An average filter takes an area of pixels surrounding a central pixel, averages all these pixels together, and replaces the central pixel with the average.

By taking the average of the region surrounding a pixel, smoothing it and replacing it with the value of its local neighborhood. This allows to reduce noise and the level of detail, simply by relying on the average. To accomplish average blur by convolving original image with a $M \times N$ normalized filter where M and N are both odd integers.

This kernel is going to slide from left-to-right and from top-to-bottom for each and every pixel in input image. The pixel at the center of the kernel an odd number is then set to be the average of all other pixels surrounding it. Let's go ahead and define a 9×9 average kernel that can be used to blur the central pixel with a 9 pixel radius.

An alternative is to give pixels different weights, where pixels farther from the central pixel contribute less to the average. It is also define an average kernel

4.7 GUIDED FILTERING

The guided filter computes the filtering output by considering the content of a guidance image which can be the input image itself or another different image. The guided filter can be used as an edge preserving smoothing operator like the popular bilateral filter, but has better behavior near edges. The guided filter is also a more generic concept of smoothing, it can transfer the structure of the guidance image to the filtering output. The guided filter naturally has fast and non –approximate linear time algorithm regardless of the kernel size and the intensity rage. Currently it is the one of the fastest edge preserving filter.

4.8 HIGH PASS FREQUENCY REGION

A high pass filter can be used to make an image appear sharper. These filters emphasize fine details in the image exactly the opposite of the low pass filter. High pass filtering works in exactly the same way as low pass filtering. It uses a different convolution kernel. There is no change in intensity. But if one pixel is brighter than its immediate neighbors, it gets boosted.

The low pass filtering smooth out noise while high pass filtering does just the opposite. It amplifies noise, if the original image is not too noisy otherwise the noise will overwhelm the image. Maximum image includes a very useful range restricted filter option. High pass filter only the brightest parts of the image, where the signal-to-noise ratio is highest.

High pass filtering can also cause small faint details to be greatly exaggerated. An over processed image will look grainy and unnatural and point sources

will have dark donuts around them. So while high pass filtering can often improve an image by sharpening detail. It can actually degrade the image quality significantly.

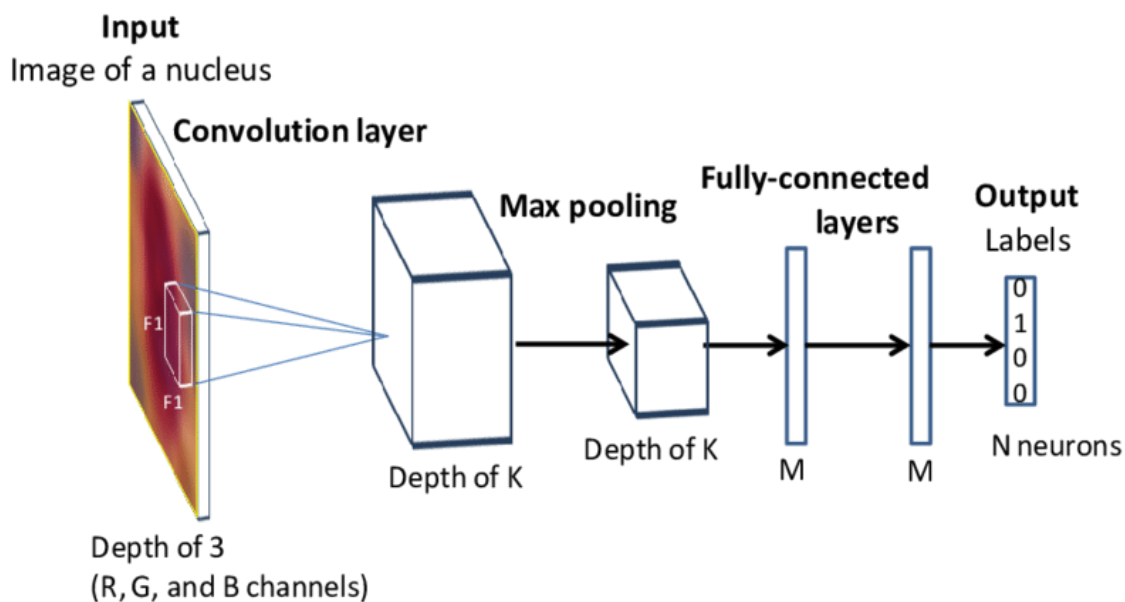
CHAPTER 5

INTRODUCTION TO CNN

5.1 CONVOLUTIONAL NEURAL NETWORK (CNN)

Convolution Neural Networks (CNN) is the current state of the art models architecture for visual recognition. CNN consists of three layers, namely convolution, max-pooling and fully connected layers. The CNN architecture is formed by stacking these layers sequentially. CNN applies multiple filters to the raw input image to extract the high level features. CNN transform the original image layer by layer from the original pixel values to the final class scores.

5.2 CNN ARCHITECTURE



5.2.1 CONVOLUTION LAYER

The convolution layer's parameters consist of a set of learnable filters. The size of every filter is smaller than the input width and height of volume. Slide each

filter across the width and height of the input volume and compute the dot products between entries of the filter and the input at any position. It will produce a 2-dimensional activation map that gives the responses of that filter at every spatial position. The network will learn the filters that activate when they see some type of visual feature.

In the first convolution layer it detects the edges from raw pixel data. In the second convolution layers it detects shapes or blobs from the previous layer output. In the subsequent convolution layer it detects the high level features such as facial structures from the last layer. The last layer in CNN is a classifier that uses these higher-level features to make predictions.

Convolution Layer accepts a volume size of $W1*H1*D1$ and requires four hyper parameters such as number of filters K , filter size F , Stride S and amount of zero padding P . The output volume size of the single convolution layer is $W2*H2*D2$.

Where,

$$W2 = ((W1 - F + 2P) / S) + 1$$

$$H2 = ((H1 - F + 2P) / S) + 1$$

$$P = (F - 1) / 2$$

$$D2 = K$$

5.2.2 MAX-POOLING LAYER

The Max-pooling layer is inserted in between the Convolution layers in CNN architecture. The Max-pooling layer is used to reduce the spatial size of Convolution image. The Max-pooling layer operates independently on every

depth slice of the input and it resizes the image using MAX operation. The pooling layer accepts input volume size of $W1*H1*D1$ and it requires two hyper parameters such as Filter Size F and Stride S . It produces the output volume size of $W2*H2*D2$. where,

$$W2 = (W1 - F) / S + 1$$

$$H2 = (H1 - F) / S + 1$$

$$D2 = D1$$

5.2.3 FULLY CONNECTED LAYER

Fully connected layers are used at the end of the network after feature extraction and consolidation has been performed by the convolution and pooling layers. They are used to create final nonlinear combinations of features and for making predictions by the network.

5.3 RELULAYER

RELU is the abbreviation of Rectified Linear Filter Unit, which applies the non saturating activation function. It effectively removes negative values from an activation map by setting them to zero. It increases the non-linear properties of the decision filter and of the overall network without affecting the receptive fields of the convolution layer. Other functions are also used to increase nonlinearity, for example the saturating hyperbolic tangent and the sigmoid function. RELU is often preferred to other functions because it trains the neural network several times faster without a significant penalty to generalization accuracy.

5.4 RECEPTIVE FIELD

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from every element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically the subarea is of a square shape. The input area of a neuron is called its receptive field. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolution layer, the receptive area is smaller than the entire previous layer.

5.5 IMAGE RECOGNITION

CNNs are often used in image recognition systems. When applied to facial recognition, CNNs achieved a large decrease in error rate. Another is a 97.6 percent recognition rate on 5,600 still images of more than 10 subjects. CNN used to assess video quality in an objective way after manual training, The resulting system had a very low root mean square error. The image net large scale recognition challenge is a benchmark in object classification and detection, with millions of images and hundreds of object classes. That performance of Convolution Neural Networks on the ImageNet tests was close to that of humans. They also have trouble with images that have been distorted with filters, an increasing common phenomenon with modern digital cameras. For example, they are not good at classifying objects into fine grained categories such as the particular breed of dog or species of bird, whereas Convolution Neural Networks handle this.

5.6 VIDEO ANALYSIS

Video is more complex than images since it has another dimension. However, some extensions of CNN into the video domain have been explored. One approach is to treat space and time as equivalent dimensions of the input and perform convolutions in both time and space. Another way is to fuse the features of two Convolution Neural Networks, one for the spatial and one for the temporal stream. Long Short Term Memory (LSTM) recurrent units are typically incorporated after the CNN to account for inter frame or inter clip dependencies. unsupervised learning schemes for training Spatio-temporal features have been introduced, based on Convolutional Gated Restricted Boltzmann machines and Independent Subspace Analysis.

BATCH NORMALIZATION

Accelerating Deep Network Training by Reducing Internal Covariate Shift , batch normalization layers (or BN for short), as the name suggests, are used to normalize the activations of a given input volume before passing it into the next layer in the network.

5.7 IMAGE QUALITY METRICS

5.7.1 PEAK SIGNAL TO NOISE RATIO (PSNR)

Peak Signal-To-Noise Ratio, often abbreviated PSNR, is an engineering term for the ratio between the maximum possible power of a signal and the power of corrupting noise that affects the fidelity of its representation. Because many signals have a very wide dynamic range, PSNR is usually expressed in terms of the logarithmic decibel scale. PSNR is most easily defined via the Mean Squared Error (MSE).

$$\text{PSNR} = 10 \log \left(\frac{\text{MAX}^2}{\text{MSE}} \right)$$

$$= 28.72$$

Mean Square Error

$$\text{MSE} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} |I(i, j) - K(i, j)|^2$$

$$= 237.76$$

5.7.2 STRUCTURAL SIMILARITY INDEX METHOD

SSIM is used for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference

The structural similarity (SSIM) index is a method for predicting the perceived quality of digital television and cinematic pictures, as well as other kinds of digital images and videos.

SSIM is used for measuring the similarity between two images. The SSIM index is a full reference metric in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference. SSIM is designed to improve on traditional methods such as PSNR and MSE.

$$\text{SSIM} = \frac{(2\mu_x\mu_y+c1)(2\sigma_{xy}+c2)}{(\mu_x^2+\mu_y^2+c1)(\sigma_x^2+\sigma_y^2+c2)}$$

$$= 0.88$$

CHAPTER 6

SOFTWARE CONFIGURATION

6.1 OPEN CV2

Open CV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. Open CV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, Open CV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc. Open CV has more than 47 thousand people of user community and estimated number of downloads exceeding 14 million.

The library is used extensively in companies, research groups and by Governmental bodies. Along with well-established companies like Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, Toyota that employ the library, there are many startups such as Applied Minds, Video Surf and Zeitera, that make extensive use of Open CV. Open CV's deployed uses span the range from

stitching street view images together, detecting intrusions in surveillance video in Israel, monitoring mine equipment in China, helping robots navigate and pick up objects at Willow Garage, detection of swimming pool drowning accidents in Europe, running interactive art in Spain and New York, checking runways for debris in Turkey, inspecting labels on products in factories around the world on to rapid face detection in Japan.

It has C++, Python, Java and MATLAB interfaces and supports Windows, Linux, Android and Mac OS. Open CV leans mostly towards real-time vision applications and takes advantage of MMX and SSE instructions when available.

A full-featured CUDA and Open CL interfaces are being actively developed right now. There are over 500 algorithms and about 10 times as many functions that compose or support those algorithms. Open CV is written natively in C++ and has a Template interface that works seamlessly with STL containers.

6.2 PYTHON OVERVIEW

Python is a widely used high level general purpose programming language. It was developed by Guido van Rossum in 1985- 1990. Being an interpreted, interactive, object oriented and high level programming language, Python source code available under the GNU General Public License (GPL)

6.3 TENSORFLOW

TensorFlow is a Python library for fast numerical computing. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow.

The numerical libraries intended for use in Deep Learning like Theano, TensorFlow was designed for use both in research and development and in production system. It can run on single CPU systems, GPUs as well as mobile devices and large scale distributed systems of hundreds of machines.

6.4 KERAS

Keras is an open source neural network library written in Python. It is capable of running on top of TensorFlow, Microsoft Cognition Toolkit, Theano or PlaidML. Designed to enable fast experimentation with deep neural networks, it focuses on being user friendly, modular and extensible. It was developed as part of the research effort of project ONEIRO (Open-ended Neuro-Electronic Intelligent Robot Operating System) and its primary author and maintainer is Francois Chollet, a Google engineer. Chollet also is the author of the Xception deep neural network model.

6.5 NUMPY

Numpy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high level mathematical functions to operate on these arrays. The ancestor of Numpy, Numeric, was originally created by Jim Hugunin with contributions from several other developers. In 2005, Travis Oliphant created Numpy by incorporating features of the competing Numarray into Numeric, with extensive modifications. Numpy is open source software and has many contributors.

6.6 MATPLOTTING

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely - histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab.

CHAPATER 7

RESULT AND DISCUSSION

7.1 INPUT (RAINY IMAGE)



Figure 7.1: Input rainy Image

Rainy videos captured by outdoor surveillance equipments may degenerate the performance of subsequent video processing tasks, like human detection person reidentification stereo correspondence object tracking and recognition.

7.2 BINARY IMAGE



Figure 7.2: Binary image

A binary image is a digital image that has only two possible values for each pixel. Typically, the two colors used for a binary image are black and white. The color used for the object in the image is the foreground color while the rest of the image is the background color.

7.3RAIN PIXEL DETECTION

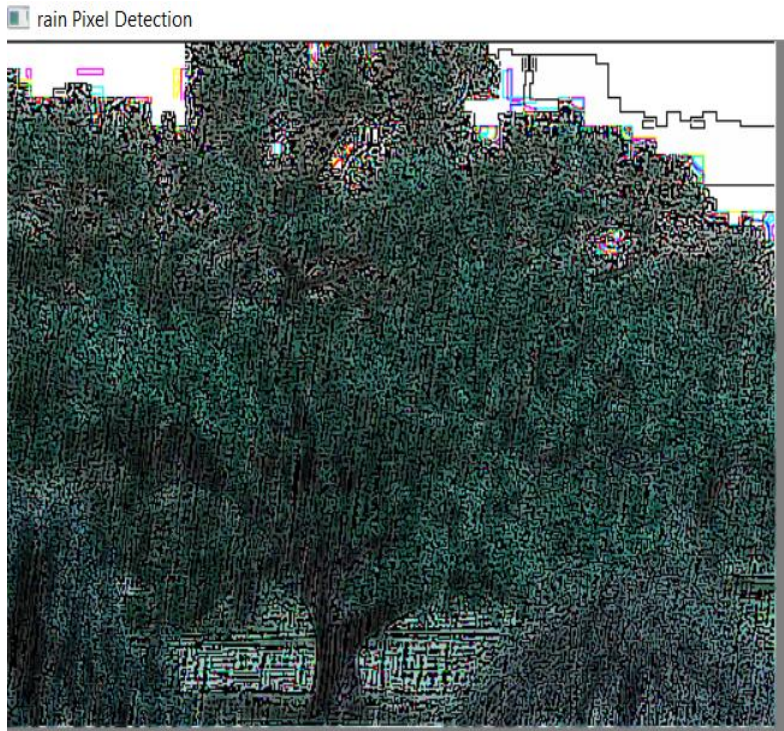


Figure 7.3 Detection of rain pixels using average filter.

The rain pixels will be assigned with higher values than their neighboring pixels. There often exists an edge-jump between rain streaks and other objects. An image patch that includes rain streaks will usually result in larger average absolute horizontal gradients than a patch without rain streaks

7.4 BASE LAYER

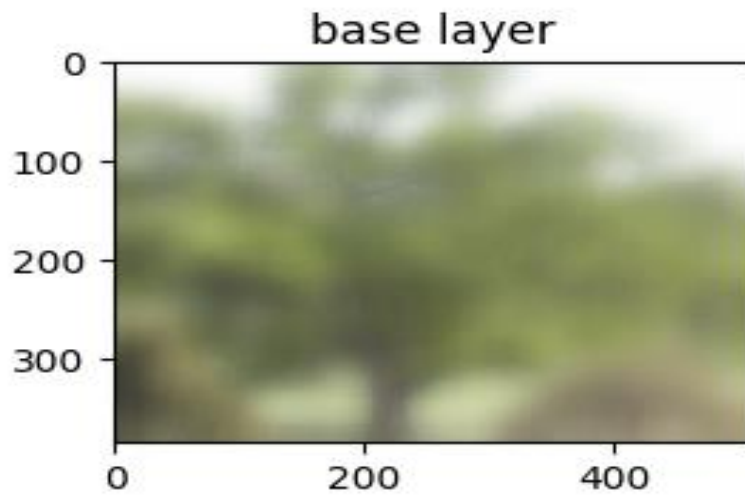


Figure 7.4: Result of low pass filter obtained by guided filter

The guided filter computes the filtering output by considering the content of a guidance image which can be the input image itself or another different image. The guided filter can be used as an edge preserving smoothing operator like the popular bilateral filter, but has better behavior near edges.

7.5 HIGH FREQUENCY REGION

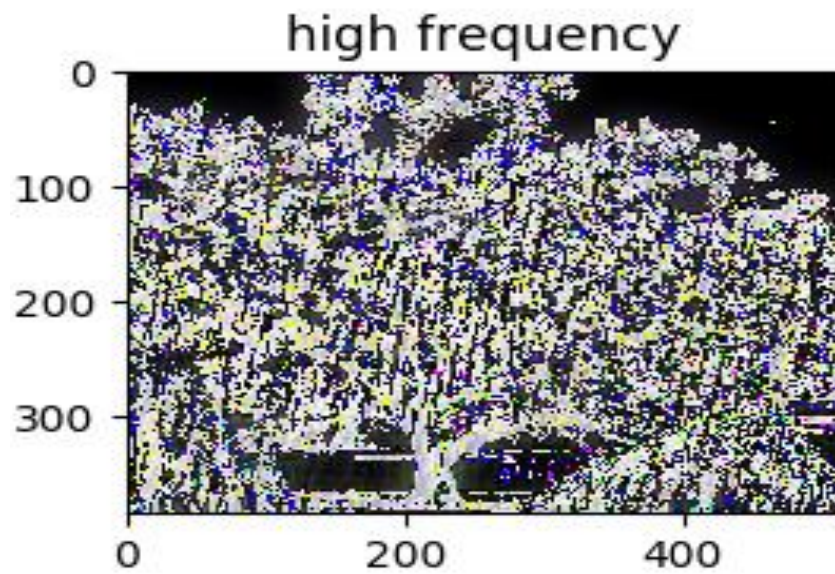


Figure 7.5 :Detailed layer

The difference between the input image and the low pass filter output obtained by guided filter.

7.6 DERAINED OUTPUT

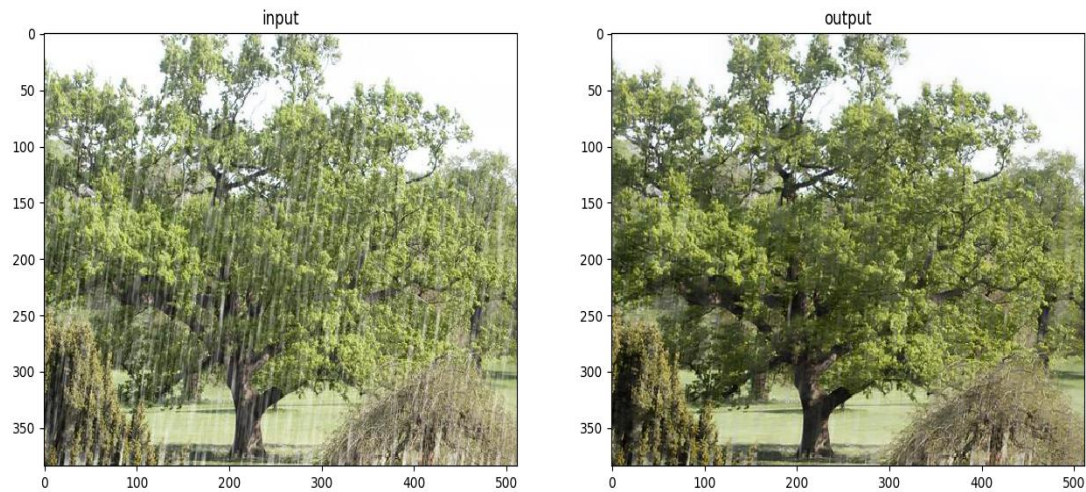


Figure 7.6: Result obtained from deep residual network using CNN

The CNN architecture is formed by stacking these layers sequentially. CNN applies multiple filters to the raw input image to extract the high level features. CNN transform the original image layer by layer from the original pixel values to the final class scores

7.7 SSIM

MSE: 237.76, SSIM: 0.88



Figure 7.7: comparison of original image and contrast image

SSIM is used for measuring the similarity between two images. The SSIM index is a full reference metric; in other words, the measurement or prediction of image quality is based on an initial uncompressed or distortion-free image as reference.

CHAPTER 8

CONCLUSION

This paper explained the ways and means of removing rain streaks in the image using deep residual network. RGB color models are used in various ways to reproduce a broad array of colors in the image. The rain pixel detection is obtained by using the mean filtering technique and the edge can be preserved by edge detection technique called guided filtering. The base layer is obtained by decomposition of the original image. The high frequency detail layer content of an image and regressing on the negative residual information has benefits for de-raining performance. Since it makes the training process easier by reducing the mapping range. An end to end deep learning architecture is used for removing rain streaks from the video or image based on the convolution neural network. In the first convolution layer it detects the edge from pixel data. In second convolution layer it detects shape from the previous layer output. The subsequent convolution layer detects the high level features such as facial structure from the last layer. The last layer in CNN is a classifier that uses higher level features to make predictions. Thus the rain streaks can be completely removed by using deep residual network

REFERENCES

- [1] Y. Chang, L.X. Yan, and S. Zhong. Transformed low rank model for line pattern noise removal. In the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 1726–1734, 2017.
- [2] B.H. Chen, S.C. Huang, and S.Y. Kuo. Error-optimized sparse representation for single image rain removal. *IEEE Transactions on Industrial Electronics*, 64(8):65736581, 2017.
- [3] J. Chen, C.H. Tan, J.H. Hou, L.P. Chau and H. Li. Robust video content alignment and compensation for rain removal in a CNN framework. *arXiv preprint arXiv:1803.10433*, 2018.
- [4] L.J. Deng, T.Z. Huang, X.L. Zhao, and T.X. Jiang. A directional global sparse model for single image rain removal. *Applied Mathematical Modelling*, 59:662679, 2018.
- [5] S.L. Du, Y.G. Liu, M. Ye, Z.Y. Xu, J. Li, and J.-G. Liu. Single image deraining via decorrelating the rain streaks and background scene in gradient domain. *Pattern Recognition*, 79:303–317, 2018.
- [6] X.Y. Fu, J.B. Huang, D.L. Zeng, Y. Huang, X.H. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages

1715–1723, 2017.

[7] X.Y. Fu, J.B. Huang, X.H. Ding, Y.H. Liao, and J. Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017.

[8] S.H. Gu, D.Y. Meng, W.M. Zuo, and L. Zhang. Joint convolutional analysis and synthesis sparse representation for single image layer separation. In the *IEEE International Conference on Computer Vision (ICCV)*, pages 1717–1725. IEEE, 2017.

[9] T.X. Jiang, T.Z. Huang, X.L. Zhao, L.J. Deng, and Y. Wang. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4057–4066, July 2017.

[10] T.X. Jiang, T.Z. Huang, X.L. Zhao, L.J. Deng, and Y. Wang. Fastderain: A novel video rain streak removal method using directional gradient priors. *arXiv preprint arXiv:1803.07487*, 2018.

[11] Y. Li, R.T Tan, X.J. Guo, J.B. Lu, and M.-S Brown. Single image rain streak decomposition using layer priors. *IEEE Transactions on Image Processing*, 26(8):3874–3885, 2017.

[12] M.H. Li, Q. Xie, Q. Zhao, W. Wei, S.H. Gu, J. Tao, and D.Y. Meng. Video rain streak removal by multiscale convolutional sparse coding. In the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

- [13] J.Y. Liu, W.H. Yang, S. Yang, and Z.M. Guo. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
- [14] S.Y. Li, W.Q. Ren, J.W. Zhang, J. Yu, and X.J. Guo. Fast single image rain removal via a deep decomposition-composition network. arXiv preprint arXiv:1804.02688, 2018.
- [15] R. Qian, Robby T. Tan, W.H. Yang, J.J. Su, and J.Y. 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.
- [16] W.-H. Ren, J.D. Tian, Z. Han, A. Chan, and Y.D. Tang. Video desnowing and deraining based on matrix decomposition. In the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 4210–4219, 2017.
- [17] D.W. Ren, W.M. Zuo, D. Zhang, L. Zhang, and M.H. Yang. Simultaneous fidelity and regularization learning for image restoration. arXiv preprint arXiv:1804.04522, 2018.
- [18] W. Wei, D.Y. Meng, Q. Zhao, and Z.B. Xu. Semi-supervised CNN for single image rain removal. arXiv preprint arXiv:1807.11078, 2018.
- [19] S.J. Song, C.L. Lan, J.L. Xing, W.K. Zeng, and J.Y. Liu. Spatiotemporal attention-based lstm networks for 3d action recognition and

detection. *IEEE Transactions on Image Processing*, 27(7):3459–3471, 2018.

[20] Y.L. Wang, S.C. Liu, C. Chen, and B. Zeng. A hierarchical approach for rain or snow removing in a single color image. *IEEE Transactions on Image Processing*, 26(8):3936–3950, 2017.

[21] W. Wei, L.X. Yi, Q. Xie, Q. Zhao, D.Y. Meng, and Z.B. Xu. Should we encode rain streaks in video as deterministic or stochastic? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2516–2525, 2017.

[22] W.H. Yang, R.T. Tan, J.S. Feng, J.Y. Liu, Z.M. Guo, and S.C. Yan. Deep joint rain detection and removal from a single image. In the *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[23] H. Zhang and V.M Patel. Density-aware single image deraining using a multi stream dense network. *arXiv preprint arXiv:1802.07412*, 2018.

[24] H. Zhang and V.M Patel. Convolutional sparse and low-rank coding based rain streak removal. In the *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1259–1267. IEEE, 2017.

[25] K. Zhang, W.M Zuo, and L. Zhang. Learning a single convolutional Super resolution network for multiple degradations. *arXiv preprint arXiv:1712.06116*, 2017.

[26] L. Zhu, C.W. Fu, D. Lischinski, and P.A. Heng. Joint bilayer optimization for single-image rain streak removal. In the *IEEE*

International Conference on Computer Vision (ICCV), Oct 2017.

[27] X. Zhang, C. Zhu, S. Wang, Y.P. Liu, and M. Ye. A bayesian approach to camouflaged moving object detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(9):2001–2013, 2017

