



# Deep learning-based image de-raining using discrete Fourier transformation

Analyzing behaviour of deep CNNs towards uncorrelated transformed domain data

Prasen Kumar Sharma<sup>1</sup> · Sathisha Basavaraju<sup>1</sup> · Arijit Sur<sup>1</sup>

Accepted: 31 August 2020 / Published online: 16 September 2020  
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

## Abstract

Single image rain streak removal is a well-explored topic in the field of computer vision. The de-raining problem is modeled as an image decomposition task where a rainy image is decomposed into rain-free background image and rain streak map. Unlike most of the existing de-raining methods, this paper attempts to decompose the rainy image in the frequency domain. The idea is inspired by pseudo-periodic characteristics of the noise signal (here the rain streaks) which leave some traces in the frequency domain, and the same can be utilized to predict the noise signal. In this paper, a deep learning-based rain streak prediction model is proposed which learns in discrete Fourier transform Oppenheim and Schafer (Discrete-Time Signal Processing, Prentice Hall, Upper Saddle River, 1989) domain. To the best of our knowledge, this is the first approach where compressed domain coefficients are directly used as input to a deep convolutional neural network. The proposed model has been tested on publicly available synthetic datasets Fu et al. (in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. <https://doi.org/10.1109/CVPR.2017.186>, Yang et al. (in: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017. <https://doi.org/10.1109/CVPR.2017.183>), Yeh et al. (in: 2015 IEEE International Conference on Consumer Electronics-Taiwan, 2015. <https://doi.org/10.1109/ICCE-TW.2015.7216999>) and results are found to be comparable with the state of the art methods in the spatial domain. The presented analysis and study have an obvious indication to extend transform domain input to train the deep learning architecture especially image de-noising like problems.

**Keywords** Image de-raining · Deep learning · Convolutional neural networks · Discrete Fourier transformation

## 1 Introduction

Rain streak removal from a single image or video has been given considerable importance in recent computer vision literature. In the real world scenario, rain streaks may cause problems in many image-based applications such as surveillance, satellite-based image tracking, autonomous car driving, etc. It can also be taken as a noise removal problem from a single image or video. Rain streak removal from a sin-

gle image is an image decomposition problem where a rainy image is decomposed to rain-free background image and rain streak map. There have been many methods for noise or rain streak removal where the majority of them work in the spatial domain. It has been observed that there are some traces of the noise in the frequency domain due to its pseudo-periodic nature. These traces may be exploited to predict the noise in the transformed domain. In this paper, the rain streaks are modeled as a pseudo-periodic noise and, its frequency response is used to train a deep network to predict the rain streak map. Interestingly, it shows comparable results with state-of-the-art rain streak removal approaches in the spatial domain.

It is observed from the recent literature that spatial domain approaches are reported to remove rain streak from a single image. Fu et al. [8,9] devised a network based on negative residual learning using Deep Residual Network (ResNet)

✉ Prasen Kumar Sharma  
kumar176101005@iitg.ac.in  
Sathisha Basavaraju  
b.sathisha@iitg.ac.in  
Arijit Sur  
arijit@iitg.ac.in

<sup>1</sup> Multimedia Laboratory, Department of Computer Science and Engineering, Indian Institute of Technology, Guwahati, India

[15] which can be added to rainy images to get their de-rained versions. Shen et al. [31] used the concepts of Haar [13] wavelets and Dark channel [14] prior for image de-raining. In [21,22,40], authors have proposed the networks that separate rain map from a rainy image to obtain rain-free background by considering the apriori knowledge such as (a) centralized sparse representation, (b) based on predicted rain direction, (c) rain streak layer and (d) Gaussian mixture models. Zhang et al. [38] learns the sparsity and low-rank representation-based convolutional kernels to determine the clear image and rain streaks. The work in [4,17,18,26,37] uncouple a rainy image into its high and low-frequency components with bilateral filtering and further split the high-frequency component into rain textures and non-rain geometric details. It is achieved using structured dictionary learning, the histogram of oriented gradients (HOG), eigencolors and depth of field. Chang et al. [3] extracts the periodic noise which follows line patterns such as rain streaks, stripes, fences etc. Chen et al. [5] introduced a model based on the Conditional Generative Adversarial Network [11] to estimate the high quality rain-free image. Gu et al. [12] decompose the rainy image into two layers, one is approximated by analysis sparse representation to portray large-scale structures and the other by synthesis sparse representation to exhibit the finer textures in the image to remove the rain streaks. Yang et al. [35] proposed a model which simultaneously detects and remove the rain streaks by using a binary map. The rain streak exists in the pixel if the corresponding value in the binary map is '1' and '0' otherwise. They simulate the heavy rain by modeling the appearance of rain streak accumulation, various shapes, and directions. Ren et al. [28] model the rain streaks in a matrix decomposition framework by dividing the rain into sparse and dense. Yeh et al. [36] divide a rainy image into the high and low-frequency components using the Gaussian filter. The rain streaks in low-frequency components are removed by using non-negative matrix factorization whereas canny edge detection is used to remove rain streaks in the high-frequency component. Wang et al. [33] proposed a model based on the observation that most of the rain streaks exist in the high-frequency component of the rainy image. The rain-free details from the high-frequency component are obtained using dictionary-based learning method. Shen et al. [31] made an argument that the spectrum of an image in Fourier domain loses a lot of great properties such as local receptive field, which makes it difficult to use a convolutional neural network. However, in this paper, we try to devise a suitable model for Fourier domain signals (Fig. 1).

## 1.1 Motivation and contribution of the proposed scheme

It is observed in the above discussion that the rain streak removal problem mostly depends on the spatial domain fea-

tures of an image. The present-day deep framework-based schemes have used the image in the spatial domain as an input for a better correlation between background image objects. In general, transform domain image is not useful to train the deep network because the spatial correlation has been lost due to signal transformation. Interestingly, it has been observed that the noise signal can be predicted from the transform domain especially when it has some periodic or pseudo-periodic structure. In other words, the pseudo-periodic noise leaves some traces in the transformed domain which can be exploited to predict the actual noise. This observation motivates us to propose a deep convolution neural network which uses transformed domain coefficients as an input rather than image pixels. To best of our knowledge, this is the first attempt to train a deep network using purely transformed domain features for image de-raining. This paper makes the following contributions:

1. This work is the first attempt to predict rain streaks in an image by a deep CNN using purely transformed coefficients as an input.
2. It has been analytically shown that the rain streak information is retained in the frequency domain.
3. We devise a novel deep learning framework for single image de-raining which can use transform domain coefficients as an input.

The rest of the paper is organized as follows: Sect. 2 describes preservation of rain streak information in the transformed domain. Section 3 describes the proposed network and its objective function. Section 4 shows the experiments conducted and results. Section 5 describes the effects of different modules of CNN when a transformed domain signal is given as an input. The paper is concluded in the Sect. 6.

## 2 Proposed method

We briefly discuss the theory of image transformation into the Fourier domain. Let  $I$  be an image,  $K$  be a kernel and  $f_{\text{map}}$  be the cross-correlation map [10, p. 329] constructed by using convolution operation as

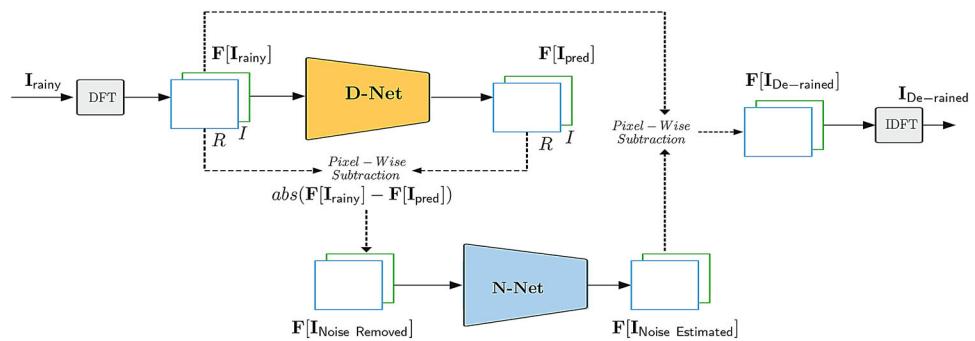
$$f_{\text{map}} = I * K \quad (1)$$

where  $*$  is a convolution operator. Eq. 1 based on the Convolution Theorem [19], can be expressed in the Fourier domain as [27]

$$\mathbf{F}[f_{\text{map}}] = \mathbf{F}[I * K] = \mathbf{F}[I] \odot \mathbf{F}[K] \quad (2)$$

where  $\mathbf{F}$  is the discrete Fourier transformation and  $\odot$  is Hadamard point-wise multiplication operator [6]. Let  $\mathbf{I}$  be

**Fig. 1** The CNN framework of the proposed single-image rain streak removal. The details of D-Net and N-Net are given in Fig. 6



an image of dimension  $M \times N$ . The discrete Fourier transformation [25] of an image  $\mathbf{I}$  can be calculated as

$$\mathbf{F}[u, v] = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \mathbf{I}[m, n] \exp \left\{ -j2\pi \frac{Nm u + M n v}{MN} \right\} \quad (3)$$

where  $j = \sqrt{-1}$ ,  $[u, v]$  are horizontal and vertical frequency locations in frequency domain,  $\exp(j\mathbf{x}) = \cos \mathbf{x} + j \sin \mathbf{x}$  based on Euler's identity [7], for each pixel location  $(x, y)$  in spatial domain. The result of  $\mathbf{F}$  on an image has complex numbers of the form  $a_{u,v} + j.b_{u,v}$  where  $a_{u,v}, b_{u,v}$  are real and imaginary coefficients respectively. For processing, we can either use the pair of real and imaginary parts or the pair of magnitude and phase parts, of  $\mathbf{F}$ . The magnitude and phase spectrums can be calculated as

$$\begin{aligned} \mathbf{M}_{u,v} &= \sqrt{a_{u,v}^2 + b_{u,v}^2} \\ \mathbf{P}_{u,v} &= \tan^{-1} \left[ \frac{b_{u,v}}{a_{u,v}} \right] \end{aligned} \quad (4)$$

where  $\mathbf{M}$  and  $\mathbf{P}$  are the respective functions to calculate the magnitude and phase spectrums. Inverse discrete Fourier transformation of  $\mathbf{F}$  in Eq. 3 to recover the original image, can be calculated as

$$\mathbf{I}[m, n] = \frac{1}{H} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} \mathbf{F}[u, v] \exp \left\{ j2\pi \frac{Nm u + M n v}{MN} \right\} \quad (5)$$

where  $H = M \times N$ . The original image can also be reconstructed from the magnitude  $\mathbf{M}$  and phase  $\mathbf{P}$  spectrums by first calculating the value of  $\mathbf{F}$  to be used in Eq. 5 as

$$\mathbf{F} = \mathbf{M} \times \exp(j \times \mathbf{P}) \quad (6)$$

We have used the magnitude spectrum of the rainy image to prove that rain streak information is preserved in the transformed domain and real, imaginary coefficients are given as

input to the deep network. The DFT<sup>1</sup> of an image consists of redundant frequencies because (a) DFT is Periodic, i.e.  $\mathbf{F}[u, v] = \mathbf{F}[u + Nk, v + Ml]$  for all  $k, l \in \mathbb{Z}$ , (b) DFT holds Conjugate Symmetry, i.e.  $\mathbf{F}[u, v] = \mathbf{F}^*[ -u + pN, -v + qM ]$  for any integer  $p, q$ , for an image of size  $M \times N$  [20]. We have not removed these redundancies before processing because their presence may not incur any misinterpretation.

## 2.1 Rain streaks in Fourier domain

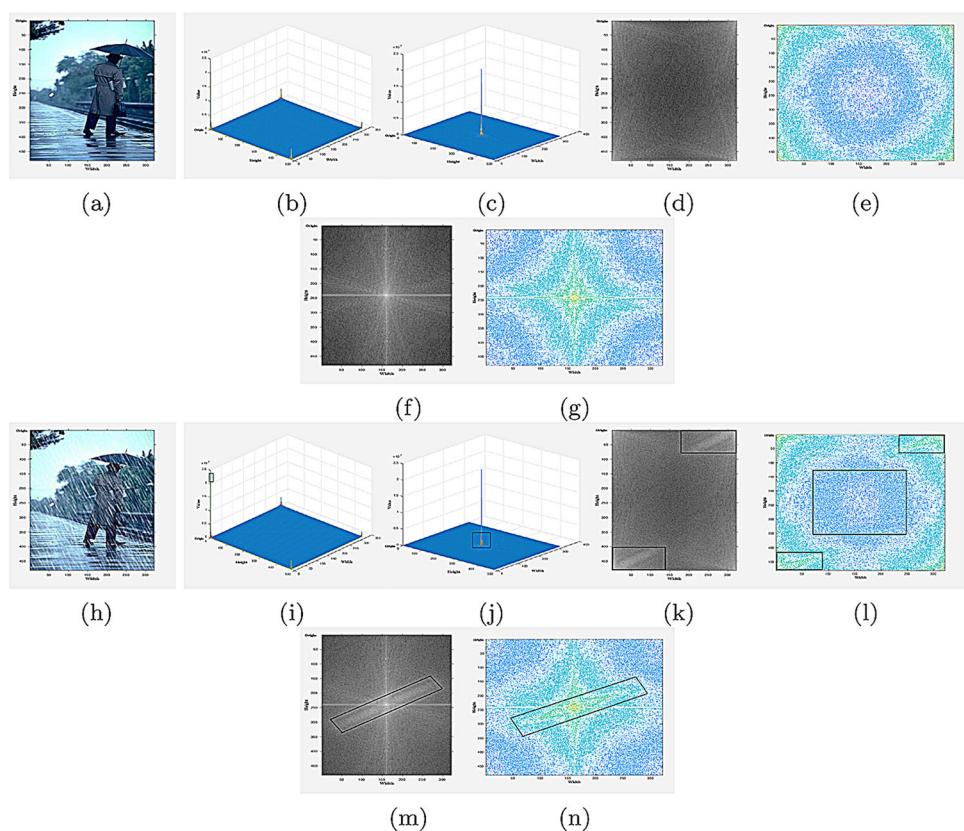
When an image is transformed from the spatial to the Fourier domain, the transformed signal is difficult to understand. However, the pseudo-periodic rain streaks among the pair of rainy and clean images can be visualized by using some transformation on Fourier data. We calculate the magnitude spectrums of a rainy and clean image using Eq. 4 as shown in Fig. 2. Both the images are first converted to grayscale, and their transformed DFT coefficients have been obtained using Eq. 3. For better visual representation, magnitude spectrums are scaled down by a logarithmic transformation as

$$\mathbf{M}_{\mathbf{I}_*}^{\text{scaled}} = \eta \times \log_e(\mathbf{M}_{\mathbf{I}_*}^{\text{original}}) \quad (7)$$

where  $\mathbf{M}_{\mathbf{I}_*}^{\text{original}}$  is unscaled,  $\mathbf{M}_{\mathbf{I}_*}^{\text{scaled}}$  is the scaled magnitude spectrums respectively and  $\eta$  is a scaling parameter which is set as 12 in our experimentation. It is observed in Fig. 2 that the transformed domain rain information can be more distinguishable if DC of the image signal is shifted to the center. Therefore, a DC shift to the center has been done by swapping all quadrants row and column wise before calculating the magnitude spectrums. Let the transformed space of scaled magnitude image with shifted DC component be  $S^*$ . In Fig. 2, the difference(here rain streaks) between rainy and clean images are noticeable in  $S^*$  space as indicated in black colored boxes. The difference between the unscaled magnitude spectrums of rainy, clean images with unshifted DC components is not visible to the human eye as shown in Fig. 2i, b respectively. When we shift the DC component to

<sup>1</sup> DFT : Discrete Fourier Transformation.

**Fig. 2** A flow of visualizations. **a** Clean image, **b** Unscaled magnitude image of (a) with unshifted DC component, **c** Unscaled magnitude image of (a) with DC component shifted to center, **d** Scaled magnitude image of (a) with unshifted DC component, **e** Contour plot of (d), **f** Scaled magnitude image of (a) with DC component shifted to center, **g** Contour plot of (f). Similar series of figures goes for rainy image (h)–(n) in the bottom row



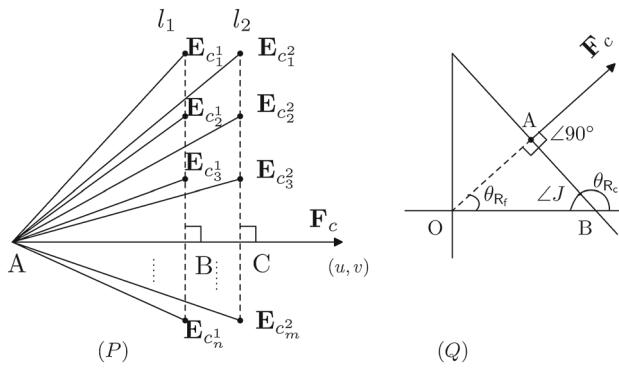
center as shown in Fig. 2j, c, the difference is slightly better visible. When we scale down the magnitude spectrums using Eq. 7 and do not shift the **DC** components, as shown in Fig. 2k, d, the difference is much more visible compared to previous images. We then shift their **DC** components to center and scale down the magnitude images for better understanding and visualization as shown in Fig. 2m, f. It is now easy to differentiate between a rainy and clean image in the Fourier domain, which are further transformed to  $S^*$  space, to show the preservation of rain streaks. It can be observed from Fig. 2 that there exist several high-intensity spots in magnitude spectrum of the rainy image in  $S^*$  space, bounded in the black colored box which is absent in the magnitude spectrum of the clean image. Each high-intensity spot in  $S^*$  space denotes a two-dimensional sinusoidal wave which corresponds to one periodic noise based on the assumption that pseudo-periodic noise in a rainy image consists of some combination of multiple periodic noises. The high-intensity spot is nearer to the center if the frequency of its corresponding sinusoid is low and farther otherwise. Therefore we may conclude that these high-intensity spots may represent rain streaks in the transformed domain and removing these high-intensity spots from the magnitude spectrum of the rainy image may remove most of the rain streaks in the spatial domain. However, there may exist some high-intensity spots in the magnitude spectrum which may contain some rain

information, but they are not very sensitive to the human eye. One of the primary goals of this paper is to train a deep network by exploiting this information to predict the rain map. To formally define the rain-streak information such as *relationship between the direction* of rain-streaks in spatial and transformed domains, consider a two-dimensional sinusoid with variables  $(x, y)$  in Euclidian space and  $(u, v)$  in frequency space as

$$\begin{aligned} f_{cp} &= \exp^{j2\pi(x.u+y.v)} \\ &= \cos 2\pi(x.u + y.v) + j \sin 2\pi(x.u + y.v) \end{aligned} \quad (8)$$

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

$$\begin{aligned} f_{cp} &= \exp^{j2\pi(x.u+y.v)} \\ &= \exp^{j2\pi w(x \cdot \frac{u}{w} + y \cdot \frac{v}{w})} \\ &= \exp^{j2\pi w(E_c \cdot F_c)} \end{aligned} \quad (9)$$



**Fig. 3** Sinusoids depicting the rain-streaks in spatial and transformed domain

where  $w = \sqrt{u^2 + v^2}$ ,  $\mathbf{F}_c$  is unit vector along the direction of  $(u, v)$  and  $\mathbf{E}_c^l = (x, y)^T$  is a spatial vector on the line  $l$ . The dot product  $\mathbf{E}_c^l \cdot \mathbf{F}_c$  represents the projection  $\mathbf{AB}$  of spatial point  $\mathbf{E}_c^l$  onto the direction of  $\mathbf{F}_c$  as shown in Fig. 3(P). All points on a straight line  $l$  perpendicular to the direction of  $\mathbf{F}_c$  have same projection. Hence  $\exp(j2\pi(x.u+y.v))$  represents a two-dimensional planar sinusoid with frequency  $w$  and whose direction is along the vector  $\mathbf{F}_c$ , i.e.  $\theta_{R_f} = \tan^{-1}(\frac{v}{u})$ . As shown in Fig. 3(Q), let  $\theta_{R_c}$  denote the rain streak direction in spatial domain and  $\theta_{R_f}$  in  $S^*$  space.  $\angle J = 180^\circ - \theta_{R_c}$ . Using the summation property of triangles on  $\triangle OAB$ ,  $\theta_{R_f}$  can be written as  $\theta_{R_c} - 90^\circ$  (Fig. 4).

## 2.2 Fourier domain input to deep CNNs

The input to the deep network is one of the major concerns in this work. Intuitively, both magnitude and phase spectrums are required to model the rain streak map in Fourier domain. It is clear from the Fig. 5 that phase spectrum also contributes to reconstructing the rain map. In Fig. 5b is reconstructed using magnitude spectrum of clean image and phase of rainy image by using inverse discrete Fourier transformation based on Eqs. 5, 6. Figure 5c is reconstructed using magnitude spectrum of rainy image Fig. 5a and phase of clean image Fig. 5e. Figure 5b, c still contains rain streaks when compared to original clean image Fig. 5d. But it can be observed that most of the rain information is captured in the magnitude spectrum, compared to phase, since Fig. 5b has fewer rain streaks than Fig. 5c. Therefore for rain streak removal, both magnitude and phase spectrums have to be given into the network<sup>2</sup>. To reduce the computational cost, instead of magnitude and phase information, real and imaginary coefficients are given as input to the deep network. Let  $\mathbf{I}_{rainy}$  be the rainy image,

<sup>2</sup> Figure 5a, e are first converted into YCbCr color space. Magnitude and phase are then calculated by performing DFT [25] on Y channel for each image. Chrominance values of Fig. 5a has been used to construct Fig. 5b, c. Chrominance values of Fig. 5e has been used to construct Fig. 5d.

$\mathbf{I}_{clean}$  be the clean image and  $\mathbf{I}_{rain map}$  be the rain streak map associated with  $\mathbf{I}_{rainy}$ .  $\mathbf{I}_{rainy}$  can be written as

$$\mathbf{I}_{rainy} = \mathbf{I}_{clean} + \mathbf{I}_{rain map} \quad (10)$$

When we convert the RGB rainy image to YCbCr colorspace, it is observed that most of the rain streak information exists in Y channel only<sup>3</sup>. Therefore input to the network is discrete fourier transformation of Y channel based on Eq. 3 as

$$\Psi(\mathbf{I}_{rainy}) = \mathbf{F}[Y_{rainy}] = \mathbf{F}_R[Y_{rainy}] \circ \mathbf{F}_I[Y_{rainy}] \quad (11)$$

where  $\circ$  is concatenation operation depthwise,  $\mathbf{F}$  is a two-dimensional discrete Fourier transformation defined in Eq. 3 and  $\mathbf{F}_R, \mathbf{F}_I$  are real and imaginary parts respectively. In later part of the paper,  $\mathbf{F}[Y_{rainy}]$  is also referred as  $\mathbf{F}[\mathbf{I}_{rainy}]$ .

## 2.3 Noise residual in Fourier domain

The noise residual based on Eq. 10 can be defined as

$$\mathbf{I}_{rain map} = abs(\mathbf{I}_{rainy} - \mathbf{I}_{clean}) \quad (12)$$

where  $\mathbf{I}_{rain map}$  is called as noise/rain map residual and  $abs$  is absolute difference. Once the model predicts rain map residual, it is pixel-wise subtracted from the rainy image to get the de-rained image. We train our model to learn the rain map residual from rainy image in fourier domain. The concept of noise residual described in Eq. 12 can also be used in frequency domain since DFT holds the linearity property which states that,

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

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

where  $p, q$  are two-dimensional discrete signals and  $\mathbf{F}$  is discrete Fourier transformation with some constant  $c$ . Considering above, Eq. 10 in frequency domain can be expressed as,

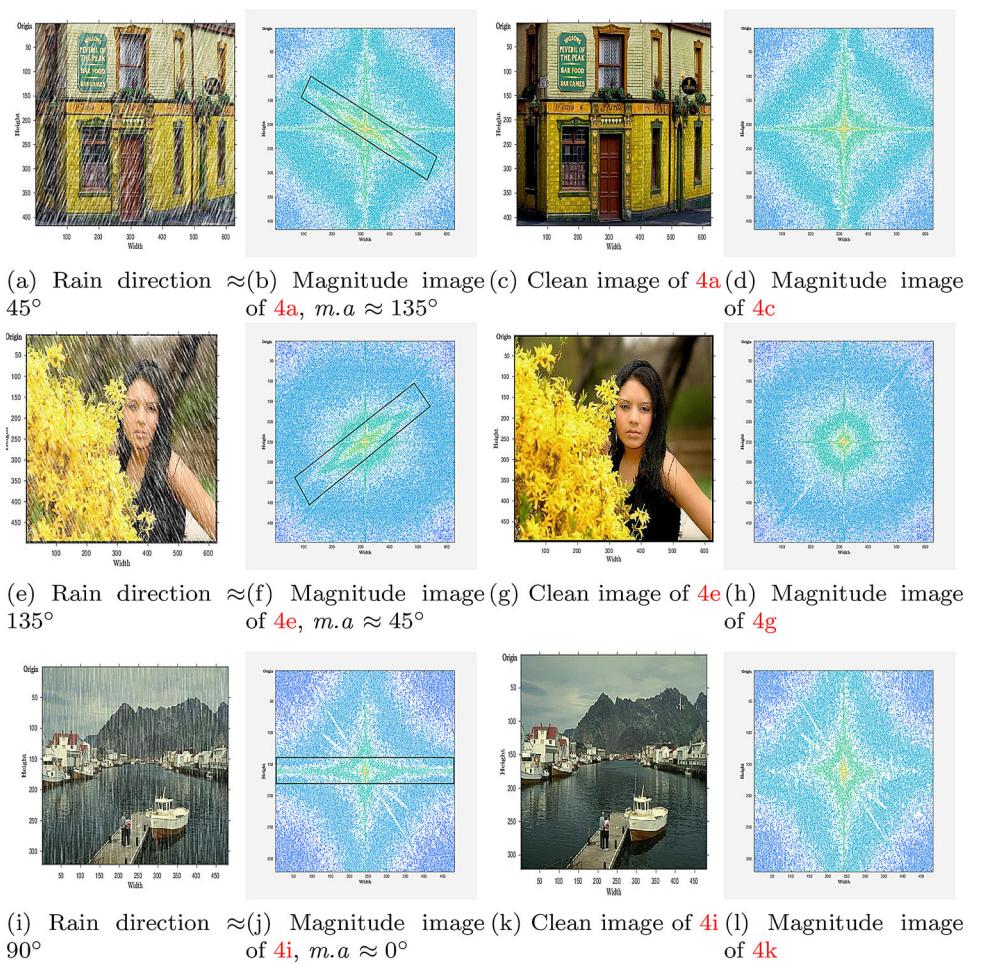
$$\begin{aligned} \mathbf{F}[\mathbf{I}_{rainy}] &= \mathbf{F}[\mathbf{I}_{clean} + \mathbf{I}_{rain map}] \\ &= \mathbf{F}[\mathbf{I}_{clean}] + \mathbf{F}[\mathbf{I}_{rain map}] \end{aligned} \quad (13)$$

The rain map residual as defined in spatial domain using Eq. 12, can be formulated in frequency domain as

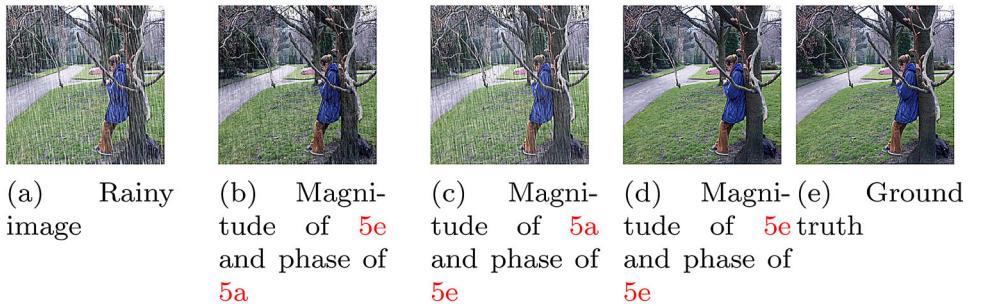
$$\begin{aligned} \mathbf{F}[\mathbf{I}_{rain map}] &= abs(\mathbf{F}[\mathbf{I}_{rainy}] - \mathbf{F}[\mathbf{I}_{clean}]) \\ &= abs(\mathbf{F}_R[\mathbf{I}_{rainy}] - \mathbf{F}_R[\mathbf{I}_{clean}]) \circ \\ &\quad abs(\mathbf{F}_I[\mathbf{I}_{rainy}] - \mathbf{F}_I[\mathbf{I}_{clean}]) \end{aligned} \quad (14)$$

<sup>3</sup> The quantitative analysis has been given in Appendix 6.1.

**Fig. 4** Different rain directions and their corresponding orientations in the contour plots of magnitude spectrums which are in  $S^*$  space. The mentioned angles are approximations



**Fig. 5** Reconstruction of images using different phase and magnitude spectrums. PSNR [16], SSIM [34]



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

### 3 Proposed networks

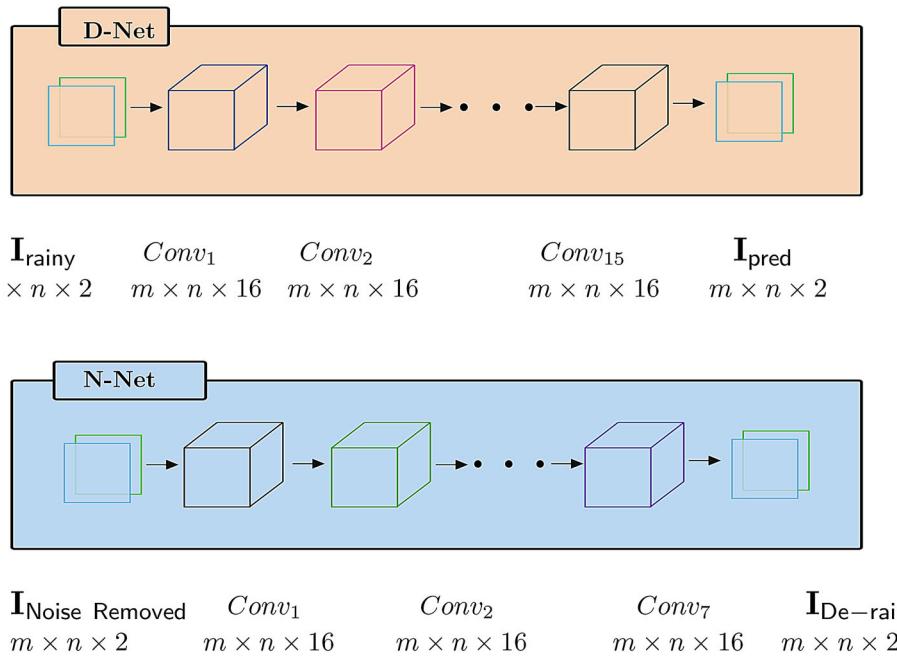
We fix the size of input patch based on Eq. 11, i.e.  $X \in \mathbb{R}^{128 \times 128 \times 2}$ . Given  $X$ , the networks are trained to learn the

parameter  $\Theta$  of a function  $h(\cdot|\Theta)$ , which maps  $X$  to the output  $Y \in \mathbb{R}^{128 \times 128 \times 2}$  such that

$$Y = h(X|\Theta) = f_L(f_{L-1}|\Theta_L) \quad (15)$$

where  $f_0=X$  each operation  $f_i(\cdot|\Theta_i)$  is referred as an  $i^{th}$  convolution layer of the network for  $i > 0$ . The proposed network denoted as RRSIFT<sub>1</sub> shown in Fig. 1 is comprised two major modules **D-Net** and **N-Net**. The **D-Net** takes input as described in Eq. 11 and predict the real, imaginary coefficients of rain streak less image denoted by  $\mathbf{F}[\mathbf{I}_{\text{pred}}]$  in Fig. 1. The initial estimate of rain streak map can be calculated using below equation as

**Fig. 6** The detailed architectures of D-Net and N-Net



$$\mathbf{F}[I_{\text{Noise Removed}}] = \text{abs}(\mathbf{F}[I_{\text{rainy}}] - \mathbf{F}[I_{\text{pred}}]) \quad (16)$$

The range of DFT coefficients is  $(-\infty, \infty)$  hence it may be difficult for a deep network to estimate the coefficients of rain streak map from a rainy image. Therefore an initial estimate of rain streak is calculated using **D-Net** and given as an input to **N-Net** to predict the complete rain streak map. The final predicted rain streak map by **N-Net** in Fourier domain is subtracted from the input of **D-Net** to get the final DFT coefficients of rain-free image. The de-rained image in spatial domain can be recovered using IDFT. The detailed architectures of **D-Net** and **N-Net** are described in following subsections.

### 3.1 D-Net

**D-Net** as shown in Fig. 6 is comprised of 16 convolution layers  $l_{1,2,\dots,16}$  and predicts the real and imaginary parts of Y channel of de-rained image based on the Eq. 15 with  $L = 16$ . The denoiser D-Net is parameterized as follows.  $l_1$  has input of the shape  $m \times n \times 2$  with 16 convolution filters of shape  $3 \times 3$ .  $l_{2:15}$  comprise of 16 convolution filters at each layer with shape  $3 \times 3$ .  $l_{16}$  outputs the real and imaginary parts of Y channel of initially predicted de-rained image of shape  $m \times n \times 2$  with 2 convolution filters of shape  $3 \times 3$ . Each filter has a stride of 1 and padding = *same*.

### 3.2 N-Net

**N-Net** as shown in Fig. 6 is comprised of 8 convolution layers  $l_{1,2,\dots,8}$  and predicts the real and imaginary parts of rain streak

map based on Eq. 15 with  $L = 8$ . The noise estimator N-Net is parameterized as follows.  $l_1$  has input of the shape  $m \times n \times 2$  with 16 convolution filters of shape  $3 \times 3$ .  $l_{2:7}$  comprise of 16 convolution filters at each layer with shape  $3 \times 3$ .  $l_8$  outputs the real and imaginary parts of final predicted rain streak map of shape  $m \times n \times 2$  with 2 convolution filters of shape  $3 \times 3$ . Each filter has a stride of 1 and padding = *same*. We have not used the pooling layers as it might cause in losing minor rain streak information due to down-sampling. We have not used any non-linearity in the network and have not normalized the input. The detailed discussion is given in the Sect. 5.

### 3.3 Loss functions

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

$$\begin{aligned}
 f_{\text{D-Net}}^l(\mathbf{F}[I_{\text{rainy}}]) &= \mathbf{W}_{\text{D-Net}}^l * f_{\text{D-Net}}^{l-1}(\mathbf{F}[I_{\text{rainy}}]) \\
 &\quad + \mathbf{b}_{\text{D-Net}}^l \\
 f_{\text{N-Net}}^l(\mathbf{F}[I_{\text{Noise Removed}}]) &= \mathbf{W}_{\text{N-Net}}^l * f_{\text{N-Net}}^{l-1}(\mathbf{F}[I_{\text{Noise Removed}}]) \\
 &\quad + \mathbf{b}_{\text{N-Net}}^l
 \end{aligned} \tag{17}$$

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

**Table 1** Quantitative results evaluated in terms of average SSIM [34] and PSNR [16] (dB) on the test datasets

Dataset	Models	PSNR [16]	SSIM [34]	FoM†
TD-Zhang et al. [39]	Input	21.15	77.81	49.48
	Luo et al. [23]	<b>21.44</b>	<b>78.96</b>	<b>50.20</b>
	Li et al. [21]	22.75	83.52	53.135
	Fu et al. [8]	<b>22.07</b>	84.22	53.145
	Yang et al. [35]	24.32	86.22	55.27
	Fu et al. [9]	27.33	89.78	58.55
	Zhu et al. [40]	23.05	85.22	54.135
	Zhang et al. [39]	27.95	90.87	59.41
	D-Net	<b>21.80</b>	<b>78.97</b>	<b>50.38</b>
	D-Net + N-Net	<b>22.22</b>	<b>78.46</b>	<b>50.34</b>
TD-Fu et al. [9]	Input	21.63	81.57	51.60
	Fu et al. [9]	27.56	91.57	59.56
	D-Net	<b>22.21</b>	<b>82.71</b>	<b>52.46</b>
	D-Net + N-Net	<b>22.59</b>	<b>82.31</b>	<b>52.45</b>
TD-Yang et al. [35] - H	Input	12.13	50.44	31.28
	Yang et al. [35]	23.45	74.90	49.17
	D-Net	<b>14.44</b>	<b>53.80</b>	<b>34.12</b>
	D-Net + N-Net	<b>13.87</b>	<b>52.50</b>	<b>33.18</b>
TD-Yang et al. [35] - L	Input	25.52	90.54	58.03
	Yang et al. [35]	36.11	97.00	66.55
	D-Net	<b>23.05</b>	<b>89.75</b>	<b>56.40</b>
	D-Net + N-Net	<b>23.92</b>	<b>89.51</b>	<b>56.71</b>

$\text{FoM}^\dagger = \frac{\text{SSIM} + \text{PSNR}}{2}$ . SSIM [34] values shown here have been multiplied by 100

Results lesser than proposed model are shown using bold and underlined

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

$$\begin{aligned} L_{\text{D-Net}} &= \sum_{i=1}^N \left\| \mathbf{F}[\mathbf{I}_{\text{pred}}^i] - \mathbf{F}[\mathbf{I}_{\text{clean}}^i] \right\|_2^2 \\ L_{\text{N-Net}} &= \sum_{i=1}^N \left\| \mathbf{F}[\mathbf{I}_{\text{Noise Estimated}}^i] - \mathbf{F}[\mathbf{I}_{\text{rain map}}^i] \right\|_2^2 \end{aligned} \quad (18)$$

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

## 4 Results

We have implemented both the modules in Tensorflow [1] framework. We have used the SSIM [34], PSNR [16] as evaluation metrics implemented in MATLAB 2018a. The synthetic datasets of rainy and clean images given by the authors of [9, 39] and [35] have been used for training and testing. Fu et al. [9] have selected clean images from various sources such as UCID [29], BSD [2] and Google images in order to generate rainy images with different orientation

and density of rain streaks. Yang et al. [35] have divided the dataset into two categories, one with light and another with heavy rain density. Zhang et al. [39] have included the medium rain density images in addition to light and heavy. We have randomly selected our training data only from the dataset available by Fu et al. [9] for both the modules. We have augmented the rainy and clean images from our selected training dataset into the disjoint patches of size  $128 \times 128$ , thus creating a total of 1,20,000 patches. We have trained our both modules on randomly selected 1,00,000 patches and validated on 20,000. We have selected three test datasets(TD). TD-Fu et al. [9] is taken randomly which has 2800 rainy images. TD-Yang et al. [35] (Heavy<sup>4</sup> and Light) and TD-Zhang et al. [39] are given by the authors of [35] and [39] respectively. For real-world case comparison, we have adopted images provided by the authors of [32]. The quantitative comparison has been shown in Table 1 on all test datasets. We have compared our models with the quantitative comparison given by Zhang et al. [39]. It can be observed that the proposed DFT-based approach for rain streak removal achieves a comparable result with state of the art approaches on the test datasets. It is important to note that

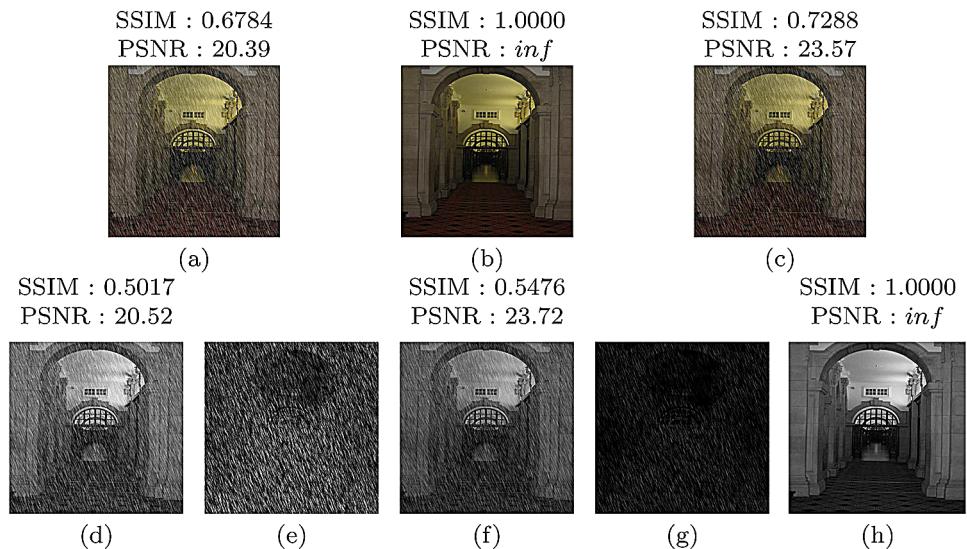
<sup>4</sup> Rain-streaks in this test-set may contradict with the real-rain.

**Fig. 7** Qualitative results on real-world rainy images. **Top** row shows rainy images, whereas **Bottom** row shows our results. TVE ( $\times 10^6$ ) denotes the total variation error that describes the amount of noise present in an image



**Fig. 8** Qualitative results on TD-Zhang et al. [39] dataset using **D-Net**. (a) Rainy image, (b) Clean image of (a), (c) Predicted de-rained image of (a), (d) Grayscale channel of (a), (e) Rain present in (d), (f) Grayscale channel of (c), (g) Rain present in (f), (h) Grayscale channel of (b). Rain present map here has been calculated by taking absolute difference of rainy and clean images, i.e.

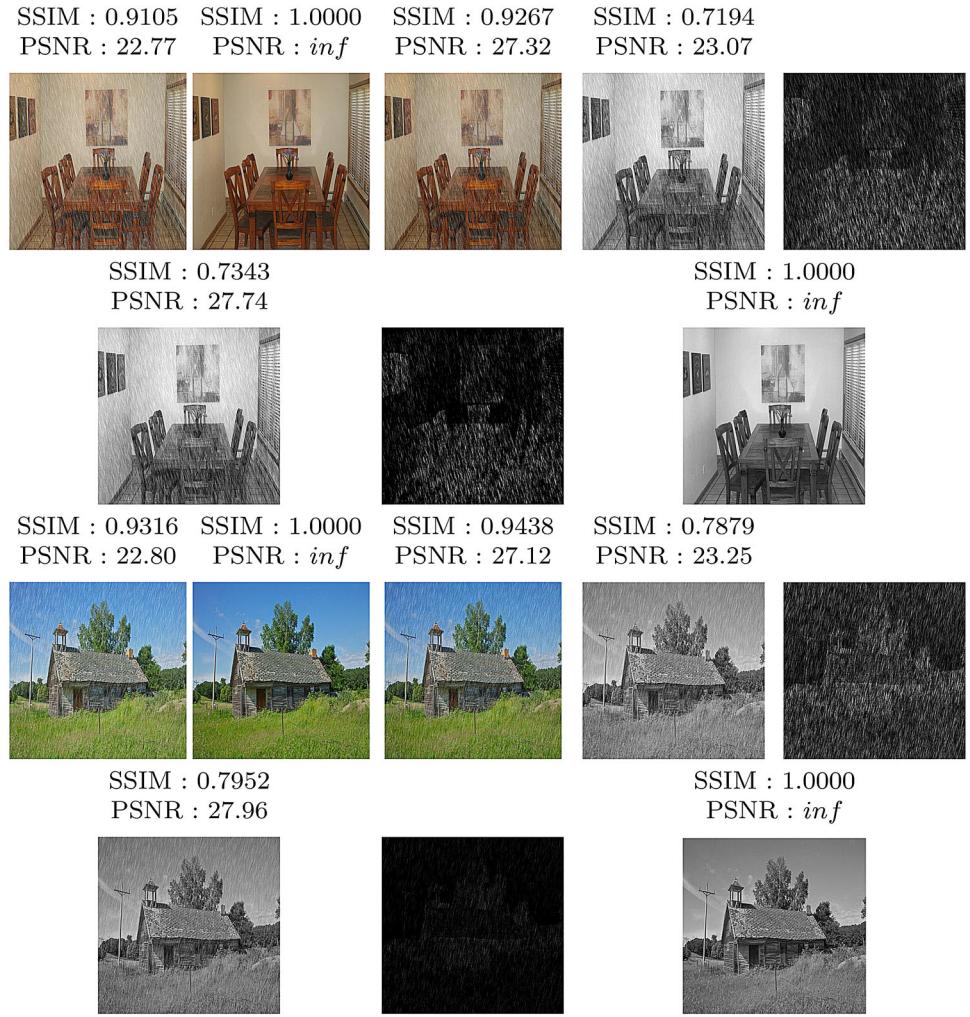
$|I_{\text{rainy or predicted}} - I_{\text{clean}}|$ .  
PSNR [16] is measured in dB



while the existing architectures incorporate and learn spatial domain features in their architecture, this is the first work achieving comparable performance using frequency domain input in deep convolutional neural networks. We have shown the qualitative results on real-world rainy images in Fig. 7. Figures 8 and 9 shows the qualitative results achieved by our proposed methods on the test dataset TD-Zhang et al. [39]. It is observed that complete rain streaks have not been

removed from the images when compared with the results obtained by spatial domain state of the art methods. However, a visual improvement along with the reduction in rain streaks can be observed in the de-rained image compared to the original rainy image.

**Fig. 9** Qualitative results on TD-Zhang et al. [39] dataset using the model **D-Net + N-Net**. Top Row : (a) Rainy image, (b) Clean image of (a), (c) Predicted de-rained image of (a), (d) Grayscale channel of (a) , (e) Rain present in (d), (f) Grayscale channel of (c), (g) Rain present in (f), (h) Grayscale channel of (b). Bottom Row: goes the same as top. Rain streak map here has been calculated by taking absolute difference of rainy and clean images, i.e.  $|I_{\text{rainy or predicted}} - I_{\text{clean}}|$ . PSNR [16] is measured in dB



**Table 2** Quantitative results on the test set TD-Fu et al. [9] of experimental models which takes normalized input and have Left: nonlinearities at each layer, Right : nonlinearities at each layer except the last

Epoch	Mode	PSNR [16]	SSIM [34]	Epoch	Mode	PSNR [16]	SSIM [34]
1	Train	5.64	2.19	1	Train	7.74	6.98
	Test	5.75	4.25		Test	6.87	7.36
18	Train	7.76	6.90	4	Train	7.71	6.91
	Test	6.86	7.27		Test	6.87	7.36
37	Train	7.70	6.74	6	Train	7.75	7.00
	Test	6.84	7.21		Test	6.87	7.37

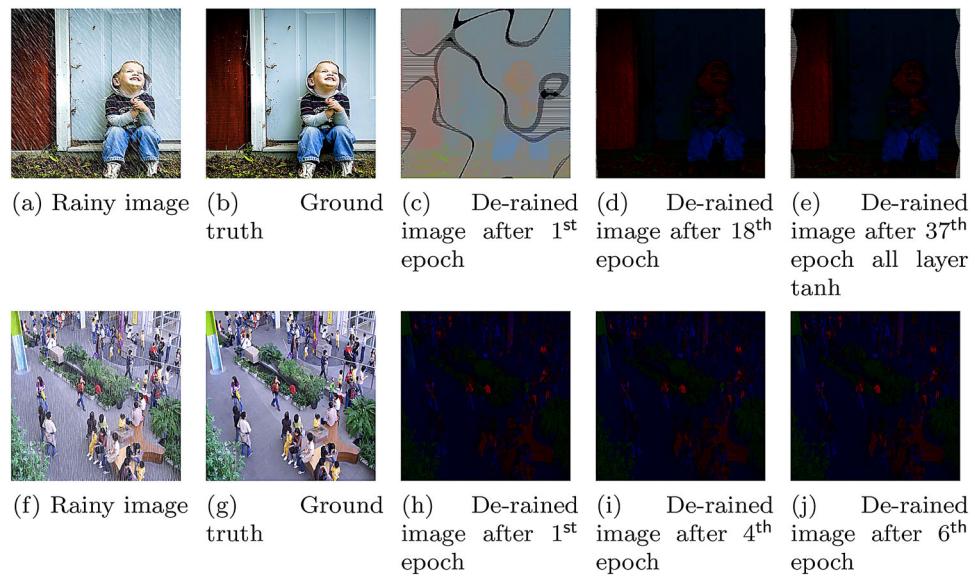
## 5 Discussion

### 5.1 Normalization of input data

It is observed that when input data is de-correlated, a linear model is enough to fit. The nonlinearity in the model may reduce the model capability and may get underfit. In general, input values to deep networks are normalized between a certain range to reduce the range of values to be predicted by the model. In this case, the input and values to be predicted by the proposed model lie between  $-\infty$  to  $\infty$  which make normal-

ization of the data difficult. We tried to normalize the input data based on the individual minimum and maximum values of input images and fed into the experimental networks A, B whose quantitative results are described in Table 2. Model A is topologically similar to **D-Net** with each convolutional layer has hyperbolic tangent as a nonlinear activation function. The input and target are normalized between [-1,1]. While testing, the result is upscaled by using the minimum and maximum DFT values of the input image which incurs in the loss. Model B is similar to A except the last convolution layer which does not have any activation function and pre-

**Fig. 10** Reconstruction of de-rained images in RGB colorspace after specified epoch using normalized input and activation function at, top row: each convolution layer, bottom row: except at last convolution layer



**Table 3** Quantitative results on the testset TD-Fu et al. [9] of experimental models with different number of layers

Layer	Configuration	PSNR [16]	SSIM [34]
1	<i>Conv k3-n2-s1-l<sub>1</sub></i>	22.39	82.13
4	<i>Conv k3-n16-s1-l<sub>1:3</sub></i>	22.44	82.19
	<i>Conv k3-n2-s1-l<sub>4</sub></i>		
8	<i>Conv k3-n16-s1-l<sub>1:7</sub></i>	22.54	82.29
	<i>Conv k3-n2-s1-l<sub>8</sub></i>		
16	<i>Conv k3-n16-s1-l<sub>1:15</sub></i>	22.21	82.71
	<i>Conv k3-n2-s1-l<sub>16</sub></i>		

dicts the un-normalized output. Both the models suffer from the underfitting problem as can be observed from Table 2. The loss in the reconstructed image is due to normalization techniques and also because of nonlinearity which is described in Sect. 5.3. Qualitative results of Table 2 are given in Fig. 10.

## 5.2 Single layer versus multilayer

We carried out experiments with the different number of layers in the **D-Net** with detailed configurations given in the Table 3. It can be observed that there is no significant improvement when the number of convolution layers is increased from 1 to 16. Therefore it may be inferred that the series of convolution layers without any nonlinearity and downsampling in between, acts as a single convolutional layer.

## 5.3 Non-linearity

The nonlinearity in model is used to make the network complex so that it can learn more complex functions which is

**Table 4** Quantitative results on the testset TD-Fu et al. [9] of experimental models with different nonlinear activation functions

Activation	Mode	PSNR [16]	SSIM [34]
tanh	Train	7.81	7.09
	Test	6.87	7.37
Sigmoid	Train	7.81	7.09
	Test	6.87	7.37
ReLU	Train	<b>18.18</b>	<b>68.63</b>
	Test	<b>16.24</b>	<b>71.51</b>

Best results are shown in bold

incapable for a linear model. It also helps in filtering out the unwanted information to be sent into the subsequent layers in a sequential model. To note the affect of nonlinearity, we used three different activation functions namely hyperbolic tangent, sigmoidal and ReLU in an experimental models and quantitative results are given in Table 4. It can be seen from the Sect. 5.2 that single layer network behaves almost similar to multilayer in special cases, we trained three networks each with single convolutional layer and different activation functions mentioned in Table 4 and can be summarized as follows

1. Hyperbolic tangent transforms the DFT values from  $(-\infty, \infty)$  to  $(-1, 1)$  which is too much random in nature. Therefore the transformed data might not preserve any information for the model to learn and the model may underfit which has happened when we use hyperbolic tangent as an activation function as mentioned in Table 4.
2. Sigmoidal function transforms the DFT values from  $(-\infty, \infty)$  to  $(0, 1)$  which is again too much random similar to hyperbolic tangent and may result in losing the

- negative frequencies due to its range. Therefore it also results in underfit as mentioned in Table 4.
3. ReLU transforms the DFT values from  $(-\infty, \infty)$  to  $\max(0, x)$  where  $x$  is the DFT value to be transformed. It can be observed that using ReLU will only result in losing negative DFT coefficients. Therefore there might be some information left after transformation which a model can attempt to learn. In this case, there is no underfitting as mentioned in Table 4. It is comparatively less random than hyperbolic tangent and sigmoid functions, therefore, perform better.

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

## 6 Conclusion

We have presented the deep learning architectures which take DFT coefficients as an input. While our proposed network D-Net predicts the initial estimate of rain streak map, D-Net+N-Net learns the final estimate of rain map which can be subtracted pixel-wise from rainy image DFT coefficients to get de-rained image DFT coefficients. We have shown that when the input is in the Fourier domain, the presence of nonlinearity in the network may result in underfit. We have used the synthetic datasets of rainy and clean images provided by the authors of [9, 35, 39] for single image de-raining. We have shown that rain streaks information is preserved in the transformed domain and deep CNN can be trained to utilize such features for image de-raining problem.

**Acknowledgements** Authors would like to thank the anonymous reviewers for their insightful comments and suggestions. Authors would also like to acknowledge the funding agency, Ministry of Human Resource Development, Government of India.

## Compliance with ethical standards

**Conflict of interest** Authors of this manuscript declare that they have no conflict of interest.

## Appendix

### 6.1 Quantitative analysis of rain-streaks present in Y channel compared to Cb & Cr

Deep learning models for image restoration, in RGB color-space due to its highly correlated nature, may induce white pixel artifacts and color saturation, especially in the case of

image de-raining [30]. YCbCr is a more suitable color-space for image restoration when the noise in an image exhibits linear or a pseudo-periodic nature [24, 30]. To quantify the noise present in the Y channel compared to chrominance channels, we have adopted the concept of sparsity and conducted the following experiment on the test-set TD- Zhang et al. that consists of 1200 rainy-clean image pairs.

1. We convert the rainy and clean images into YCbCr color-space.
2. Obtains the pixel-wise difference between corresponding Y Cb Cr channels of rainy and clean images.
3. Measure the sparsity ratio based on the following equation

$$\mathcal{S} = \frac{\text{No. of Zero-pixels}}{\text{Total No. of pixels}} \quad (19)$$

4. High sparsity indicates the low rain-noise present in the channel compared to its clean counterpart. We have obtained the following results. The same has been added in the appendix with qualitative results.

Metric	Y	Cr	Cb
$S$ (avg.)	0.0449	0.4481	0.4040

5. It can be observed that the Y channel has the lowest sparsity ratio that indicates the highest rain-streak noise present compared to other channels.
6. Although the values are test-set specific, it may obey for any rain-streak removal test-set.

### 6.2 Run-time comparison

We have implemented the proposed model using Tensorflow framework [1]. It takes  $\sim 0.56$  s to test an image of size  $512 \times 512$  on an 8 GB GPU. We have also compared the proposed model with existing schemes based on the run-time per image, and results are shown in Table 5.

**Table 5** Run-time comparison of the proposed work with existing schemes

Methods	Run-time (secs)
Luo et al. [23]	189.3
Fu et al. [8]	2.8
Yang et al. [35]	600.6
Fu et al. [9]	0.3
Zhu et al. [40]	1.4
Zhang et al. [39]	0.2
Proposed	0.560

## References

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). URL <https://www.tensorflow.org/>, software available from tensorflow.org
- Arbelaez, P., Maire, M., Fowlkes, C., Malik, J.: Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 898–916 (2011). <https://doi.org/10.1109/TPAMI.2010.161>
- Chang, Y., Yan, L., Zhong, S.: Transformed low-rank model for line pattern noise removal. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp 1735–1743 (2017). <https://doi.org/10.1109/ICCV.2017.191>
- Chen, D.Y., Chen, C.C., Kang, L.W.: Visual depth guided color image rain streaks removal using sparse coding. *IEEE Trans. Circuits Syst. Video Technol.* **24**(8), 1430–1455 (2014). <https://doi.org/10.1109/TCSVT.2014.2308627>
- Chen, Q., Yi, X., Ni, B., Shen, Z., Yang, X.: Rain removal via residual generation cascading. In: 2017 IEEE Visual Communications and Image Processing (VCIP), pp 1–4 (2017). <https://doi.org/10.1109/VCIP.2017.8305092>
- Davis, C.: The norm of the schur product operation. *Numer. Math.* **4**(1), 343–344 (1962). <https://doi.org/10.1007/BF01386329>
- Dunham, W.: Euler: the Master of Us All. No. v. 22 in Dolciani Mathematical Expositions, Mathematical Association of America (1999). <https://books.google.co.in/books?id=uKOVNvGOKhQC>
- Fu, X., Huang, J., Ding, X., Liao, Y., Paisley, J.: Clearing the skies: a deep network architecture for single-image rain removal. *IEEE Trans. Image Process.* **26**(6), 2944–2956 (2017). <https://doi.org/10.1109/TIP.2017.2691802>
- Fu, X., Huang, J., Zeng, D., Huang, Y., Ding, X., Paisley, J.: Removing rain from single images via a deep detail network. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1715–1723 (2017). <https://doi.org/10.1109/CVPR.2017.186>
- Goodfellow, I., Bengio, Y., Courville, A.: Deep Learning. MIT Press, Cambridge (2016)
- Goodfellow, I.J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Proceedings of the 27th International Conference on Neural Information Processing Systems—Volume 2, MIT Press, Cambridge, MA, USA, NIPS’14, pp 2672–2680 (2014). <http://dl.acm.org/citation.cfm?id=2969033.2969125>
- Gu, S., Meng, D., Zuo, W., Zhang, L.: Joint convolutional analysis and synthesis sparse representation for single image layer separation. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp 1717–1725 (2017). <https://doi.org/10.1109/ICCV.2017.189>
- Haar, A.: Zur theorie der orthogonalen funktionensysteme. *Math Ann* **69**(3), 331–371 (1910). <https://doi.org/10.1007/BF01456326>
- He, K., Sun, J., Tang, X.: Single image haze removal using dark channel prior. *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(12), 2341–2353 (2011). <https://doi.org/10.1109/TPAMI.2010.168>
- He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition (2015). [arXiv:1512.03385](https://arxiv.org/abs/1512.03385)
- Hore, A., Ziou, D.: Image quality metrics: Psnr vs. ssim. In: Proceedings of the 2010 20th International Conference on Pattern Recognition, IEEE Computer Society, Washington, DC, USA, ICPR ’10, pp 2366–2369 (2010). <https://doi.org/10.1109/ICPR.2010.579>
- Huang, D.A., Kang, L.W., Wang, Y.C.F., Lin, C.W.: Self-learning based image decomposition with applications to single image denoising. *IEEE Trans. Multimed.* **16**(1), 83–93 (2014). <https://doi.org/10.1109/TMM.2013.2284759>
- Kang, L.W., Lin, C.W., Fu, Y.H.: Automatic single-image-based rain streaks removal via image decomposition. *IEEE Trans. Image Process.* **21**(4), 1742–1755 (2012). <https://doi.org/10.1109/TIP.2011.2179057>
- Katznelson, Y.: An introduction to harmonic analysis. Cambridge Mathematical Library (1976)
- Lee, J.H., Heo, M., Kim, K.R., Kim, C.S.: Single-image depth estimation based on Fourier domain analysis. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2018)
- Li, Y., Tan, R.T., Guo, X., Lu, J., Brown, M.S.: Rain streak removal using layer priors. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2736–2744 (2016). <https://doi.org/10.1109/CVPR.2016.299>
- Li, Y., Tan, R.T., Guo, X., Lu, J., Brown, M.S.: Single image rain streak decomposition using layer priors. *IEEE Trans. Image Process.* **26**(8), 3874–3885 (2017). <https://doi.org/10.1109/TIP.2017.2708841>
- Luo, Y., Xu, Y., Ji, H.: Removing rain from a single image via discriminative sparse coding. In: 2015 IEEE International Conference on Computer Vision (ICCV), pp 3397–3405 (2015). <https://doi.org/10.1109/ICCV.2015.388>
- Lian, N.-X., Zagorodnov, V., Tan, Y.-P.: Edge-preserving image denoising via optimal color space projection. *IEEE Trans. Image Process.* **15**(9), 2575–2587 (2006)
- Oppenheim, A.V., Schafer, R.W.: Discrete-Time Signal Processing. Prentice Hall, Upper Saddle River (1989)
- Park, K., Yu, S., Jeong, J.: A contrast restoration method for effective single image rain removal algorithm. In: 2018 International Workshop on Advanced Image Technology (IWAIT), pp 1–4 (2018). <https://doi.org/10.1109/IWAIT.2018.8369644>
- Pratt, H., Williams, B., Coenen, F., Zheng, Y.: Fcnn: Fourier convolutional neural networks. In: Ceci, M., Hollmén, J., Todorovski, L., Vens, C., Džeroski, S. (eds.) Machine Learning and Knowledge Discovery in Databases, pp. 786–798. Springer, Cham (2017)
- Ren, W., Tian, J., Han, Z., Chan, A., Tang, Y.: Video desnowing and deraining based on matrix decomposition. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2838–2847 (2017). <https://doi.org/10.1109/CVPR.2017.303>
- Schaefer, G., Stich, M.: UCID: an uncompressed color image database. In: Yeung, M.M., Lienhart, R.W., Li, C.S. (eds.) Storage and Retrieval Methods and Applications for Multimedia 2004, vol 5307, pp 472–480 (2003). <https://doi.org/10.1117/12.525375>
- Sharma, P.K., Jain, P., Sur, A.: Dual-domain single image de-raining using conditional generative adversarial network. In: 2019 IEEE International Conference on Image Processing (ICIP), pp 2796–2800 (2019). <https://doi.org/10.1109/ICIP.2019.8803353>
- Shen, L., Yue, Z., Chen, Q., Feng, F., Ma, J.: Deep joint rain and haze removal from single images (2018). [arXiv:1801.06769](https://arxiv.org/abs/1801.06769)
- Wang, T., Yang, X., Xu, K., Chen, S., Zhang, Q., Lau, R.W.: Spatial attentive single-image deraining with a high quality real rain dataset. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2019)
- Wang, Y., Chen, C., Zhu, S., Zeng, B.: A framework of single-image deraining method based on analysis of rain characteristics. In: 2016 IEEE International Conference on Image Processing (ICIP), pp 4087–4091 (2016). <https://doi.org/10.1109/ICIP.2016.7533128>
- Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *Trans. Image Proc.* **13**(4), 600–612 (2004). <https://doi.org/10.1109/TIP.2003.819861>

35. Yang, W., Tan, R.T., Feng, J., Liu, J., Guo, Z., Yan, S.: Deep joint rain detection and removal from a single image. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 1685–1694 (2017). <https://doi.org/10.1109/CVPR.2017.183>
36. Yeh, C.H., Liu, P.H., Yu, C.E., Lin, C.Y.: Single image rain removal based on part-based model. In: 2015 IEEE International Conference on Consumer Electronics—Taiwan, pp 462–463 (2015). <https://doi.org/10.1109/ICCE-TW.2015.7216999>
37. Yu, S., Ou, W., You, X., Mou, Y., Jiang, X., Tang, Y.: Single image rain streaks removal based on self-learning and structured sparse representation. In: 2015 IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), pp 215–219 (2015). <https://doi.org/10.1109/ChinaSIP.2015.7230394>
38. Zhang, H., Patel, V.M.: Convolutional sparse and low-rank coding-based rain streak removal. In: 2017 IEEE Winter Conference on Applications of Computer Vision (WACV), pp 1259–1267 (2017). <https://doi.org/10.1109/WACV.2017.145>
39. Zhang, H., Patel, V.M.: Density-aware single image de-raining using a multi-stream dense network (2018). [arXiv:1802.07412](https://arxiv.org/abs/1802.07412)
40. Zhu, L., Fu, C.W., Lischinski, D., Heng, P.A.: Joint bi-layer optimization for single-image rain streak removal. In: 2017 IEEE International Conference on Computer Vision (ICCV), pp 2545–2553 (2017). <https://doi.org/10.1109/ICCV.2017.276>

**Publisher's Note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Prasen Kumar Sharma** is a Research scholar with the Department of Computer Science and Engineering at Indian Institute of Technology Guwahati, India. He has received his M.Tech degree in Information Technology from Indian Institute of Technology Kharagpur in 2017 and B.Tech. degree in Information Technology from JNTU, Hyderabad in 2014. His research interests include Image restoration using Deep Learning framework.



**Sathisha Basavaraju** is a Research scholar with the Department of Computer Science and Engineering at Indian Institute of Technology Guwahati, India. He has received his M.Tech degree in Computer Science and Engineering from Indian Institute of Technology Guwahati in 2012 and B.E. degree in Information Science and Engineering from Visvesvaraya Technological University, Belagavi in 2007. His research interests include Image Memorability, Image processing and reconstruction.



**Arijit Sur** received his Ph. D. degree in Computer Science and Engineering from Department of Computer Science and Engineering, Indian Institute of Technology Kharagpur. He has obtained his M. SC. in Computer and Information Science and M. Tech in Computer Science and Engineering both from Department of Computer Science and Engineering, University of Calcutta. He is currently working as an Associate Professor at Department of Computer Science and Engineering, Indian Institute of Technology Guwahati. He is a recipient of the Infosys Scholarship during his Ph. D. tenure at IIT Kharagpur. He got Microsoft Outstanding Young Faculty Program Award at Dept. of CSE, IIT Guwahati. His current research interest is Multimedia Security such as Image and Video Watermarking, Steganography & Steganalysis and Reversible Data Hiding.

## Terms and Conditions

Springer Nature journal content, brought to you courtesy of Springer Nature Customer Service Center GmbH (“Springer Nature”). Springer Nature supports a reasonable amount of sharing of research papers by authors, subscribers and authorised users (“Users”), for small-scale personal, non-commercial use provided that all copyright, trade and service marks and other proprietary notices are maintained. By accessing, sharing, receiving or otherwise using the Springer Nature journal content you agree to these terms of use (“Terms”). For these purposes, Springer Nature considers academic use (by researchers and students) to be non-commercial.

These Terms are supplementary and will apply in addition to any applicable website terms and conditions, a relevant site licence or a personal subscription. These Terms will prevail over any conflict or ambiguity with regards to the relevant terms, a site licence or a personal subscription (to the extent of the conflict or ambiguity only). For Creative Commons-licensed articles, the terms of the Creative Commons license used will apply.

We collect and use personal data to provide access to the Springer Nature journal content. We may also use these personal data internally within ResearchGate and Springer Nature and as agreed share it, in an anonymised way, for purposes of tracking, analysis and reporting. We will not otherwise disclose your personal data outside the ResearchGate or the Springer Nature group of companies unless we have your permission as detailed in the Privacy Policy.

While Users may use the Springer Nature journal content for small scale, personal non-commercial use, it is important to note that Users may not:

1. use such content for the purpose of providing other users with access on a regular or large scale basis or as a means to circumvent access control;
2. use such content where to do so would be considered a criminal or statutory offence in any jurisdiction, or gives rise to civil liability, or is otherwise unlawful;
3. falsely or misleadingly imply or suggest endorsement, approval , sponsorship, or association unless explicitly agreed to by Springer Nature in writing;
4. use bots or other automated methods to access the content or redirect messages
5. override any security feature or exclusionary protocol; or
6. share the content in order to create substitute for Springer Nature products or services or a systematic database of Springer Nature journal content.

In line with the restriction against commercial use, Springer Nature does not permit the creation of a product or service that creates revenue, royalties, rent or income from our content or its inclusion as part of a paid for service or for other commercial gain. Springer Nature journal content cannot be used for inter-library loans and librarians may not upload Springer Nature journal content on a large scale into their, or any other, institutional repository.

These terms of use are reviewed regularly and may be amended at any time. Springer Nature is not obligated to publish any information or content on this website and may remove it or features or functionality at our sole discretion, at any time with or without notice. Springer Nature may revoke this licence to you at any time and remove access to any copies of the Springer Nature journal content which have been saved.

To the fullest extent permitted by law, Springer Nature makes no warranties, representations or guarantees to Users, either express or implied with respect to the Springer nature journal content and all parties disclaim and waive any implied warranties or warranties imposed by law, including merchantability or fitness for any particular purpose.

Please note that these rights do not automatically extend to content, data or other material published by Springer Nature that may be licensed from third parties.

If you would like to use or distribute our Springer Nature journal content to a wider audience or on a regular basis or in any other manner not expressly permitted by these Terms, please contact Springer Nature at

[onlineservice@springernature.com](mailto:onlineservice@springernature.com)