ISSN: 2366-1313

A NOVEL IMAGE DERAINING USING MULTI SCALE PROGRESSIVE FUSION TECHNIQUE

B.Usha Rani[1], K.Koushik Panda[2] , K.Mahendra[3] ,W.Akhil[4],Y.Ganesh[5],S.Sai Kumar[6]

[1,2,3,4,5]Aditya Institute of Technology and Management, Tekkali

Abstract: Rain drops have different distance from their position to camera, rain drop in the air appear with varying degrees of blurring and resolution. A rain image and its multi-scale (or multiresolution) counterparts both show similar rain patterns, making possible to use this information for rain drop presentation. The multi scale progressive technique for single rain image drop removal is the name of the framework in which we examine the multi scale representation for rain drop from the view of input image scales and pyramidal deep features because multiscale progressive approach had better results in different image processing models. We use recurrent calculation to collect the texture for similar rain drops at various locations. This enables us to examine the information at the dimension to define the target rain. In addition, we build multi-scale hierarchal structures and add the attention mechanism to direct the careful integration of these associated data from various scales. The training is boosted by this multi scale progressive fusion technique in addition to the cooperative representation. Our suggested strategy receives the most cutting-edge outcomes after being thoroughly tested on numerous benchmark datasets. Additionally, we perform tests on combined detraining, detection, and segmentation tasks, which sparks a fresh line of inquiry into task-driven image de-raining.

Keywords: Image De-raining, Convolution Neural Network, Deep Learning.

1.Introduction: considerable difficulties caused by rain streaks in pictures and films, particularly in outdoor vision applications like traffic control, surveillance, and self-driving cars. In addition to impairing visual quality, these rain streaks also hinder the operation of several computer vision algorithms, which has an impact on real-world applications in industries like entertainment, remote sensing, and weather forecasting. The goal of the project is to create a practical and efficient way to improve image quality and clarity in bad weather, which is essential for automated systems and human perception alike.

considerable difficulties caused by rain streaks in pictures and films, particularly in outdoor vision applications like traffic control, surveillance, and self-driving cars. In addition to impairing visual quality, these rain streaks also hinder the operation of several computer vision algorithms, which has an impact on real-world applications in industries like entertainment, remote sensing, and weather forecasting. The goal of the project is to create a practical and efficient way to improve image quality and clarity in bad weather, which is essential for automated systems and human perception alike. This work presents a new method for image de-raining, focusing on the difficulty of eliminating rain streaks from individual pictures.

This project innovates by concentrating on single-image processing as opposed to conventional techniques that deal with video sequences. Using convolutional neural networks and deep learning, suggested approach seeks to rewrite the rules for picture deraining. To provide excellent results, it integrates deep learning techniques with domain knowledge specific to image processing in a unique way. This method not only increases the potential of many computer vision applications in bad weather, but it also promises to improve the quality of photos impacted by rain.

2.Literature survey:

Methods including Bossu et al. [1] presents a computer vision-based system that can identify whether it is raining or snowing. A traditional Gaussian Mixture Model is used in image sequences to distinguish the foreground from the background. Due to the dynamic nature of rain and snow, the foreground model is utilized for their detection. To choose the possible rain streaks, size- and photometry-based selection criteria are presented. Jie et al. [2] has proposed a. Removing rain is crucial to enhancing the resilience of systems that rely on outdoor vision. Existing techniques for removing rain have problems when it comes to either complicated dynamic pictures captured by quickly moving cameras or heavy downpours with opaque occlusions. In this case, a unique Derain algorithm breaks down the picture into depth-consistent units using super pixel (SP) segmentation. Xu et al. [3] has proposed Rain streaks can be eliminated from an image using a deep network design called DerainNet. The mapping link between rainy and clean picture detail layers is directly learned from data using a deep convolutional neural network (CNN). For training, we create photos with rain because we lack the ground truth equivalent to real-world images of it raining.

Li-Wei Kang, et al. [4] proposed Using morphological component analysis to formulate rain removal as an image decomposition problem, we offer a framework for rain removal based on a single image. The suggested method divides a picture using a bilateral filter into its low- and high-frequency (HF) components before performing a traditional image decomposition methodology directly. Wei-Sheng Lai, et al. [5] To gradually reconstruct the sub-band residuals of high-resolution pictures, the Laplacian Pyramid Super-Resolution Network (LapSRN) is suggested. Using coarse-resolution feature maps as input, our model predicts the high-frequency residuals at each pyramid level and then upsamples to the finer level using transposed convolutions. The method drastically lowers the computational complexity because it does not require bicubic interpolation as a pre-processing step.

Xue et al. [6] Lightweight pyramid networks for image deraining. IEEE Transactions on Neural Networks and Learning Systems, 2019. The Laplacian pyramid to predict the clean Gaussian pyramid Xia et al. [7] Recurrent Squeeze-and- Excitation Context Aggregation Net for Single Image Deraining It includes the exponentially increasing dilation and removal of the BN layer.Rui et al [8]Attentive generative adversarial network for raindrop removal from a single image The method utilizes a generative adversarial network, along with the input image to generate a raindrop-free image through a contextual autoencoder

3. Materials and Methodology:

3.1. Data Collection: The foundation of the training process is data gathering and preprocessing. A sizable dataset made up of clean and artificially rained versions of the photographs is assembled Pre-processing is done on these pictures to guarantee consistency in format, resolution, and size. The deep learning model is then trained on this dataset, with the goal of lowering the loss function, which is a representation of the variation between the output of the network and the clean image. Iterative training involves continuously adjusting layers, filters, and hyperparameters to maximize the model's performance. A battery of tests is used to validate the model with both artificial and real-world images. This is a critical stage in evaluating the model's generalizability and efficacy over a range of contexts. The derained images are statistically assessed for quality using performance metrics like Structural similarity index (SSIM) and Peak signal-to-noise ratio (PSNR).

3.2. Selection of Implementation Platform: Python is the language utilized, and Jupyter Notebook is the platform used for this machine learning implementation.

3.3. Data Preparation: This entails collecting a wide range of photos, including both dry and wet ones. To add more variation, artificial rain effects can be added to the dataset. Shrinking photographs to a consistent size, normalizing pixel values, and maybe using data augmentation methods like flipping, rotating, or adding noise are some examples of standardizing image formats.

3.4. Modelling: The figure 1 shows the overall methodology of the paper.

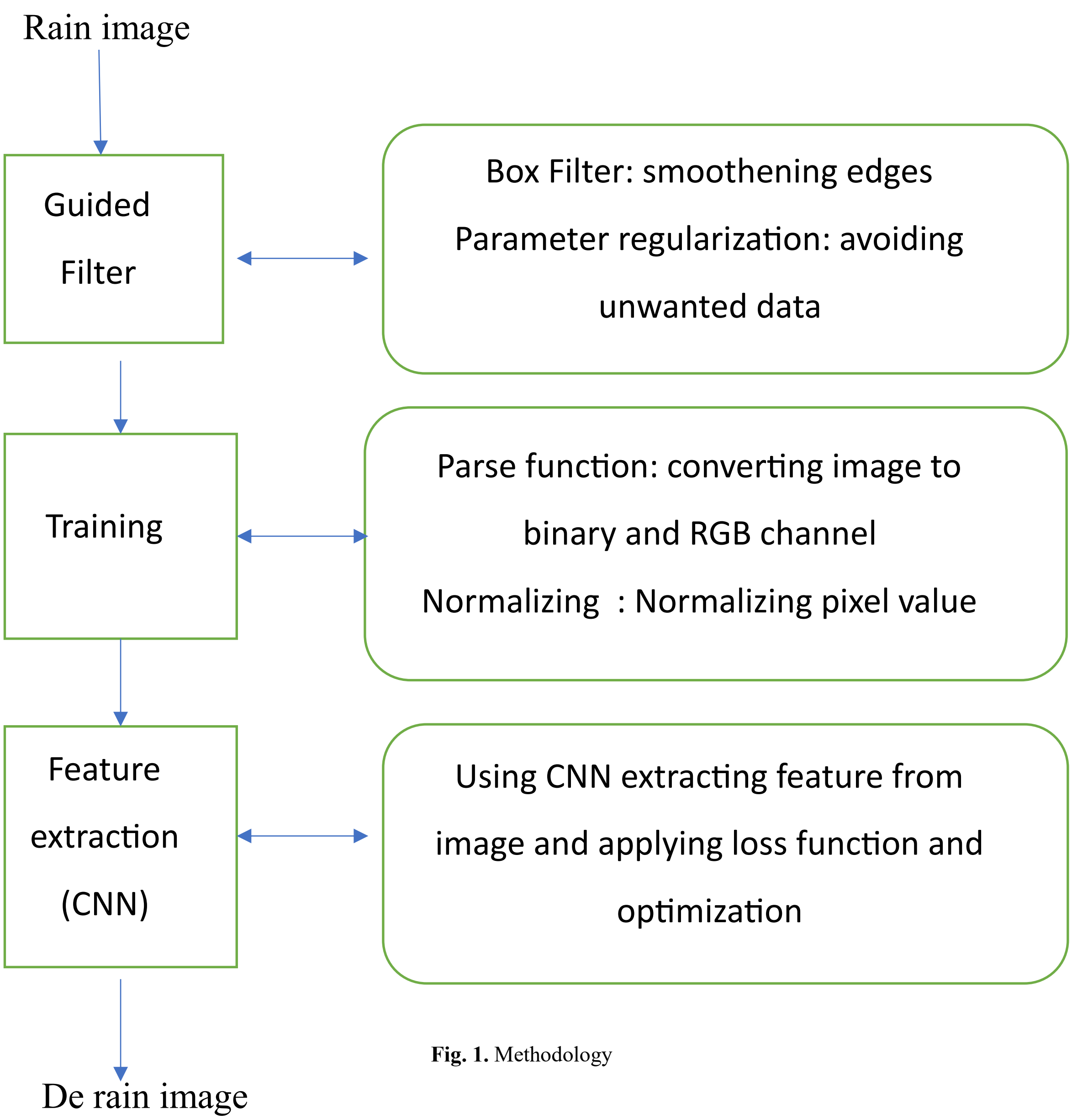


Fig. 1. Methodology

ISSN: 2366-1313

3.5 Guided filter: The function guidedfilter, which embodies the guided filter algorithm, accepts an input picture , a guiding image , a filter radius , and a supplementary epsilon parameter are included. The following is a concise representation of the algorithm:

3.5.1 Application of Guided Filters:

The application of the guided filter to the input image forms the basis of the process(x) and the directing picture (y):

3.5.2 Mathematical Formulation:

The guided filter is based on mathematical relationships that are essential to its performance. A box filter applied to a tensor of ones yields N, the normalization factor. The input tensor X's mean is represented by mean_X, which is obtained by dividing X by N after a box filter is applied. The link between a guiding tensor Y and X is captured by the covariance term cov_{XY}, which is computed as the product of mean_X and mean_Y minus the box filter of X * Y divided by N. These foundational elements include variables like A and B, which are essential to the guided filter's complex calculation.

N = box_filter(ones((1, 1, W, H)), r)

mean_x = box_filter(x, r) / N

covxy = box_filter(x * y, r) / N - mean_x * mean_y

3.6 Network Architecture: DerainNet:

As a preprocessing phase, a guided filter is used by the DerainNet model to anticipate rain details. Convolutional layers are used in the design in both the input and detail levels to capture complex characteristics. TensorFlow's guided filter helps get a base layer that is essential to improving the model's performance. describes the DerainNet model's architecture. The output and detail layers are obtained by applying convolutional layers after the guidedfilter function has produced a base layer.

3.7 loss function:

Mean Squared Error (MSE) The difference between the anticipated rain details and the ground truth labels is measured in the code using the Mean Squared Error (MSE) loss. The average squared difference between matching elements of the actual and predicted values is determined by the regression loss function, or MSE, which is frequently used.

loss = tf.reducemean(tf.square(detailslabel - detailsoutput))

detailslabel gives the base rain details, and detailsoutput is the predict output from the DerainNet model. The tf.square function gives the element wise squares of the difference, and tf.reducemean calculate mean for all elements, gives the MSE loss.

3.8 optimization:

The Adam optimizer is used to improve the model's parameters and minimize the MSE loss. an adaptive learning rate optimization technique. It accelerates the rate of convergence by varying the learning rates for each parameter separately.

3.8.1 Parameter tuning : Adjusting ParametersA key component of successful optimization is knowing and adjusting the network's parameters. These factors include the patch size, number of channels, and learning rate, which affect the step size during optimization. They also include architecture-specific characteristics like these. In order to balance convergence and model complexity, these parameters need to be carefully adjusted.In order to ensure that the DerainNet model learns to efficiently derain images during training, the optimization procedure involves the use of the Adam optimizer in conjunction with MSE loss and careful parameter tweaking. To achieve best performance, these components must be regularly monitored and adjusted.

3.9 Image Processing Loop:

The DerainNet model must be initialized, which is the next critical step. By utilizing TensorFlow functions, we established the framework for effective image processing. Our methodology becomes versatile for training and using pre-trained models with a TensorFlow session and a saver for model checkpoints. This flexibility is essential to our research's dynamic nature. The iterative image processing loop is the beating heart of our methodology. We use DerainNet on every image in the dataset and use metrics like Peak Signal-to-Noise Ratio (PSNR) and Structural Similarity Index (SSIM) to assess the quality of the derained photos. The loop guarantees a thorough evaluation of the model's performance on a variety of images. Saved processed photos offer real results for additional examination. Our methodology includes a targeted step to examine and assess a particular image as an extra level of examination. This targeted research expands on our knowledge of our model's potential and constraints by offering in-depth insights into how it functions under particular circumstances.A comprehensive method for picture deraining with DerainNet is formed by combining dataset preparation, model initialization, the image processing loop, and particular image evaluation. Each step is accompanied by snippets of code, providing an open and repeatable process for further study and development in this area.

4.Comparative analysis:

Ref	Dataset	Method	PSNR/SSIM
Rui et al. [8]	YOLO-V3	Attentive generative adversarial network	20.43
Xia et al. [7]	Rain800	Recurrent squeeze and Excitation context	24.73
Wei et al. [5]	BSDS100	Deep Laplacian pyramid Network	25.72
Xue et al. [6]	Rain100H	Lightweight Pyramid Network	23.73/0.81
Xu et al. [3]	UCID	A deep network architecture for single-image rain removal	0.82(SSIM)/24.31

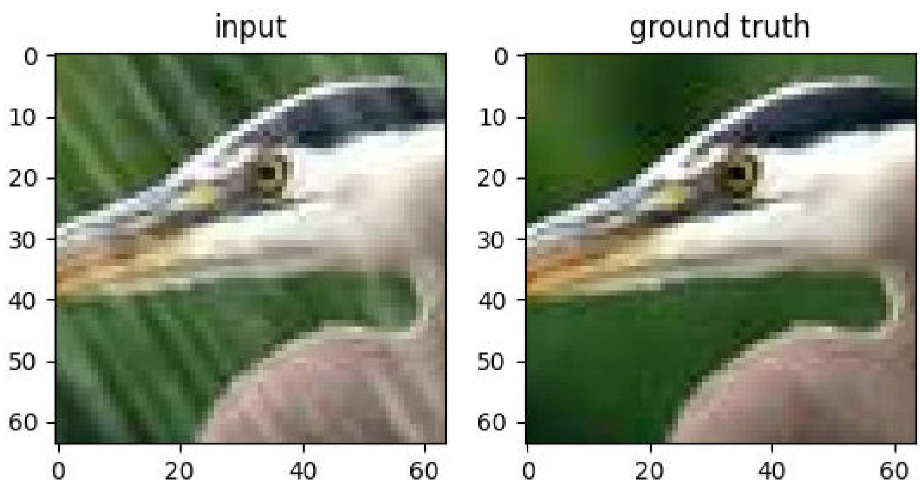
ISSN: 2366-1313

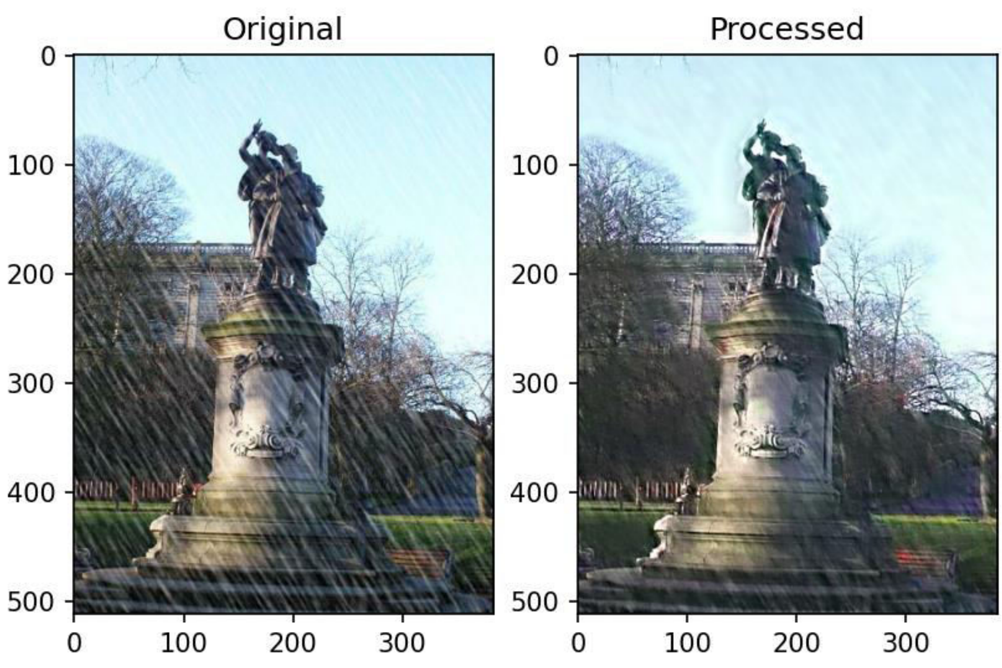
Our study	**UCID**	**Single Image De-raining Using Multiscale Progressive Fusion Network**	**0.86(SSIM)/27.71**

UCID dataset used in [3] and got 0.82 SSIM value and 24.31 PSNR value same UCID dataset is used in our model and we got 0.86 SSIM and 27.71 PSNR value our model comparatively done better than [7],[5],[6] .

5.Results:

Below images are output screens of our rained image: Figure shows a original image and a specific image after processed**.** We show a side-by-side comparison of the original and derained versions of the





photographs in the results section. This graphic illustration demonstrates how successful the deraining procedure is. Furthermore, we offer a quantitative assessment of the restoration accomplished for every image pair by providing Peak Signal-to-Noise Ratio (PSNR) values. We assess the deraining algorithm's effectiveness both visually and numerically, talking about how it can reduce the effects of rain-induced degradation while improving image clarity and preserving crucial information.

Rainy images has noise and details are not clearly shown but in derained images we found clarity and details .

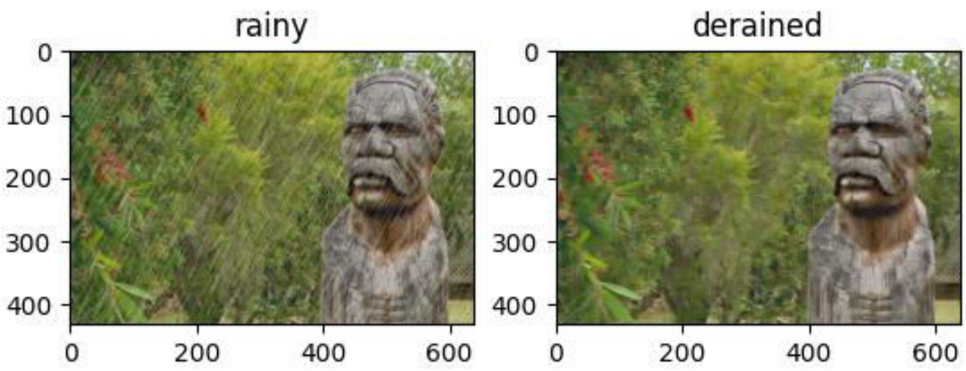


Figure shows PSNR and SSIM values of specific image

```
Image 90: SSIM = 0.9070357467885323, PSNR = 29.183856147974595
Image 91: SSIM = 0.8668262514298637, PSNR = 26.727816940118707
Image 92: SSIM = 0.8678467590856841, PSNR = 27.77935477070162
Image 93: SSIM = 0.798976129587032, PSNR = 27.395843553706545
Image 94: SSIM = 0.8940057389915722, PSNR = 26.402142491812675
Image 95: SSIM = 0.8614700406541976, PSNR = 27.97898354564708
Image 96: SSIM = 0.9085877931945183, PSNR = 26.484551499775975
Image 97: SSIM = 0.8391973317159281, PSNR = 26.77835038705335
Image 98: SSIM = 0.8089525716373789, PSNR = 25.142094988822112
Image 99: SSIM = 0.6837040582403127, PSNR = 27.75725030975874
Image 100: SSIM = 0.789873701432133, PSNR = 26.059914330307468
Specific Image: SSIM = 0.8391976765530395, PSNR = 26.778346504219105
```

In order to measure the improvement in image quality, PSNR values are given. We explore both the numerical and visual components of this technique, offering a thorough grasp of how the deraining process affects image restoration. PSNR value helps to showcase the reduced noise of our image . PSNR and SSIM values are given for specified image too

6.Conclusion

The creation of the single-image deraining project using a deep Convolutional Neural Network (CNN) represent a notable leap forward in the realm of image processing. This project effectively showcased the capability of a deep learning-based approach to remove rain streaks from individual images, significantly elevating image quality. The model's noteworthy attributes, including high accuracy, efficiency, and the preservation of original image integrity, highlight its potential applications across diverse fields such as photography, film, surveillance, and autonomous vehicle navigation.

A pivotal success of this endeavor lies in its ability to address shortcomings inherent in traditional deraining methods, particularly in terms of processing speed and suitability for single-image applications. The user-friendly interface developed for the system further enhances its practicality, making it accessible to users with varying backgrounds. Positive feedback from initial users serves as a testament to the system's potential impact and usability.

Nevertheless, the project encountered challenges, notably in the acquisition of diverse and comprehensive datasets essential for training the model. Additionally, exploring the interpretability of the AI's decision-making process emerged as a crucial aspect requiring further attention to instill trust and reliability in the system. These challenges underscore the ongoing efforts to refine and optimize the project for broader implementation and effectiveness.

References:

[1] Peter C Barnum, Srinivasa Narasimhan, and Takeo Kanade. Analysis of rain and snow in frequency space. International journal of computer vision, 86(2-3):256, 2010.

[2] Jer´ emie Bossu, Nicolas Hauti ´ ere, and Jean-Philippe Tarel. ` Rain or snow detection in image sequences through use of a histogram of orientation of streaks. International journal of computer vision, 93(3):348–367, 2011.

[3] Holger Caesar, Jasper Uijlings, and Vittorio Ferrari. Cocostuff: Thing and stuff classes in context. In IEEE Conference on CVPR, pages 1209–1218, 2018.

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

[5] Yi-Lei Chen and Chiou-Ting Hsu. A generalized lowrank appearance model for spatio-temporally correlated rain streaks. In IEEE International Conference on ICCV, pages 1968–1975, 2013.

[6] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. IEEE Trans. Image Process. 26(6):2944–2956, 2017.

[7] Xueyang Fu, Jiabin Huang, Delu Zeng, Yue Huang, Xinghao Ding, and John Paisley. Removing rain from single images via a deep detail network. In IEEE Conference on CVPR, pages 3855–3863, 2017.

[8] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. Lightweight pyramid networks for image deraining. IEEE Transactions on Neural Networks and Learning Systems, 2019.

[9] Kshitiz Garg and Shree K Nayar. When does a camera see rain? In IEEE International Conference on ICCV, volume 2, pages 1067–1074, 2005.

[10] Daniel Glasner, Shai Bagon, and Michal Irani. Superresolution from a single image. In IEEE International Conference on ICCV, pages 349–356, 2009.

[11] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. Meshcnn: a network with an edge. ACM Transactions on Graphics (TOG), 38(4):90, 2019.

[12] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In IEEE International Conference on ICCV, Oct 2017.

[13] Behzad Kamgar-Parsi and Azriel Rosenfeld. Optimally isotropic laplacian operator. IEEE Trans. Image Process., 8(10):1467–1472, 1999.

[14] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. IEEE Trans. Image Process., 21(4):1742–1755, 2011.

[15] Wei-Sheng Lai, Jia-Bin Huang, Narendra Ahuja, and MingHsuan Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In IEEE Conference on CVPR, pages 624–632, 2017.

[16] J. H. Kim, C. Lee, J. Y. Sim, and C. S. Kim. Single-image deraining using an adaptive nonlocal means filter. In IEEE ICIP, 2013.

[17] J. H. Kim, J. Y. Sim, and C. S. Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. IEEE Transactions on Image Processing, 24(9):2658–2670, 2015.

[18] A. Krizhevsky, I. Sutskever, and G. Hinton. ImageNet classificationwithdeepconvolutionalneuralnetworks. InNIPS, 2012.

[19] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradientbased learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, 1998.

[20] Y.Li,R.T.Tan,X.Guo,J.Lu,andM.S.Brown. Rainstreak removal using layer priors. In CVPR, 2016.

[21] Y.Luo,Y.Xu,andH.Ji. Removingrainfromasingleimage via discriminative sparse coding. In ICCV, 2015. [26] J.S.Ren,L.Xu,Q.Yan,andW.Sun. Shepardconvolutional neural networks. In NIPS, 2015.

[22] W. Ren, S. Liu, H. Zhang, J. Pan, X. Cao, and M. H. Yang. Single image dehazing via multi-scale convolutional neural networks. In ECCV, 2016.

[23] V. Santhaseelan and V. K. Asari. Utilizing local phase information to remove rain from video. International Journal of Computer Vision, 112(1):71–89, 2015. [29] G. Schaefer and M. Stich. UCID: An uncompressed color image database. In Storage and Retrieval Methods and Applications for Multimedia, 2003.

[24] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In ICLR, 2015.

[25] C.-H. Son and X.-P. Zhang. Rain removal via shrinkage of sparse codes and learned rain dictionary. In IEEE ICME, 2016.

[26] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In ICCV, 1998.

[27] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, 13(4):600–612, 2004.

[28] J. Xie, L. Xu, E. Chen, J. Xie, and L. Xu. Image denoising and inpainting with deep neural networks. In NIPS, 2012. [35] S. You, R. T. Tan, R. Kawakami, Y. Mukaigawa, and K. Ikeuchi. Adherent raindrop modeling, detection and removal in video. IEEE Transactions on Pattern Analysis and Machine Intelligence, 38(9):1721–1733, 2016.

[29] https://drive.google.com/file/d/10cu6MA4fQ2Dz16zfzyrQWDDhOWhRdhpq/view