Indian Institute of Information Technology, Allahabad

Department of Information Technology

Mini Project Report
(B.TECH Semester - 5)

Image Deraining

Guided By:

Dr. Navjot Singh
(Assistant Professor, Department of Information Technology)

Prepared By:

Ayan Ahmad - IIT2021090
Adeshpal Singh - IIT2021018
Dheeraj Devnani - IIT2021080
Ujjwal Pandram - IIT2021034
Devangi Gajjar - IIB2021037

Abstract — **This research focuses on image deraining, employing Generative Adversarial Networks (GANs) and image filtering methods. The study compares the generated derained images against ground truth using metrics like PSNR, SSIM, and MSE. The project explores the effectiveness of GANs, including CycleGANs, and traditional filters, emphasizing the role of CNNs and Cycle Consistency Loss. The findings offer insights into the trade-offs between computational complexity and visual fidelity, contributing to advancements in image deraining for applications in adverse weather conditions.**

Keywords — Image Filtering, PSNR, SSIM, MSE, GAN, CycleGAN, Generator, Discriminator, unpaired image-to-image mapping, Cycle Consistency Loss, adversarial losses, Identity Loss, CNN

I. Introduction

Images captured during rainy days often endure a discernible decline in scene visibility. The imperative task at hand for single image deraining algorithms is to transcend this degradation, aspiring to generate crisp and clear images from their rain-laden counterparts. Beyond the realm of mere visual enhancement, the application of image deraining holds the promise of significantly elevating both the perceptual quality of images for human observers and the operational efficiency of various computer vision applications.

Figure 1

The potential impact is far-reaching, extending its reach into critical domains like outdoor surveillance systems and intelligent vehicles. In these scenarios, where adverse weather conditions can impede conventional algorithms, the prowess of effective image deraining becomes particularly evident. As depicted in Figure 1, the substantial degradation caused by heavy rain poses a formidable challenge for face detection and verification algorithms, underscoring the urgency to develop robust solutions.

This research embarks on a comprehensive exploration of single image deraining methodologies, employing a synthesis of Generative Adversarial Networks (GANs) and traditional image filtering methods.. The focal points of our investigation include evaluating the effectiveness of GANs, with specific attention to CycleGANs, traditional filters, and the pivotal role played by Cycle Consistency Loss, Adversarial Loss and Identity Loss.

By navigating through the intricate trade-offs between computational complexity and visual fidelity, our findings aspire to contribute to the advancement of image deraining techniques. The insights garnered hold promise for applications in adverse weather conditions, positioning this research at the intersection of theoretical advancements and practical applicability.

From a mathematical standpoint, the composition of a rainy image involves the delineation of two distinct components: one corresponding to the rain streaks and the other to the clean background image. This separation unfolds through the expression:

$$x = y + w$$

Here, x signifies the rainy image, y represents the clean background image, and w encapsulates the rain streaks. Consequently, the process of image de-raining can be conceptualized as the task of disentangling these two integral components from a given rainy image.

II. Background and Related Work

A. Single Image Deraining

Sparsity-based Methods: Tackling the challenging nature of single image de-raining (SID), the approach presented in [1] employed learned dictionary atoms to sparsely represent clean backgrounds and rain-streaks. Another discriminative strategy [2] utilized discriminative codes and learned dictionary atoms with mutual exclusivity but encountered artifacts around rain-streak components.

Low-rank Representation-based Methods: To capture shared patterns and orientations in rain streak components, Chen et al. proposed a low patch-rank prior. However, this method introduced a risk of blurring important details. Addressing this concern, Zhang et al. introduced a convolutional coding-based method [3], leveraging convolutional low-rank filters to capture rain pixels.

Gaussian Mixture Model-based Methods: Li et al. [4] incorporated patch-based Gaussian Mixture Model (GMM) priors within an image decomposition framework, accommodating various orientations and scales of rain streaks.

Deep Learning-based Methods: Building on the success of convolutional neural networks (CNNs) in computer vision tasks, recent studies have explored CNN-based approaches for SID [5]. Fu et al.'s [6] two-step procedure involves decomposing rainy images into background-based and detail layers.

B. Generative Adversarial Networks (GANs)

Generative Adversarial Networks (GANs) are a class of machine learning models comprising a generator and discriminator. The generator creates data, such as images, from random noise, while the discriminator evaluates whether the generated data resembles real examples. Trained in tandem through a competitive process, the generator aims to produce increasingly realistic outputs, and the discriminator strives to distinguish between real and generated data. This adversarial dynamic leads to the refinement of the generator's ability to produce authentic-looking outputs. GANs have proven effective in various applications, from image synthesis to data generation, showcasing their versatility and generative prowess.

The Real-ESRGAN model [7] builds upon the ESRGAN architecture, incorporating a generator with residual-in-residual dense blocks for super-resolution tasks. It extends the original ESRGAN design to handle ×2 and ×1 scaling factors and employs pixel-unshuffle to optimize GPU memory consumption. The U-Net discriminator, influenced by prior works, enhances discriminative power for intricate outputs, providing detailed feedback to the generator. Spectral normalization stabilizes training dynamics and mitigates over-sharp artifacts introduced by GAN training. The training process involves two stages: a PSNR-oriented model (Real-ESRNet) is initially trained, followed by the Real-ESRGAN model with a combination of L1, perceptual, and GAN losses.

The DerainCycleGAN network [8] comprises three components: (1) U-ARSE for extracting rain streak masks from rainy images; (2) generators GN and GR for generating rain-free and rainy images; and (3) discriminators DN and DR for distinguishing real and generated images. Key Concept: Competitive Training, Generator aims to produce increasingly realistic data and Discriminator learns to better discern real from fake data. It features two branches: (1) rainy to rainy cycle-consistency and (2) rain-free to rain-freecycle-consistency. The Unsupervised Attention-guided Rain Streak Extractor (U-ARSE) uses visual attention in a multi-stage process to precisely extract rain streaks, considering both rainy and rain-free images simultaneously. The generators and discriminators are designed for the rainy (R) and rain-free (N) domains, incorporating adversarial losses, Identity Loss and cycle-consistency constraints. The objective function includes terms for adversarial, attention, cycle-consistency, perceptual, GMM, and reconstructive losses. During testing, the network utilizes U-ARSE and GN to generate rain-free images from given rainy images.

III. Architecture

A. Conventional Image Filters

Considering a filter J_{M_i}, with tunable parameters M_i , the filtering results vary with different settings of M_i to a certain extent. This variation in essence provides a range of approximations covering various input situations. To express the space of optimal solutions, we sample this continuous variation to form the set of 'filtered basis' (FB), written as $F = \{J_{M_i}(O) | i = 1, 2, ..., n\}$. Our goal is to learn the best blending of the filtered basis to produce the optimal solution Gc. If denoting the blending function as φ(·), the process can be simply expressed as

$$Gc = \varphi(F)$$

and the loss function can be formulated as

$$Lc = L(\varphi(F), Gc)$$

where φ(·) is defined as a shallow fully connected network to weigh different channels.

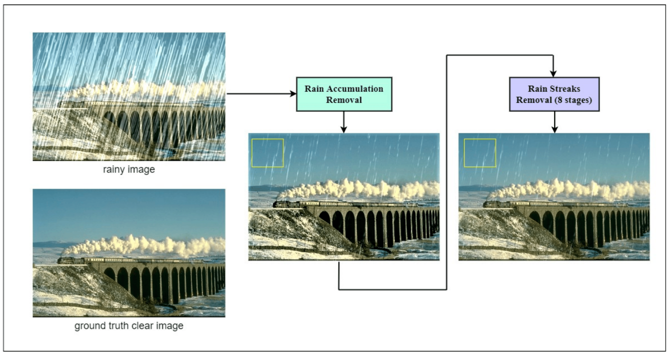


Image Enhancement Filters:
Contrast Enhancement: This step enhances the contrast of the image using CLAHE (Contrast Limited Adaptive Histogram Equalization).
Bilateral Filter: The code applies a bilateral filter to the image. This is achieved using the cv2.bilateralFilter function.
Morphological Operations: Morphological operations are used to try to remove rain streaks.Specifically, an opening operation is performed using a kernel of size (1, 5).
Blending: The code blends the luminance (Y channel) of the processed image with the chrominance (Cr and Cb channels) of the original image.

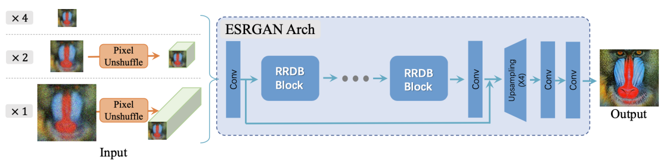
Image Derain Filters:
Gaussian Blur Filter, Median Blur Filter, Bilateral Filter,
Blur Filter, Box Filter, Dilation Filter, Erosion Filter Morphological Gradient Filter, Sobel Filter, Laplacian Filter, Canny Edge Detection Filter.

B. Generative Adversarial Networks (GANs)

GANs are powerful neural network architectures.

Comprise two main components - Generator: Creates synthetic data and Discriminator: Distinguishes real from fake data. Key Concept: Competitive Training: Generator aims to produce increasingly realistic data and Discriminator learns to better discern real from fake data.

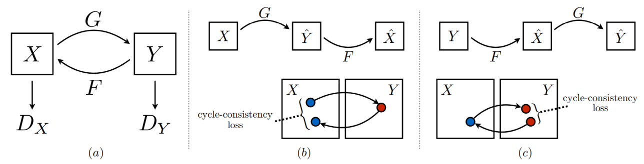
B1. Real-ESRGAN



We adopt the same generator (SR network) as ESRGAN, i.e., a deep network with sev- eral residual-in-residual dense blocks (RRDB). We also extend the original ×4 ESRGAN archi- tecture to perform super-resolution with a scale factor of

×2 and ×1. As ESRGAN is a heavy network, we first employ the pixel-unshuffle (an inverse operation of pixel- shuffle) to reduce the spatial size and enlarge the chan- nel size before feeding inputs into the main ESRGAN archi- tecture. Thus, the most calculation is performed in a smaller resolution space, which can reduce the GPU memory and computational resources consumption.

B2. CycleGAN



CycleGAN aims to minimize the combination of adversarial losses, cycle consistency losses, and identity losses, ensuring that the generators produce realistic translations between domains while maintaining cycle consistency and identity mappings.
Generators: G_{r2c} translates rainy images to clean images, while G_{c2r} translates clean images back to rainy images. Discriminator: D_r distinguishes real rainy images from translated/clean images, and D_c distinguishes real clean images from translated/rainy images.
Adversarial loss in GANs measures the competition between the generator and discriminator: the generator aims to produce data indistinguishable from real data, while the discriminator aims to distinguish real from fake. This loss function guides GAN training, leading to the creation of realistic synthetic data.

$$\begin{aligned}\mathcal{L}_{\text{GAN}}(G, D_Y, X, Y) =& \mathbb{E}_{y\sim p_{\text{data}}(y)}[\log D_Y(y)] \\ &+ \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log(1 - D_Y(G(x))],\end{aligned} \tag{1}$$

Cycle consistency loss in CycleGAN enforces that translating an image from domain A to domain B and back to domain A should result in an image similar to the original. It ensures that image content is preserved during unpaired image translation, enhancing coherence and meaningful translations. This loss term is essential for maintaining image quality and semantic consistency.

$$\begin{aligned}\mathcal{L}_{\text{cyc}}(G, F) =& \mathbb{E}_{x\sim p_{\text{data}}(x)}[\|F(G(x)) - x\|_1] \\ &+ \mathbb{E}_{y\sim p_{\text{data}}(y)}[\|G(F(y)) - y\|_1].\end{aligned} \tag{2}$$

IV. Implementation

A. Dataset:

Rain100H, Rain100L, Our dataset

Most existing datasets for Single Image Deraining consist of artificial rainy images with well-designed rain streaks, lacking realism compared to actual rain streaks.

We have created a script that transforms clear images into rainy scenes. By overlaying randomly selected rain streaks onto these images, it generates a rainy effect. The script processes a batch of clear images, applies different rain streaks to each, and saves the resulting 'rainy' images.

input img

output img

B. Conventional Filters:

Image Enhancement Filters - Contrast Enhancement, Bilateral Filter, Morphological Operations, Blending.

Filters Explored:

a. Gaussian Blur Filter: Applied using cv2. GaussianBlur with a kernel size of (5, 5) and a standard deviation of 0.
b. Median Blur Filter: Applied using cv2 .medianBlur with a kernel size of 5.
c. Bilateral Filter: Applied using cv2.bilateralFilter with a kernel size of 5, a sigma color value of 75, and a sigma space value of 75.
d. Dilation Filter: Applied using cv2. dilate with one iteration.
e. Erosion Filter: Applied using cv2. erode with one iteration.
f. Morphological Gradient Filter: Applied using cv2.morphologyEx with the morphological operation cv2.MORPH GRADIENT.
g. Sobel Filter: Applied using cv2. Sobel for edge detection.
h. Laplacian Filter: Applied using cv2. Laplacian for edge detection.

C. CycleGAN

Collect unpaired datasets for the two domains.Design U-Net architecture for generators (G_XtoY and G_YtoX).Design CNN architectures for discriminators (D_X and D_Y).Implement adversarial loss for realistic image generation.Implement forward and backward cycle-consistency losses. Combine adversarial and cycle-consistency losses in the overall loss function.Train generators and discriminators alternately.Set learning rates for generators and discriminators. Load batches of images from both domains during training. Optionally augment data for increased variability. Train for a sufficient number of epochs. Optionally visualize generated images during training. Use the trained generators to translate images between domains.

a. **Adversarial Loss**: Incorporating adversarial loss in CycleGAN amplifies the realism of generated images. Through a competitive process, the generator network G aims to produce rain-free images G(y) that are indistinguishable from real rain-free images x, compelling the discriminator D to discern between authentic and generated images by optimizing the log probabilities of correctly identifying both real x and generated G(y) images.

Adversarial loss $\mathcal{L}_{\text{GAN}}(G, D)$ is formulated as:

$$\mathcal{L}_{\text{GAN}}(G, D) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[\log(1 - D(G(y)))]$$

where x represents real rain-free images, y represents rainy images, $G(y)$ denotes the generated rain-free images, and $D(.)$ is the discriminator function.

b. **Cycle Consistency Loss:**By imposing cycle consistency loss in CycleGAN, the network ensures the fidelity of image transformations. This loss enforces that the transformation from rainy images y to rain-free G(y) and vice versa (from rain-free x to F(x)) maintains consistency with the original images, minimizing the L distance between the reconstructed and original images.

Cycle consistency loss $\mathcal{L}_{\text{cycle}}(G, F)$ for generators G and F is expressed as:

$$\mathcal{L}_{\text{cycle}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[||F(G(x)) - x||_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[||G(F(y)) - y||_1]$$

where F is the generator for the reverse transformation.

c. **Identity Loss:**Integrating identity loss in CycleGAN safeguards the integrity of already clear images during transformations. It discourages unnecessary alterations in rain-free images x and rainy images y, minimizing the L distance between the generated G(x) and F(y) images and their original counterparts.

Identity loss $\mathcal{L}_{\text{identity}}(G, F)$ for generators G and F:

$$\mathcal{L}_{\text{identity}}(G, F) = \mathbb{E}_{x \sim p_{\text{data}}(x)}[||G(x) - x||_1] + \mathbb{E}_{y \sim p_{\text{data}}(y)}[||F(y) - y||_1]$$

d. **Generator Loss:**The Generator Loss in CycleGAN amalgamates multiple components, each contributing uniquely to the training objective. It combines adversarial loss to enhance realism, cycle consistency loss to ensure transformation fidelity, and identity loss to preserve the integrity of already clear images. By assigning appropriate weights to these components, the generator G optimizes a composite loss function to synthesize high-quality, authentic rain-free images.

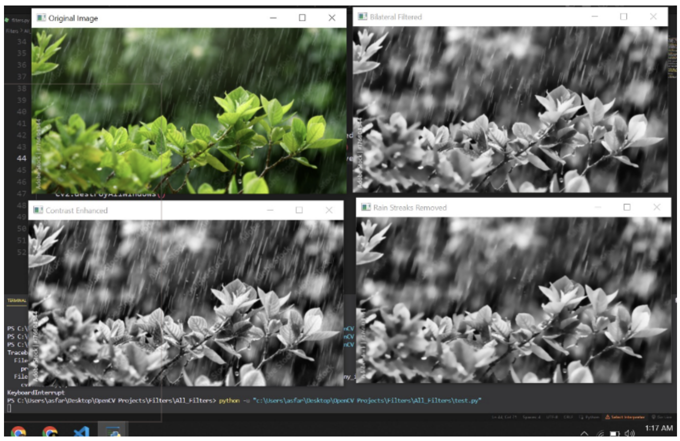
e. **Discriminator Loss:**The Discriminator Loss in CycleGAN focuses on improving the discriminator D's acumen in discerning authentic from generated images. It optimizes by minimizing the log probabilities of correctly identifying real rain-free images x and maximizing the log probabilities of correctly identifying generated rain-free images G(y), thereby refining the discriminatory ability to distinguish between real and synthetic outputs.

D. ESRGan

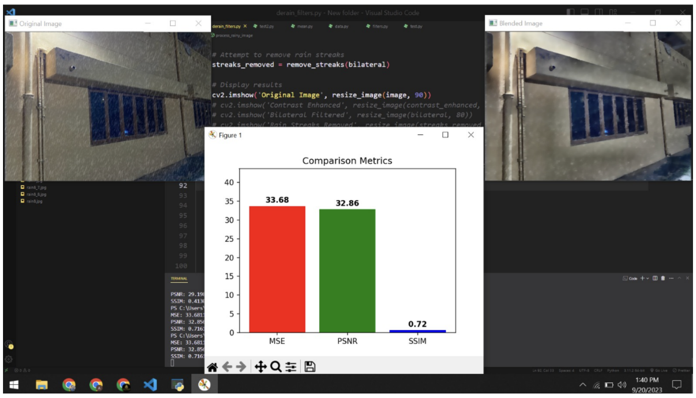
Architecture:
Generator Network (G): Utilizes CNN-based architecture with residual blocks and upscaling modules.
Discriminator Network (D): PatchGAN discriminator assessing image patches for real vs. generated images.
Loss Functions:
Perceptual Loss: Extracts feature maps using pre-trained networks (e.g., VGG), encouraging similarity between generated and real images' high-level features.
Adversarial Loss: Guides generator to produce realistic images, while discriminator distinguishes real from generated images.
Training Process:
Data Preparation: Collects pairs of low-resolution and high-resolution images.
Generator Training: Optimizes by minimizing combined adversarial and perceptual losses.
Discriminator Training: Enhances discernment between real and generated images.
Adversarial Training: Generator aims for realism to fool discriminator; discriminator improves classification.

V. Results

1. Conventional Filters



Input image and subsequent output images using different filters.



Evaluation Matrix using PSNR, MSE and SSIM

2. CycleGAN

Evaluation metrics. For images with ground truth, we evaluate each method by two commonly used metrics, i.e., Peak Signal-to-Noise Ratio (PSNR) [21] and Structural Similarity Index (SSIM) [17]. For the cases without ground truth, i.e., SS-TL-Data, we only provide visual results.

Compared methods. The deraining result of our net is compared with those of two model-driven algorithms (i.e., DSC [11] and GMM [10]), four supervised deep nets (i.e., DetailNet [4], JORDER [15], RESCAN [12], and PReNet [24]), one semi-supervised deep net SS-TL [27], and two unsupervised deep nets.

Datasets		Rain100L		Rain100H	
Metrics		PSNR	SSIM	PSNR	SSIM
Model-based methods	DSC [11]	27.34	0.849	30.07	0.866
	GMM [10]	29.05	0.872	32.14	0.916
Supervised methods	DetailNet [4]	32.38	0.926	34.04	0.933
	JORDER [15]	36.61	0.974	33.92	0.953
	RESCAN [12]	38.52	0.981	36.43	0.952
	PReNet [24]	37.45	0.979	36.66	0.961
Semi-supervised method	SS-TL [27]	32.37	0.926	34.02	0.935
Unsupervised methods	**CycleGAN (Our Model)**	**30.56**	**0.91**	**29.45**	**0.88**

VI. Use Cases

The application domains for single image deraining algorithms are diverse, extending their impact across various fields.

- Enhanced Vision in Autonomous Vehicles: Deraining algorithms contribute to improving the visibility of images captured in adverse weather conditions, ensuring better performance and safety for autonomous vehicles navigating through rain.
- Surveillance Systems: In surveillance scenarios, where clear visibility is crucial, deraining techniques enhance image quality, aiding in the accurate detection and identification of objects and individuals.
- Outdoor Photography: Single image deraining plays a pivotal role in outdoor photography by restoring images affected by rain, allowing photographers to capture and preserve the aesthetic appeal of scenes even during inclement weather.
- Traffic Management: Deraining algorithms contribute to improved visibility in traffic monitoring systems, facilitating more accurate analysis and decision-making in real-time traffic management applications.
- Meteorological Monitoring: In meteorological applications, clear and undistorted images are crucial for accurate analysis. Deraining techniques ensure that rain-induced degradation does not compromise the quality of meteorological images.

- Reliable Image and Video Analysis: For applications involving image and video analysis, such as object recognition or tracking, deraining ensures that the input data is free from the distortions caused by rain, leading to more reliable analysis outcomes.

VII. Limitations

The challenges and constraints associated with single image deraining algorithms are as follows.

- Non-Uniform Rain Patterns: Deraining struggles with diverse rain patterns, impacting its ability to uniformly remove rain streaks across images with varying conditions.
- Real-Time Processing Demands: Computational constraints hinder real-time deraining, limiting applications like autonomous vehicles and surveillance systems that demand swift image processing.
- Limited Training Data: Deraining models face limitations in generalization due to insufficient training data, impacting performance in unforeseen rain scenarios.
- Over-Smoothing: Some deraining methods risk over-smoothing, sacrificing fine details in the scene during rain removal.
- Artifacts: Deraining processes may introduce artifacts, such as ringing or unnatural patterns, diminishing the quality and interpretability of derained images.

VIII. Future Scope

In the future, the scope of this research can be broadened to address emerging challenges and enhance the applicability of the deraining algorithm.

- Generalization for Diverse Rain Patterns: Develop deraining models with improved adaptability to diverse and non-uniform rain patterns, ensuring robust performance across a broader range of scenarios.
- Real-Time Optimization:: Focus on optimizing deraining algorithms to meet real-time processing demands, enhancing their applicability in time-sensitive applications like autonomous vehicles and surveillance.
- Expanding to Handle Various Weather Conditions: Extend deraining capabilities to address different weather conditions beyond rain, providing a comprehensive solution for diverse environmental challenges.
- Enhancing Video Deraining Capabilities: Explore advancements in deraining techniques specifically tailored for video content, ensuring effective removal of rain streaks while maintaining temporal coherence.

IX. References

[1] L.-W. Kang, C.-W. Lin, and Y.-H. Fu, “Automatic single-image-based rain streaks removal via image decomposition,” IEEE TIP, vol. 21, no. 4, pp. 1742–1755, 2012.

[2] Y. Luo, Y. Xu, and H. Ji, “Removing rain from a single image via discriminative sparse coding,” in ICCV, 2015, pp. 3397–3405.

[3] H. Zhang and V. M. Patel, “Convolutional sparse and low-rank codingbased rain streak removal,” in 2017 IEEE WACV. IEEE, 2017, pp. 1–9

[4] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, “Rain streak removal using layer priors,” in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016, pp. 2736–2744.

[5] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, “Removing rain from single images via a deep detail network,” in 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017, pp. 1715–1723.

[6] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, “Clearing the skies: A deep network architecture for single-image rain removal,” IEEE Transactions on Image Processing, vol. 26, no. 6, pp. 2944–2956, 2017

[7] X. Wang, L. Xie, C. Dong, & Y. Shan. (2021). Real-ESRGAN: Training Real-World Blind Super-Resolution with Pure Synthetic Data. arXiv preprint arXiv:2107.10833

[8] Wei, Y., Zhang, Z., Wang, Y., Xu, M., Yang, Y., Yan, S., & Wang, M. (2021). DerainCycleGAN: Rain Attentive CycleGAN for Single Image Deraining and Rainmaking. (arXiv:1912.07015)

[9] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing Rain from Single Images via a Deep Detail Network. In CVPR, pp.1715-1723, 2017

[10] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain Streak Removal Using Layer Priors. In CVPR, 2016.

[11] Y. Luo, Y. Xu, and H. Ji. Removing Rain from a Single Image via Discriminative Sparse Coding. In ICCV, 2015.

[12] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan. Deep Joint Rain Detection and Removal from a Single Image. In CVPR, pp. 1685-1694, 2017.

[13] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeezeand-excitation context aggregation net for single image deraining. In ICCV, pp. 262-277, 2018.

[14] W. Wei, D. Meng, Q. Zhao, Z. Xu, Y. Wu. Semi-supervised transfer learning for image rain removal. In CVPR, pp. 3877-3886, 2019.