Trinity College Dublin
Coláiste na Tríonóide, Baile Átha Cliath
The University of Dublin

School of Computer Science and Statistics

Single Image DeRaining

Pranav Raviraj Shetty Nachiketh Janapareddy

Instructor: Subrahmanyam Murala

December 21, 2023

A project report submitted in partial fulfilment
of the requirements for course
CS7GV1 computer vision)

Declaration

I hereby declare that this project report is entirely my own work and that it has not been submitted as an exercise for a degree at this or any other university.

I have read and I understand the plagiarism provisions in the General Regulations of the University Calendar for the current year, found at `http://www.tcd.ie/calendar`.

I have completed the Online Tutorial on avoiding plagiarism 'Ready Steady Write', located at `http://tcd-ie.libguides.com/plagiarism/ready-steady-write`.

I consent / do not consent to the examiner retaining a copy of the thesis beyond the examining period, should they so wish (EU GDPR May 2018).

I agree that this thesis will not be publicly available, but will be available to TCD staff and students in the University's open access institutional repository on the Trinity domain only, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement. **Please consult with your supervisor on this last item before agreeing, and delete if you do not consent**

Signed: ______________________ Date: ______________________

This must be on a separate page.

1. Abstract

Single-image deraining is challenging due to the unpredictable nature of rain models. Current methodologies often adhere to narrow rain model assumptions, which fall short in diverse real-world conditions, leading to complex optimization or repetitive refining steps. This significantly limits their practicality and efficiency, especially for applications where efficiency is crucial. In this paper, we tackle single-image deraining as a broad image enhancement problem and introduce a model-independent deraining method, Efficient-DeRain. This method processes rainy images remarkably fast, while achieving comparable de-raining results. A novel pixel-wise dilation filtering technique is introduced, employing pixel-specific kernels generated by a kernel prediction network. This enables efficient multi-scale kernel prediction for each pixel. Additionally, an innovative data augmentation strategy, RainMix, is proposed to train the network more effectively for real-world rainy images, thereby bridging the synthetic-real data gap. Extensive evaluations on both synthetic and real-world rainy datasets have demonstrated the method's effectiveness and efficiency.

2. Introduction

Rain patterns captured by outdoor vision systems often induce sudden intensity changes in images or videos, impairing the performance of visual perception systems in tasks like object tracking, semantic segmentation, and pedestrian detection. Given its frequent occurrence, integrating a deraining feature in all-weather vision systems is almost essential. A deraining method aims to enhance image quality for downstream vision tasks by removing rain streaks from corrupted image/video data.

Efficient on-chip deraining is vital for real-time, efficiency-sensitive applications, such as autonomous driving or navigation. Practical applications demand deraining techniques that balance low overhead with high performance and efficiency. Despite recent progress in deraining, most methods focus on deciphering physical rain and background models, addressing optimization challenges to remove rain streaks, and leveraging deep learning with certain assumptions about rain patterns. However, the efficiency aspect has been relatively overlooked, limiting their applicability in real-time scenarios.

Current methods typically rely on specific rain generation hypotheses and rain-background models. Their goal is to reverse this rain-adding phase through intensive iterative optimization and refinement. However, these methods often rest on limited rain model assumptions, failing to accurately represent real-world rain patterns. This study introduces an efficient, universal derain method by approaching single-image deraining from a new angle. The proposed approach is model-independent and doesn't rely on assumptions about rain generation. Experimental evidence shows that assumptions about the rain model are

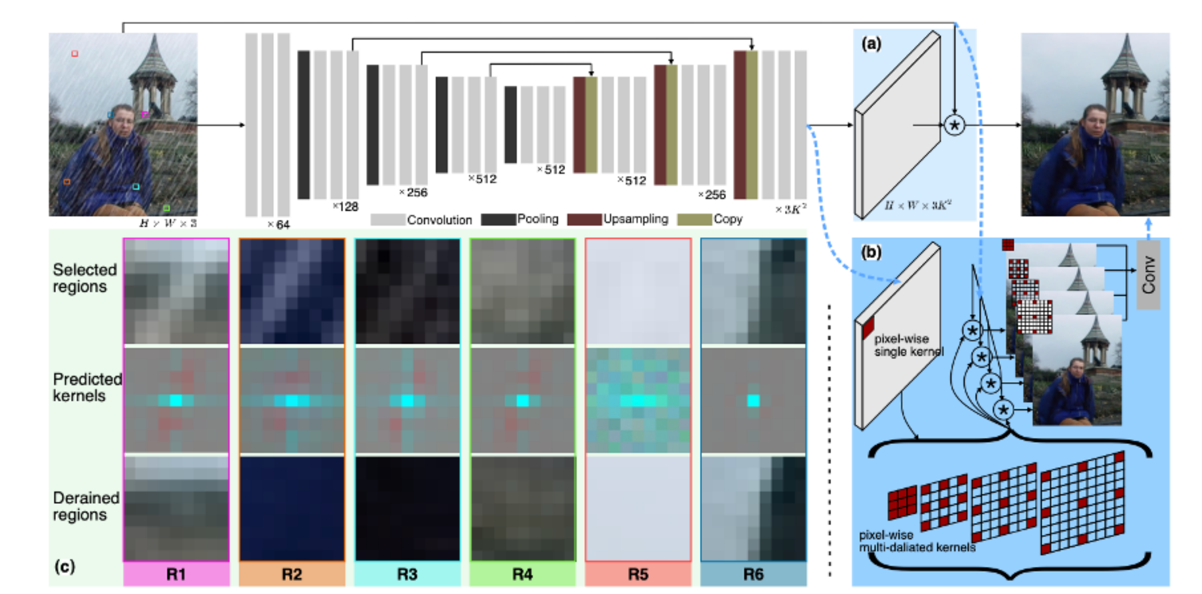

Figure 1: Pipeline of EfficientDeRain. (a) represents the pixel-wise image filtering introduced in Sec. 3.1 where each pixel is processed by an exclusive kernel predicted by a kernel prediction network. (b) we further extend this simple structure and propose the pixel-wise dilation filtering in Sec. 3.2 to handle multi-scale rain streaks. (c) we show that the predicted kernels can be adapted to different image contents while recovering the object boundary in the original image.

not always necessary for effective deraining and can sometimes hinder performance. This method achieves high-efficiency deraining in a single-stage process without the need for progressive refinement or iterative optimization.

The key contributions of our study are:

- Introducing pixel-wise dilation filtering, where rainy images are filtered using pixel-specific kernels predicted by a kernel prediction network.
- Presenting RainMix, an effective data augmentation technique for training networks to handle real rainy images, narrowing the gap between synthetic and real data.
- Demonstrating the advantages of our approach on synthetic and real-world rainy datasets, achieving efficient and high-performance deraining. EfDeRain, operates significantly faster than the state-of-the-art RCDNet while maintaining similar performance metrics, and outperforms RCDNet in displaying real rainy images when combined with RainMix.

3. Related work (Literature review)

Contemporary deraining algorithms are generally classified into two types: single-image deraining and video-based deraining, with the former being the focus of this discussion. This section reviews the key deep learning-based single-image deraining methods that are pertinent to our study.

For single-image deraining, Wang et al. (2020a) [9] propose the Rain Convolutional Dictionary Network (RCD-Net). This network employs the proximal gradient method as its optimization strategy, encoding rain shapes using an intrinsic convolutional dictionary learning mechanism. RCD-Net iteratively addresses the deraining problem in stages, alternating between updating the background layer and the rain map through two sub-nets (M-net and B-net) at each stage.

Du et al. (2020) [2] suggest the Conditional Variational Image Deraining (CVID) network, leveraging a conditional variational auto-encoder (CVAE) architecture. The CVID network utilizes a variational approach, allowing the trained decoder to generate multiple predictions of derained images from an input rainy image. The final single-image output is derived by integrating these forecasts.

Yang et al. (2019) [12] introduce a multi-task network, Joint Rain Detection and Reconstruction (JORDER), to address the inverse single-image deraining challenge. JORDER concurrently learns three elements: the rain streak layers, the binary rain-streak map, and the clean background. Contextual dilated networks are employed to capture local contextual information, enhancing resistance to rain streaks. The network employs a recurring approach to gradually process rain streaks of various directions and shapes.

Wang et al. (2019) [10] propose the Spatial Attentive Network (SPANet), based on a two-round four-directional RNN architecture. SPANet utilizes four spatial attentive blocks to progressively identify rain streaks, complemented by conventional residual blocks for feature extraction. Additional residual blocks reconstruct a clean background by utilizing the learned negative residuals.

Overall, these methods often necessitate specific prior knowledge of rain streaks or significantly depend on a supposed rain model to formulate their algorithms. Many are based on a progressive or recurrent framework, enabling early-stage outcomes to be refined in subsequent deraining stages. However, this study posits that such stringent requirements and limitations may render the deraining process less effective and generalizable, complicating its real-world application.

4. Proposed work

0.4.1 Pixel-wise Image Filtering for Deraining

We introduce a model-independent deraining technique based on image filtering in this section. Rain typically leads to image degradation effects like motion blur, occlusion, and fog. Addressing it with image filtering methods, known for their efficiency and effectiveness in diverse degradations, is logical. This approach utilizes pixel-wise filtering for an input rainy image Ir $\in R^{H\times W}$.

$$\hat{\mathbf{I}} = \mathbf{K} \circledast \mathbf{I}^{\mathrm{r}},$$

Pixel-wise filtering involves handling each pixel with a unique kernel, and K$\in R^{H\times W\times K\times K}$ contains all pixel kernels, with Ir $\in R^{H\times W}$ being the estimated derained image. For deraining the p-th pixel of Ir, the exclusive kernel at the p-th point in K, denoted as Kp $\in R^{K\times K}$, is used. Here, p represents the pixel's 2D coordinates. We forecast the rained pixel as follows, where t ranges between (- K/2, -K/2) and (K/2,K/2).

$$\hat{\mathbf{I}}(\mathbf{p}) = \sum_{\mathbf{t},\mathbf{q}=\mathbf{p}+\mathbf{t}} \mathbf{K}_p(\mathbf{t})\mathbf{I}^{\mathrm{r}}(\mathbf{q}),$$

Addressing the following challenges is essential for effective deraining with pixel-wise filtering: efficiently estimating spatially-variant, scale-variant, and context-aware kernels, as rain can cause different effects like fog, blur, and streak occlusion in various parts of an image. For instance, rain streaks may vary in size, orientation, and transparency across the image and could be semantically related to certain image aspects, such as scene depth (Hu et al., 2019) [5].

Therefore, pixel-wise kernels must accommodate the spatial and scale variances of rain streaks and scene information. Given these requirements, hand-crafted kernel designs are impractical. In Sect. 4.2, we introduce a multi-dilated-kernel prediction network, employing a deep neural network (DNN) to predict multi-scale kernels for each pixel from a rainy image input. This approach also aims to bridge the gap between synthetic and actual rain for training a robust deraining DNN.

0.4.2 Learnable Pixel-wise Dilation Filtering

Network for kernel prediction- Inspired by recent advances in image denoising (Bako et al., 2017 [1]; Mildenhall et al., 201 [7]), we propose using the wet image as input to estimate the pixel-wise kernels K for deraining. KPN() signifies a network similar to UNet, and its architecture is depicted in Fig. 1. The kernel prediction network is trained offline on rainy-clean image pairs, allowing it to predict spatially variant kernels that adapt to the varying thickness and intensity of rain streaks while preserving object boundaries. As shown in Fig. 1, our method effectively removes rain streaks and recovers occluded borders in various regions, demonstrating the predicted kernels' ability to discern rain streak locations.

Algorithm 1: Learning EfficientDeRain via RainMix

Input: KPN$(\cdot)$, fusion Convolution Conv$(\cdot)$, Loss function $\mathcal{L}$, Rainy Images $\mathcal{I}^{\mathrm{r}}$, Clean Images $\mathcal{I}$, real rain streak set $\mathcal{R}$, Operation set $\mathcal{O} = \{\mathrm{rot}, \mathrm{shear}_{x/y}, \mathrm{trans}_{x/y}, \mathrm{zoom}_{x/y}\}$.
Output: Pre-trained Network KPN$(\cdot)$ and Conv$(\cdot)$.

Function `RainMix` $(\mathcal{R})$ **:**
- Sample a rain map $\mathbf{R}^{\mathrm{org}} \sim \mathcal{R}$;
- Initialize an empty map $\mathbf{R}^{\mathrm{mix}}$;
- Sample mixing weights $(w_1, w_2, w_3, w_4) \sim$ Dirichlet;
- **for** $i = 1$ to 4 **do**
  - Sample operations $(\mathrm{o}_1, \mathrm{o}_2, \mathrm{o}_3) \sim \mathcal{O}$;
  - Combine via $\mathrm{o}_{12} = \mathrm{o}_2\mathrm{o}_1$ and $\mathrm{o}_{123} = \mathrm{o}_3\mathrm{o}_2\mathrm{o}_1$;
  - Sample $\mathrm{o} \sim \{\mathrm{o}_1, \mathrm{o}_{12}, \mathrm{o}_{123}\}$;
  - $\mathbf{R}^{\mathrm{mix}} += w_i \mathrm{o}(\mathbf{R}^{\mathrm{org}})$
- Sample a blending weight $w \sim$ Beta;
- **return** $\mathbf{R} = w\mathbf{R}^{\mathrm{org}} + (1-w)\mathbf{R}^{\mathrm{mix}}$;

End function;

for $i = 1$ to $|\mathcal{I}^{\mathrm{r}}|$ **do**
- Generate rain map via $\mathbf{R} =$ `RainMix`$(\mathcal{R})$;
- Sample an image pair via $(\mathbf{I}^{\mathrm{r}}, \mathbf{I}) \sim (\mathcal{I}^{\mathrm{r}}, \mathcal{I})$;
- Sample $\mathbf{X} \sim \{\mathbf{I}^{\mathrm{r}}, \mathbf{I}\}$ and Perform $\mathbf{I}^{\mathrm{rm}} = \mathbf{R} + \mathbf{X}$;
- Predict pixel-wise kernels via $\mathbf{K} = \mathrm{KPN}(\mathbf{I}^{\mathrm{rm}})$;
- Derain via Eq. 4 and get $\hat{\mathbf{I}} = \mathrm{Conv}(\{\hat{\mathbf{I}}_l\})$;
- Calculate Eq. 5 and do back-propagation;
- Update parameters of KPN$(\cdot)$ and Conv$(\cdot)$;

$$\mathbf{K} = \mathrm{KPN}(\mathbf{I}^{\mathrm{r}}), \tag{3}$$

The network assigns higher weights to non-rainy pixels and lower ones to rainy pixels, validating our approach's effectiveness. For image fusion and multi-dilated filtering, we leverage the concept of dilated convolution to expand each predicted kernel to three scales. This enables handling multi-scale rain streaks without losing efficiency. When rain streaks cover a large portion of the image, a larger scale kernel can efficiently derain using related distant pixels. Direct multi-scale kernel prediction is straightforward but requires more resources. As an alternative, we expand pixel-wise filtering to convolution layers through pixel-wise dilation filtering.

$$\hat{\mathbf{I}}_l(\mathbf{p}) = \sum_{\mathbf{t},\mathbf{q}=\mathbf{p}+l\mathbf{t}} \mathbf{K}_p(\mathbf{t})\mathbf{I}^{\mathrm{r}}(\mathbf{q}), \tag{4}$$

Here, l is the dilation factor used to adjust the filter's application range. We consider four scales: l=1,2,3,4. Applying Equation 4 yields four derained images (I1,I2,I3,I4), which are then combined using a 3×3 convolution layer to produce the final result. Function loss: For network training, we employ L1 and SSIM loss functions. Assuming the clean image I as ground truth and the derained image I hat, the loss function is as follows, with λ=0.2 for all experiments.

$$\mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}) = \|\hat{\mathbf{I}} - \mathbf{I}\|_1 - \lambda \, \mathrm{SSIM}(\hat{\mathbf{I}}, \mathbf{I}) \tag{5}$$

where we fix $\lambda = 0.2$ for all experiments.

0.4.3 RainMix: Bridging the Gap to Real Rain

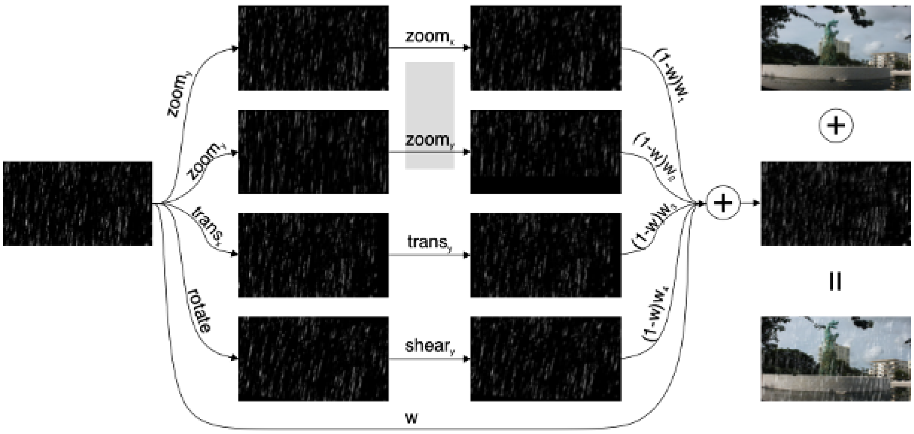

Figure 2: An example of RainMix to generate a rainy image.

Closing the gap between artificially generated rainy visuals and real-world scenarios remains a challenge. This section introduces RainMix, an innovative approach to address this issue. Garg and Nayar (2006) [4] conducted extensive research on the appearance of rain streaks, creating a dataset of real rain streaks under varying lighting and viewing conditions. However, it's challenging to assert that this dataset exhaustively covers all real-world rain scenarios due to the influence of natural elements like wind and light refraction.

Algorithm 1 illustrates our RainMix-based learning algorithm. To generate a new rain image for training, RainMix creates a rain map at each training iteration, which is added to clean or rainy images. It randomly selects a rain map from the real rain streak dataset (Garg and Nayar, 2006) [4] and performs three transformations on the rain map using randomly chosen operations. The transformed rain maps are then blended using Dirichlet distribution weights, followed by combining with the initial rain map using Beta distribution weight. This process intuitively simulates various real-world rain appearances. Figure 2 showcases an example of using RainMix to create a rainy image.

5. Experimental Discussion

0.5.1 Setups

Datasets. We perform comparison and analysis tests on two well known datasets, including the synthetic Rain1400 dataset (Fu et al. 2017) [3] and the Rain100H (Yang et al. 2017, 2019) [12] datasets, in order to thoroughly validate and assess our approach.

Metrics. For every dataset, we apply the widely used quantitative evaluation metrics of structural similarity (SSIM) and peak signal to noise ratio (PSNR). Better deraining outcomes are generally indicated by bigger PSNR and SSIM values.

Baseline. In order to be as thorough as possible, we conduct an evaluation in order to compare with a total of nine state-of-the-art deraining methods, i.e., nine baselines for the derain streak task (removing rain streak). These baselines include the following: rain convolutional dictionary network (RCDNet) (Wang et al. 2020a [9]), joint rain detection and removing (JORDERE) (Yang et al. 2019 [12]), spatial attentive deraining method (SPANet) (Wang et al. 2019 [10]), progressive image deraining network (PReNet) (Ren et al. 2019 [8]), semi-supervised transfer learning for rain removal (SIRR) (Wei et al. 2019 [11]), recurrent squeeze-and-excitation context aggregation net (RESCAN) (Li et al. 2018b [6]), deep detail network (Fu et al. 2017 [3]), and Clear (Fu et al. 2017 [3]).

0.5.2 Comparison on Rain100H and Rain1400 Dataset

In an evaluation across the Rain100H and Rain1400 datasets, our approach, named EfDeRain, demonstrates superior performance in image quality metrics. On the Rain100H dataset, known for its densely rain-streaked images, EfDeRain presents competitive PSNR and SSIM scores, closely rivaling the top-performing method RCDNet, affirming its robustness in severe weather conditions. Specifically, it garners a remarkable SSIM score of 0.967, reflecting high structural similarity to the ground truth. Similarly, on the Rain1400 dataset, where rain is less pervasive, EfDeRain maintains high performance with a PSNR of 31.8 and an SSIM of 0.9547, showcasing consistent effectiveness across different rain intensities. This performance is indicative of EfDeRain's adeptness at not only removing rain but also preserving and accentuating fine image details, such as textures and edges, which is critical in scenarios where maintaining image integrity is paramount. The results substantiate that EfDeRain can adeptly differentiate between rain and intrinsic image features, thereby applying tailored kernel predictions to each pixel for optimal rain removal and detail preservation.

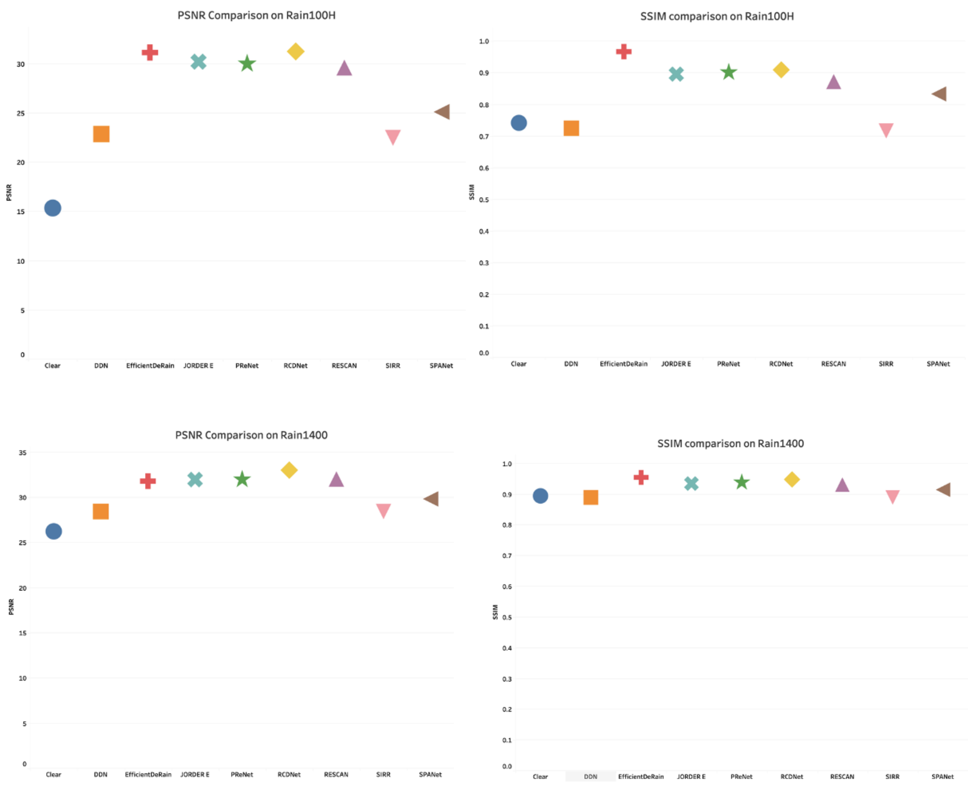

Figure 3: Comparing method EfDeRain with nine baseline methods on Rain100H (Yang et al. 2017, 2019) and Rain1400 (Fu et al. 2017) with PSNR and SSIM

6. Ablation Study

In this subsection, we scrutinize the impact of integrating RainMix into our training regimen, initially on the Rain100H dataset. Our analysis contrasts the performance of two advanced iterations of our approach: version four (v4), which incorporates RainMix, against version three (v3), which proceeds without it. The accompanying figures offer a visual assessment of both versions, with particular attention to the deraining efficacy and detail preservation.

Our findings are as follows: Both v3 and v4 demonstrate proficient rain removal; however, v4 distinctly outperforms v3 in safeguarding image details, as indicated by its elevated PSNR and SSIM metrics. This disparity underscores the significance of RainMix in training, enhancing the model's discernment between rain interference and integral image elements.

For instance, the comparison showcases that v4 not only more effectively eradicates rain but also better retains and accentuates crucial details within the scene. The textural clarity of the animals in the second scene and the architectural nuances in the third scene are appreciably more pronounced in v4 compared to v3, supporting the assertion that RainMix contributes positively to both deraining and overall image fidelity.

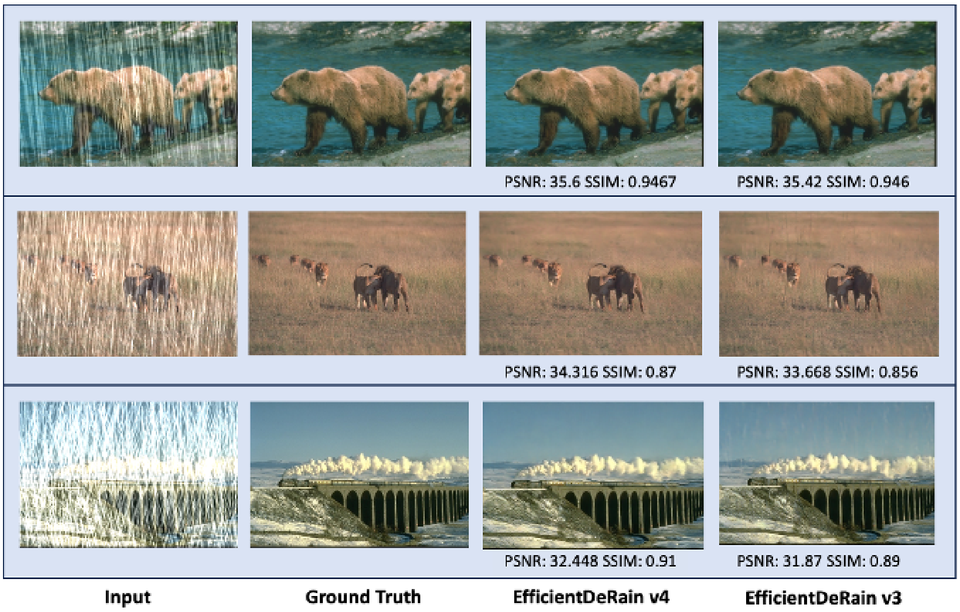

Figure 4: Rain-100H: Three visualization results of four variants of EfDeRain: v3 and v4 have the same structure but v4 uses RainMix for training.

In our next analysis, we explore the incremental advancements achieved by integrating enhancements into EfficientDeRain, as applied to the Rain1400H dataset. Our focus is on the nuanced differences between two iterations of our model: version four (v4), embodying the latest refinements, versus version three (v3), the preceding benchmark. The images provided enable a discerning visual comparison, with an emphasis on the capacity for rain streak removal and detail retention.

Our comparative analysis of the EfficientDeRain model on the Rain1400H dataset reveals a nuanced advancement from version three (v3) to version four (v4). While both iterations adeptly remove rain streaks, v4 offers a discernible, albeit modest, improvement in detail preservation, as indicated by the slightly elevated PSNR and SSIM metrics. This consistent enhancement across various scenes, including the enhanced textural sharpness in the animal scene and the more intricate portrayal of the coastal landscape, confirms v4's refined ability to differentiate between rain disturbances and integral image features. Although the improvements are not revolutionary, they affirm the progressive refinement inherent in v4, contributing to better deraining efficacy and overall image quality.

These examinations confirm that the strategic implementation of RainMix in training our model equips it with heightened adaptability and robustness, preparing it to tackle a wide array of rain conditions, thereby solidifying its position as an effective tool for image restoration in diverse and challenging weather scenarios.

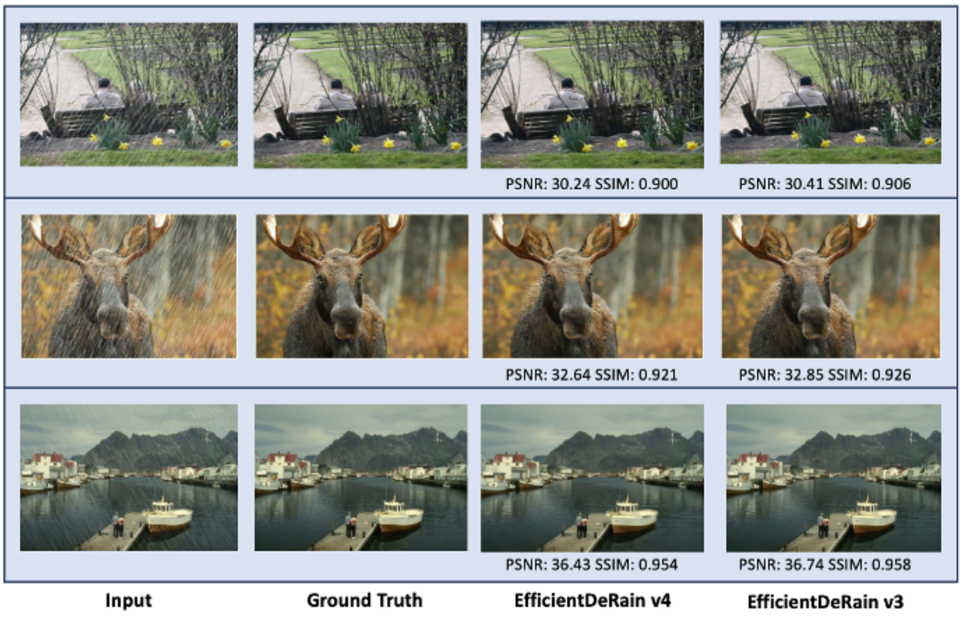

Figure 5: Rain-1400: Three visualization results of four variants of EfDeRain: v3 and v4 have the same structure but v4 uses RainMix for training.

Dataset	Version	Value
rain_1400	V3 PSNR	32.29
	V3 SSIM	0.956
	V4 PSNR	31.81
	V4 SSIM	0.954
rain_100H	V3 PSNR	30.35
	V3 SSIM	0.961
	V4 PSNR	31.12
	V4 SSIM	0.967

Table 1: Ablation Study on Rain100H and Rain1400

6. Application of the Proposed Method

Utilising the EfficientDeRain technique for single-image deraining demonstrates noteworthy progress in image processing. It sets itself apart by using a network that painstakingly examines and processes every pixel using a range of multi-scale, pre-trained kernels. This focused method makes it possible to accurately exclude rain from photos, which is a frequent hindrance in many computer vision applications.

Experimented on demanding datasets such as Rain100H and Rain1400, EfficientDeRain has proven to be a reliable method for removing rain from photos while maintaining important features necessary for further image processing applications. The use of the RainMix strategy during training empowers the network to handle diverse rain patterns, ensuring adaptability across various real-world scenarios.

This method is particularly beneficial for applications where visual clarity is crucial, such as in autonomous vehicle systems, outdoor surveillance, and digital photography enhancement. By improving the reliability of visual information under inclement weather conditions, EfficientDeRain holds the potential to revolutionize image-based systems and technologies.

7. Conclusion

In this report, we present our implementation of the model-free deraining approach known as EfficientDeRain. Through our execution, we observed that the method not only exhibits remarkable performance but also operates with a high degree of efficiency compared to leading methods. This implementation showcases two key aspects that contribute to its performance: Firstly, the innovative pixel-wise dilation filtering, which processes each pixel with multi-scale kernels derived from a pre-trained kernel prediction network. Secondly, we adopted the RainMix data augmentation technique, effectively narrowing the divide between synthetic and real-world data. Our comprehensive evaluation on widely recognized synthetic datasets, specifically Rain100H and Rain1400, confirms the method's strengths in both efficiency and deraining capability.

Bibliography

[1] S. Bako, T. Vogels, B. McWilliams, M. Meyer, J. Novák, A. Harvill, P. Sen, T. DeRose, and F. Rousselle. Kernel-predicting convolutional networks for denoising monte carlo renderings. *ACM Transactions on Graphics (TOG)*, 36(4):97, 2017.

[2] Y. Du, J. Xu, X. Zhen, M.-M. Cheng, and L. Shao. Conditional variational image deraining. *IEEE Transactions on Image Processing*, 2020.

[3] 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*, 26(6):2944–2956, 2017.

[4] K. Garg and S. K. Nayar. Photorealistic rendering of rain streaks. In *ACM SIGGRAPH 2006 Papers*, pages 996–1002. ACM, 2006.

[5] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng. Depth-attentional features for single-image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8022–8031, 2019.

[6] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 254–269, 2018.

[7] B. Mildenhall, J. T. Barron, J. Chen, D. Sharlet, R. Ng, and R. Carroll. Burst denoising with kernel prediction networks. In *CVPR*, pages 2502–2510, 2018.

[8] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[9] H. Wang, Q. Xie, Q. Zhao, and D. Meng. A model-driven deep neural network for single image rain removal. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3103–3112, 2020.

[10] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12270–12279, 2019.

[11] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu. Semi-supervised transfer learning for image rain removal. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[12] W. Yang, R. T. Tan, J. Feng, Z. Guo, S. Yan, and J. Liu. Joint rain detection and removal from a single image with contextualized deep networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 42(6):1377–1393, 2019.