IMPROVING IMAGE DE-RAINING USING REFERENCE-GUIDED TRANSFORMERS

Zihao Ye[1], *Jaehoon Cho*[2], *Changjae Oh*[1]

[1]School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2]Hyundai Motor Company, Seoul, Korea

ABSTRACT

Image de-raining is a critical task in computer vision to improve visibility and enhance the robustness of outdoor vision systems. While recent advances in de-raining methods have achieved remarkable performance, the challenge remains to produce high-quality and visually pleasing de-rained results. In this paper, we present a reference-guided de-raining filter, a transformer network that enhances de-raining results using a reference clean image as guidance. We leverage the capabilities of the proposed module to further refine the images de-rained by existing methods. We validate our method on three datasets and show that our module can improve the performance of existing prior-based, CNN-based, and transformer-based approaches.

Index Terms— Image de-raining, transformers

1. INTRODUCTION

Image de-raining is an essential task in computer vision as rain streaks can decrease visibility and deteriorate the robustness of most outdoor vision systems. De-raining has been widely applied in a wide range of practical applications, including autonomous driving [1, 2] and surveillance systems [3, 4], as an essential pre-processing step.

Early approaches that solve the task with hand-crafted priors such as sparse coding [5] and Gaussian mixture model [6] are formulated to explicitly model the physical characteristics of rain streaks. However, they often fail under complex rain conditions and show over-smoothed images [7]. The advent of Convolutional Neural Networks (CNNs) has led to substantial advances in single image de-raining [8, 9, 10, 11, 12, 13]. CNNs, however, have limited receptive fields, which means that the pixel value estimation for each spatial location primarily depends on small local surroundings. Therefore, due to the limited capacity for modeling long-range spatial context [14], CNN-based methods often struggle with accurately detecting heavy rain streaks, leading to blurred results [15]. Transformer-based methods [16, 17, 18, 19, 20] have emerged as a promising alternative as they can better capture non-local information, enhancing image reconstruction quality. However, these approaches do not model local image details well, which are crucial for achieving clear image restoration [19].

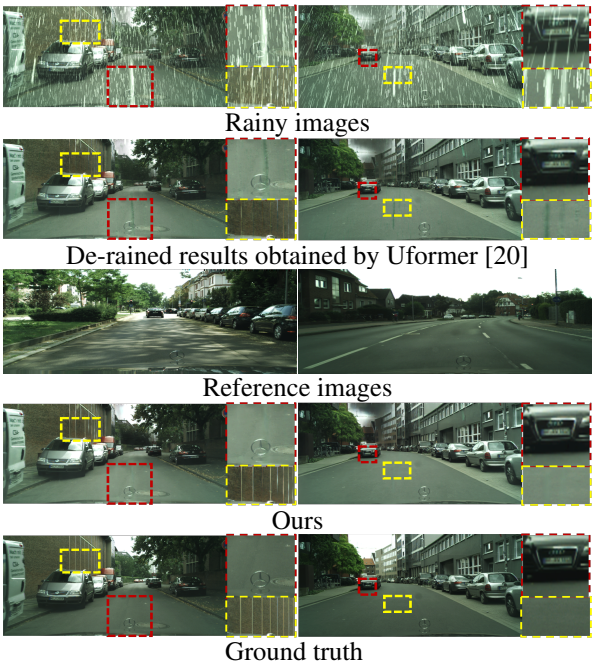

Fig. 1. Sample de-rained images from Cityscapes-Rain [21]. Unlike existing methods, our reference-guided de-raining filter enhances the de-rained results using a reference clean image as guidance.

This limitation arises from the self-attention mechanism in Transformers, which does not adequately handle the local invariant properties, in contrast to CNNs. While single image de-raining methods have made significant progress, there remains room for improvement in their ability to handle diverse and challenging rain conditions.[1]

In this paper, we propose a novel framework for image de-raining. We use existing de-raining models as baselines and present a reference-guided de-raining filter that extracts useful feature information from a reference clean image to compensate for the baseline results. The key insight is to transfer useful features from a reference clean image. Our framework consists of a feature extractor, a feature attention module, and a feature fusion module. Given a rainy image and a reference rainy image as input, we first estimate the de-rained images

[1]More results and the code: `http://ziiihooo.com/blog/2024/derain/`

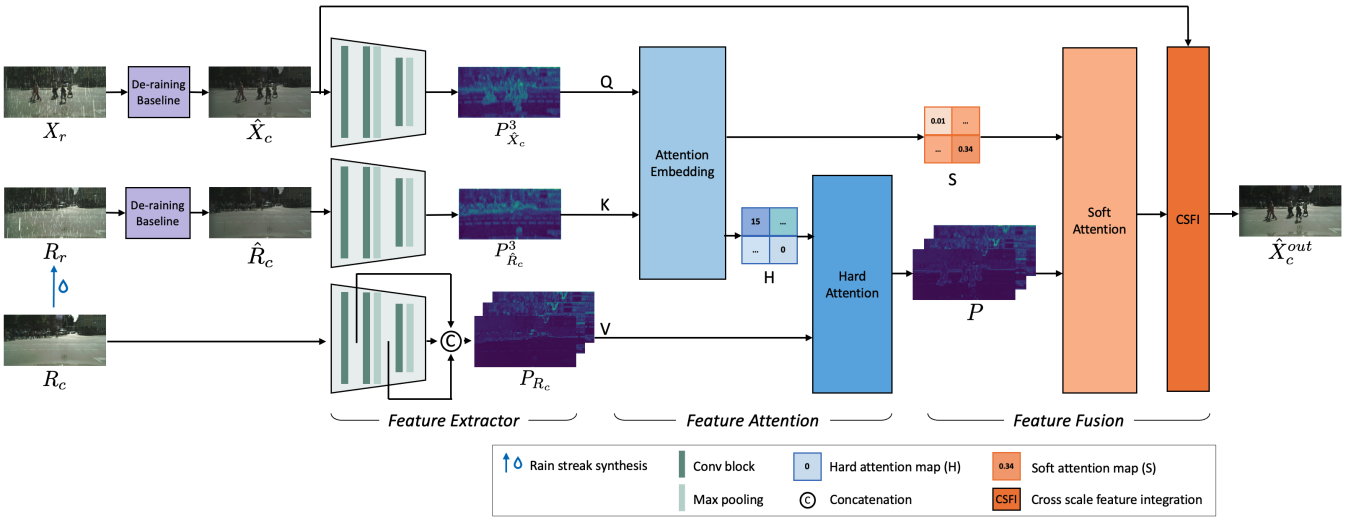

Fig. 2. Overview of our framework. We first obtain a input rainy image, X_r, and a synthesized reference rainy image, R_r. Using an existing de-raining model, we obtain the input de-rained image, $\hat{X}_c$, and the reference de-rained image, $\hat{R}_c$. These two de-rained images and the reference clean image, R_c are used as input to our reference-guided de-raining filter. By capturing the useful information from the features from R_c, and transferring it to $\hat{X}_c$, we can generate the enhanced de-raining output, $\hat{X}_c^{out}$.

using an existing de-raining model. We use these results with a reference clean image as input for the feature extractor that extracts the multi-scale features of each image. The feature attention module uses these features as input to estimate useful features from a reference rain/clean image. At this point, the feature attention module computes the most relevant feature patch from the reference clean image. Finally, we introduce the feature fusion module to aggregate multi-scale features.

We summarize our main contributions as follows:

- We propose a novel framework to integrate existing de-raining methods into a reference-guided de-raining filter that captures useful features by leveraging reference images.
- Our method can be used with a wide range of existing methods in a plug-and-play manner.
- Experimental results show that our method improves the performance of existing methods, from a prior-based to a state-of-the-art method.

2. PROBLEM FORMULATION

Given an input rainy image X_r, existing learning-based single image de-raining methods aim to learn a model $f_{\theta_S}(\cdot)$, parameterized by θ_S, that can generate an estimated clean image $\hat{X}_c = f_{\theta_S}(X_r)$ in which the rain streaks are removed. The model $f_{\theta_S}(X_r)$ is learned by minimizing the error between $\hat{X}_c$ and the ground-truth clean image X_c^{gt} using a loss function $\mathcal{L}$ as:

$$\arg\min_{\theta_S} \mathcal{L}(f_{\theta_S}(X_r), X_c^{gt}). \quad (1)$$

In this paper, we propose to further employ a reference clean image R_c as guidance to improve the result of existing de-raining models and generate the enhancement output $\hat{X}_c^{out}$. We present a model $g_{\theta_R}(\cdot)$, parameterized by θ_R, that aims to extract useful feature information from R_c to compensate for $\hat{X}_c$, resulting in the final enhancement output $\hat{X}_c^{out} = g_{\theta_R}(\hat{X}_c, R_c)$. Namely, our model learns to minimize the following loss function:

$$\arg\min_{\theta_R} \mathcal{L}(g_{\theta_R}(\hat{X}_c, R_c), X_c^{gt}). \quad (2)$$

3. METHOD

Figure 2 shows the overview of our framework. The proposed reference-guided de-raining filter (RDF), $g_{\theta_R}(\cdot)$, is designed to extract useful feature information from R_c to compensate for $\hat{X}_c$ obtained from an existing baseline model. Given $\hat{X}_c$, we first perform image retrieval to find R_c from an image database. We then obtain the synthesized rainy image R_r by synthesizing the rain streaks to R_c [22]. By using R_r as input to the baseline model, we can estimate the reference de-rained image $\hat{R}_c$. By capturing the similarity between the two de-rained images, $\hat{X}_c$ and $\hat{R}_c$, RDF aims to transfer the useful information from a reference clean image R_c to $\hat{X}_c$, generating the enhancement output $\hat{X}_c^{out}$.

RDF mainly consists of three components: feature extractor, feature attention and feature fusion. The feature extractor first projects the images $\hat{X}_c$, $\hat{R}_c$, and R_c into the features

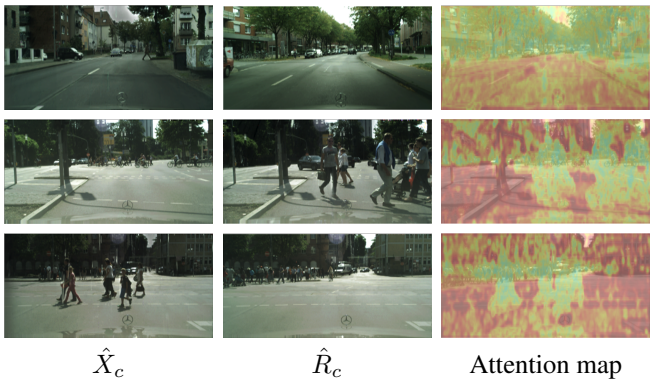

Fig. 3. Attention maps from the feature attention module. Using the feature of a input de-rained image, $\hat{X}_c$, as a query, and the feature of a reference de-rained image, $\hat{R}_c$, as a key, we compute attention weights that are utilized to select the useful features from the reference image. The attention maps are color-coded, where warmer colors indicate higher values.

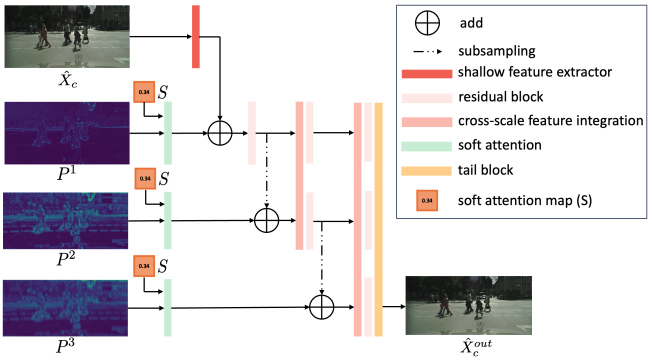

Fig. 4. Feature fusion module. The de-rained images are first projected into the feature space using a shallow feature extractor. The features at each level are then compensated sequentially from level 1 (fine-level) to Level 3 (coarse-level).

$P_{\hat{X}_c}$, $P_{\hat{R}_c}$, and P_{R_c}, respectively. The feature attention module firstly computes the attention weights taking $P_{\hat{X}_c}$ as query (Q) and $P_{\hat{R}_c}$ as key (K). Attention embedding module outputs the highest relevance as soft attention maps S at patch level and indexes corresponding to highest relevance at patch level as hard attention maps H. In the hard attention module, H is further used to select the most relevant patch from the paired value (V), P_{R_c}, estimating the feature P, the useful feature extracted from the reference clean image. In the feature fusion module, P is re-weighted at the patch level using the soft attention maps S. The re-weighted feature is then integrated with the de-rained image through the Cross-Scale Feature Integration (CSFI) stage. Finally, the fused feature is back-projected into the image space to produce the enhancement output $\hat{X}_c^{out}$.

3.1. Feature extractor

The feature extractor module, including several convolution blocks, is designed to project images into a feature space. Specifically, the feature extractor maps a single image into three distinct feature levels. The Level-1 feature, $P^1_{\{\cdot\}} \in \mathbb{R}^{(B,C,H,W)}$, preserves the input image size but with a higher-dimensional channel space C. The Level-2 and Level-3 features are represented at lower resolutions and increased channel dimensions, specifically denoted as $P^2_{\{\cdot\}} \in \mathbb{R}^{(B,2C,H/2,W/2)}$ and $P^3_{\{\cdot\}} \in \mathbb{R}^{(B,4C,H/4,W/4)}$, respectively. In this context, the feature $P_{\{\cdot\}} \triangleq \{P^1_{\{\cdot\}}, P^2_{\{\cdot\}}, P^3_{\{\cdot\}}\}$ aggregates these three-level features extracted from an input of the feature extractor.

3.2. Feature attention

The feature attention module receives three different features, $P_{\hat{X}_c}$, P_{R_c}, and $P_{\hat{R}_c}$. This module maps the query $P_{\hat{X}_c}$ to useful feature patches in the context of the key $P_{\hat{R}_c}$ and value P_{R_c}, subsequently outputting the useful features P along with the corresponding relevance maps at the patch level. During the attention embedding stage, only the Level-3 features $P^3_{\hat{X}_c}$ and $P^3_{\hat{R}_c}$ are utilized to compute relevance for patches, as they encapsulate more abstract information and a wider receptive field. Following this, the hard attention module selects the most relevant feature patch of P_{R_c} for the patches of $P_{\hat{X}_c}$, using the relevance just computed at every feature level. In Figure 3, we visualize the attention maps, where we could see that similar areas are highly noticed, which are used to compensate for de-rain results later.

3.3. Feature fusion

In Figure 4. the CSFI module represents a well-established method for blending features with various scales [23, 24]. In this paper, we leverage this approach to fuse P, the useful features extracted from the reference, with the original de-rained image. The useful feature is incrementally added to the original de-rained image $\hat{X}$, guided by relevance maps. During the compensation stage, the CSFI module is employed in conjunction with the residual block to facilitate information sharing across all feature levels. The level-1 feature, characterized by its relatively precise information and intricate details, is compensated to the image first. Conversely, the level-3 feature, which encapsulates more abstract information, is compensated to the image last. This systematic process results in a projection from the feature space to the image space, generating the final output $\hat{X}_c^{out}$.

Table 1. Quantitative evaluation on the three datasets: BDD100K-Rain, synthesized using SyRaGAN [22] and BDD100K [25], Cityscapes-Rain [21], and KITTI-Rain [21]. Experiments include a prior-based model, GMM [6], a CNN-based model, PReNet [12], and a transformer-based model, Uformer [20], and their improvement using our module highlighted in blue.

Methods	BDD100K-Rain		KITTI-Rain		Cityscapes-Rain	
Name	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
GMM [6]	28.37	0.8590	17.08	0.4818	23.333	0.7830
PReNet [12]	33.38	0.9474	22.71	0.7497	23.80	0.9529
Uformer [20]	36.30	0.9619	31.59	0.9694	23.98	0.9509
GMM + **Ours**	$31.44_{+3.07}$	$0.9003_{+0.0412}$	$25.23_{+8.15}$	$0.7933_{+0.3114}$	$23.48_{+0.14}$	$0.8869_{+0.1039}$
PReNet + **Ours**	$33.72_{+0.34}$	$0.9487_{+0.0013}$	$26.92_{+4.21}$	$0.8551_{+0.1054}$	$24.98_{+1.18}$	$0.9595_{+0.0066}$
Uformer + **Ours**	$36.37_{+0.07}$	$0.9627_{+0.0008}$	$33.05_{+1.46}$	$0.9761_{+0.0067}$	$25.64_{+1.66}$	$0.9601_{+0.0091}$

3.4. Loss

During the training phase, our approach consists of initialization and fine-tuning stages. In the initialization stage, the model is trained to transfer useful features from clean object images using an L1 loss function. The model learns the transformation from a derained to a clean image in this stage, ensuring the model has the capability to perform deraining on simple rainy images. In the fine-tuning stage, the model is further trained to transfer useful features from reference clean images to better simulate real-world application scenarios. For the independent objectives of these two stages, we employ the loss functions tailored to each stage's specific requirements. During the initialization stage, the loss function is defined as the $\ell 1$ reconstruction loss:

$$\mathcal{L}(\hat{X}_c^{out}, X_c) = \|\hat{X}_c^{out} - X_c\|_1. \tag{3}$$

In the fine-tuning stage, we apply the MS-SSIM-L1-Loss [26]:

$$\mathcal{L}(\hat{X}_c^{out}, X_c) = \alpha_1 \|\hat{X}_c^{out} - X_c\|_1 + \alpha_2 (1 - \text{SSIM}(\hat{X}_c^{out}, X_c)), \tag{4}$$

where α_1 and α_2 are the hyperparameters to control the effect of each loss.

4. VALIDATION

4.1. Experiment setup

In our framework, we use existing de-raining models as baselines. We adopted three baseline models, including the prior-based (GMM [6]), CNN-based (PReNet [12]), and transformer-based (Uformer [20]) models. Except for the prior-based method that does not require training, we used public codes for training each baseline on each dataset.

For the dataset, we require both a clean/rainy image pair and a reference clean/rainy image pair that contain similar scenes. However, existing rain benchmarks, such as Rain100L [27] and DID [28], have limited similar scenes, providing unreasonable reference images. Therefore, we constructed the dataset as follows:

- BDD100K-Rain: We used BDD100K [25], a large-scale driving scene dataset, and synthesized the rain streaks by using SyRaGAN [22] and obtained 256×256 images.
- Cityscapes-Rain/KITTI-Rain: This dataset includes 256×1024 images and constructed by [21], which renders rain streaks to evaluate bad weather.

For image retrieval, we implemented the reference image retrieval method at the image hash level. First, all images in the dataset are projected into the image perceptual hash space. Then, for every image that requires compensation, the nearest neighbor image is selected as the reference image.

We utilized 8 A100 GPUs and PyTorch for our experiments. The channel numbers for levels 1, 2, and 3 are set to 64, 128, and 256, respectively. The convolution blocks in the feature extractor module are initialized with VGG-19 [29]. At the feature attention stage, the patch sizes for levels 1, 2, and 3 are set to 12, 6, and 3, respectively. Within the feature fusion module, the features for levels 1, 2, and 3 are configured as $(64, H, W)$, $(128, H/2, W/2)$, and $(256, H/4, W/4)$, respectively. To ensure that our model can capture useful features and compensate accurately, we initially employ the input clean image X_c as the reference image for initialization training using the loss functions in (3). The initialization training is done for each model on each dataset. After the initialization training stage, X_c is replaced with the actual reference image in the subsequent training stage for fine-tuning. In the fine-tuning stage, we set $\alpha_1 = 0.6$ and $\alpha_2 = 0.4$ and use the loss function in (4).

4.2. Discussion

Table 1 presents the quantitative results of the baselines and the improvements achieved using our method. Although each baseline employs a different methodology for image deraining, our method can universally enhance the performance of these baseline models. This suggests that our reference-guided de-raining filter effectively extracts useful features from reference images. The improvement is most significant on the KITTI-Rain dataset, as this dataset provides better

Table 2. Effect of reference images. PReNet [12] trained on BDD100K-Rain [25, 22] is used as a backbone while changing the reference images to the ground truth clean image, Gaussian noise image, and our reference image obtained by image retrieval.

Reference Type	PSNR	SSIM
Ground truth	$35.50_{+2.06}$	$0.9736_{+0.0257}$
Noise image	$33.37_{-0.07}$	$0.9470_{-0.0009}$
Reference image	$33.78_{+0.34}$	$0.9491_{+0.0013}$

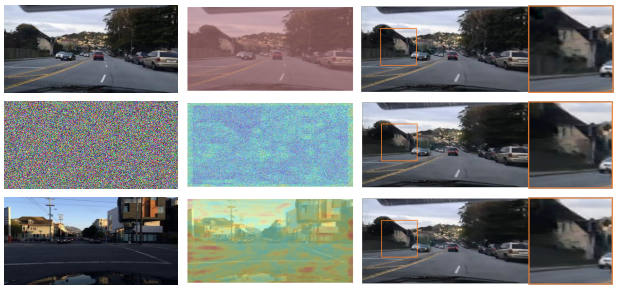

Fig. 5. Effect of reference images on the attention maps and de-raining results. De-raining images are obtained by using (from top to bottom) the ground-truth clean image, Gaussian noise, and our reference image obtained by image retrieval.

reference images. Our pipeline also demonstrates various degrees of compensation effects on different baselines. For Uformer [20], which achieves the best results among the three baseline models, our method shows relatively small improvements. On the other hand, for GMM [6], which is the earliest method among the three baseline models, our method shows more substantial improvements.

We further evaluate the effect of reference images on the BDD100K-Rain dataset, specifically to analyze how a reference image can contribute to the de-raining process. Using PReNet as a baseline model, we analyze the effect of reference images using three different image types: ground truth clean images, Gaussian-noise images, and our reference images collected using image retrieval. As shown in Table 2, ground truth images, which encapsulate all the useful information, yield the best results and significantly enhance the performance. Noise images, that include less relevant information, produce the worst outcomes. Reference images containing similar scenes provide results that fall between the upper bound set by the ground truth images and the lower bound established by the noise images. The results indicate that our model can transfer useful features from reference images, and the degree of enhancement primarily depends on the quality of the reference images.

5. CONCLUSION

This paper introduces a novel framework for image de-raining that leverages a reference-guided de-raining filter, a transformer network that enhances existing de-raining results using a reference clean image as guidance. Our framework, as plug-and-play de-raining enhancement, shows performance improvements of prior, CNN, and transformer-based models across multiple datasets. As future work, we will integrate our method with text-to-image generation models that can synthesize clean images and use these images as references for de-raining.

6. REFERENCES

[1] H. Huang, A. Yu, and R. He, "Memory oriented transfer learning for semi-supervised image deraining," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[2] Q. Guo, J. Sun, F. Juefei-Xu, L. Ma, X. Xie, W. Feng, Y. Liu, and J. Zhao, "Efficientderain: Learning pixel-wise dilation filtering for high-efficiency single-image deraining," in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.

[3] M. Li, X. Cao, Q. Zhao, L. Zhang, and D. Meng, "Online rain/snow removal from surveillance videos," *IEEE Transactions on Image Processing*, vol. 30, pp. 2029–2044, 2021.

[4] S. Li, W. Ren, F. Wang, I. B. Araujo, E. K. Tokuda, R. H. Junior, R. M. Cesar-Jr, Z. Wang, and X. Cao, "A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives," *International Journal of Computer Vision*, vol. 129, no. 4, pp. 1301–1322, 2021.

[5] L. Kang, C. Lin, and Y. Fu, "Automatic single-image-based rain streaks removal via image decomposition," *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1742–1755, 2011.

[6] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[7] J. Cho, S. Kim, and K. Sohn, "Memory-guided image de-raining using time-lapse data," *IEEE Transactions on Image Processing*, vol. 31, pp. 4090–4103, 2022.

[8] R. Li, R. T. Tan, and L. Cheong, "All in one bad weather removal using architectural search," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[9] H. Wang, Y. Wu, Q. Xie, Q. Zhao, Y. Liang, S. Zhang, and D. Meng, "Structural residual learning for single image rain removal," *Knowledge-Based Systems*, vol. 213, pp. 106595, 2021.

[10] 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*, vol. 42, no. 6, pp. 1377–1393, 2020.

[11] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, "Lightweight pyramid networks for image deraining," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 31, no. 6, pp. 1794–1807, 2020.

[12] 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/CVF Conference on Computer Vision and Pattern Recognition*, 2019.

[13] J. Cho, S. Kim, D. Min, and K. Sohn, "Single image deraining using time-lapse data," *IEEE Transactions on Image Processing*, vol. 29, pp. 7274–7289, 2020.

[14] J. Liang, J. Cao, G. Sun, K. Zhang, L. Van Gool, and R. Timofte, "Swinir: Image restoration using swin transformer," in *Proceedings of the IEEE/CVF international Conference on Computer Vision*, 2021.

[15] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin, "Non-locally enhanced encoder-decoder network for single image de-raining," in *Proceedings of the 26th ACM international Conference on Multimedia*, 2018.

[16] H. Chen, Y. Wang, T. Guo, C. Xu, Y. Deng, Z. Liu, S. Ma, C. Xu, C. Xu, and W. Gao, "Pre-trained image processing transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[17] Y. Liang, S. Anwar, and Y. Liu, "Drt: A lightweight single image deraining recursive transformer," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

[18] Y. Li, K. Zhang, J. Cao, R. Timofte, and L. Van Gool, "Localvit: Bringing locality to vision transformers," *arXiv preprint arXiv:2104.05707*, 2021.

[19] X. Chen, H. Li, M. Li, and J. Pan, "Learning a sparse transformer network for effective image deraining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2023.

[20] Z. Wang, X. Cun, J. Bao, W. Zhou, J. Liu, and H. Li, "Uformer: A general u-shaped transformer for image restoration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[21] M. Tremblay, S. S. Halder, R. de Charette, and J. Lalonde, "Rain rendering for evaluating and improving robustness to bad weather," *International Journal of Computer Vision*, vol. 129, no. 2, pp. 341–360, 2021.

[22] J. Choi, D. H. Kim, S. Lee, S. H. Lee, and B. C. Song, "Synthesized rain images for deraining algorithms," *Neurocomputing*, vol. 492, pp. 421–439, 2022.

[23] F. Yang, H. Yang, J. Fu, H. Lu, and B. Guo, "Learning texture transformer network for image super-resolution," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[24] K. Sun, Y. Zhao, B. Jiang, T. Cheng, B. Xiao, D. Liu, Y. Mu, X. Wang, W. Liu, and J. Wang, "High-resolution representations for labeling pixels and regions," *arXiv preprint arXiv:2005.09228*, 2020.

[25] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020.

[26] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on Computational Imaging*, vol. 3, no. 1, pp. 47–57, 2017.

[27] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, "Multi-scale progressive fusion network for single image deraining," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020.

[28] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

[29] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations*, 2015.