Pattern Recognition Letters 158 (2022) 24–33

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Laplacian encoder-decoder network for raindrop removal

Simone Zini*, Marco Buzzelli

Department of Informatics, Systems and Communication, University of Milano – Bicocca, Viale Sarca 336, Building U14, Milan 20126, Italy

ARTICLE INFO

Article history:
Received 19 January 2022
Revised 21 March 2022
Accepted 12 April 2022
Available online 14 April 2022

Edited by: Maria De Marsico

Keywords:
Rain removal
Raindrop removal
Image restoration

ABSTRACT

Digital raindrop removal is a branch of image restoration that aims at identifying adherent droplets on a glass surface and replacing them with plausible content. When successfully performed, raindrop removal was proven in the past to positively affect both the perceived appearance of the scene, and the performance of computer vision tasks such as semantic segmentation and object detection. In this paper, we design and implement a new encoder-decoder neural network for supervised raindrop removal. Our network, given a rainy input image, produces as output the Laplacian pyramid of a rain-free version of the input, making it possible to handle the variety of appearances of rain droplets by processing different frequency bands independently. To this end, we define and experimentally prove the effectiveness of a custom loss function that combines the errors of the different Laplacian frequency bands. We test our model for raindrop removal on a standard dataset, using multiple objective metrics to provide a detailed analysis of its performance. We confirm the superiority of our proposal in a comparison with other methods from the state of the art.

1. Introduction

Adverse weather conditions negatively impact the perceived visibility of a scene. In a street-driving scenario, for example, rain droplets adhering to the glass surface of a car windshield might occlude crucial elements such as obstacles, pedestrians, or traffic signs, and are generally distracting to the driving experience. In addition to hindering human vision, rain-induced artifacts are also found to affect computer vision: several works in the scientific literature quantify the benefits of digital rain removal on a wide variety of tasks, ranging from object detection [1], to semantic segmentation [2], to optical character recognition [3]. The problem of image regression, in its most general formulation, has been addressed with a wide variety of approaches through the years: from handcrafted solutions [4], to the more recent exploitation of Convolutional Neural Networks (CNN) [5]. The latter has involved general-purpose methods for image-to-image translation [6], as well as domain-specific architectures such as the ones described in Section 2. Compared to other artifacts such as rain streaks and rain mist, rain droplets impose a significant and specific set of challenges, such as large occlusion areas and a wide variety of appearances, as we show in Section 3. In this work we specifically focus on raindrop removal by designing a Laplacian encoder-decoder neural network. Our solution allows us to control the image reconstruction process by producing the different levels of a Laplacian pyramid decomposition of the expected clear (i.e. rain-free) image. This approach avoids relying on commonly-used attention maps which are inherently limited by misalignements between the rainy and clear image, a phenomenon observed in Alletto et al. [7]. Our model is trained with multiple losses, evaluating the partial reconstruction at each level of the pyramid. In the training procedure, we modify the derivation tree in order to prevent redundant gradient flow for pyramid levels that impact more than one loss component. This novel formulation, and its integration with the Laplacian decomposition, was found to be optimal after comparative evaluation with several other alternatives, as reported in the experimental results. Future developments of the proposed method are also suggested based on an in-depth analysis of the relationship between Laplacian levels and rain removal.

2. Related works

The digital removal of rain-induced artifacts has been actively studied through the years, resulting in an extensive scientific production. A recent review by Yang et al. [8] presented a comprehensive analysis, ranging from solutions based on explicit raindrop-appearance models [9–11], to data-driven ones typically relying on deep learning [12,13]. Wang et al. [14] produced a similar overview of existing approaches, with particular attention to rain removal in video sequences, providing direct links to papers, source codes,

* Corresponding author.
E-mail addresses: s.zini1@campus.unimib.it (S. Zini), marco.buzzelli@unimib.it (M. Buzzelli).

https://doi.org/10.1016/j.patrec.2022.04.016
0167-8655/

project pages, datasets, and metrics. In this paper, we focus on single-shot rain removal. Rain-related artifacts can be organized in three macro categories according to [1]: rain droplets, rain streaks, and rain mist. An image is said to be affected by "rain droplets" (also referred to as raindrops) when the scene is observed through wet glass: typically, a car windshield right after it rained. This particular interpretation of the problem is the one addressed in this paper, therefore an analysis of corresponding state of the art solutions is provided in the current section. The term "rain streaks" refers instead to the visual artifacts of a scene directly observed when rain is currently pouring [15–17]. In this case, the terminal velocity of falling rain produces motion-blurred rain streaks overimposed to the image. Finally, the task of rain-streak removal is often treated in conjunction with the correction of "rain-induced mist": in the same scenario, in fact, rain streaks that are far away from the camera are not individually discernible, and produce a global appearance equivalent to that of airborne water [18]. Additionally, some recent works have focused on the digital removal of snow flakes, training regression models either through adversarial techniques [19], or more traditional learning procedures [20].

The specific field of raindrop removal is relatively recent, compared to rain streak and mist removal. Wu et al. [4] developed an handcrafted approach to the problem: they focused on droplet detection, by analyzing colour, texture, and shape statistics of raindrop images. Based on these features, their solution is to produce a first set of candidate raindrop regions, which is subsequently pruned through a learning-based verification algorithm. The authors then resorted to existing image inpainting solutions in order to restore the selected image areas. A relevant contribution to the field has then been given by Qian et al. [21], who in 2018 published a high-quality dataset that has since become the *de facto* standard for this area of research. The authors also introduced a so-called "attentive generative network", trained in an adversarial configuration. They injected a visual-attention map to both the generative and discriminative component of the network, in order to focus the image processing mainly on corrupted areas. However, whenever attention maps are designed to target explicit raindrop masks (a function of the difference between rainy image and clear reference), they are inherently limited by misalignements between the two images and moving objects, as observed by Alletto et al. [7]. They consequently developed a physically-accurate computer-graphics engine to augment images with artificial raindrops. Such technique allowed them to exploit existing datasets unrelated to rain removal, in order to train a model that is able to simultaneously locate and remove raindrops in a self-supervised manner. Their solution, based on a conditional generative adversarial network, is mainly developed for application to video sequences by exploiting motion cues. Quan et al. [22] devised a so-called "double attention mechanism" to guide the learning and inference of a Convolutional Neural Network in the task of raindrop removal. Their approach relies upon the generation of a shape-driven attention map, to locate raindrops based on a-priori knowledge on their shape properties. Such attention map was applied using a channel recalibration mechanism, to properly weight the intermediate activations of their neural model. Hao et al. [23] released a dataset of images augmented with physics-based synthetic raindrops, as well as the associated raindrop masks. They defined a neural network for raindrop detection which explicitly models the refraction and blurring components of the raindrop itself. Shao et al. [24] explicitly modelled the blur level of rain droplets using a soft mask populated through an iterative procedure, and fuse it with the input image through an attention mechanism. They also exploit the a multi-scale analysis based on the observation that different scale versions of a rainy image have similar raindrop patterns. In developing our final solution, we experimented with different strategies, and eventually defined a neural network that while not relying on attention maps still produces competitive or even superior results when compared to such methods.

More specifically, our approach to the digital removal of rain droplets leverages a Laplacian decomposition of the input image, in order to address the problem at different scales. Decomposing the input image with various representations has been successfully exploited in the past for rain streak removal while not for raindrop removal. Kang et al. [25] applied an image decomposition based on morphological component analysis, specifically resorting to bilateral filtering. They decomposed the image into a low-frequency and high-frequency part, and focused on processing only the high-frequency component: they exploited dictionary learning and sparse coding to further decompose it into rain and non-rain components, in order to effectively remove the former. Similarly, Sun et al. [26] also devised an approach that relies on image decomposition for dictionary-based removal of rain streaks, but embedded and formulated the decomposition-basis selection as an optimization problem instead of exploiting bilateral filtering. Fu et al. [27] focused on reducing the computational complexity of Convolutional Neural Networks for rain streaks removal by representing the input image as a Gaussian-Laplacian pyramid, and by designing a so-called "Lightweight Pyramid Network" (LPNet) based on a recursive and residual structure.

To the best of our knowledge, this is the first time that image decomposition is exploited for raindrop removal. In Sections 3 and Section 4.3 we show that this application is particularly suitable, as the various appearances of rain droplets can be individually handled by exploiting the Laplacian decomposition.

3. Proposed method for raindrop removal

Raindrops adhering to a transparent surface in front of the camera (like a car windshield, or the camera lens itself) degrade the quality of the information contained in the picture to different extents, depending on the camera focus:

1. In-focus raindrops. The degradation appears as blur in sparse image areas, affecting low and high frequencies.
2. Out-of-focus raindrops. We mainly identify two effects:
   - A degradation related to the refraction phenomena introduced by the convex shape of the drop, that affects the low-frequencies.
   - A drop contour degradation that is manifested as artifacts in the high frequencies.

An example of in-focus and out-of-focus raindrops can be seen in the second row of Fig. 1. To model the degradation distribution over different frequencies of the input image, we exploit its Laplacian pyramid decomposition [28], whose effect is also depicted in figure.

3.1. Laplacian-based image restoration

Given an input image I_{rainy}, our encoder-decoder network G is designed and trained to generate the corresponding levels $\hat{y}_i$ of its Laplacian pyramid decomposition, free of rain artifacts:

$$\hat{Y} = \{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_N\} = G(I_{\text{rainy}}) \tag{1}$$

where $\hat{y}_N$ is the tallest level of the Laplacian pyramid, corresponding to the low frequencies component. The final recomposed output I_{derained} is then computed as:

$$I_{\text{derained}} = L_{\hat{Y}}(1) \tag{2}$$

$$L_Y(j) = \begin{cases} y_N & \text{if } j = N \\ y_j + \text{upsample}(L_Y(j+1)) & \text{otherwise} \end{cases} \tag{3}$$

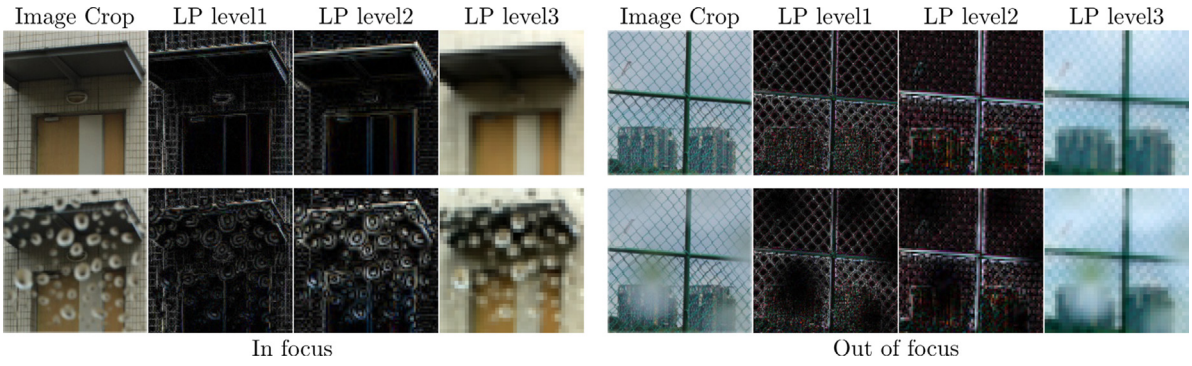

Fig. 1. Different kinds of raindrop and their impact on the overall image. The ones in camera focus tend to introduce artifacts related to the sharp edges of the single raindrops in combination with the refraction phenomenon. The ones out of focus tend to remove information where the drops are located, by blurring the corresponding image areas.

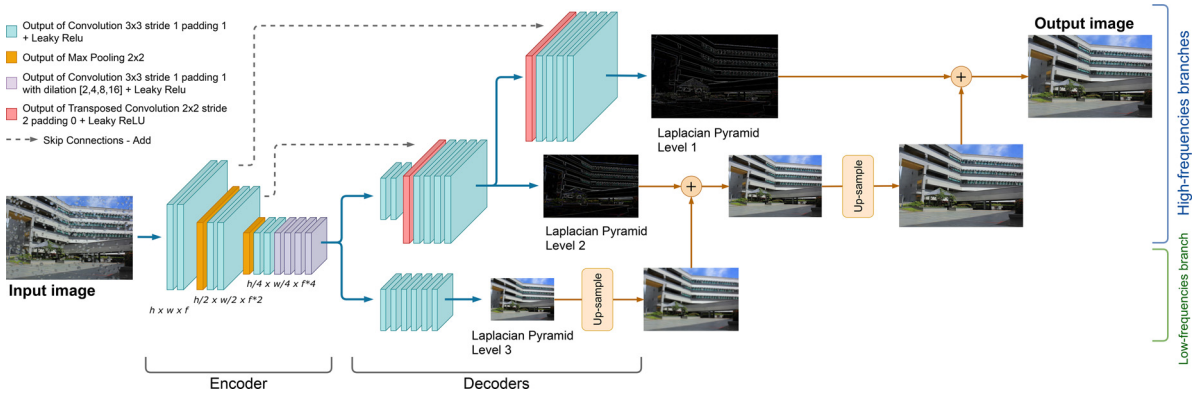

Fig. 2. Architecture of the proposed Laplacian Raindrop Removal CNN. The number of features in output after the first convolution is set to $f = 64$. The output of the different levels is combined by up-sampling the lower levels and summing them to the higher frequencies, in order to obtain the final output.

The proposed network architecture, depicted in Fig. 2, is divided in two main components: an encoder for the input I_{drop}, and a novel decoder composed of multiple output branches, in relation to the specific formulation of Laplacian pyramid levels.

The design of the encoder is partially inspired from the U-net model [29], with some relevant variations in the convolution and general structure. More specifically, the encoder is a sequence of two CONV-lReLU-CONV-lReLU layers with a MaxPool operation, to extract features and to reduce the spatial dimension. The activations are not reduced to spatial dimensions 1×1 as in the original U-Net architecture, but only reduced down by a factor of 4 (given by the presence of the two MaxPooling operation), to avoid losing spatial information in the encoded features, which serves an important role in image restoration.

The deepest part is a sequence of two CONV-lReLU blocks and four CONV-lReLU blocks with dilation [30]: these last six blocks of layers have been added with respect to the original U-Net encoder structure, to increase the model receptive field without reducing further the spatial feature dimensionality. The dilation spacing increases as a power of two from the first layer to the last one (2, 4, 8, 16). The depth of output features after the first convolution is set to $f = 64$, and the following ones are derived as indicated in Fig. 2.

The decoder, which addresses the actual restoration of the information at different frequency bands, has been designed in relation to the number N of levels of the Laplacian pyramid which we intend to reconstruct. The following description is provided for a general number of levels N, although we set $N = 3$ on the basis of preliminary experiments. Given the dimension of the images used in training and the memory limitations of our hardware configuration, we adopted $N = 3$, to not reduce too much the spatial dimensions of the images in higher levels of the laplacian pyramid and to not increase the model occupation in memory. The decoder is composed of branches of two types: one dedicated to restoration of low-frequencies, two dedicated to high-frequencies. The low-frequencies branch is a concatenation of six CONV-lReLU with a final CONV(1×1) layer with a Sigmoid activation function to map from the features space to the RGB colour space. The output of this branch corresponds to the deepest level of the Laplacian pyramid, which is a low-resolution version of the rain-free image, and which will be combined with the highest levels generated by the model according to Eq. (1). The high-frequencies branches are designed to restore the details and the fine structures in the image. The corresponding Laplacian pyramid levels all share common characteristics: values centered around zero and a general appearance that is not as intelligible as that of the lower frequencies. For this reason, the structure of this part of the model is composed of multiple sub-branches that incrementally enhance the features from the deepest to the highest level of the laplacian pyramid. Each higher branch is an extension with respect to the previous level in the Laplacian pyramid, i.e. it takes the features decoded by the preceding level to restore its own. The decoder blocks are composed of four CONV-lReLU layers plus a transposed convolution, to upsample the features for the higher Laplacian level, and a CONV(1×1) with a Tanh activation function to map from feature space to RGB colour space.

3.2. Laplacian loss function

Given a target rain-free image I_{clear}, we extract the corresponding Laplacian pyramid levels Y for comparison with the restored output $\hat{Y}$. Our loss function $Loss_d$ reconstructs the restored image

Table 1
Study on the training configuration: results achieved training the proposed model using the different loss function configurations. Evaluation performed on *test_a* from the dataset by Qian et al. [21]. Best result in bold.

Training configuration	PSNR	SSIM
a: One loss per image, classical encoder-decoder	30.14	0.9198
b: One loss per image, reconstructed image	29.76	0.9200
c: One loss per level, pyramid levels	30.56	0.9252
d: One loss per level, reconstructed levels	**31.12**	**0.9297**

up to each level, and compares it with the corresponding reconstructed target using the L1 norm ($||\cdot||_1$):

$$Loss_d = \sum_{i=1}^{N} ||L_Y(i) - L_{\hat{Y}}(i)||_1 \tag{4}$$

Instead of directly comparing the generated frequencies with the target ones, we reconstruct the image up to the specific level at which the comparison takes place. This strategy, experimentally validated in Section 4.2, is motivated by two purposes:

- Levels balance: the comparison at each branch is always performed on complete RGB images. This guarantees a magnitude of error similar between the different branches, without the necessity to re-weight to the different components of the loss for regularization purposes.
- Reconstruction context: instead of comparing images composed only of details taken out of their original context, the comparison using the reconstructed level can highlight differences in relation to the context in which the details are located. This is expected to help the training in detecting structures introduced by raindrops, in contrast to the ones coming from elements of the actual scene.

It should be noted that, with the formulation expressed in Eq. (4), the evaluation of every Laplacian level impacts all the lower levels during the gradient back-propagation. Therefore, higher levels effectively influence the learning process multiple times. In order to prevent this phenomenon, the flow of gradients during the training process has been modified, by inhibiting the back-propagation on the lower branches, and maintaining it only for the current branch.

4. Experiments

4.1. Experimental setup

To train and test our model for raindrop removal, we adopted the dataset and metodology presented by Qian et al. [21]. This dataset was collected by placing a glass panel in front of a camera, and taking pictures before and after spraying the glass with water. The dataset contains a total 1119 pairs of different outdoor scenes. The dataset is divided into three main folders: the *train* folder, containing 861 pairs of images, and two test folders: *test_b* (249 image pairs) and *test_a* (58 image pairs, a subset of well-aligned images from *test_b*). The folder *test_a* is commonly used for methods assessment and comparisons [21–23,31,32]. The folder *test_b* (without the images contained in *test_a*) is commonly used for internal validation.

We trained our encoder-decoder network with images from the *train* folder, cropped at dimension 256×256 pixels. The crops have been collected using a sliding window with overlap equal to 128 pixels, generating a total 20,664 training samples. To further augment the dataset we randomly applied online flipping and rotation (90°, 180°, 270°) to the images at training time. For validation and test, we used respectively *test_b* and *test_a* folders [21–23]. Our model is written in PyTorch v1.4.0, trained using an NVIDIA Titan V GPU with 12 GB of RAM. We adopted the Adam optimizer [33] with $\beta_1 = 0.9, \beta_2 = 0.999$ with a starting learning rate $lr = 2 \times 10^{-5}$ decreased by a factor $10\times$ after 300 epochs of training, and weight decay set to 10^{-8}.

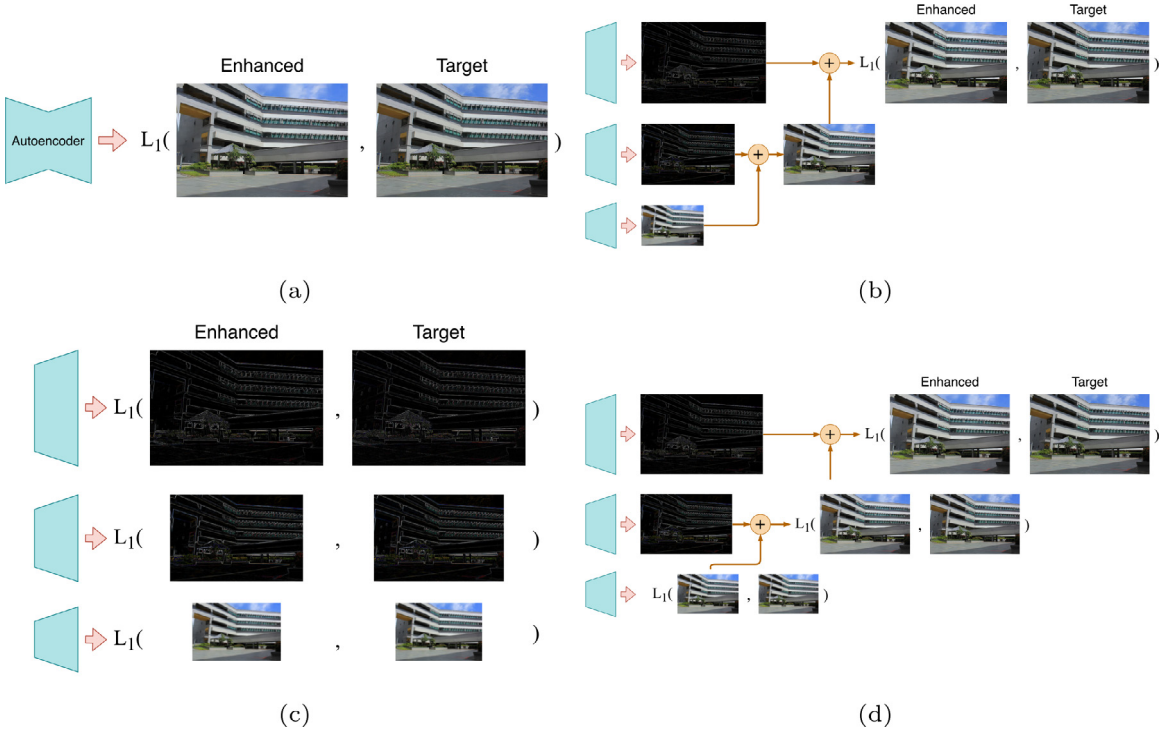

Fig. 3. We analyzed different training configurations for our encoder-decoder network: (a) comparison between the output of a classic encoder-decoder model and the target, (b) comparison between the reconstructed output and the target, (c) comparison between each level output and the corresponding target level, (d) comparison between each reconstructed level of the pyramid and the corresponding target.

Table 2
Effect of replacing each Laplacian level with its perfect ground truth version. For the "Rainy" columns, "base images" refers to the original images in *test_a* from Qian et al. [21]. For "Derained", "base images" refers to the output of our rain removal network.

	Rainy		Derained	
	PSNR	SSIM	PSNR	SSIM
Base images	24.10	0.8511	31.12	0.9297
Perfect level 1	24.61 (+2.1%)	0.9181 (+7.9%)	32.12 (+3.2%)	0.9781 (+5.2%)
Perfect level 2	24.60 (+2.1%)	0.8749 (+2.8%)	31.22 (+0.3%)	0.9402 (+1.1%)
Perfect level 3	28.95 (+20.1%)	0.8885 (+4.4%)	33.92 (+9.0%)	0.9389 (+1.0%)

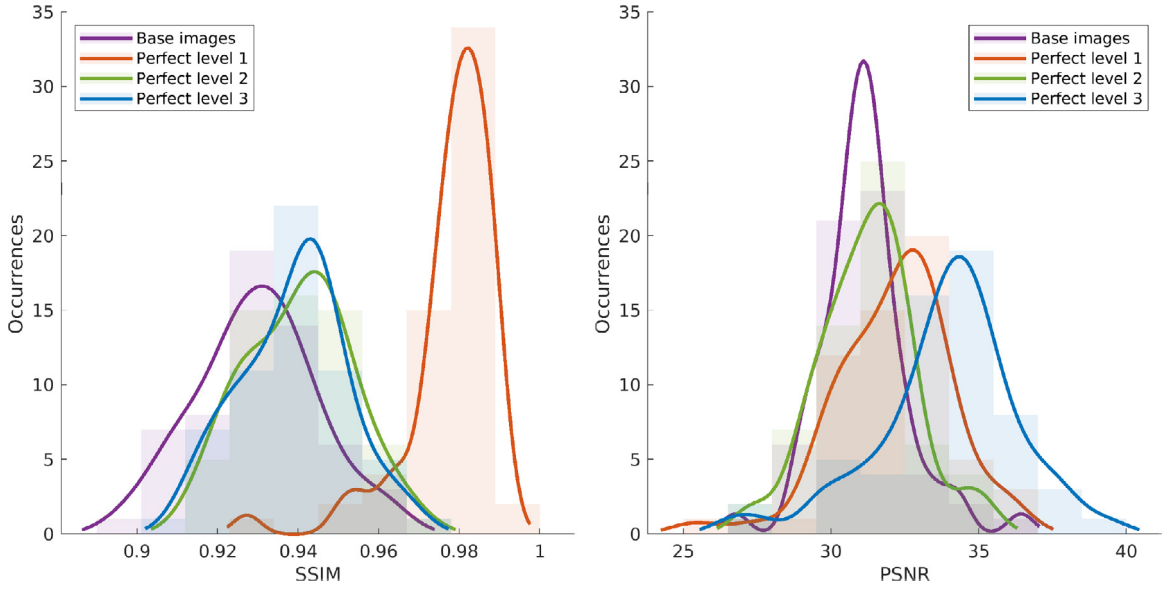

Fig. 4. SSIM and PSNR distributions corresponding to replacing each Laplacian level of our derained images ("base images") with a perfect version from the ground truth. The comparison is always performed with the full ground truth. Kernel Density Estimation [37] is applied to the distributions to facilitate interpretability.

Table 3
Quantitative evaluation of methods for raindrop removal on *test_a* and *test_b* from the dataset by Qian et al. [21]. Results on *test_b* are reported from [24]. Best result in bold, second-best underlined (excluding models trained with different training data).

	test_a		*test_b*	
Method	PSNR	SSIM	PSNR	SSIM
Eigen et al. [38]	28.59	0.6726	–	–
Pix2pix - Isola et al. [6]	30.14	0.8299	23.50	0.7150
AttentiveGAN - Qian et al. [21]	**31.57**	0.9023	24.92	0.8090
Peng et al. [31]	30.72	0.9262	–	–
Quan et al. [22]	31.44	<u>0.9263</u>	–	–
Hao et al. [23]	30.17	0.9128	–	–
Porav et al. [32]	<u>31.55</u>	0.9020	–	–
Alletto et al. [7]*	31.94*	0.9450*	–	–
DURN - Liu et al. [39]	31.24	0.9259	25.32	0.8173
Shao et al. [24]	31.47	0.9235	<u>25.35</u>	**0.8197**
Ours	31.12	**0.9297**	**25.40**	<u>0.8185</u>

* Solution trained on a different training set.

Quantitative evaluation is performed using two full-reference quality assessment metrics: Peak Signal-to-Noise Ratio (PSNR) [34] and Structural Similarity Index Measure (SSIM) [35], both computed on the luminance channel of images in the YCbCr colour space. To be noted that SSIM was proven to be better correlated with human opinion scores, compared to PSNR [36].

4.2. Evaluation of alternative training configurations

We compared the Laplacian loss function $Loss_d$ defined in Section 3.2 based on our encoder-decoder network, with a baseline devoid of any Laplacian decomposition, and with two alternative loss functions that do exploit the decomposition, but combine the resulting levels in different ways. All four configurations, depicted in Fig. 3, exploit the L1 norm to perform the output-target comparisons, and are described in the following:

- Configuration *a*. We exclude our Laplacian decomposition in order to provide an experimental baseline. The loss function compares the target with the output of a classical encoder-decoder model. Here the decoder defined in Section 3 is completely replaced with a specular version of our encoder.
- Configuration *b*. We exploit our Laplacian encoder-decoder. The loss function only compares the final fully-reconstructed output $L_{\hat{Y}}(1)$ with the target image $L_Y(1)$:

$$Loss_b = ||L_Y(1) - L_{\hat{Y}}(1)||_1 \tag{5}$$

- Configuration *c*. We exploit our Laplacian encoder-decoder. The loss function evaluates the output of each level i individually, and sums all the contributions. The comparison between target and output is performed directly on pyramid levels, without reconstructing the image.

$$Loss_c = \sum_{i=1}^{N} ||y_i - \hat{y}_i||_1 \tag{6}$$

- Configuration *d*. We exploit our Laplacian encoder-decoder. The loss function evaluates the output of each level individually, and sums all the contributions. For each level i, an intermediate image $L_{\hat{Y}}(i)$ is reconstructed for comparison with the cor-

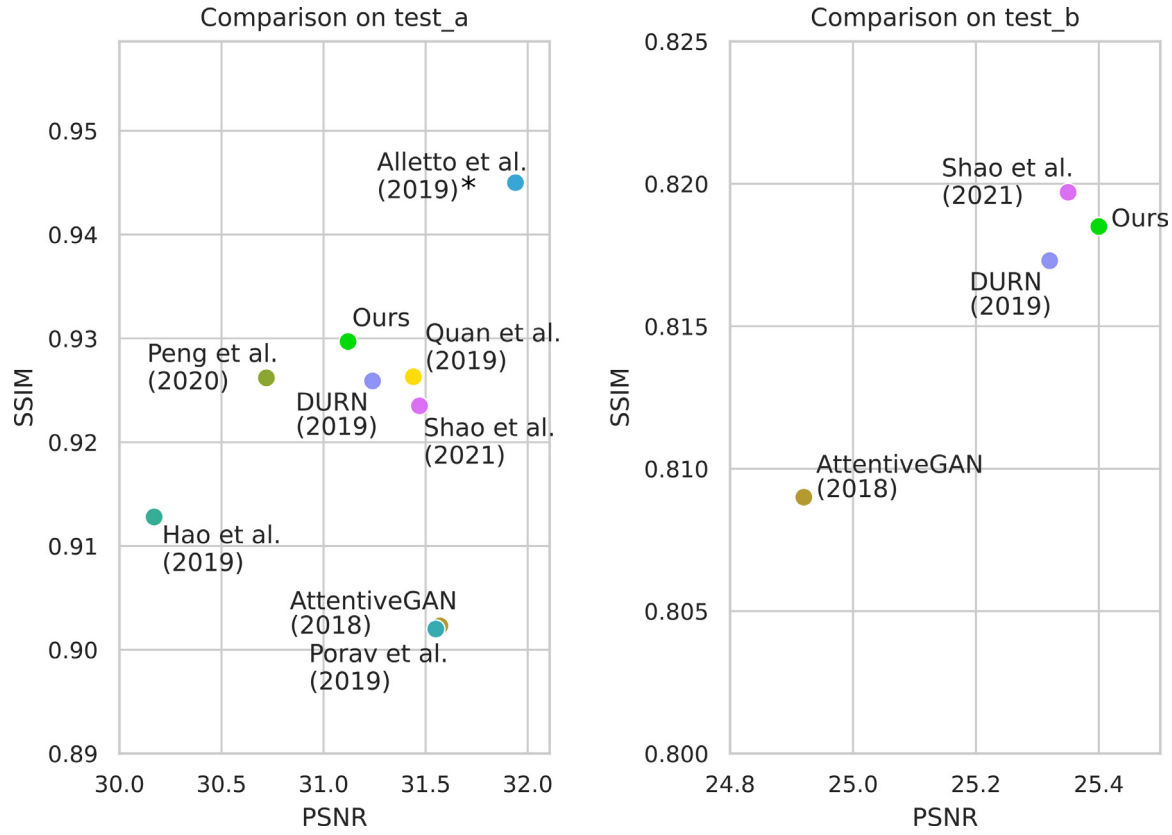

Fig. 5. PSNR-SSIM comparison of the state-of-the-art-models and our proposed method on *test_a* and *test_b* from Qian et al. [21]. A higher value means better visual results. *: Solution trained on a different training set.

Table 4
Comparison of average inference time for different methods. The reported models have been evaluated on an NVIDIA Titan V GPU.

	Pix2Pix [6]	AttentiveGAN [21]	DURN [39]	Quan et al. [22]	Ours
Time (s)	0.010	0.253	0.0165	0.206	0.054

responding target. This is our definitive configuration, relying upon Eq. (4).

Table 1 reports the results of the correspondingly-trained models in terms of PSNR and SSIM. It is possible to observe how the use of a loss function that is aware only of the final result (configuration *b*), is not enough to fully exploit the power of the Laplacian decomposition. Such a model, without a control on the output of the single levels, obtains worse results with respect to a single encoder-decoder trained with the same distance function and with no Laplacian decomposition (configuration *a*). Configuration *c* compares the results of each branch with the corresponding target version, but without the Laplacian reconstruction step for the corresponding levels. In this case, we obtained an improvement with respect to both single-loss configurations *a* and *b*. In this configuration, the training of each branch is directly related to the reconstruction of a certain frequency band: each level is thus focused on the restoration of certain details, without considering the other branches' contribution to the final restored image. However, due to the different nature of the images generated at the different branches (low frequencies and high frequencies), the magnitude of the loss evaluated at each level during training is different. Without any weighting-based regularization, therefore, this configuration is potentially suboptimal. The final version (configuration *d*) evaluates the results of each branch, with respect to the lower levels results. Instead of simply comparing the branches' output, we first reconstruct the image up to the interested level, and only then the loss is calculated. In this way, we are able to compare the results of each layer with the corresponding targets, and at the same time, the losses at the different levels have similar magnitude. Moreover, with this kind of image evaluation the details introduced by each branch are compared with the target in relation to the general context of the image in which they are located, instead of comparing only the map of details modified by the neural network. This helps the neural network to better identify the presence of raindrops that must be removed, in comparison with textures coming from the original scene.

4.3. Laplacian decomposition assessment

We assess the impact of Laplacian decomposition on rain removal, by decomposing *test_a* rainy images and replacing each level independently with the corresponding ground truth.

We then compare each resulting version with the full ground truth. We use PSNR and SSIM metrics in order to adhere to the de-facto standard practice adopted by the specialized literature, also noting that SSIM is known to be well-correlated with human judgement [36]. Results are shown as "Rainy" in Table 2.

PSNR reports the greatest potential advantage when resolving the problem at low frequencies (level 3). Conversely, the higher frequencies (level 1) have potentially the greatest impact on SSIM, which was in fact specifically designed to capture structural simi-

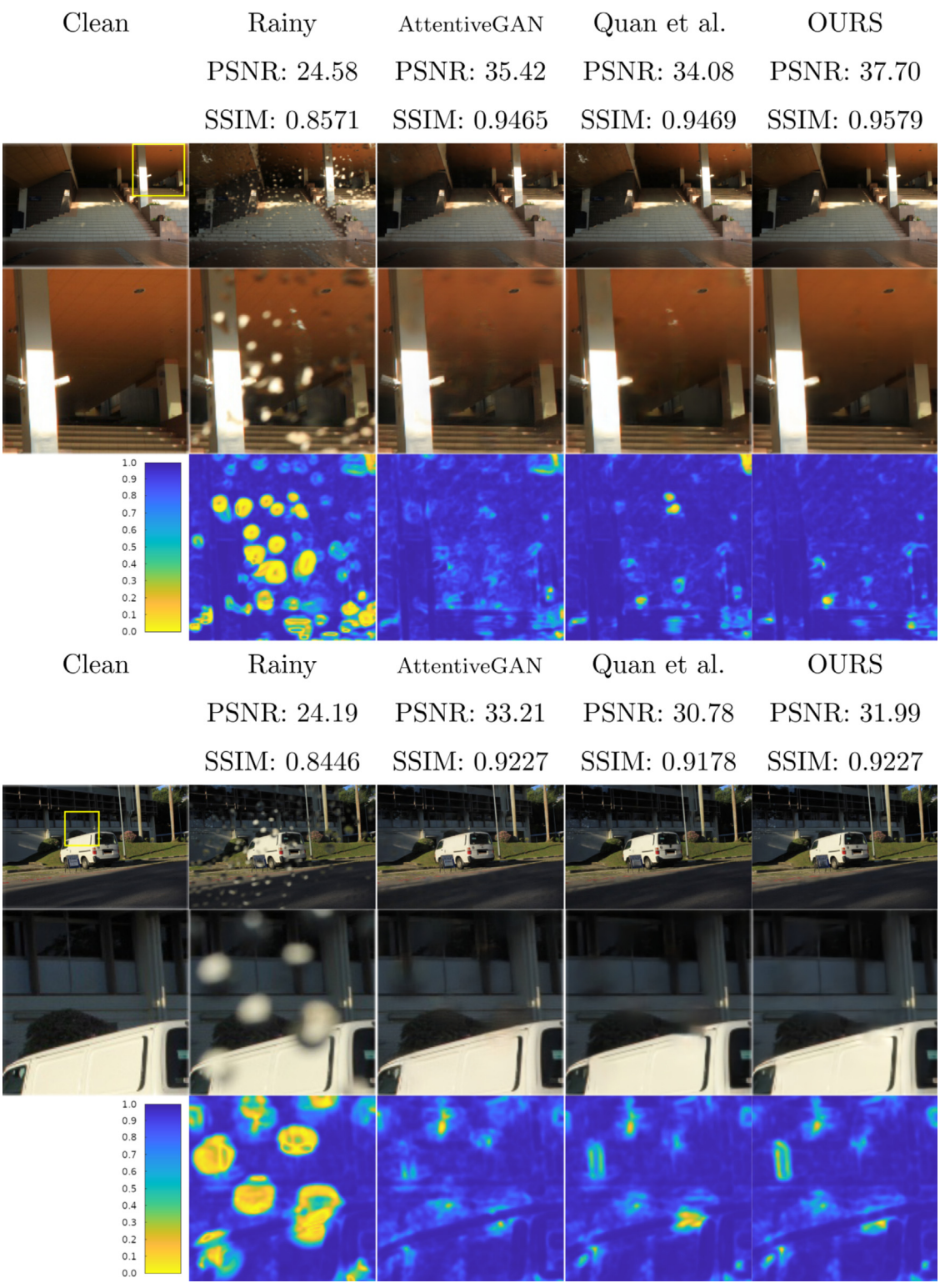

Fig. 6. Visual comparison of methods for raindrop removal. Our proposed model correctly restores information on uniform areas and near edges coming from the original scene. Zoomed crops and the corresponding SSIM maps are reported to facilitate the results interpretation.

larity. To be noted that the upper bound of SSIM is 1, while PSNR has no upper bound. This first evaluation provides an indication of how different errors are distributed across multiple levels. We can also observe that our solution, reported as the "Derained" columns for the base images, outperforms all the individual level replacements for the rainy images, showing that it effectively brings an improvement at more than one level.

We can then quantify the upper bound of improving our current solution one level at a time, i.e. we determine the potential impact of perfectly restoring either of the levels from our derained images. We do so, once again, by replacing each level individually with the corresponding one from the ground truth images. The results are shown in Table 2, with the "Derained" columns, and in Fig. 4 for a view of the entire distribution. For SSIM, the largest

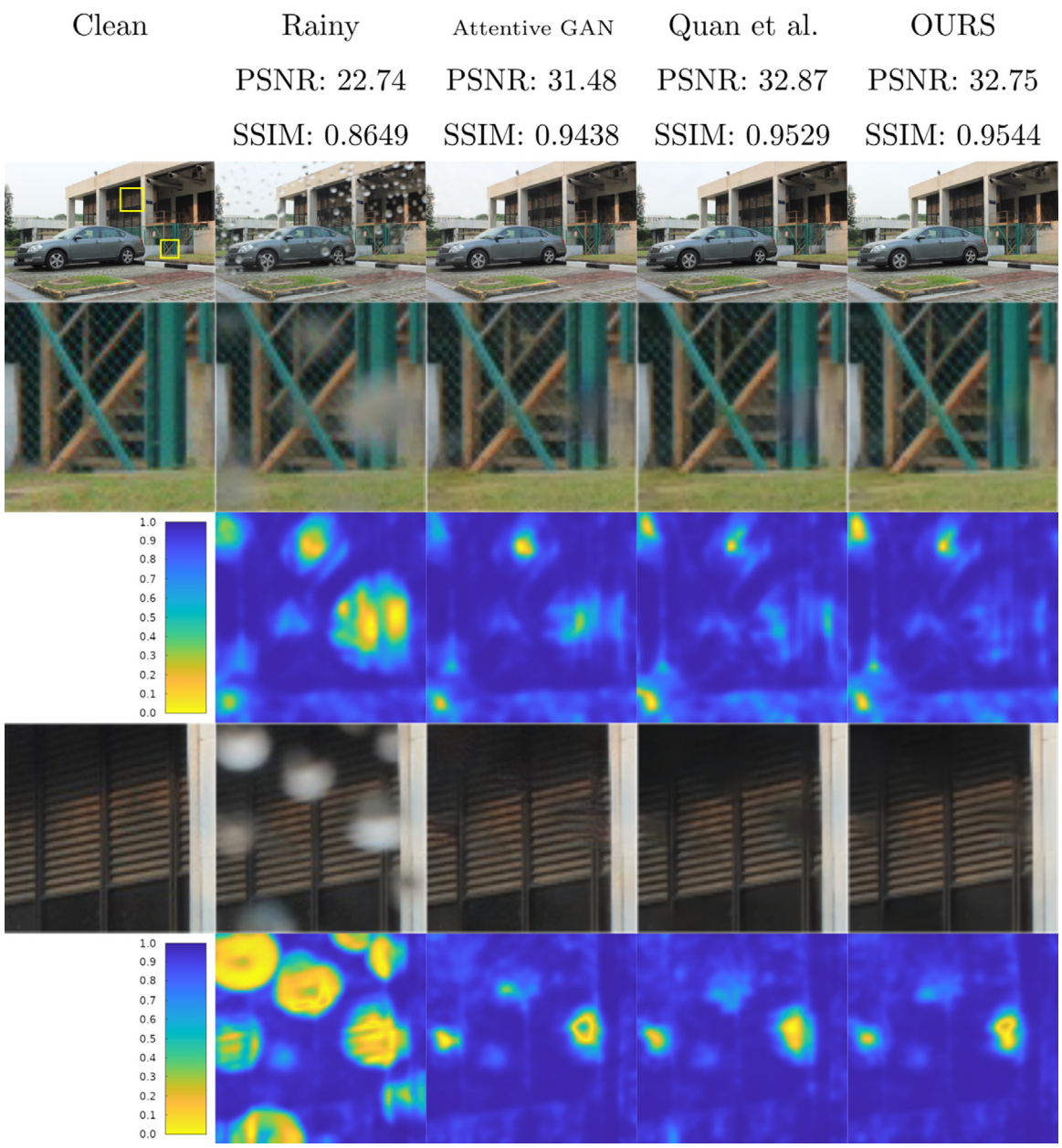

Fig. 7. Visual comparison of methods for raindrop removal on heavily-textured areas. Our model correctly reconstructs some of the complex structures occluded by out-of-focus raindrops. Zoomed crops and the corresponding SSIM maps are reported to facilitate the results interpretation.

possibility for improvement still appears to be working on level 1 (high frequencies). Interestingly enough, for PSNR we observe a large potential improvement by working both on level 3 and level 1. In general, this suggests to focus on details at high frequencies, which would have a positive impact on both evaluation metrics, and which is left as a direction for future research.

4.4. Comparison with the state of the art

In order to analyze the performance of the proposed solution, we compared it with state of the art approaches specifically designed for single-image raindrop removal. We selected an image-to-image general purpose method named Pix2Pix [6], the method by Eigen et al. [38] which is the first attempt in raindrop removal from a single image, AttentiveGAN by Qian et al. [21] which adopts Generative Adversarial Networks to address the restoration process, and methods by Quan et al. [22], Hao et al. [23], DURN by Liu et al. [39], Peng et al. [31], Shao et al. [24], which represent the latest attempts in raindrop removal, exploiting different types of attention mechanisms and/or estimated raindrop maps. The comparison, done in terms of standard measures PSNR and SSIM on *test_a* and *test_b* from Qian et al. [21], is reported in Table 3, while Fig. 5 presents a visualization in Cartesian representation. For both metrics, the higher, the better restoration.

As it can be seen in Table 3, our method outperforms the state of the art solutions on the standard test set *test_a* in terms of SSIM, and achieves comparable performance for PSNR. Alletto et al. [7] report the results of their method on the *test_a* set of the same dataset in terms of PSNR and SSIM values as 31.94 and 0.945 respectively, thus obtaining extremely good performance. However, since their solution was trained on different data, the results are in our opinion not directly comparable. To further analyze the performance of the proposed model on a larger dataset, we also evaluate our solution on the *test_b* set from Qian et al. [21], comparing

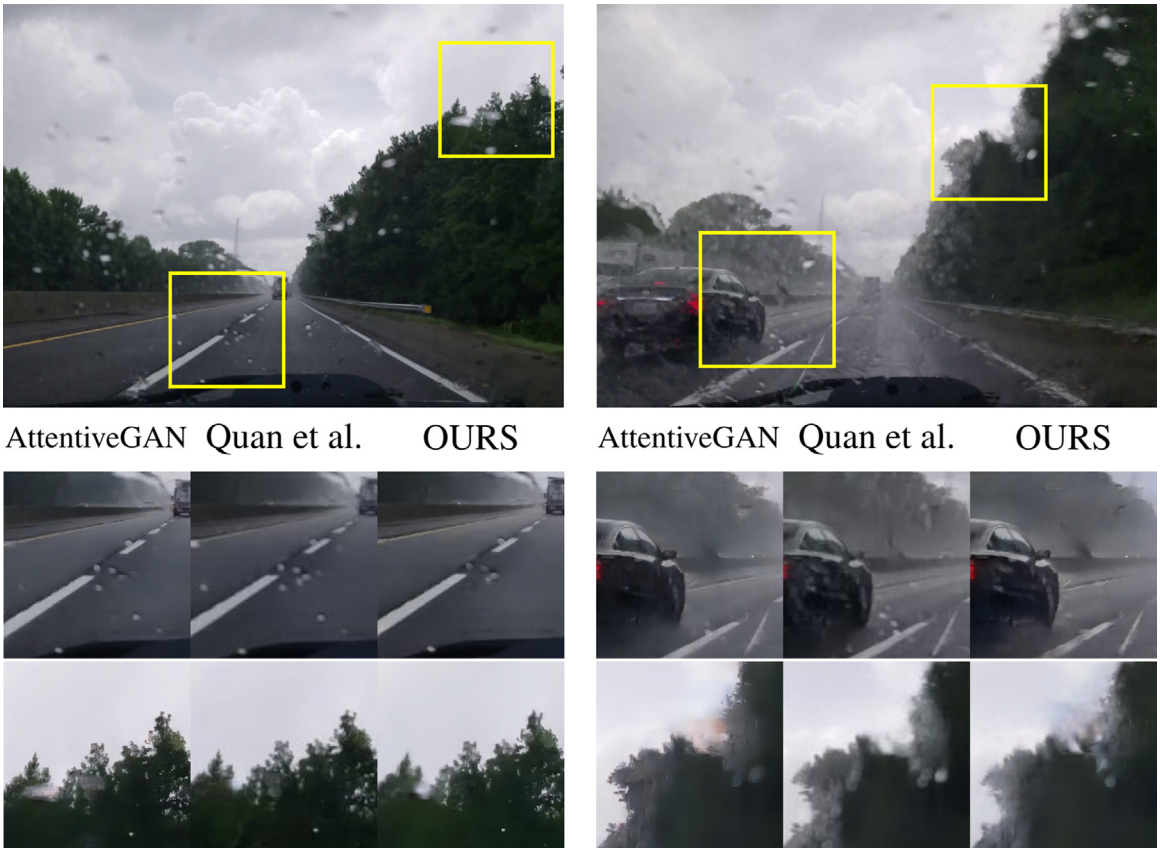

Fig. 8. Results of rain removal on out-of-dataset images acquired with a car dash camera during a storm. Our model is able to restore images from a real-world scenario, removing raindrops and restoring details and structures corrupted by the presence of raindrops. Image credit Eli Christman.

it with the results from other methods as reported by Shao et al. [24]. As can be seen in Table 3, our model outperforms the state of the art in terms of PSNR, while the SSIM index reaches comparable results with the model by Shao et al. [24]. Furthermore, it is interesting to notice the performance drop of AttentiveGAN [21], which can be associated with sub-optimal generalization effectiveness, observed when testing the model with a higher number of images. In this sense, our proposed solution shows a more stable behaviour, being capable to better generalize, and perform generally better than the best performing ones on the smaller set *test_a*.

An additional term for comparison is to account for the average running time during the inference phase. In Table 4 we report the average inference time on *test_a* of our model compared with various methods of which the inference code is available. All of the models have been tested using an NVIDIA Titan V GPU. It is possible to observe that our proposed solution is faster than the method by Qian et al. [21], Quan et al. [22], while being in the same order of magnitude as the other compared methods.

Concerning a visual comparison of the models, we report some processed images from the *test_a* set in Figs. 6 and 7. The comparison was performed against the methods lying on the Pareto front of Fig. 5 (determined without Alletto et al. [7]) and restricting the choice to those whose code and models are publicly available: AttentiveGAN [21] and Quan et al. [22] The images were selected in order to highlight a variety of scene and droplet types. It is possible to observe that our encoder-decoder network produces a satisfactory restoration of homogeneous areas, as well as regions occluded by large out-of-focus droplets, while maintaining little-to-no artifacts related to refraction phenomena. To further prove the effectiveness of the proposed solution, we tested the model on an out-of-dataset scenario. In Fig. 8 we report two frames from a video sequence captured using a car dash camera during a storm, comparing once again our approach with AttentiveGAN [21] and Quan et al. [22]. As can be seen, the proposed solution is able to remove raindrops from the input images, which is particularly evident in the second reported frame, at the same time preserving details (trees from the first image) and avoiding the introduction of colour artifacts (trees from the second image).

5. Conclusions

We presented an encoder-decoder neural network for adherent raindrop removal, motivated by the perspective of improving the visibility of an acquired scene. Our neural architecture takes advantage of image decomposition, by generating the Laplacian pyramid levels of a rain-free version of the input image. This formulation deconstructs a problem that is inherently characterized by a variety of appearances, and allows our model to address each frequency band with a different strategy. We designed a loss function that takes into account the different nature of each Laplacian level, and we showed its suitability in a comparison against other possible loss functions. The effectiveness of this solution was also demonstrated with respect to existing state of the art methods for raindrop removal.

To direct future research, we conducted investigative experiments to understand what components of the image offer the greater chances at improving the model performance. The conclusion is that both SSIM and PSNR measures would benefit significantly by focusing on the lowest level of the Laplacian pyramid, i.e. by improving the reconstruction of high frequencies. The same analysis could be extended to include other metrics, either aiming

at better correlation with human perception, or aim at characterizing the effect on computer vision.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Li, I.B. Araujo, W. Ren, Z. Wang, E.K. Tokuda, R.H. Junior, R. Cesar-Junior, J. Zhang, X. Guo, X. Cao, Single image deraining: a comprehensive benchmark analysis, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 3838–3847.

[2] S. Zini, M. Buzzelli, On the impact of rain over semantic segmentation of street scenes, in: Workshop on Metrification and Optimization of Input Image Quality in Deep Networks, ICPR 2020, Springer, 2021, pp. 597–610.

[3] S. Zini, S. Bianco, R. Schettini, CNN-based rain reduction in street view images, in: Proceedings of the 2020 London Imaging Meeting, 2020, pp. 78–81.

[4] Q. Wu, W. Zhang, B.V. Kumar, Raindrop detection and removal using salient visual features, in: 2012 19th IEEE International Conference on Image Processing, IEEE, 2012, pp. 941–944.

[5] T.-H. Le, P.-H. Lin, S.-C. Huang, LD-Net: an efficient lightweight denoising model based on convolutional neural network, IEEE Open J. Comput. Soc. 1 (2020) 173–181.

[6] P. Isola, J.-Y. Zhu, T. Zhou, A.A. Efros, Image-to-image translation with conditional adversarial networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1125–1134.

[7] S. Alletto, C. Carlin, L. Rigazio, Y. Ishii, S. Tsukizawa, Adherent raindrop removal with self-supervised attention maps and spatio-temporal generative adversarial networks, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019. 0–0

[8] W. Yang, R.T. Tan, S. Wang, Y. Fang, J. Liu, Single image deraining: from model-based to data-driven and beyond, IEEE Trans. Pattern Anal. Mach. Intell. 43 (2020) 4059–4077.

[9] K. Garg, S.K. Nayar, Vision and rain, Int. J. Comput. Vis. 75 (1) (2007) 3–27.

[10] Y. Luo, Y. Xu, H. Ji, Removing rain from a single image via discriminative sparse coding, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 3397–3405.

[11] W. Yang, R.T. Tan, J. Feng, J. Liu, Z. Guo, S. Yan, Deep joint rain detection and removal from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1357–1366.

[12] X. Fu, J. Huang, X. Ding, Y. Liao, J. Paisley, Clearing the skies: a deep network architecture for single-image rain removal, IEEE Trans. Image Process. 26 (6) (2017) 2944–2956.

[13] H. Zhang, V. Sindagi, V.M. Patel, Image de-raining using a conditional generative adversarial network, IEEE Trans. Circuits Syst. Video Technol. 30 (11) (2019) 3943–3956.

[14] H. Wang, Y. Wu, M. Li, Q. Zhao, D. Meng, A survey on rain removal from video and single image, arXiv preprint arXiv:1909.08326 (2019).

[15] X. Bi, J. Xing, Multi-scale weighted fusion attentive generative adversarial network for single image de-raining, IEEE Access 8 (2020) 69838–69848.

[16] Y. Mi, S. Yuan, X. Li, J. Zhou, Dense residual generative adversarial network for rapid rain removal, IEEE Access 9 (2021) 24848–24858.

[17] K. Jiang, Z. Wang, P. Yi, C. Chen, Z. Han, T. Lu, B. Huang, J. Jiang, Decomposition makes better rain removal: an improved attention-guided deraining network, IEEE Trans. Circuits Syst. Video Technol. 31 (2020) 3981–3995.

[18] H. Dong, J. Pan, L. Xiang, Z. Hu, X. Zhang, F. Wang, M.-H. Yang, Multi-scale boosted dehazing network with dense feature fusion, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 2157–2167.

[19] D.-W. Jaw, S.-C. Huang, S.-Y. Kuo, Desnowgan: an efficient single image snow removal framework using cross-resolution lateral connection and GANs, IEEE Trans. Circuits Syst. Video Technol. 31 (4) (2020) 1342–1350.

[20] Y.-F. Liu, D.-W. Jaw, S.-C. Huang, J.-N. Hwang, Desnownet: context-aware deep network for snow removal, IEEE Trans. Image Process. 27 (6) (2018) 3064–3073.

[21] R. Qian, R.T. Tan, W. Yang, J. Su, J. Liu, Attentive generative adversarial network for raindrop removal from a single image, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 2482–2491.

[22] Y. Quan, S. Deng, Y. Chen, H. Ji, Deep learning for seeing through window with raindrops, in: Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 2463–2471.

[23] Z. Hao, S. You, Y. Li, K. Li, F. Lu, Learning from synthetic photorealistic raindrop for single image raindrop removal, in: Proceedings of the IEEE International Conference on Computer Vision Workshops, 2019. 0–0

[24] M.-W. Shao, L. Li, D.-Y. Meng, W.-M. Zuo, Uncertainty guided multi-scale attention network for raindrop removal from a single image, IEEE Trans. Image Process. 30 (2021) 4828–4839.

[25] L.-W. Kang, C.-W. Lin, Y.-H. Fu, Automatic single-image-based rain streaks removal via image decomposition, IEEE Trans. Image Process. 21 (4) (2011) 1742–1755.

[26] S.-H. Sun, S.-P. Fan, Y.-C.F. Wang, Exploiting image structural similarity for single image rain removal, in: 2014 IEEE International Conference on Image Processing (ICIP), IEEE, 2014, pp. 4482–4486.

[27] X. Fu, B. Liang, Y. Huang, X. Ding, J. Paisley, Lightweight pyramid networks for image deraining, IEEE Trans. Neural Netw. Learn. Syst. (2019).

[28] P. Burt, E. Adelson, The Laplacian pyramid as a compact image code, IEEE Trans. Commun. 31 (4) (1983) 532–540.

[29] O. Ronneberger, P. Fischer, T. Brox, U-net: convolutional networks for biomedical image segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2015, pp. 234–241.

[30] F. Yu, V. Koltun, Multi-scale context aggregation by dilated convolutions, arXiv preprint arXiv:1511.07122 (2015).

[31] J. Peng, Y. Xu, T. Chen, Y. Huang, Single-image raindrop removal using concurrent channel-spatial attention and long-short skip connections, Pattern Recognit. Lett. 131 (2020) 121–127.

[32] H. Porav, T. Bruls, P. Newman, I can see clearly now: image restoration via de-raining, in: 2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019, pp. 7087–7093.

[33] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).

[34] Z. Wang, A.C. Bovik, Modern image quality assessment, Synth. Lect. Image, Video, Multimed. Process. 2 (1) (2006) 1–156.

[35] Z. Wang, A.C. Bovik, H.R. Sheikh, E.P. Simoncelli, Image quality assessment: from error visibility to structural similarity, IEEE Trans. Image Process. 13 (4) (2004) 600–612.

[36] I. Bakurov, M. Buzzelli, R. Schettini, M. Castelli, L. Vanneschi, Structural similarity index (SSIM) revisited: adata-driven approach, Expert Syst. Appl. 189 (2022) 116087.

[37] M. Rosenblatt, Remarks on some nonparametric estimates of a density function, Ann. Math. Stat. 27 (1956) 832–837.

[38] D. Eigen, D. Krishnan, R. Fergus, Restoring an image taken through a window covered with dirt or rain, in: Proceedings of the IEEE International Conference on Computer Vision, 2013, pp. 633–640.

[39] X. Liu, M. Suganuma, Z. Sun, T. Okatani, Dual residual networks leveraging the potential of paired operations for image restoration, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7007–7016.