Fast Image-Anomaly Mitigation for Autonomous Mobile Robots

Cover Letter

Gianmario Fumagalli, Yannick Huber, Marcin Dymczyk, Roland Siegwart, Renaud Dubé

March 9, 2021

We are grateful to all the Reviewers for their helpful suggestions and kind dedication. We have carefully read all the reviews and addressed many comments and questions, both related to the content of the paper and to its form (grammar mistakes and typos).

Some of the Reviewers complained about the fact that we defined our dataset as “the largest dataset for the task” and that it is not public. Our statement was limited to the publicly available datasets, and, since the dataset released by Porav et al. [1] contains less than 5 thousand image pairs (even though the actual dataset comprises 50k images), we considered ours as the largest. However, to prevent misconfusion for the future readers, we changed our statement to “one of the largest publicly available datasets” and we clarified that it will be made publicly available upon publication.

Other comments were related to the difficult reproducibility of the results. We have added the number of filters in each layer in the explanation of the generator and we introduced a new figure that displays in detail the aggregation process of the feature maps, other than changing its description in the text. We hope that such changes will help future readers to understand our architecture better and to reproduce our results.

We have also implemented other detailed suggestions provided by single reviewers. We have

- Revised the explanation about the use of a “lightweight discriminator”
- Clarified why images of our dataset are grayscale
- Pointed out more what are the differences between our data acquisition process and the one described in Porav et al. [1]
- Added references to the metrics used
- Changed vertical spaces below certain figures

We hope that the changes we made will strengthen our contributions, help understanding our work and improve readability of the paper.

Sincerely,
Gianmario Fumagalli, Yannick Huber, Marcin Dymczyk, Roland Siegwart and Renaud Dubé

References

[1] Horia Porav, Tom Bruls, and Paul Newman. I can see clearly now:Image restoration via de-raining. In2019 International Conference onRobotics and Automation (ICRA), pages 7087–7093. IEEE, 2019

Comments to author (Associate Editor)
==========

The paper presents a new deep network for fast anomaly mitigation. Overall, I agree with the reviewers on two main problems of the current version of the paper: i) The main claim of the paper is the inference speed, but it's not clear which part of the theory brings this improvement, ii) The claim about "biggest dataset" seems to be a false claim, as Porav et al. provided a 10 times bigger dataset. It would be more beneficial for the community if the source code and/or dataset are publicly available. Currently, I see it's very difficult to reproduce the results of the paper, as the network architecture is not described in detail.

Comments on Video Attachment:

Good video demonstration
Comments to author (Editor)
==========

Contribution of the paper should be strengthened to respond to negative comments and concerns.

Comments on Video Attachment:

Good video

Reviewer 2 of ICRA 2021 submission 1802

Comments to the author
======================

The paper "Fast Anomaly Mitigation for Autonomous Mobile Robots" proposes an GAN-based method to remove raindrops from images to counter adversarial effects caused by the drops on downstream tasks. To this end, they train a generator that is able to remove raindrops and use an enhancer to improve the results. The paper furthermore proposes a way to generate out-of-focus synthetic raindrops. Results on existing datasets show improvements mainly in terms of the runtime of the approach at a similar level of quality.

The papepr is well-written and easy to follow. Figures provide context and help the understanding and the qualitative impression of the results. Generally, the topic seems relevant as it provides a way to cope with image artifacts they could affect the image perception pipeline. The proposed method follows along the work of Porav et al. and extends it a bit by the augmentation and enhancer, while making the generator network more efficient by removing layers. The proposed method needs an enhancer to regain the losses incurred by having a smaller generator as shown by the ablation studies. Since the work basically follows Porav et al. in respect to raindrop removal, rain drop generation, technical details of the used device to record data, I would have expected a stronger and more explicit differentiation of the paper from this particular approach. It should be clear what are the advances. From my point of view, the contribution boils down to simpler generator (discriminator), image enhancer network to improve the results, and the data recording (which will not be available?) Regarding this it is not surprising that the approach is then faster, but also a bit worse then the other approaches. As the authors will not provide code or data, I currently doubt that it is possible to reproduce the employed architecture. Especially the added part with the augmentation is a bit foggy.

In summary, the paper mainly lacks details on the important parts (augmentation) and it is unclear what is the real gain of the paper for the community as the generated data will not be available and the actual details of the approach/architecture/raindrop generation are missing that

would allow reproducing the approach. Either be more specific on what architecture is used or provide more information in the paper, since provided details and Figure 1 is not enough.

More detailed comments regarding some other points:

- Authors claim to have "The largest dataset for the task of removing camera-
attached anomalies, with rainy or dusty images with the corresponding clean ground truth." and state in section that they collected 5k images. However, the dataset from Porav et al. contains 50k images. How does this relate to the claim that it's the largest dataset.
- Section III.A. The generator employs some convolutional aggregation module that somehow aggregated upsampled feature maps but it is unclear how this aggregation works from the description ("convolutional aggregation module after each of them, which takes as input the concatenation between these feature maps and the ones from the previous layer, bilinearly downsampled by a factor of two."). As I understood it, the downsampling happens via strided convolutions (?) and then the feature maps from before the strided convolution are added via bilinear downsampling (???) to the downsampled feature volume? First, for downsampling is no bilinear interpolation needed, since we can just omit every second entry in the feature map if I understood it correctly.
- Generally, it is hard to understand and reproduce the architecture without code or a more explicit table over the different hyper parameters. Either the figure should contain some information or an table with explicit in and out channels is needed to be able to reproduce the results.
- Authors claim lightweight architecture, but it would be helpful to know exactly how big is the difference in terms of number of parameters.
- Making the discriminator more efficient seems odd as it is only needed for training. A better discriminator should also result in better performance of the generator. Is the need for the enhancer may be caused by the less effective discriminator, which is simply not able to provide good enough results?
- Authors mention in the introduction "dust" as a possible anomaly ("The majority of samples is degraded by raindrops adherent on glass and a few are spoiled with dust and dirt to show how our architecture can be adapted to clean images affected by other, less researched anomalies."), but this

is not mentioned at all in the experimental evaluation, where only raindrops are removed.
- Figure 3. Are these images grayscale? Why?
- The synthetic rain drop generation is only described with very few lines (basically 2 lines). Either code/pseudo code would be needed to reproduce the procedure or even understand what is actually happening. Is it based on the Porav method? Are there changes needed to their method?
- Experiments: The data acquisition process is described and the setup, but is the data used for something? Ablation is happening on Qian et al. data. And will the data be released?
- The actual device looks very similar to Porav et al. Are there innovations? Is the Ouster LiDAR used for something?
- Random initialization with normal distributions is a particularly bad choice for ReLU CNNs. (see

He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015 IEEE International Conference on Computer Vision (ICCV)

X. Glorot and Y. Bengio. Understanding the difficulty oftraining deep feedforward neural networks. InInternationalConference on Artificial Intelligence and Statistics, pages249256, 2010.)

minor issues:
- Although some research has moved towards building models that can adapt to uncertain conditions and challenging environments [33], DeRaindrop [23] and Porav et al. [22] have proven that a de-noising pre-processing step is more effective. -- "proven" suggest that this is and will be the case. However, it's more "experimentally showed"
- I prefer to have author names or proper nouns instead of references as nouns, as it improves the reading experience, e.g., "We developed a light-weight generator adapted from Porav et al. [22] that ..." directly makes it clear that you are speaking about the paper mentioned before.
- The \vspace{-X} in front of the Introduction seems too large (the section title is far to close to the abstract). You could add a vspace{} in the figure and relax the vspace under the abstract.
- It is mentioned that the encoder produces a feature map that is fed into 9 residual blocks. Figure 2 has after the feature volume (orange) only 6 "blocks". How does this fit together?
- Metrics should be mentioned at least with a reference to

find the actual definition used in the paper or better shortly introduced.
- I wonder how the performance of the approaches only with synthetically generated raindrops is? Wouldn't this show the real advantage of the proposed approach for generating better rain drops?

Comments on the Video Attachment
==========

Good overview of the method and the results.

Reviewer 3 of ICRA 2021 submission 1802

Comments to the author
======================

The article proposed an end-to-end approach to mitigate rain effect from a single image. The proposed model improves efficiency of the model by a significant amount while preserving the performance evaluated with two metrics (SSIM & PSNR).
The article is well written and easy to follow, and here are a few questions that could help a larger audience do not have much deep learning experiences.
1) The enhancer (Figure 2) was left outside the discriminator, seems not utilizing its full power (for the discriminator). If this is a design choice, could you add a few comments to clarify?
2) Could you provide a bit more details about the two metrics being used here (SSIM & PSNR), e.g. a few more references, and especially how to best understand the deltas in table III, since the improvement on SSIM is less than PSNR in terms of the numbers/percentages.
3) It's great that the article included runtime comparisons on GPU vs. CPU, however given the increased applications on portable devices, it will attract more attentions from industry if there will be runtime on ARM based CPUs in future.

It would be great to share more details about the dataset which was collected for training, such as whether or not it would be public available; what is the ratio between easy/medium/hard cases; what are the different types of whether included in this dataset; length of each sequence; typical objects in the view; It could attract a lot of eyes since the author mentioned 'largest dataset' in abstract, and many of the readers may have similar questions.

Comments on the Video Attachment
================================

Nice video demonstrating the general idea of the paper.
Only comment is the image sequences seems to have some jumps, instead of smooth motion of the camera.

Reviewer 5 of ICRA 2021 submission 1802

Comments to the author
======================

This paper presents a framework for the removal of anomalies, such as raindrops, from images. It borrows components from two earlier publications, a U-net-like generator network in combination with a discriminator network from [22] and an enhancer network from [24]. At least the first two networks are, however, more light-weight than the original, allowing for a 15-fold improvement in speed. To somewhat improve the reconstruction quality, a convolutional aggregation module is also
proposed.

Unfortunately, almost everything that is claimed as contributions in this paper are also very vaguely described.
Its impossible to see where the 15-fold increase in speed comes from. In fact, it looks as if the proposed system includes more components than the one in [22]. There are small variations in the specifics, but the increase in speed is so large that it ought to be a major theme to explain how that is at all possible, at least assuming that it runs on the same hardware using a similar codebase. What are the computational costs of the different networks and where do you get these gains in speed?

The lack of specifics when it comes to speed could have been excused if the reconstruction quality is improved with respect to the earlier publications, but that is not the case. In terms of quality, the proposed system performs worse than the two baseline methods. Even if this is the case, its still interesting to see what trade-offs can be made to find the right balance between accuracy and speed, but then the explanation of speed gains need to be clear.

The proposed synthetic drop generation technique is also vaguely described. If it is claimed as a contribution, it better be described well enough for it to be possible to recreate. Now you are left with a couple of convincing images but without a full understanding of how it was done. It would have been interesting to see a comparison in terms of reconstruction quality for the proposed drop generator and the one in [22].

Another claim is the largest dataset for the task of removing camera-attached anomalies, but the dataset in [22] is larger than the one here, a dataset collected using a very similar binocular camera setup. Given how easy it is to create new data with such a setup, since it doesnt require annotations, the size of the dataset is of less interest.

Something that seems to be missing is a discriminator loss that includes the clean images, such a L_disc in [22]. Otherwise, the discriminator will always output 1 and it would lose its function. It also seems that D(A) in (2) should really be D(G(A)). Other than that, the paper is quite well written and relatively easy to follow.

If the speed has indeed been improved 15-fold with respect to [22], it would have been better to focus the study on the gradual improvements made to optimize the most time-consuming parts of the system, with an analysis of how the reconstruction quality is affected by these improvements.

Comments on the Video Attachment
==========

The video is a nice summary of the paper, with a description of the proposed solutions, as well as illustrative results.

Reviewer 11 of ICRA 2021 submission 1802

Comments to the author
======================

This is an excellent, interesting, relevant and well developed paper on a predictive technique for removing anomalies from video feeds using GANs. The literature review for the
paper is extensive and complete. It lays a good groundwork for the rest of the paper and covers a wide range of appropriate techniques, and provides grounding for the interested reader for further references.

In terms for further improvements to the paper it would be interesting to see a systematic study, potentially lab-based, to attempt to quantify the limits of the algorithm in providing correction. The further work that is considered in the paper, about distinguishing between removable and non-removable marks is an interesting development and would be highly welcomed. It would be interesting to extend this work to consider some of the implications on the reconfigurable nature of the decisions that will be made.

Additionally it would be interesting to consider just how much the real-time characteristics are affected by the different types of anomalies that are present; e.g. the tradeoff between light marks and heavy marks, or even the "heaviness" of the rain and what implications this may then have on the operation of the robot.

Overall this is a very interesting paper that is highly relevant both academically and industrially. It would be interesting to read this as an extended journal paper on the back of this conference paper.

Comments on the Video Attachment
================================

This is an interesting and useful video. Captioning is very useful, and could benefit from extending slightly, just to incorporate a little more information.

Reviewer 13 of ICRA 2021 submission 1802

Comments to the author
======================

This letter proposed a denoising algorithm to remove raindrop from the image for the application of mobile robots. The method is based on the well-known generative adversarial network where the generator that transforms a rainy image to clean image. The novelty of the letter is to incorporate an aggregation scheme to the enhancer that improves the performance of the neural network.

However, the title does not match the work demonstrated in this paper -- please consider use the keyword camera anomaly.

III. A. '... where A stands...' change A's font in description to match that in the equation (1). You also have to explain what is D and G here.

'Here, n_{FM} stands for...' the same problem, you have to explain what is D and match the font between the description and that in the equation.

III. B. '...To synthesize out-of-focus raindrops we introduce multiple
shiftings in random directions where our camera anomalies are situated.' Does the shift only apply to the raindrop or the whole image? Please clarify this.

Figure 4. I am expecting to see a blurry raindrop as foreground and the road should be sharp. I agree fixed-focus cameras are common for robotics but the whole image is out of focus. Could you explain how to include the out of focus effect in your dataset?

IV. A. '... Moreover, the filters can be easily replaced...' Does your algorithm can also deal with other camera anomalies such as dirt mentioned here? The title and the abstract of the letter seem to claim the application of the algorithm is a general-purpose camera anomaly correction. However, the only case that has been discussed in this letter is the raindrop. It would be more interesting and more relevant to see if this algorithm is applicable to other camera anomalies such as scratches, dirt, running rain, salt and surface ice.

IV. C. Table I what is the GPU and CPU time of your network for G, G+E? How will the enhancer or aggregation scheme slow down the algorithm?

This letter presented a solid work on the image denoise algorithm for the raindrop on the camera. The contribution is relevant to the robotics application and the influence of the work could further benefit the community if relevant code and data synthetic technique are open-sourced.

Comments on the Video Attachment
==========

The video attachment is clear and informative.

Fast Anomaly Mitigation for Autonomous Mobile Robots

Gianmario Fumagalli[1,2], Yannick Huber[1], Marcin Dymczyk[1], Roland Siegwart[2], Renaud Dubé[1]
[1]Sevensense Robotics AG, {firstname.lastname}@sevensense.ch
[2]Autonomous Systems Lab, ETH Zürich, {firstname.lastname}@mavt.ethz.ch

***Abstract*— Camera anomalies like rain or dust can severely degrade image quality and its related tasks, such as localization and segmentation. In this work we address this important issue by implementing a pre-processing step that can effectively mitigate such artifacts in a real-time fashion, thus supporting the deployment of autonomous systems with limited compute capabilities. We propose a shallow generator with aggregation, trained in an adversarial setting to solve the ill-posed problem of reconstructing the occluded regions. We add an enhancer to further preserve high-frequency details and image colorization. We also produce the largest dataset to date to train our architecture and use realistic synthetic raindrops to obtain an improved initialization of the model. We benchmark our framework on existing datasets and on our own images obtaining state-of-the-art results while enabling real-time performance, with up to 40x faster inference time than existing approaches.**

I. INTRODUCTION

With the continuous development of autonomous robots, their deployment outside labs is rapidly increasing.While representing a great step towards the future, leaving self-driving machines operate in a less-controlled, less-predictable environment poses completely new challenges. Particularly crucial are the ones related to the so-called camera anomalies, which are impurities that can affect the camera lenses and deteriorate the captured images, as shown in Figure 1, left column. For instance, the vision system of robots moving in outdoor terrains can be spoiled by lifted soil, dust and by the weathering, rain especially [7].

Raindrops on lenses have been largely the most studied camera anomaly [29, 30, 38], due to the higher probability to occur and the wider impact they have on the image. Indeed, their rounded shapes lead to a fish-eye-lens effect, that refracts the light in a different way with respect to the background and displays different – sometimes far – parts of the scene. Considering that the focal point is usually at infinity, with the unaffected regions being in focus and the raindrops looking extremely blurred, it is not surprising that the degradation of the image would be consistent, thus reducing dramatically the performance of vision-related tasks, such as classification and segmentation.

Although some research has moved towards building models that can adapt to uncertain conditions and challenging environments [33], DeRaindrop [23] and Porav *et al.* [22] have proven that a de-noising pre-processing step is more effective. These architectures are able to produce realistic de-rained samples, but at the cost of not being real-time capable on platforms with limited computing power, like autonomous robots. Following their steps, our approach towards the task aims at developing an effective and efficient pre-processing step able to restore the original clean image, while enabling real-time performance. We developed a light-weight generator adapted from [22] that estimates the occluded regions, coupled with an enhancer module [24] that preserves input colors and details. Our architecture is trained in an adversarial setting [6] with a discriminator [19] that has proven effective for this task [22, 23]. With our framework, we can compete with state-of-the-art methods on datasets from [22, 23] and outperform all existing methods with similar results by a large margin in terms of inference speed. In addition, we trained the model on our own dataset captured with the novel sensor shown in Figure 5. We collected the largest available dataset for the task, comprising of pairs of anomaly-affected and clean images, as described in Section IV-A. The majority of samples is degraded by raindrops adherent on glass and a few are spoiled with dust and dirt to show how our architecture can be adapted to clean images affected by other, less researched anomalies. Alongside with it, we provide a realistic synthetic drop generation that mimics the physical photometric effect of raindrops on camera lens. With respect to our reference model [22], we improved by creating proto-raindrops that

Fig. 1: Fast camera-anomaly mitigation produced by our GAN architecture. Rain-affected samples (left column) are fed into our generator that, alongside the enhancer module, learns to restore the original clean image (right column) in a real-time fashion.

are also full-fledged out of focus. A foreground layer of computer-generated raindrops can be applied to huge existing datasets, like ImageNet [4] or MS-COCO [20], to obtain cost-free rainy samples. We used our synthetic images in support of real data, as a pre-training step to produce a precise initialization for the weights before the actual training. Section III-B exposes the process of creating our synthetic raindrops and displays their use for our purposes. To summarize, our main contributions are:

- A light architecture that can perform anomaly mitigation with state-of-the-art performance in real-time;
- A realistic synthetic-drop-generation technique that outputs rainy images with out-of-focus raindrops;
- The largest dataset for the task of removing camera-attached anomalies, with rainy or dusty images with the corresponding clean ground truth.

II. RELATED WORK

Several attempts have been done throughout the years to reduce the effect of camera anomalies, such as snow, fog, haze, dirt, rain streaks and raindrops, with both deep-learning (DL) and non-deep-learning techniques.

A. Non-DL methods

Some researchers in this field, especially in early works, have tackled the problem using a non-deep-learning approach. Their main limitation is that they all require high-framerate videos instead of single images, thus reducing their applicability in a real-world scenario. In [41] the authors implement a rain removal algorithm that leverages both temporal and chromatic properties of pixels across a video sequence, stating that a pixel is never covered by rain throughout the entire video and the changes in RGB values are approximately the same. It also has the limitation that it only works for static cameras. Roser and Geiger [29] developed a rain-detection and image-reconstruction algorithm using photometric properties (optical path) applied to a spherical raindrop model, which is fitted multiple times in the image to detect raindrop regions. Image reconstruction is done evaluating the intensity level of pixels near raindrops. In their subsequent work [30] they change the raindrop model to cubic Bezier Curves, which is more realistic. However, the change in shape and size of raindrops is too high to be adequately fitted. In [15] stereo video sequences are needed to perform spatio-temporal frame warping with a median filter across three selected frames to reconstruct the clean image, while in [38] raindrops are detected and removed by looking at local properties, i.e. spatial derivatives and changes in optical flow. Others [16, 28] model background fluctuations with a multi-label Markov field and reconstruct the image via a low-rank representation (SVD decomposition) of the background. The latest work by [12] applies a split augmented Lagrangian Shrinkage algorithm to directional and temporal gradients to efficiently remove rain streaks in videos.

B. DL methods

Latest research is mainly focusing on deep-learning techniques for their capacity to store information during training [1] allowing to perform anomaly mitigation on single images. Apart from the work by Eigen *et al.* [5], where a Convolutional Neural Network (CNN) is used to restore an image taken through a window covered by rain or dirt, the great majority of deep-learning techniques involve the use of Generative Adversarial Networks [6], as they have proved successful in several image-to-image translation problems [11, 42]. Severeal works in this field address the task of image de-noising applied to de-hazing [2, 18, 21, 26, 27, 37, 39] and rain streak deletion [3, 25, 34, 36, 40] and can be related to the equally important topic of camera-anomaly mitigation. Qian *et al.* [23] inject visual attention in both generator and discriminator architectures to focus on raindrop regions. The probabilities of belonging to a raindrop are estimated using a recurrent neural network [14, 31] alongside with an LSTM module [10]. The remaining part of the generator consists of a convolutional autoencoder, and the discriminator is represented by a CNN. The authors train the architecture on their static dataset and produce state-of-the-art results at the cost of slow inference time, computational complexity and complex training scheme due to the attention module. Porav *et al.* [22] remove the attention and develop a simpler architecture similar to [35]. The generator employs the autoencoder structure with skip-connection layers to preserve input structure and details. It is trained in an adversarial setting alongside a Patch-GAN discriminator [19]. They also published a dataset that allowed us to compare our results with theirs. Again, while being simpler, their architecture is still heavy and does not ensure real-time performance, which is critical for real-world robotic applications.

III. METHODOLOGY

A. Architecture

We address the task of image de-raining as an image-to-image translation problem, where rainy and clear images are regarded as two different image styles. Our architecture comprises a generator and an enhancer. The former tries to fool the discriminator during training by producing fake de-rained images in an adversarial fashion; the latter aims to improve the resulting images by preserving input color and details. Figure 2 displays our full architecture.

Generator Our generator architecture is inspired from Pix2Pix HD [35] and on [22], although much lighter. It has an encoder-decoder structure with a sequence of residual blocks [9] in between. The encoder employs two down-sampling convolutional layers with the aim to enlarge the receptive field and capture the information at multiple scales. To further refine the resulting feature maps we added a convolutional aggregation module after each of them, which takes as input the concatenation between these feature maps and the ones from the previous layer, bilinearly downsampled by a factor of two.

The encoder outputs feature maps with a quarter of the original size, that are then fed into a series of 9 residual

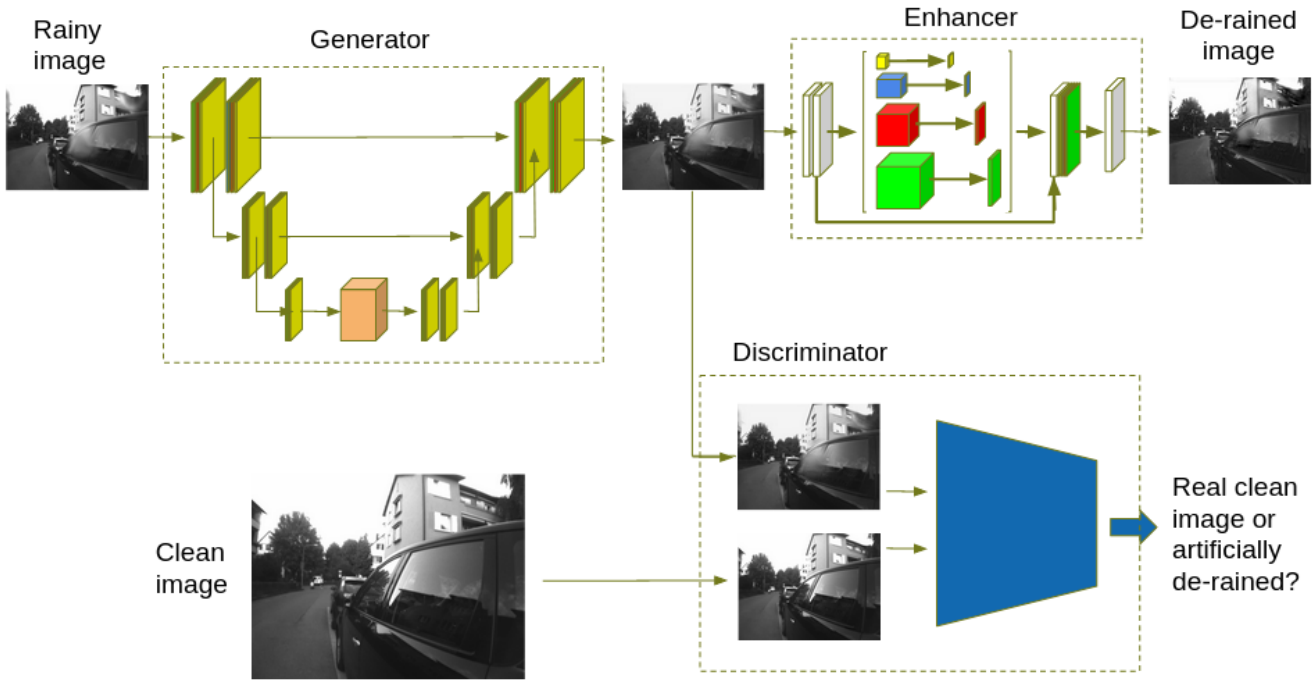

Fig. 2: Our full pipeline for anomaly removal. The affected image is first run through the generator that performs the first mitigation. Then, after concatenation with the original image, it is fed into the enhancer to produce the final output.

blocks that perform the actual image restoration, focusing on raindrops and trying to estimate the occluded regions. Its output is passed through a decoder with a similar structure as the encoder, with the only difference that skip-connection layers are added as input to the aggregation blocks to preserve useful input information.

Discriminator Our discriminator helps the generator to output more realistic results during training [6]. It is similar to PatchGAN [19], which has proven successful for this task [11, 22, 35]. We use only 3 convolutional layers, as it represents the best trade-off between speed and accuracy. This way, the discriminator classifies 14x14 patches as real or fake, instead of the whole image, thus helping the generator to produce more detailed outputs.

Enhancer Reducing the size of the network comes at the cost of reduced performance, especially in terms of details and colorization. To overcome this issue, we leverage the enhancer module recently proposed by Qu *et al.* [24]. Although originally introduced for image de-hazing, it can still be applied to image de-raining, as proved in our experiments. It contains a 4-scale pyramid pooling that processes feature maps at resolutions reduced by factors of 4, 8, 16 and 32 in parallel, which allows the model to capture information at different levels of detail. A 1x1 convolution is applied to each of them to weight the channels adaptively. After upsampling to the original resolution, they are concatenated with the original feature maps using a 3x3 convolution to produce the final outputs.

Losses Our objective function combines several losses to take into account the different qualities we require for our outputs. The first two are applied to the generator and discriminator and preserve global context information, while the last two work on the enhancer and help with details and colors. The *adversarial loss* [6] is defined as

$$\mathcal{L}_{GAN} = (1 - D(G(\mathcal{A})))^2, \tag{1}$$

where A stands for an image affected by anomalies. It is employed in the interplay between generator and discriminator, where the former tries to fool the latter by producing realistic results. For this reason, it has to be applied directly to the output of the generator, and not on the enhanced results. The same happens with the *feature matching loss* [35]

$$\mathcal{L}_{FM} = \sum_{n=1}^{n_{FM}} \frac{\|D(\mathcal{C})_n - D(\mathcal{A})_n\|_1}{2^{n_{FM}-n}}. \tag{2}$$

Here, n_{FM} stands for the number of selected discriminator layers and C for the clean image. This loss penalizes the distance between the intermediate features extracted by the discriminator from both the ground-truth and the reconstructed images. This way, it ensures that the generator outputs fake samples that are close to reality even at multiple scales. On the other hand, the *perceptual loss* [13]

$$\mathcal{L}_{VGG} = \sum_{n=1}^{n_{VGG}} \frac{\|VGG(\mathcal{C})_n - VGG(E(G(\mathcal{A})))_n\|_1}{2^{n_{VGG}-n}} \tag{3}$$

is applied between enhanced and clean image. It computes the absolute error between activations of neurons at certain layers of a VGG network [32] pretrained on ImageNet [4]. By doing so, we encourage the network to preserve high-level perceptual features when restoring the image. The

Fig. 3: Samples from our dataset.

fidelity loss

$$\mathcal{L}_{FID} = \|\mathcal{C} - E(G(\mathcal{A}))\|_2 \tag{4}$$

measures the L2 pixel-wise difference between the enhanced output and the clean image. The simplicity of our network allows the minimization of the sum of all these losses in a unique step, which differs from [24] where a complex training scheme is proposed. Thus, the objective function to be minimized during training is:

$$\mathcal{L}_G = \mathcal{L}_{GAN} + \mathcal{L}_{FM} + \mathcal{L}_{VGG} + \mathcal{L}_{FID}. \tag{5}$$

B. Synthetic Data

We model our synthetic drops similar to Porav *et al.* [22]. However, we noticed that their syn-drops are in focus and with sharp borders, in contrast to what happens in reality. To synthesize out-of-focus raindrops we introduce multiple shiftings in random directions where our camera anomalies are situated. Moreover, we increase the brightness inside some random raindrop regions, as we experienced when recording our real dataset. An example of our improvements with respect to our reference are displayed in Figure 4.

Images corrupted with our computer-generated raindrops are used to train the network and the resulting weights are re-used as initialization for the training on our real dataset, to provide a task-related initial configuration. If applied with this modality, our synthetic raindrops proved to be more realistic than the ones produced by our reference model [22]. Please refer to Section IV-D for a quantitative evaluation.

Fig. 4: Differences between our synthetic raindrops (right) and our reference model [22] (left). Ours are out of focus and with different brightness with respect to the background, which is more common for fixed-focus cameras used for mobile robotics.

IV. EXPERIMENTS

A. Dataset

Here we present the device used to collect our dataset, shown in Figure 5. We used the Alphasense Core multi-camera sensor[1] manufactured by Sevensense Robotics. The two front cameras Sony IMX-287 are spaced by a baseline of 140 mm and are synchronized to record images of the same scene, to ensure we have both the ground-truth clean image and the affected one. Instead of directly spoiling one camera lens, we have built a frame to sustain two pieces of glass, each placed in front of every camera. This way, we could affect only one of the two cameras with the anomaly, corrupting one glass panel and keeping the other clean. The glasses are mainly steady, with small variations in the angles with respect to the camera axes, which enrich the variability of the dataset. Moreover, the filters can be easily replaced when too dirty, making the whole data-collection process very fast. With this sensor, we collected 5613 image pairs of size 540x360 (after cropping) walking around the cities of Zurich and Milan, at different times of the day and with several weather conditions, with the aim to increase the variation of the scene. The majority of the samples are spoiled by water sprayed directly on the glass, mimicking raindrops with diverse shapes and sizes. Furthermore, we interchanged between left and right camera for the ground truth image due to a slight difference in illumination levels between the two, which automatically makes the training on

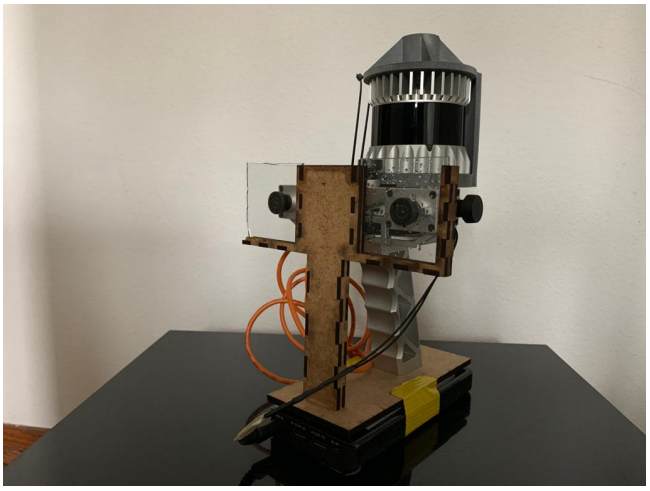

Fig. 5: Setup of our device for data collection. A wooden frame allows to place a glass panel in front of each of the two anterior cameras of the Alphasense Core multi-camera sensor. One is kept clean and the other one is corrupted with anomalies.

[1] https://github.com/sevensense-robotics/alphasense_core_manual

(a) Input (b) G (c) G+E (d) G+E+A (e) Ground Truth

Fig. 6: Outputs of the different stages of our architecture. Rows correspond to easy, medium or difficult task, depending on how raindrops affect the images.

(a) Input (b) Ground Truth (c) DeRaindrop [23] (d) Porav *et al.* [22] (e) Ours

Fig. 7: Visual comparison between our results and the state of the art.

this dataset more robust to this intrinsic noise. The images are rectified [8] to reproduce the same scene, except for the occluded regions. In Figure 3 some samples from our dataset are displayed.

B. Training Details

We use the standard training scheme for image-to-image translation problems [11]. The discriminator is trained on a previously de-rained image and then the generator and the enhancer are updated in one step, minimizing our objective function, in contrast with Qu *et al.* [24], where a dedicated training scheme is designed to incorporate the enhancer. We adopt Adam [17] optimizer with learning rate $\lambda = 0.0002$, batch size of 8 and train for 200 epochs, using early stopping on validation set with patience 10. Experiments have been performed on an Nvidia RTX 2080 GPU and an Intel Core i7 10th Gen CPU.

C. Ablation Study

In this section, we state dissimilarities and improvements across incremental configurations of our architecture. We will consider three variants, tested on the dataset provided by Qian *et al.*:

- G is the output of the generator, without enhancement and aggregation scheme;
- G + E is the image produced by the combination of enhancer module and generator;
- G + E + A incorporates the aggregation scheme into

Method	SSIM	PSNR
Input	0.851	24.09
G	0.869	27.12
G+E	0.898	29.15
G+E+A	**0.909**	**29.84**

TABLE I: Results on the dataset by Qian *et al.* [23] produced by different stages of our architecture. The experiments are run with synthetic initialization.

Initialization	SSIM	PSNR
Raw	0.760	19.34
Random	0.787	22.36
Synthetic [22]	0.791	22.15
Our synthetic	**0.813**	**24.77**

TABLE II: Effect of different initialization strategies on our full architecture.

the generator.

Visual results can be found in Figure 6. For the easy task – top row – where the input image is faintly affected by raindrops, G is sufficient to perform the anomaly mitigation. On the medium task – central row – instead, brighter spots can be clearly distinguished on the right wall of G, as clear signs of attempts to remove raindrops. On the other hand, G+E, although slightly over-colored, is able to produce adequate results, because of the input information directly injected into the enhancer. Nevertheless, it fails on the hard task – bottom row – where the raindrops introduce too much noise into the input image and, consequently, into the enhancer. In this case, the counter-effect produced by the generator needs to be stronger. With the introduction of the aggregation scheme (G+E+A), the network leverages feature maps learned at previous stages to retain relevant information and output successfully de-rained images.

On a quantitative point of view, as reported in Table I, the visual results are confirmed by the changes in Structural SIMilarity (SSIM) and Peak Signal-To-Noise Ratio (PSNR), two widely-accepted metrics for this task. The enhancer module leads to the highest delta in both metrics, stating the importance of input information to output realistic images. The aggregation scheme further improves the results by providing the generator with more power to balance input noise introduced into the enhancer by heavily corrupted samples.

D. Impact of synthetic data

To investigate how realistic and effective our synthetic raindrops are, we corrupt images from existing datasets ([22, 23]) and use them to pre-train our architecture. Then, we initialize the training on real data with the weights learned during the pre-training phase. We compared this setup with the same procedure applied to the synthetic dataset provided by Porav *et al.* [22] and with a random initialization drawn from a Gaussian distribution with $\mu = 0$ and $\sigma = 0.02$. Results are shown in Table II. Our synthetic initialization outperformed both the others, proving on one hand that artificial anomalies can be useful for this task by improving the random initialization; on the other hand that ours are more realistic with respect to our reference model.

Method	SSIM	PSNR	TIME GPU(s)	TIME CPU(s)
DeRaindrop [23]	**0.921**	31.50	0.262	4.317
Porav *et al.* [22]	0.902	**31.55**	0.091	2.780
Ours	0.909	29.84	**0.006**	**0.542**

TABLE III: Comparison with the state of the art. Our architecture largely outperforms the others in terms of time taken for a forward pass, while still producing comparable results.

E. Comparison with state of the art

Figure 7 shows visual outputs from our architecture compared to the state of the art. Again, we distinguish between easy, medium and hard task. In the first two rows, all reconstructed images present little or no difference between each other and the ground truth. Only in the hard task - bottom row - looking carefully, the output of DeRaindrop looks cleaner and with less artifacts. This slight difference is reflected in the metrics, as depicted in Table III. Our model ranks second in SSIM and third in PSNR. However, our architecture outperforms the others by a great margin in terms of average time required to de-noise the images. With just *6 milliseconds* needed to run on the GPU, we are 15 to 40 times faster then the other works. And 5 to 8 times faster on the CPU, where it takes around half a second to run. This means that our network is the most suitable for a practical application on real robots, as it represents a good trade-off between speed and effectiveness.

V. CONCLUSION

In this work we have presented our framework for fast camera-anomaly deletion and image restoration. We added aggregation modules and an enhancing block to a shallow GAN architecture to preserve state-of-the-art performance, while achieving real-time capabilities, which is our main focus. The proposed method can be run as a pre-processing step in every pipeline that involves classification, segmentation or localization tasks for autonomous robots in challenging conditions, where rain or dirt could spoil the camera lens and degrade performance. Our pipeline is suitable to be combined with other architectures in a real-time fashion, also considering the limited computing power of autonomous robots.

In the future, we will inspect other anomalies, e.g. scratches, dust and fingerprints and will combine our architecture with a classification pipeline. This way, the robot could distinguish whether the camera is clean or affected by anomalies and, if that is the case, run it through the mitigation pipeline. Another classification stage would decide which anomaly is corrupting the images and leverage the corresponding pre-trained model.

REFERENCES

[1] Harold C Burger, Christian J Schuler, and Stefan Harmeling. Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE conference on computer vision and pattern recognition*, pages 2392–2399. IEEE, 2012.

[2] Bolun Cai, Xiangmin Xu, Kui Jia, Chunmei Qing, and Dacheng Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11):5187–5198, 2016.

[3] Dongdong Chen, Mingming He, Qingnan Fan, Jing Liao, Liheng Zhang, Dongdong Hou, Lu Yuan, and Gang Hua. Gated context aggregation network for image dehazing and deraining. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1375–1383. IEEE, 2019.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[5] David Eigen, Dilip Krishnan, and Rob Fergus. Restoring an image taken through a window covered with dirt or rain. In *Proceedings of the IEEE international conference on computer vision*, pages 633–640, 2013.

[6] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

[7] Jinwei Gu, Ravi Ramamoorthi, Peter Belhumeur, and Shree Nayar. Removing image artifacts due to dirty camera lenses and thin occluders. In *ACM SIGGRAPH Asia 2009 papers*, pages 1–10. 2009.

[8] Richard I Hartley. Theory and practice of projective rectification. *International Journal of Computer Vision*, 35(2):115–127, 1999.

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[10] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1125–1134, 2017.

[12] Tai-Xiang Jiang, Ting-Zhu Huang, Xi-Le Zhao, Liang-Jian Deng, and Yao Wang. Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Transactions on Image Processing*, 28(4):2089–2102, 2018.

[13] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.

[14] Michael I Jordan. Serial order: A parallel distributed processing approach. In *Advances in psychology*, volume 121, pages 471–495. Elsevier, 1997.

[15] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. Stereo video deraining and desnowing based on spatio temporal frame warping. In *2014 IEEE International Conference on Image Processing (ICIP)*, pages 5432–5436. IEEE, 2014.

[16] Jin-Hwan Kim, Jae-Young Sim, and Chang-Su Kim. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *IEEE Transactions on Image Processing*, 24(9):2658–2670, 2015.

[17] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[18] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, pages 4770–4778, 2017.

[19] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *European conference on computer vision*, pages 702–716. Springer, 2016.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[21] Xiaohong Liu, Yongrui Ma, Zhihao Shi, and Jun Chen. Griddehazenet: Attention-based multi-scale network for image dehazing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7314–7323, 2019.

[22] Horia Porav, Tom Bruls, and Paul Newman. I can see clearly now: Image restoration via de-raining. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 7087–7093. IEEE, 2019.

[23] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2482–2491, 2018.

[24] Yanyun Qu, Yizi Chen, Jingying Huang, and Yuan Xie. Enhanced pix2pix dehazing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8160–8168, 2019.

[25] Dongwei Ren, Wangmeng Zuo, Qinghua Hu, Pengfei Zhu, and Deyu Meng. Progressive image deraining networks: A better and simpler baseline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3937–3946, 2019.

[26] Wenqi Ren, Si Liu, Hua Zhang, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale convolutional neural networks. In *European conference on computer vision*, pages 154–169. Springer, 2016.

[27] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2018.

[28] Weihong Ren, Jiandong Tian, Zhi Han, Antoni Chan, and Yandong Tang. Video desnowing and deraining based on matrix decomposition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4210–4219, 2017.

[29] Martin Roser and Andreas Geiger. Video-based raindrop detection for improved image registration. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 570–577. IEEE, 2009.

[30] Martin Roser, Julian Kurz, and Andreas Geiger. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. In *Asian Conference on Computer Vision*, pages 235–244. Springer, 2010.

[31] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning internal representations by error propagation. Technical report, California Univ San Diego La Jolla Inst for Cognitive Science, 1985.

[32] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[33] Abhinav Valada, Rohit Mohan, and Wolfram Burgard. Self-supervised model adaptation for multimodal semantic segmentation. *International Journal of Computer Vision*, pages 1–47, 2019.

[34] Tianyu Wang, Xin Yang, Ke Xu, Shaozhe Chen, Qiang Zhang, and Rynson WH 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.

[35] Ting-Chun Wang, Ming-Yu Liu, Jun-Yan Zhu, Andrew Tao, Jan Kautz, and Bryan Catanzaro. High-resolution image synthesis and semantic manipulation with conditional gans. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8798–8807, 2018.

[36] Yanyan Wei, Zhao Zhang, Haijun Zhang, Richang Hong, and Meng Wang. A coarse-to-fine multi-stream hybrid deraining network for single image deraining. In *2019 IEEE International Conference on Data Mining (ICDM)*, pages 628–637. IEEE, 2019.

[37] Dong Yang and Jian Sun. Proximal dehaze-net: A prior learning-based deep network for single image dehazing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 702–717, 2018.

[38] Shaodi You, Robby T Tan, Rei Kawakami, Yasuhiro Mukaigawa, and Katsushi Ikeuchi. Adherent raindrop modeling, detection and removal in video. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1721–1733, 2015.

[39] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3194–3203, 2018.

[40] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *IEEE transactions on circuits and systems for video technology*, 2019.

[41] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. In *2006 IEEE international conference on multimedia and expo*, pages 461–464. IEEE, 2006.

[42] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.