Data-Driven Single Image Deraining: A Comprehensive Review and New Perspectives

Zhao Zhang, *Senior Member, IEEE*, Yanyan Wei, Haijun Zhang, *Senior Member, IEEE*, Yi Yang, *Senior Member, IEEE*, Shuicheng Yan, *Fellow, IEEE*, and Meng Wang, *Fellow, IEEE*,

Abstract—Due to the powerful fitting ability of neural networks and massive training data, data-driven Single Image Deraining (SID) methods have obtained significant performance, and most of the existing studies focus on improving the deraining performance by proposing different kinds of deraining networks. However, the generalization ability of current SID methods may still be limited in the real scenario, and the deraining results also cannot effectively improve the performance of subsequent high-level tasks (such as object detection). The main reason may be because the current works mainly focus on designing new deraining neural networks, while neglecting the interpretation of the solving process. To investigate these issues, we in this paper re-examine the three important factors (i.e., *data*, *rain model* and *network architecture*) for the SID task, and specifically analyze them by proposing new and more reasonable criteria (i.e., *general vs. specific*, *synthetical vs. mathematical*, *black-box vs. white-box*). We will also study the relationship of the three factors from new perspectives of data, and reveal two different solving paradigms (*explicit vs. implicit*) for the SID task. Besides, we also profoundly studied one of the three factors, i.e., *data*, and measured the performance of current methods on different datasets via extensive experiments to reveal the effectiveness of SID data. Finally, with the above comprehensive review and in-depth analysis, we also draw some valuable conclusions and suggestions for future research.

Index Terms—Comprehensive review; data-driven single image deraining; data; rain model; network architecture; in-depth analysis.

1 INTRODUCTION

SINGLE image deraining (SID) or rain removal from a single image has been emerging as an important task in different areas of image processing, pattern recognition and computer vision, which has a wide range of applications relating to single image processing and restoration. However, the quality of the images captured in rain days by outdoor vision system, e.g., autonomous driving, person/vehicle tracking and surveillance, are usually degenerated by the rain streak, rain drop and rain accumulation (see Fig.1). This will directly degrade the subsequent high-level vision tasks, e.g., object detection [1, 2], image recognition [3, 4] and saliency detection [5, 6]. Besides, due to the irregular and complex rain information in practice and the ill-posed properties, the task of SID, which aims at estimating the rain-free background from a single image degraded by rain streak, rain drop or (and) rain accumulation, is still a challenging and unmanageable issue to date. Unlike the video deraining that can leverage temporal redundancy and dynamics of rain, SID mainly exploits the spatial information of neighboring pixels, and the visual properties of rain

Fig. 1: The first row denotes different types of visibility degradation caused by rain, where (a) rain streak, (b) rain drop, and (c) rain accumulation. The second row shows the visual impact of rain on traffic and surveillance, namely, from the vehicle perspective (d), and monitoring perspective (e).

- Z. Zhang, Y. Wei and M. Wang are now with the School of Computer Science and Information Engineering; also with the Key Laboratory of Knowledge Engineering with Big Data (Ministry of Education); also with the Intelligent Interconnected Systems Laboratory of Anhui Province, Hefei University of Technology, Hefei 230601, China.
E-mail: cszhang@gmail.com, weiyanyan@mail.hfut.edu.cn, eric.mengwang@gmail.com.
- H. Zhang is with the Department of Computer Science, Harbin Institute of Technology (Shenzhen), Xili University Town, Shenzhen, China.
E-mail: hjzhang@hit.edu.cn.
- Y. Yang is with the Centre for Artificial Intelligence, University of Technology Sydney, Sydney, NSW, Australia.
E-mail: Yi.Yang@uts.edu.au.
- S. Yan is with Sea AI Lab (SAIL), Singapore; also with the National University of Singapore, Singapore 117583.
E-mail: shuicheng.yan@gmail.com.

Manuscript received July 1, 2021; revised August 1, 2021.

and background scenes. As such, the SID tasks will be confronting with more difficulties than the video deraining task [7, 8, 9, 10].

To study the SID task, existing methods can be generally divided into two basic categories, i.e., traditional non-data-driven methods (such as filter-based, prior-based and model-based ones) [11, 12, 13, 14, 15, 16, 17] and deep learning-based data-driven methods [18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28]. For filter-based methods, researchers proposed to filter the rain image to obtain the rain-removed image [11, 12]. For prior- and model-based type, some classical methods include sparse representation [13, 14], low-rank representation [15, 16] and gaussian mixture model [17]. Specifically, sparse representation aims at finding the sparsest representation of the input data in the form of a linear combination

of basic elements as well as the basic elements themselves. These elements are called atoms which compose a dictionary. Low-rank representation is a rank minimization problem, in which the cost function measures the fit between a given data matrix and an approximating matrix (i.e., optimization variable), subject to a constraint that the approximating matrix has a reduced rank.

With the rapid development of deep learning and deep neural networks [3, 29, 30], data-driven deraining methods have obtained impressive performance improvement over those traditional deraining methods. In this field, Rain14000 [18] and Rain100H/Rain100L [19] datasets were firstly proposed in 2017 for the task of data-driven SID. After that, more and more synthetic SID datasets have been constructed, and meanwhile the study on data-driven SID enters into a new period. In addition to synthesizing rain image data, different kinds of side information have been developed based on the original data, such as rain streak, rain mask, rain density and image depth. By designing specific SID methods [19, 20, 21, 23, 26] that can take advantage of these side information, state-of-the-art deraining performance has been obtained. It is noteworthy that, for the topic on SID, there are three important influencing factors that have to be discussed, i.e., **Data**, **Rain model** and **Network architecture**:

- **Data.** Data are the core factor of all the data-driven SID methods. Massive training data can usually enable the deraining networks to deliver significant SID performance. However, due to the lack of image pairs in the real world, most current SID methods use the synthetic data for both training and testing.

- **Rain model.** Rain model mainly investigates how to model the rain mathematically and how to synthesize the rain image. Since rain is a complex optical phenomenon and the rain removal problem is usually ill-posed, rain model indicates how we view the phenomenon and how we wish to solve the problem.

- **Network architecture.** The deraining network structure is the most changeable and charming factor in the SID task. As such, researchers have designed new modules and network architectures of different properties to improve the deraining performance.

In recent studies, the performance on the SID task is getting better, however this is usually in the case of processing synthetic image data rather than real data. Specifically, the generalization ability of current SID methods is still limited in the real-world scenario. Furthermore, the rain removal task still cannot effectively improve the subsequent high-level vision tasks (e.g., object recognition and detection [2, 31]). To this end, several researchers have conducted experiments to enable the SID methods to have better application potential. For example, a large-scale benchmark dataset called MPID [32], which contains both synthetic and real rain images with various rain types, was recently proposed to evaluate the performance of existing SID methods. The experimental evaluation and result analysis can reveal the performance gap between the synthetic and real-world data to some extent.

However, it is noteworthy that most of the existing studies and analysis on SID, reviewed in [32, 33, 34, 35], mainly focus on measuring the *effectiveness of SID methods*, and discuss the three influencing factors independently without clearly discussing their relationships and mutual influence on the task of SID. In this paper, we therefore re-define the classification of the three factors, figure out the relations among them, and reveal different solving paradigms for addressing the SID tasks. Besides, we also proposed to evaluate the *effectiveness of SID data* based on new evaluation criteria. Overall, the main contributions of this paper are summarized as follows:

- 1) **New divisions on the three factors:** We provide new, reasonable and easily understandable divisions on the data (*general vs. specific*), rain model (*synthetical vs. mathematical*) and network architecture (*black-box vs. white-box*) to study the three influencing factors on SID.
- 2) **In-depth analysis:** We figure out the relationships between the three factors and reveal two different solving paradigms (*explicit vs. implicit*) for SID. Besides, we also analyze, survey and category the current SID methods according to different properties and perspectives.
- 3) **Novel experimental design:** We design novel forms of simulations to evaluate the effectiveness of SID data by providing novel quantitative evaluation criteria. To the best of our knowledge, this is the first work to evaluate and quantify the existing mainstream SID datasets.
- 4) **Instructive conclusion:** Based on the above analysis and results, we derive some practical and instructive conclusions on choosing data appropriately, which can help the related researchers who are confused about choices.

The rest of this paper is organized as follows. In Section 2, we introduce the related survey papers on rain removal task and describe the difference to our work. Section 3 illustrates the three influencing factors, i.e., data, rain model and network architecture from new divisions. Section 4 explores the relationship between the three factors and discusses the solving paradigms of SID methods. In Section 5, we summarize the recent related data-driven SID methods, in terms of training strategy, network pipeline, domain knowledge, data preprocessing and objective function. In Section 6, we design novel forms of experiments to rank the public paired datasets and provide a detailed analysis. Finally, Section 7 derives some valuable conclusions and discusses some future directions.

2 RELATED WORK

It is noteworthy that there are four relevant survey papers on SID, proposed in recent years, i.e., [32, 33, 34, 35]. These surveys discuss and analyze the SID methods from different perspectives, and we summarize the four related surveys in Table 1. From the comparison, we can conclude that:

- 1) Wang *et al.* [35] provided an inexhaustive review for the current SID techniques and mainly categorized them into three classes, that is, early filter-based, conventional prior-based and recent deep learning-based data-driven approaches. Furthermore, inspired by the rationality of current deep learning-based method PReNet [25] and insightful characteristics underlying rain shapes, they also build a specific coarse-to-fine deraining network architecture, which can finely deliver rain structures and progressively removes the rain streaks from input image.

- 2) Li *et al.* [32] presented a comprehensive study and evaluation of 6 existing SID algorithms, and introduced a new largescale benchmark dataset (MPID) that contained both synthetic and real-world rain images of various rain types. To be specific, this MPID dataset contains three types of synthetic rain models (rain streak, rain drop and rain mist), as well as a rich variety of evaluation criteria (two full- and three no-reference objective evaluations, subjective evaluation and task-specific evaluation). The evaluation and analysis can indicate the performance gap between the synthetic rain images and real-world images to some extent.

- 3) Wang *et al.* [33] simultaneously reviewed both the video deraining and SID. Specifically, for the parts on SID, they mainly

TABLE 1: A brief introduction to four related survey papers on SID.

Paper	Description	Key Words	Year
Wang <i>et al.</i> [35]	Shortly discuss three categories of deraining approaches: filter-based, conventional prior-based, and deep learning-based approaches, and then proposing a new SID method.	Single image deraining, A new SID method	2020
Yang <i>et al.</i> [34]	Discuss two categories of deraining approaches: model-based and data-driven approaches from many perspectives.	Single image deraining, Detailed analysis	2020
Wang <i>et al.</i> [33]	Present a comprehensive review for current rain removal methods for video and a single image.	Single image/Video deraining, Lots of experiments, Public github repository	2019
Li <i>et al.</i> [32]	Present a comprehensive study and evaluation of existing single image deraining algorithms, using a new largescale benchmark consisting of both synthetic and real-world rain images of various rain types.	A new benchmark dataset, Lots of experiments, Public Github repository	2019

divide the existing methods into three categories according to the ways of problem solving, i.e., filter-based, prior-based and deep learning-based data-driven methods. However, the data-driven SID methods only occupy a relatively small proportion of the survey paper. Specifically, the analysis part does not leave much space to introduce the data-driven methods, and the experimental part only evaluates seven data-driven methods, which are only trained on three datasets and tested on five datasets. Based on the evaluation results, they summarized the deficiencies of current deraining methods, and also presented some remarks to illumine some meaningful future research directions along this line.

4) Different from [33], Yang *et al.* [34] mainly focuses on discussing the SID problem. Specifically, they divided the existing SID methods into model-based and data-driven methods, and further divide the deep learning-based data-driven methods into four sub-categories, i.e., deep convolutional neural networks (CNN-based), generative adversarial networks (GAN-based), semi-/unsupervised learning-based and benchmarks. Subsequently, they described the related methods in the four sub-categories in detail, including the network architectures, basic blocks, loss functions and datasets. To evaluate the performance of each method, they selected eight deep learning-based data-driven methods and evaluated them using three synthetic datasets. However, this survey paper did not provide the detailed evaluations, but only provided a graph of performance. In addition, they also proposed some future directions, including the integration of physics model and real images, rain modeling, evaluation methodology, and more related tasks and real-world applications.

Remarks. After a brief introduction to the existing four survey papers, we can present the difference between our work and theirs. The survey [35] tends to proposing a new SID method and spends less space on reviewing current methods, so we will not discuss more with this work here. Next, we will mainly compare our work with [32, 33, 34]. Firstly, the experiments in [32] focused on fairly evaluating the performance of SID methods using a new benchmark dataset. It clearly reveals the shortcomings of current rain removal issue, including the high complexity of the rain removal tasks, lack of appropriate evaluation metrics, poor generalization on real image, and little help for upstream tasks. Note that, in a traditional sense, [32] is not a real survey paper, since it is groundbreaking work, but it only reveals some new features of SID tasks. Inspired by [32], we have designed more experiments to reveal the two key issues mentioned above, which have not been paid much attention in current rain removal researches. From the contents of [33] and [34], it is clear that they both are lack of the investigation and analysis on the relationships between used data and network structures. Particularly,

their conducted experiments cannot deduce the following several aspects: (1) superiority of different data, since appropriate data in training phase can enhance the representation and generalization abilities of the networks, which can serve better for the real scenario and real emerging applications; (2) pros and cons of different solving paradigm, while we investigate it carefully via experiments on side information, different operations, and explicit or implicit paradigms. Specifically, different from [33] and [34] that also focus on the video deraining and traditional SID methods, we mainly focus on the data-driven SID task, where the related methods are categorized from the perspective of data into general and specific ones based on the relationship between data and networks. Note that the conclusions of the two categories are in fact derived by a large number of experimental verifications, and moreover some discussions and suggestions are also provided on this basis. We believe this survey paper will be able to provide a more insight view for understanding the data-driven SID methods from a new perspective of data.

One of the most important purposes of the SID task is that it can be potentially used in various emerging applications of realistic scenarios, so the deraining results on real rain images (without ground truth) will determine the actual rain removal ability of each method. However, most existing data-driven SID methods are supervised, which need all paired data for training, so they cannot be trained directly on real images. Specifically, supervised methods are usually trained on the synthetic datasets and are tested on real rain images, i.e., the test result on real images represents the generalization ability of each method. Note that this result will be directly determined by the properties of synthetic datasets, including the rain direction, density and shapes, etc. In other words, to solve the real rain removal task well, the characteristics of synthetic data should be closer to those of the real rain as much as possible. However, most existing data-driven SID methods usually focus on the performance evaluation on synthetic datasets to reflect the deraining ability of the proposed rain models, and are only tested on some real rain images to evaluate the so-called generalization ability. Clearly, this evaluation process is understandable and can be performed easily, however, the key and most difficulty point is actually how to define a “best” synthetic dataset. In other words, how to select data and rain model appropriately, and then design the corresponding network structures and optimization strategies will be worthy of discussions. But to the best of our knowledge, there is no prior study or discussion on this topic yet.

Fig. 2: Illustration of representative general and specific datasets for SID, where we also show some samples of image pairs.

3 THE THREE FACTORS

In this section, we focus on discussing the divisions of the three factors based on data (*general vs. specific*), rain model (*synthetical vs. mathematical*) and network architecture (*black-box vs. white-box*) by using new and more reasonable criteria.

Fig. 3: Examples of paired data, including the rain image and corresponding clear image (the first and second in the top row), and some types of side information (see others).

3.1 Data

As far as we know from the literature, 20 paired datasets are proposed in current SID methods from 2017 to 2021. These datasets contain image pairs (i.e., rain image (**O**) and a clean ground-truth image (**B**)), as well as some side information, e.g., rain streak (**S**), rain mask (**M**), image depth (**F**), rain density (**D**), transmission map (**T**), and atmospheric light (**A**). According to whether the data contain extra side information, we propose to divide the current SID data into *general data* and *specific data*. Fig. 2 illustrates some representative general and specific datasets. Fig. 3 shows an example of paired images with side information for SID. Tabel 2 summarized the 20 mainstream paired datasets.

3.1.1 General data

The datasets of this kind only contain the image pairs (i.e., rain image and the corresponding ground-truth image), without any side information. Representative general datasets include:

- **Rain800** [22]: 800 clean images are randomly chosen from

the UCID [42] and BSD [43]. The authors add the rain streaks into these clear images using Photoshop [44]. After that, 700 pairs are chosen for training, while 100 pairs are chosen for testing.

- **Rain14000** [18]: 1,000 clean images are randomly chosen from the UCID [42], BSD [43] and Google. By using Photoshop [44], each clean image was used to generate 14 rain images with different streak orientations and magnitudes. They randomly select 12,600 image pairs for training and 1,400 for testing.

- **Rain12** [17]: It is only a test dataset that contains 12 rain images with one type of rain streak.

- **SPA-Data** [27]: 170 real rain videos are captured by using a mobile phone or collected from the Internet. Using the video deraining method [27], 29,500 image pairs are generated, split into 28,500 for training and 1,000 for testing.

- **MPID** [32]: It contains three types of rain, including the rain streak, raindrop, and rain&mist. The training set has 2,400, 861, and 700 image pairs, while the test set has 200, 149, and 70 image pairs.

- **Auto100L/Auto800** [39]: These two datasets are generated by GAN-based model based on Rain100L [19] and Zhang *et al.* [22]. By automatically adding rain streak under unsupervised mode, it makes the rain streaks have more various shapes and directions which will be adapted to the real rain streaks than original datasets that are manually created by Photoshop.

- **RainDrop** [26]: Using a camera with two pieces of glasses, the authors consist of 1,119 pairs of raindrops images with various background scenes, where 861 images are employed for training.

- **RainDS** [38]: The first real-world deraining dataset includes different types of rain (rain streak and raindrop) captured in various lighting conditions and scenes. It contains 250 real-world and 1,200 synthetic rain image pairs.

- **QSM-D-Data** [36]: It follow [41] to prepare training dataset which contains 20,800 pairs. The synthetic rain images are synthesized with rendered rain layer and the ground truth by using the screen blend mode.

- **Rain-II** [40]: The authors synthesize 400 images by [44] where the synthetic rain images possess apparent vapor, which is named as Rain-II. This dataset is only used for testing.

- **Rain20** [28]: By using the skill of [45], twenty rain images with complexity and multiformity of the rain streaks are synthe-

TABLE 2: The 20 mainstream paired datasets for the SID task, which were proposed from 2017 to 2021. * denotes the name of the dataset, defined by us. • denotes that this dataset is a real-world paired dataset

Method	Dataset	Train/Test	Type	Side Information	Public	Year
QSM [36]	QSM-Data*	20,800/100	General	-	No	2021
RICNet [37]	RainLevel5	26,870/2,880	Specific	rain speed/fog	No	2021
CCN [38]	RainDS	1,015/435	General	-	No	2021
DerainCycleGAN [39]	Auto100L	200/100	General	-	No	2021
DerainCycleGAN [39]	Auto800	700/100	General	-	No	2021
Wang <i>et al.</i> [40]	Rain-II	-/400	General	-	No	2020
Li <i>et al.</i> [21]	NYU-Rain	13,500/2,700	Specific	rain streak/ts. map/as. light	No	2019
Li <i>et al.</i> [21]	Outdoor-Rain	9,000/1,500	Specific	rain streak/ts. map/as. light	Partial	2019
SIRR [28]	Rain20*	-/20	General	-	Yes	2019
DAF-Net [20]	RainCityscapes	9,432/1,188	Specific	depth map	Yes	2019
Li <i>et al.</i> [32]	MPID	3,961/419	General	-	Partial/Cropped	2019
SPANet [27]	•SPA-Data*	28,500/1,000	General	-	Cropped	2019
DDC-Net [41]	DDC-Data*	10,400	Specific	rain streak	No	2019
DID-MDN [23]	Rain12000*	12,000/4,000	Specific	rain density	Yes	2018
Qian <i>et al.</i> [26]	RainDrop*	861/249	General	-	Yes	2018
Fu <i>et al.</i> [18]	Rain14000*	9,100/4,900	General	-	Yes	2018
ID-CGAN [22]	Rain800*	700/100	General	-	Yes	2017
JORDER [19]	Rain100H	1,800/100	Specific	rain mask/rain streak	Yes	2017
JORDER [19]	Rain100L	200/100	Specific	rain mask/rain streak	Yes	2017
Li <i>et al.</i> [17]	Rain12*	-/12	General	-	Yes	2016

sized. These testing images contain two different scenarios, that is, sparse rain streaks and dense rain streaks, and each scenario contains ten test images.

3.1.2 Specific data

Besides the essential image pairs, the datasets of this kind also contain different side information, e.g., rain streak, rain mask, image depth, rain density, transmission map, and atmospheric light. Some representative specific datasets include:

- **Rain100H** [19]: 1,900 clear images are selected from BSD [43]. The rain streaks are synthesized by photorealistic rendering techniques [46] or adding simulated sharp line streaks along a certain direction. Rain100H has 1,800 image pairs for training with five streak directions and 100 image pairs for testing.

- **Rain100L** [19]: 300 clear images are selected from BSD [43]. Rain100L has 200 image pairs with only one type of rain streak for training and 100 image pairs for testing.

- **Rain12000** [23]: By using the Photoshop software [47], each image from 4,400 clear images is selected to synthesize three rain density levels (i.e., light, medium, and heavy) rain images. After that, 12,000 image pairs are used to train, while 1,200 image pairs are used for testing.

- **RainCityscapes** [20]: 295 clear images are selected from Cityscapes [48] to synthesize rain images by using the camera parameters and scene depth information. The training set has 262 images, while the test set has 33 images.

- **NYU-Rain** [21]: 16,200 clear images are chosen from NYU-Depthv2 [49] to rendering synthetic rain streaks and rain accumulation effects based on the provided depth information. 13,500 image pairs are then used for training, with 2,700 for testing.

- **Outdoor-Rain** [21]: The clear images are chosen from [26] to synthesize the rain images by the same method as NYU-Rain. It contains 9,000 training samples and 1,500 test samples.

- **DDC-Data** [41]: The clean images are from BSD [43], and the rain streak layers are generated following [44] with varying intensities, orientations, and overlaps. Finally, a dataset containing 10,400 image pairs are fused in the way of screen blending.

- **RainLevel5** [37]: The clean images are from Cityscapes-dataset [48] and 5 levels (25, 50, 75, 100 and 200mm/hr) of rain images with corresponding fog rendering is synthesized. The dataset includes 26,870 pairs for training and 2,880 pairs for testing.

3.2 Rain Model

A rain model mainly indicates two issues: 1) how to model the rain mathematically, and 2) how to synthesize the rain image. However, in fact only some rain models can be used to synthesize the rain image, while others are only the mathematical models of rain. As such, according to whether it can generate data, we divide current rain models into *synthetical rain model* and *mathematical rain model*. Fig. 4 shows an example of generating a rain image by using a synthetic rain model.

Fig. 4: The synthetic process of a rain image based on Eqn. (4).

3.2.1 Synthetical rain model

Note that both the synthetical and mathematical rain models can model the rain, while only the synthetical rain model can be used to synthesize the rain image. In what follows, we will introduce some representative synthetical rain models in detail:

• **Rain Streak Model (RSM)**. This is the most fundamental rain model, and the formulation is described as:

$$\mathbf{O} = \mathbf{B} + \mathbf{S}, \quad (1)$$

where $\mathbf{O}$ denotes a rain image, which can be decomposed into a rain streak component $\mathbf{S}$ and a clean background $\mathbf{B}$. RSM is widely employed in data-driven SID methods, such as [22, 23, 27, 50, 51, 52]. Datasets like Rain800, Rain14000, and Rain12000 are synthesized based on this model. Note that, from Eqn. (1) to Eqn. (14), we use uniform mathematical characters.

• **Screen Blend Model (SBM)**. Different from the above RSM, SBM is a non-linear composite model which is formulated as:

$$\mathbf{O} = \mathbf{B} + \mathbf{S} - \mathbf{B} \odot \mathbf{S}, \quad (2)$$

where $\odot$ is element-wise multiplication. Unlike the additive composite character in Eqn. (1), the background and rain streak influence the appearance of each other. [41] uses SBM as the rain model and synthesizes the DDC-Data.

• **Image Depth Model (IDM)**. According to [53], the visual intensity of a rain streak depends on the scene depth d from the camera to the underlying scene objects behind the rain. The mathematical model is formulated as follows:

$$\mathbf{O} = \mathbf{S} + \alpha \mathbf{F} + (1 - \mathbf{S} - \mathbf{F}) \odot \mathbf{B}, \quad (3)$$

where α is the atmospheric light, which is assumed to be a global constant [54], $\mathbf{F}$ denotes the fog layer with a range of $[0,1]$, $\mathbf{1}$ is a matrix of ones. [20] uses IDM as the rain model and synthesizes the RainCityscapes.

• **Heavy Rain Model (HRM)**. Heavy Rain often causes rain accumulation and visual effects with haze. Rain accumulation or veiling can affect the results from water particles in the atmosphere and distant rain streaks that cannot be seen individually. The mathematical model can be expressed as follows:

$$\mathbf{O} = \mathbf{T} \odot \left(\mathbf{B} + \sum_{i=1}^n \tilde{\mathbf{S}}_i \right) + (1 - \mathbf{T}) \odot \mathbf{A}, \quad (4)$$

where each $\tilde{\mathbf{S}}_i$ is a layer of rain streaks with the same direction, i is the index of the rain-streak layers, and n is the maximum number of rain streak layers. $\mathbf{T}$ denotes the transmission map introduced by the scattering process of the tiny water particles, and $\mathbf{A}$ is the global atmospheric light of the scene. [21] uses HRM as the rain model and synthesizes the NYU-Rain and Outdoor-Rain datasets.

• **Rain Accumulation Model (RAM)**. RAM considers rain accumulation effect like Heavy Rain Model except with adding rain mask information. The mathematical model is expressed as:

$$\mathbf{O} = \mathbf{T} \odot \left(\mathbf{B} + \sum_{i=1}^n \tilde{\mathbf{S}}_i \odot \mathbf{M} \right) + (1 - \mathbf{T}) \odot \mathbf{A}, \quad (5)$$

where most of the notations are similar to Eqn. (4) except $\mathbf{M}$. $\mathbf{M}$ is the rain mask image which is a binary image, i.e., 1 for rain, and 0 denotes rain-free in pixel. [19] uses RAM as the rain model and synthesizes the Rain100H and Rain100L datasets.

3.2.2 Mathematical rain model

We introduce the representative mathematical models of rain:

• **Rain Residual Model (RRM)**. This model is similar to the RSM. The advantage of using the rain residuals is to obtain a

cleaner background than using the rain streaks during training, but this formula cannot be used to synthesize rain images:

$$\mathbf{O} = \mathbf{B} - \mathbf{R}, \quad (6)$$

where $\mathbf{R}$ indicates the residual of rain steak, which is a negative matrix. RRM is wildly used in [18, 25, 55, 56].

• **Base Detail Model (BDM)**. Instead of decomposing the rain image as a rain streak layer and a background layer, BDM decomposes the rain image into the sum of a base layer and a detail layer by using a low-pass filter:

$$\mathbf{O} = \mathbf{O}_{base} + \mathbf{O}_{detail}, \quad (7)$$

where the subscript detail denotes the detail layer that contains structure information, and base denotes the base layer which is similar to clear image. BDM is used in [57].

• **Raindrop Mask Model (RMM)**. This model formulates the raindrop-degraded image as a combination of a background image and the effect of raindrops, which is defined as follows:

$$\mathbf{O} = \mathbf{I} + (1 - \mathbf{M}) \odot \mathbf{B}, \quad (8)$$

where $\mathbf{M}$ is the binary mask, and $\mathbf{M}(x) = 1$ means the pixel x is part of a rain region, and otherwise means it is part of background regions. $\mathbf{I}$ is the effect brought by the raindrops, representing the complex mixture of the background information and the light reflected by the environment and passing through the raindrops adhered to a lens or windscreen. [26] uses RMM as the rain model.

• **Raindrop Transparency Model (RTM)**. This model consider a simple linear model with transparency, defined as

$$\mathbf{O} = \mathbf{A} \odot \mathbf{R} + (1 - \mathbf{A}) \odot \mathbf{B}, \quad (9)$$

where $\mathbf{A} \in [0,1]^{C \times M \times N}$ denotes the transparency matrix. Each entry of $\mathbf{A}$ represents the percentage of the light path covered by raindrops for the corresponding pixel. The study in [58] has adopted RTM as the rain model.

• **Rain Convolutional Dictionary (RCD)**. This model considers the problem under the conventional prior-based methodology by exploiting the prior knowledge for representing rain streaks:

$$\mathbf{O} = \mathbf{B} + \sum_{n=1}^N \mathbf{C}_n^c \otimes \mathbf{P}_n, \quad (10)$$

where $\sum_{n=1}^N \mathbf{C}_n^c \otimes \mathbf{P}_n = \mathbf{S}^c (c = 1, 2, 3)$, $\mathbf{S}^c$ denotes the c^{th} color channel of rain streak $\mathbf{S}$, and $\{\mathbf{C}_n^c\}_{n,c} \subset \mathbb{R}^{k \times k}$ is a set of rain kernels which describes the repetitive local patterns of rain streaks, and $\{\mathbf{P}_n\}_n \subset \mathbb{R}^{H \times W}$ represents the corresponding rain maps representing the locations where local patterns repeatedly appear. N is the number of kernels and $\otimes$ is the 2-dimensional (2D) convolutional operation. [59] uses RCD as the rain model.

• **Two Transmissions Model (TTM)**. Because the rain streaks and vapors are entangled with each other, this mathematical model entangles the rain streaks and vapors properly from the transmission medium perspective:

$$\mathbf{O} = (\mathbf{T}_s + \mathbf{T}_v) \odot \mathbf{B} + [1 - (\mathbf{T}_s + \mathbf{T}_v)] \odot \mathbf{A}, \quad (11)$$

where $\mathbf{T}_s$ and $\mathbf{T}_v$ are the transmission map of rain streaks and vapors, respectively. TTM is adopted in [40] as the rain model.

• **Streak Drop Model (SDM)**. In real-world rainy weather scenarios, rain streaks and raindrops may co-occur during outdoor image capture. As such, this rain model considers both the rain streak and raindrop in one formula:

$$\mathbf{O} = \alpha \mathbf{I} + (1 - \mathbf{M}) \odot (\mathbf{B} + \mathbf{S}), \quad (12)$$

where α is the global atmospheric lighting coefficient. SDM is adopted in [38] as the rain model.

3.3 Network Architecture

The network architecture is always the most critical and concerned factor for the SID task. By designing practical modules, researchers have constructed a pipeline to solve the proposed rain model. According to whether the network derives all the rain model information, we divide the current SID network architectures into *white-box network architecture* and *black-box network architecture*. Fig. 5 shows the examples of two network architectures to learn information in different ways.

Fig. 5: Examples of two different network architectures to learn to predict a clear image $\hat{\mathbf{B}}$ by different ways.

3.3.1 White-box network architecture

The term "white-box" here comes from the well-known "white-box testing", which is one of the main software testing methods. The main feature of the white-box testing is that the examiner understands the internal structure of the program. Similarly, the *white-box network architecture* means that the authors use the network to derive all the rain model information. This kind of network structures often needs the ground truth or prior knowledge. For example, [21] predicts the derained image by:

$$\hat{\mathbf{B}} = \frac{\mathbf{O} - (1 - \tilde{\mathbf{T}}) \odot \tilde{\mathbf{A}}}{\tilde{\mathbf{T}}} - \sum_i^n \tilde{\mathbf{S}}_i, \quad (13)$$

where $\tilde{\mathbf{T}}$, $\tilde{\mathbf{A}}$, and $\tilde{\mathbf{S}}_i$ are obtained by the neural network in supervised mode. The network of this kind, which can derive all the rain model information that occurred in Eqn. (4) to obtain the final derained result, is called *white-box network architecture*. In fact, many existing networks in current methods are white-box architectures, for instance [21, 22, 23, 51].

3.3.2 Black-box network architecture

Compared with the white-box network architecture, the black-box network architecture does not learn all the rain model information, but using the robust fitting and black-box capability of the deep neural network for forcible training instead. This kind of network is called *black-box network architecture*. The term "black-box" denotes that the internal structure of the program is unexplainable to the examiner. For example, the framework in [19] aims at predicting the derained result by a black-box way:

$$\tilde{\mathbf{B}} = \mathbf{O} - \partial [\mathbf{E}, \tilde{\mathbf{M}}, \tilde{\mathbf{S}}], \quad (14)$$

where ∂ represents the function of neural networks, $\mathbf{E}$ denotes the rain feature maps extracted by previous CNN, $\tilde{\mathbf{M}}$ denotes the rain mask predicted from the convolutional process of $\mathbf{E}$, and $\tilde{\mathbf{S}}$ denotes the rain streak predicted from the convolutional process based on the concatenation $[\mathbf{E}, \tilde{\mathbf{M}}]$. In Eqn. (14), only $\tilde{\mathbf{M}}$, $\tilde{\mathbf{S}}$, and $\tilde{\mathbf{B}}$ are derived in the supervised training process, while the transmission map $\mathbf{T}$ and atmospheric light $\mathbf{A}$ in Eqn. (5) were not derived. Note that the existing SID methods [19, 20, 24, 25, 26] have adopted the black-box network architectures.

4 ANALYSIS OF DATA-DRIVEN SID

We mainly analyze the intrinsic relationships among the three influencing factors, and then discuss the solving paradigms of existing SID methods based on the preceding studies.

Fig. 6: The relationship of the three factors in the SID task.

4.1 Relationship of The Three Factors

To facilitate the discussion and analysis, we denote the set of data, rain model and network architecture as $\mathbb{H}$, $\mathbb{G}$, and $\mathbb{F}$, respectively. The image pairs and side information are respectively represented as $[\mathbf{O}, \mathbf{B}, \mathbf{I}_i] \in \mathbb{H}$, where $\mathbf{O}$ and $\mathbf{B}$ represent the rain image and clear ground-truth image, $\mathbf{I}_i$ ($i = 1, \dots, n$) denotes the side information. Then, we can mathematically formulate the process of rainmaking and deraining as follows:

$$\begin{aligned} \mathbf{O} &= g(\mathbf{B}, \mathbf{I}_i), \\ \hat{\mathbf{B}}, \hat{\mathbf{I}}_i &= f(\mathbf{O}), \end{aligned} \quad (15)$$

where $g(\cdot) \in \mathbb{G}$ is a rain model for rainmaking or rain generation, by which the rain image $\mathbf{O}$ is generated, and $f(\cdot) \in \mathbb{F}$ is a network architecture for deraining, by which the derained image $\hat{\mathbf{B}}$ and side information $\hat{\mathbf{I}}_i$ are predicted. From Eqn. (15), we can see that the rain model $g(\cdot)$ is to make or generate rain, while the network architecture $f(\cdot)$ is an inverse function to remove rain, so the training goal of the deraining networks can be described as

$$f(\cdot) = g^{-1}(\cdot). \quad (16)$$

Note that the relationship of the three factors is illustrated in Fig. 6, from which we see that data are core part, while the rain

model and network architecture are closely associated with data. Specifically, the rain model defined by some prior assumptions can be used to generate data. Then, the network architecture uses the data to solve the rain model by maximum posterior probability reversely. That is, rainmaking and deraining are inverse processes in the SID task. Since data play a core role in three factors, we will mainly study the effectiveness of SID data in Section 6.

4.2 Solving Paradigms of SID

Due to the complexity of the optical characteristics of rain [53], the uncertainty of prior knowledge and the flexibility of deep learning solution, the relationship of the three factors does not always follow Eqn. (16). The relationship among the three factors is often asymmetrical or inconsistent. The complexity of the relationships leads to different solving paradigms of SID methods. According to whether the solving paradigms of SID methods follow Eqn. (16), we divide the current SID methods into *explicit solving paradigms* and *implicit solving paradigms*. Table 3 shows the three factors and the corresponding solving paradigms of existing SID methods based on our proposed division criterion. Where the term "network" indicates that whether this deraining method can learn all the information in their rain models.

TABLE 3: The solving paradigms used in current SID methods.

Paradigm	Method	Data	Rain Model	Network
<i>Explicit</i>	Li <i>et al.</i> [21]	specific	HRM	white-box
	DID-MDN [23]	specific	RSM	white-box
	ID-CGAN [22]	general	RSM	white-box
<i>Implicit</i>	DAF-Net [20]	specific	IDM	black-box
	RESCAN [24]	specific	RAM	black-box
	Qian <i>et al.</i> [26]	general	RMM	black-box
	PReNet [25]	specific	RRM	black-box
	SPANet [27]	general	RSM	black-box
	JORDER [19]	specific	RAM	black-box
	Wang <i>et al.</i> [40]	specific	TTM	white-box
	MSPFN [50]	specific	RSM	white-box
	JDNet [51]	specific	RSM	white-box
	GraNet [52]	specific	RSM	white-box
	Fu <i>et al.</i> [18]	general	RRM	white-box
	MPRNet [56]	specific	RRM	white-box
	Fu <i>et al.</i> [55]	specific	RRM	white-box
	RCDNet [59]	specific	RCD	white-box
	DDC-Net [41]	specific	SBM	white-box

4.2.1 Explicit solving paradigm

We first discuss the explicit solving paradigm of SID methods. As described in Table 3, we see that only a few existing SID methods adopt the explicit solving paradigm. Note that we divide the solving paradigms of current methods as "explicit" only based on one judgment, i.e., the relationship of the three factors in the method is consistent with Eqn. (16). For example, the studies of [22, 23] use the RSM to synthesize the Rain800 and Rain12000 datasets, then use the network architecture to solve it in a white-box mode. Similarly, the method in [21] builds the NYU-Rain and Outdoor-Rain data, while using Eqn. (4) to solve RAM.

4.2.2 Implicit solving paradigm

In contrast, most SID methods in Table 3 adopt the implicit

solving paradigm. Note that we divide the solving paradigm of a method as "implicit" based on two judgments: 1) the network architecture is not consistent with the rain model. For example, the methods in [19, 24] use RAM (Eqn. (5)) and neglect to predict the transmission maps and atmospheric light in their network architecture, while [20, 22, 25, 26, 27] cannot to predict rain streak information; 2) the training data are not consistent with the rain models. For example, the methods in [50, 51, 52] use RSM (Eqn. (1)) as their rain models, which only contains the clean image and rain streak layer, but use Rain100H and Rain100L as training data, which is synthesized by RAM (Eqn. (5)). Similarly, the methods in [18, 55, 56] use RRM (Eqn. (6)) as their rain models, which only contains the clean image and rain residual layer, but also using Rain100H and Rain100L as training data.

4.3 Division of Recent SID Methods

Based on the above in-depth analysis, we can category the recent 47 data-driven SID methods (proposed between 2017-2021) from six aspects of the three factors (i.e., *general vs. specific*, *synthetical vs. mathematical*, *black-box vs. white-box*), as shown in Fig. 7. From the classification, we can conclude that:

1) Most of the SID methods use general data (i.e., Rain800, Rain14000, SPA-Data) and synthetic rain model (i.e., RSM), which is the most basic and simplest way to solve SID problem. The main difference among those SID methods is the design of domain knowledge, which will be discussed in Section 5;

2) There are 32 black-box and 22 white-box SID methods. More researchers tend to solve rain models inaccurately, thanks to the amazing fitting learning ability of neural networks;

3) More and more SID methods [20, 21, 38, 40, 59, 60] start to use specific data (i.e., RainCityscapes, NYU-Rain, DDC-Data) and mathematical rain model (i.e., RCD, TTM, SDM) to solve SID problem, which results in obtaining better performance;

4) There are only two SID methods [26, 38] that adopt the specific data and mathematical rain model, which implies that we can pay more attention to this task.

5 MORE ANALYSIS ON SID METHODS

In addition to the above analysis based on the three factors, SID methods still have some other aspects than can be further explored, such as training strategy, network pipeline, domain knowledge, data preprocessing, and objective function. However, these issues have not been fully investigated in previous studies. This section will summarize 54 existing SID methods to describe these features. The main information is summarized in Table 4, and the statistical results are illustrated in Fig. 9.

5.1 Training Strategy

In order to improve the generalization ability to the real-world scenario, many SID methods use the real rain images as training data. According to whether unlabeled data are involved in the training process, the training strategies can be divided into three learning modes, i.e., supervised, unsupervised and semi-supervised.

- **Supervised.** In supervised mode, the network is optimized by calculating the loss between the predicted image and the labeled image by using the data with ground truth. For example, JORDER [19] can predict the rain mask, rain streak, and background information based on a fully-supervised learning mode.

- **Unsupervised.** In contrast, since there is no ground truth images for comparison in unsupervised mode, the deraining network

TABLE 4: The mainstream datasets for the SID task, which are proposed from 2017 to 2021.

Method	Mode	Pipeline	Domain Knowledge	Data Processing	Objective Function	Code	Year
RICNet [37]	S	Hybrid	PK/DK/OK	Random	MSE/MAE/Adv/SSIM/KL/AC	×	2021
JRGR [61]	S-S	Parallel	DK/OK	Random	MSE/MAE/Adv	×	2021
QSMD [36]	S	Hybrid	PK/DK/OK	Fixed	MSE/MAE/QS	×	2021
RLNet [62]	S	Hybrid	DK/OK	Fixed	MSE/MAE/SSIM	×	2021
MOSS [63]	S-S	Sequential	DK/OK	Random	MAE/TV	×	2021
DerainCycleGAN [39]	U-S	Parallel	DK/OK	Random	MSE/MAE/Adv	×	2021
Fu <i>et al.</i> [55]	S	Sequential	DK	Random	MAE	×	2021
MPRNet [56]	S	Hybrid	DK/OK	Fixed	MSE	✓	2021
CCN [38]	S	Hybrid	DK/OK	-	MAE/SSIM	×	2021
Semi-DerainGAN [64]	S-S	Parallel	DK/OK	Random	MSE/MAE/Adv/	×	2021
CVID [65]	S	Hybrid	PK/DK/OK	Random	MSE/KL	✓	2020
RCDNet [59]	S	Hybrid	PK/DK/OK	Random	MSE/KL	✓	2020
Pan <i>et al.</i> [66]	S	Parallel	DK/OK	-	MAE/Adv	✓	2020
MSPFN [50]	S	Hybrid	DK/OK	Fixed	MAE	✓	2020
RDDAN [67]	S	Sequential	DK/OK	Random	MAE/SSIM	×	2020
JDNet [51]	S	Sequential	DK/OK	Random	SSIM	×	2020
QuDeC [68]	S	Hybrid	PK/DK/OK	-	MSE/CE	×	2020
Syn2Real [69]	S-S	Parallel	PK/DK/OK	Random	MSE/MAE/GP	✓	2020
DCSFN [70]	S	Sequential	DK/OK	Random	SSIM	✓	2020
Zhu <i>et al.</i> [71]	S	Hybrid	DK/OK	Random	MAE	✓	2020
DRD-Net [72]	S	Parallel	DK/OK	-	MSE	✓	2020
Wang <i>et al.</i> [40]	S	Hybrid	DK/OK	Fixed	MSE/MAE	✓	2020
SSDRNet [73]	S	Hybrid	DK/OK	Random	MAE/SSIM	✓	2020
DTDN [74]	S	Parallel	PK/DK/OK	Fixed	MSE/Adv	×	2019
Liu <i>et al.</i> [75]	S	Sequential	DK	-	MAE/SSIM	✓	2019
ERL-Net [76]	S	Hybrid	DK/OK	Random	MAE	×	2019
GraNet [52]	S	Hybrid	DK/OK	Fixed	MAE	×	2019
LPNet [77]	S	Hybrid	PK/OK	Random	MAE/SSIM	✓	2019
PReNet [25]	S	Sequential	DK/OK	Random	SSIM	✓	2019
DAF-Net [20]	S	Hybrid	DK/OK	Fixed	MSE	✓	2019
ReHEN [78]	S	Sequential	DK/OK	Random	MSE/SSIM	✓	2019
DDC-Net [41]	S	Hybrid	DK/OK	Fixed	MSE/Adv	✓	2019
RR-GAN [79]	U-S	Parallel	DK/OK	Random	MSE/Adv	×	2019
MH-DerainNet [80]	S	Sequential	DK/OK	Random	MSE/SSIM	×	2019
SIRR [28]	S-S	Parallel	PK/DK/OK	Random	MSE/KL/GMM/TV	✓	2019
SPANet [27]	S	Sequential	DK/OK	Random	MSE/MAE/SSIM	✓	2019
ReMAEN [81]	S	Sequential	DK/OK	Random	MSE	✓	2019
UD-GAN [82]	U-S	Parallel	PK/DK/OK	Fixed	MAE/Adv	×	2019
UMRL [83]	S	Hybrid	DK/OK	Fixed	MAE	✓	2019
ID-CGAN [22]	S	Parallel	DK	Fixed	MSE/Adv	✓	2019
Li <i>et al.</i> [21]	S	Hybrid	PK/DK/OK	Fixed	MSE/MAE/Adv	✓	2019
JORDER-E [60]	S	Hybrid	DK/OK	Random	MSE/CE	✓	2019
Quan <i>et al.</i> [58]	S	Hybrid	DK/OK	Random	MAE	✓	2019
RWL [84]	S	Hybrid	PK/DK	-	MSE	×	2019
DualCNN [85]	S	Parallel	DK	Random	MSE	✓	2018
NLEDN [86]	S	Sequential	DK/OK	Fixed	MAE	✓	2018
RESCAN [24]	S	Sequential	DK/OK	Random	MSE	✓	2018
ResGuideNet [87]	S	Sequential	DK/OK	Random	MSE/SSIM	×	2018
Qian <i>et al.</i> [26]	S	Parallel	DK	Fixed	MSE/Adv	✓	2018
DID-MDN [23]	S	Hybrid	DK/OK	Fixed	MSE/CE	✓	2018
SMRNet [88]	S	Sequential	DK/OK	-	MSE	×	2017
JORDER [19]	S	Hybrid	DK/OK	Random	MSE/CE	✓	2017
DerainNet [57]	S	Parallel	PK/OK	Random	MSE	✓	2017
Fu <i>et al.</i> [18]	S	Sequential	PK/OK	Random	MSE	✓	2017

is usually optimized by using prior knowledge or self-supervision. For example, RR-GAN [79] uses the consistency regularization to learn the rain distribution without using label information.

• **Semi-supervised.** By using both labeled and unlabeled data, semi-supervised network can be optimized in the case of transfer learning, domain adaption and sharing of the network parameters. For example, SIRR [28] shares a CNN to learn both synthetic and real-world rain distribution by regularizing K-L distance.

Remarks. We count the number of each training strategy used in existing SID methods and list them in Table 5. From Table 4 and 5, we can conclude that:

1) There are only 5 semi-supervised and 3 unsupervised SID methods, and 46 supervised SID methods. The main reason is that

TABLE 5: The statistics of training strategy used in current SID methods.

Pipeline	Supervised	Semi-supervised	Unsupervised	Total
Count	46	5	3	54

the SID problem is ill-posed and is difficult to solve via weak constraints in semi/un-supervised mode;

2) Most SID methods adopt the supervised model. By involving the paired data, supervised model can significantly perform test data with the same distribution. However, these pre-trained methods often failed in the case of real-world rain images, due to the poor generalization ability;

Fig. 7: The division of recent SID methods from six aspects based on the three factors.

3) Among those 8 semi/un-supervised SID methods, 2 semi-supervised and 3 unsupervised methods are based on the GAN model [89, 90]. That is because the GAN model can provide better image generation and restoration in weak supervised mode.

4) Seven semi/un-supervised SID methods use the parallel pipeline to construct steady network architecture. Note that the parallel pipeline will be introduced below.

5.2 Network Pipeline

Due to the different training strategies and domain knowledge used in the SID tasks, data-driven SID methods have different network pipelines. According to the flow of data in the network, we can divide the network pipeline into three structures, that is, sequential, parallel and hybrid, as illustrated in Fig. 8.

- **Sequential.** This is the most basic and common network pipeline, where data flows linearly. For example, Fu *et al.* [18] uses a linear way to build an end-to-end network. An architecture with multi-streams handling the same task can also be considered as sequential mode. For example, MH-DerainNet [80] uses different kernel sizes to get features with different receptive fields.

- **Parallel.** Instead of simply using a single neural network, a parallel pipeline is applied to handle equally important tasks. For example, DerainCycleGAN [39] uses the parallel networks to process both rain and clean ground-truth images. An architecture with multi streams handling the different tasks can also be considered as parallel mode. For example, DerainNet [57] uses two streams to dispose the detail and base layers, respectively.

- **Hybrid.** A hybrid pipeline would have both sequential and parallel architectures inside. Since the modules can be assembled in more complex way, in case that more side information needs to be processed simultaneously. For example, DAF-Net [20] uses a hybrid network to handle the tasks of learning depth map and image feature at the same time; JORDER [19] firstly uses the sequential multi streams to extract feature maps with different receptive fields, and then uses three parallel convolutions to learn the rain mask, rain streak and background, respectively.

Remarks. We count the number of each network pipeline used in existing SID methods and summarize them in Table 6. From Table 4 and 6, we can conclude that:

Fig. 8: Three representative network pipelines of data-driven SID methods. The picture is taken from original papers.

TABLE 6: The statistics of network pipeline used in current SID methods.

Network Pipeline	Sequential	Parallel	Hybrid	Total
Count	16	14	24	54

1) There are 16 sequential, 14 parallel and 24 hybrid pipelines adopted in existing methods. It looks like the hybrid structure is more popular than the sequential and parallel structures;

2) More SID methods choose the sequential pipeline from 2017 to 2018, while choosing the parallel and hybrid pipelines to consider more side information from 2019 and later. Note that SID methods needing side information (i.e., rain density, image depth) prefer the parallel and hybrid pipelines. That is because additional networks need to be designed for learning the side information;

3) From 2020 to 2021, there are 12 methods using the hybrid pipelines among existing 23 SID methods, comparing with 12 with hybrid pipeline in 31 SID methods during 2017 and 2019. More and more hybrid pipeline denotes that researchers begin to design more complicated network architectures;

4) The pipeline of GAN-based methods, such as RR-GAN [79] and Qian *et al.* [26], can be regarded as parallel, since the generator and discriminator are generally two independent networks.

5.3 Domain Knowledge

Due to the complexity of the SID problem, researchers explore different domain knowledge to extract useful feature information and obtain better deraining performance. Generally, there are two main types of domain knowledge. One is prior knowledge based on the digital signal/image processing and probability statistics, such as guided filter [91], mutual information [92], and Kullback-Leibler divergence [93]. The other one is deep-learning knowledge

based on deep feature extraction, such as GAN [89], LSTM [94], high-level perceptual feature [95], and U-net [96]. Besides, some other domain knowledge, e.g., parameter initialization and learning rate strategy, are defined as other kinds of knowledge.

• **Prior Knowledge (PK).** This kind of knowledge is based on the digital signal/image processing and probability statistics. For example, guild filter is commonly used in traditional image processing. For the SID task, researchers have used the guild filter to decompose the original rain image into two parts: high-frequency component (or detail layer) and low-frequency component (or base layer). The former contains more rain information, while the latter contains more background information. For example, Fu *et al.* [18] decomposes the rain image into a base layer and a detail layer, while RWL [84] decomposes the rain image into high-frequency and low-frequency components. That decomposition is to facilitate the deraining of rain, while preserving the background. Another example is the log histograms of filtered natural images below the straight line connecting the minimal and maximal values and naming this property as sparsity prior. Therefore, sparsity has great potential to separate rain and non-rain textures during deraining. For example, QSMD [36] develops a quasi-sparse distribution to approximate the sparsity to obtain a feasible loss function.

• **Deep-learning Knowledge (DK).** It uses deep neural network to extract useful feature information. For example, many SID methods use perceptual features to preserve the image contents. Specifically, VGG-16/VGG-19 [3] pre-trained on ImageNet [97] can obtain higher-level features. The perceptual loss [95] measuring the difference of high-level features has demonstrated better visual performance than the per-pixel loss used in traditional SID methods [18, 19]. In addition, by utilizing unique modules, SID methods can obtain more representational features. Some of these modules are universal modules, such as dilated convolution [98] used in [19, 55], LSTM [94] used in [25, 26], U-net [96] used in [26, 41], SENet [99] used in [24, 72], and ShuffleNet [100] used in QSMD [36]. Others are based on attention [102] mechanism and specifically designed, such as the spatial attentive module used in SPANet [27] and confidence map network used in UMRL [83].

• **Other Knowledge (OK).** In addition to the above two kinds of domain knowledge, there are still many operations based on the understanding of SID. For example, parameter initialization makes the initial network converge quickly and can avoid falling into the local saddle point or gradient cliff initially. A common approach is to use the Gaussian distributions to initialize the network parameters, which are superior to the random initialization. Another approach is to use the Xavier [101] method to initialize the network parameter, such as ResGuideNet [87]. Some networks, such as the VGG network pre-trained on ImageNet, can initialize the parameters of networks with the same structures, such as DAF-Net [20]. In addition, some other SID methods with complicated network architectures often train in different phases. For example, Li *et al.* [21] trains the network in physics-based stage and model-free stage, respectively; Wang *et al.* [40] firstly pre-trains the ANet and SNet, and jointly trains the whole network at last. Learning rate is also considered to obtain better deraining performance. For example, CCN [38] sets the learning rate to 0.001 and adopts a cosine scheduler for training.

Remarks. We count the number of each domain knowledge used in existing methods and list them in Table 7. Besides, we also summarize the used representative deep-learning knowledge in Table 8. From Table 4, 7, and 8, we can conclude that:

1) Deep-learning knowledge is widely used in 52 data-driven

TABLE 7: The statistics of the domain knowledge used in current SID methods.

Domain Knowledge	Prior	Deep-learning	Other	Total
Count	13	52	47	54

TABLE 8: The statistics of the deep-learning knowledge used in current SID methods. From the first row to the end row denote GAN, U-net, Attention, Perceptual, Dilated Convolution, LSTM, Encoder-decoder, Multi-stage, Multi-scale, Two-branch, Multi-stream, Multi-task, Squeeze-and-excitation Network, ShuffleNet, Neural Architecture Search, and Graph Convolutional Network, respectively.

Deep-learning	GAN	U-net	Att	Per	DC	LSTM
Count	12	13	19	11	11	7
	En-De	M-G	M-C	T-B	M-M	M-T
	10	14	7	6	4	4
	SE	SN	NAS	GCN		
	6	2	1	1		

SID methods, due to the strong representation learning ability. In addition, prior knowledge is only used in 12 methods, which means that we can still explore more effective prior knowledge to help data-driven SID task;

2) Other kinds of knowledge is very important for the data-driven SID task, since almost each method use these tricks to obtain more stable and better performance;

3) Among the deep-learning knowledge, GAN [89], U-net [96], Attention [102], Perceptual [95], Dilated Convolution [98], Encoder-decoder [103] and Multi-stage are used more frequently, as can be seen from Table 8;

4) LSTM [94], Multi-scale, Two-branch, Multi-stream, Multi-task, Squeeze-and-excitation Network [99] and ShuffleNet [100] are often adopted in data-driven SID methods;

5) Neural Architecture Search [104] and Graph Convolutional Network [105] are used in CCN [38] and Fu *et al.* [55] in 2021, respectively. We suggest the researchers to pay more attention to these new technologies which have not been applied in SID tasks.

5.4 Data Preprocessing

For the SID task, data preprocessing is also worthy of investigation. Most SID methods read the image data in PNG or JPG format, and then convert it to a tensor form. According to the pattern of image clipping and augmentation, we divide the existing data preprocessing methods into fixed and random sampling.

• **Fixed Sampling.** This sampling method aim at cropping a fixed-size image from the middle or random position of the input. For example, DID-MDN [23] randomly crops a 512×512 image from the input image (or its horizontal flip) of size 586×586 . Due to the large proportion of cropping, data augmentation is rarely used. Fixed sampling can obtain an image with a larger receptive field and is suitable for the method with downsampling.

• **Random Sampling.** Different from the fixed sampling, images of small size (namely, patch) are randomly or regularly cropped from the original image. For example, Fu *et al.* [18] randomly selected 9,100 images, from which it generated three million 64×64 rainy/clean patch pairs. Random sampling has two advantages: 1) it can augment data and therefore reduce the

Fig. 9: Different categories of statistics in all methods.

underfitting phenomenon caused by the insufficient data; 2) it can greatly reduce the GPU consumption and speed up the training process. Larger batch size can be set in training to reduce the over-fitting during the optimization process.

TABLE 9: The statistics of the data preprocessing methods used in current SID studies.

Data Preprocessing	Fixed	Random	Undefined	Total
Count	16	31	7	54

Remarks. We count the number of each data preprocessing method used in current studies and list them in Table 9. From Table 4 and 9, we can conclude that:

1) Most SID methods (totally 31) chose the random sampling, while only 16 methods used the fixed sampling, from which we can infer that both data augmentation and large batch size will be more effective for the SID task;

2) Among those 11 SID methods [19, 20, 21, 23, 26, 37, 41, 60, 71, 84, 85] using side information, 5 used the fixed sampling, and 5 used the random sampling. We speculate that the correlation between data preprocessing and side information is not strong.

5.5 Objective Function

The objective function is an important part of the data-driven SID methods. The parameters of neural networks can be learned via the objective function in the training process. According to whether the constraint contains ground truth, current objective functions can be generally divided into Fidelity-Driven Metrics (with ground truth) and Probability&Statistics Model (without ground truth).

• **Fidelity-Driven Metrics.** Signal fidelity-driven matrices include Mean Absolute Error (MAE, L1), Mean Squared Error (MSE, L2) and Structural Similarity (SSIM), which are technically defined as Eqn. (17, 18, 19), respectively.

$$\mathcal{L}_{MAE}(x, \hat{x}) = \|x - \hat{x}\|_1, \quad (17)$$

$$\mathcal{L}_{MSE}(x, \hat{x}) = \|x - \hat{x}\|_2^2, \quad (18)$$

$$\mathcal{L}_{SSIM}(x, \hat{x}) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}, \quad (19)$$

where x and $\hat{x}$ are ground-truth and derained images respectively in Eqn. (17) and Eqn. (18). In Eqn. (19), μ_x and $\mu_{\hat{x}}$ denote the averages of x and $\hat{x}$, respectively. $\sigma(x)$ and $\sigma(\hat{x})$ denote the

variance of x and $\hat{x}$, respectively. c_1 and c_2 are two numbers used to stabilize the division with weak denominator.

• **Probability&Statistics Model.** In this category, many objective functions are based on the informatics, probability theory, and mathematical statistics, such as Cross Entropy loss (CE), Adversarial loss (Adv), Kullback-Leibler divergence (K-L), Gaussian Mixture Model (GMM), Gaussian Process (GP), Quasi Sparsity loss (QS), Autocorrelation loss (AC), and Total Variation (TV). These objective functions can be technically defined as Eqn. (20, 21, 22, 23, 24, 25, 26, 27), respectively.

$$\mathcal{L}_{CE} = \frac{1}{N} \sum_i -[y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)],$$

$$\mathcal{L}_{CE} = \frac{1}{N} \sum_i - \sum_{c=1}^M y_{ic} \cdot \log(p_{ic}), \quad (20)$$

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] , \quad (21)$$

$$D_{KL}(p \parallel q) = \sum_{i=1}^N p(x) \cdot \log \frac{p(x)}{q(x)},$$

$$\mathcal{L}_{HOG} = \frac{1}{K} \sum_k \text{KL}(\mathcal{O}_k^{gt}, \mathcal{O}_k^{out}), \quad (22)$$

$$\mathcal{R} \sim \sum_{k=1}^K \pi_k \mathcal{N}(\mathcal{R} \mid \mu_k, \Sigma_k),$$

$$\mathcal{L}(\mathcal{R}; \Pi, \Sigma) = - \sum_{n=1}^N \log \sum_{k=1}^K \pi_k \mathcal{N}(\mathcal{R}_n \mid 0, \Sigma_k) \quad (23)$$

$$m(v) = \mathbb{E}[f(v)],$$

$$K(v, v') = \mathbb{E}[(f(v) - m(v))(f(v') - m(v'))], \quad (24)$$

$$P_q(\mathbf{I}) = \prod_{i,k} P_q(\omega_{i,k} * \mathbf{I}), \quad (25)$$

$$\mathcal{L}_Q = \sum_{t=1}^N \sum_{i,k} |\omega_{i,k} * \mathcal{S}(\mathbf{I}_t)| + |\omega_{i,k} * [\mathbf{I}_t - \mathcal{S}(\mathbf{I}_t)]|,$$

$$\mathcal{L}_{ac} = \sum_{x,y} \sum_{i=1}^P \frac{\text{ac}_{x,y}(\max_i) - \text{mean}_{\text{ac}_{x,y}}}{P} \quad (26)$$

$$\mathcal{R}_{V\beta}(x) = \sum_{i,j} \left((x_{i,j+1} - x_{i,j})^2 + (x_{i+1,j} - x_{i,j})^2 \right)^{\frac{\beta}{2}}. \quad (27)$$

In Eqn. (20), the first and second expressions are the dichotomous and multiple classification expressions, respectively. Eqn. (21) is adversarial loss used in GAN-based methods, such as [26, 39, 79], where z is a noise, D and G are discriminator and generator, respectively. $p(x)$ and $q(x)$ are two probability distributions on random variables, and RICNet [37] use K-L divergence to calculate the distance between the ground-truth and predicted image in Eqn. (22). In Eqn. (23), $\mathcal{R}$ is rain streak, K is the number of mixture components, π_k , μ_k and Σ_k are mixture coefficients, Gaussian distribution means and variance, respectively. SIRR [28] use the negative log likelihood function to constrain the unsupervised samples, where $\Pi = \pi_1, \dots, \pi_K$, $\Sigma = \Sigma_1, \dots, \Sigma_K$, K is the number of mixture components, and N is the number of samples. In Eqn. (24), $v, v' \in \mathcal{V}$ denote the possible inputs that index the GP. In Eqn. (25), where $\mathbf{I}$ is rain image, $\omega_{i,k}$ is the k^{th} filter centered at the i^{th} pixel. $*$ is the convolution operation. QSMD [36] use it to construct a QS loss, where $\mathcal{S}(\cdot)$ denotes the network inference. In Eqn. (26), $\max_i$ represents the index of the top i -th coefficient, and $\text{mean}_{ac_x, y}$ denote the mean values of ac_x and ac_y . In Eqn. (27), the total variation function $\mathcal{R}_{V\beta}(x)$ encourages the images to consist of piece-wise constant patches, images are discrete ($X \in \mathbb{R}^{H \times W}$) and $\beta = 1$. MOSS [63] adopts a Total Variation regularizer term to smooth the recovered background image.

TABLE 10: The statistics of the objective functions used in current SID methods.

Objective Function	MSE	MAE	SSIM	Adv	CE	KL
Count	35	26	14	12	4	4
	GMM	GP	QS	AC	TV	
	1	1	1	1	2	

Remarks. We count the number of each objective function used in current SID methods and list them in Table 10. From Table 4 and 10, we can conclude that:

- 1) Fidelity-driven metrics are wildly used in data-driven SID methods. Specifically, MSE is the most popular loss function, due to the fact that 35 of 54 SID methods chose it, while MAE is the second-popular, since 26 SID methods selected it;
- 2) A single loss is used in 21 SID methods as the objective function. Specifically, MSE, MAE, and SSIM are respectively used as a single loss function in 10, 8 and 3 current SID methods, respectively. This phenomenon implies that a single loss function can also make network convergence well;
- 3) The cross entropy has been chosen by 4 SID methods. For example, it is used as a loss function to constrain the rain mask prediction ([19]) or rain density prediction ([23]);
- 4) Adversarial loss is widely selected by 12 GAN-based SID networks, such as ([26, 39, 64, 74, 79, 82]);
- 5) SSIM is used in 14 SID methods, such as [25, 27, 75, 77, 78, 87], and it was frequently used five times in 2019;
- 6) The probability&statistics models are totally used in 26 methods. Due to the complexity of rain distribution, probability&statistics model are usually hard to be guaranteed, which is usually used as an auxiliary constraint. This form of unconditional constraint may be a future research direction for developing SID.

6 DATA RANKING EXPERIMENT

In this paper, we mainly evaluate the contribution of data generated by the rain models on the SID tasks. Since data play a core role among the three factors, we will mainly focus on analyzing data and designing a group of experiments to study the impact of data and evaluate the effectiveness of SID data.

6.1 Experiment Design

The data ranking experiments are mainly prepared to figure out which paired dataset is most effective for the SID task, denoted by **EOD (Effectiveness of Data)**. A higher **EOD** score can at least reflect two things: 1) this dataset is more challenging and can potentially stimulate the network to fit continuously, without causing rapid convergence; 2) a pre-trained method on this dataset will potentially have better generalization ability in the real scenario. To obtain more fair and reliable evaluation results, we will conduct four tests from different dimensions:

- **Test-1:** Objective evaluation on paired datasets by using full-reference Image Quality Assessment (IQA);
- **Test-2:** Objective evaluation on real image datasets by using the no-reference IQA metrics;
- **Test-3:** Subjective evaluation on real image dataset by using human subjective assessment;
- **Test-4:** High-level task evaluation on real image dataset by using object detection evaluation metrics.

Note that Test-1 is mainly used to verify the difficulty of data. Test-2 and Test-3 are mainly used to evaluate the generalization abilities of data by using both objective and subjective evaluation metrics, which can also indicate how large the difference between synthetic and natural scenarios. Finally, Test-4 is conducted on a high-level task, i.e., object detection, which can reflect the impact of data on the practical vision tasks.

6.2 Dataset and Experiment Setting

Datasets. In this study, we selected 9 public paired datasets for Test-1. These datasets include Rain100H [19], Rain100L [19], Rain800 [22], Rain14000 [18], Rain12000 [23], SPA-Data [27], RainCityscapes [20], Outdoor-Rain [21] and MPID [32], respectively. For both Test-2 and Test-3, we construct a new dataset called *Real-Internet* with 385 real-world images without ground-truth, which are collected from the Internet and previous studies [19, 23, 28]. For Test-4, an object detection dataset RID [32] is used for the task-driven evaluation.

Evaluation Metrics. To evaluate the deraining results for the images with ground truth in Test-1, we use the PSNR [106] and SSIM [107] as the evaluation metrics. To evaluate the authentic images without ground-truth in Test-2, we use the no-reference IQA metrics, including SSEQ [108], NIQE [109], BRISQUE [110], and BLINDS-II [111]. To evaluate the authentic images by subjective assessment in Test-3, we follow a standard-setting as Bradley-Terry model [112] to estimate the subjective score of each method so that they can be ranked, with the same routine as [113]. For the high-level task evaluation in Test-4, we use mean Average Precision (mAP) score to evaluate the detection performance.

Compared SID methods. Three representative methods are chosen for testing: Fu *et al.* [18], RESCAN [24], and LPNet [77]. They will be trained on an NVIDIA GeForce GTX 1080i GPU with 12GB memory. We will use the original implementation code and follow the same experimental setup for these methods.

TABLE 11: The numerical evaluation results of data ranking experiment.

		Original	Rain100H [19]	Rain100L [19]	Rain800 [22]	Rain14000 [18]	Rain12000 [23]	SPA-Data [27]	RainCity [20]	Outdoor [21]	MPID [32]
<i>Fu et al. (CVPR2017) [18]</i>											
Test-1	<i>PSNR</i> ↑	-	26.78	34.54	25.23	30.05	29.20	36.93	27.52	20.19	34.65
	<i>SSIM</i> ↑	-	0.810	0.956	0.841	0.904	0.903	0.960	0.904	0.833	0.953
Test-2	<i>SSEQ</i> ↓	30.61	34.10	31.25	34.15	33.98	34.39	31.76	37.68	29.14	31.28
	<i>NIEQ</i> ↓	3.959	4.267	3.723	4.048	3.925	4.166	3.754	4.063	4.170	3.908
	<i>BRIS</i> ↓	25.69	31.36	24.88	29.17	29.08	30.97	24.06	33.61	36.41	26.20
	<i>BLIIN</i> ↓	80.46	86.09	81.67	81.39	82.12	78.38	82.30	69.29	77.54	82.71
Test-3	<i>Clear</i> ↑	-	0.27	0.07	0.15	0.12	0.14	0.07	0.04	0.05	0.08
	<i>Auth</i> ↑	-	0.12	0.09	0.15	0.18	0.15	0.15	0.02	0.03	0.11
Test-4	<i>FCNN</i> ↑	18.84	16.20	18.72	16.41	18.03	18.36	17.71	15.59	17.14	18.27
	<i>YOLO</i> ↑	19.61	18.47	19.57	17.86	18.94	18.90	19.27	12.51	14.17	19.13
<i>RESCAN (ECCV2018) [24]</i>											
Test-1	<i>PSNR</i> ↑	-	28.95	38.41	28.36	32.69	33.86	38.50	18.03	21.10	40.52
	<i>SSIM</i> ↑	-	0.867	0.980	0.885	0.941	0.948	0.972	0.773	0.834	0.984
Test-2	<i>SSEQ</i> ↓	30.61	30.26	30.36	33.46	32.18	32.15	31.62	31.25	27.13	30.07
	<i>NIEQ</i> ↓	3.959	4.007	3.698	4.087	4.233	4.108	3.924	3.871	4.080	3.821
	<i>BRIS</i> ↓	25.69	25.81	24.05	28.72	30.59	29.30	25.44	28.09	27.93	24.93
	<i>BLIIN</i> ↓	80.46	87.91	82.14	80.50	79.45	79.66	81.46	77.87	81.39	82.47
Test-3	<i>Clear</i> ↑	-	0.32	0.08	0.23	0.09	0.06	0.11	0.04	0.05	0.04
	<i>Auth</i> ↑	-	0.14	0.13	0.11	0.14	0.16	0.11	0.01	0.04	0.17
Test-4	<i>FCNN</i> ↑	18.84	18.02	18.63	18.71	18.55	18.92	18.47	17.36	18.01	18.96
	<i>YOLO</i> ↑	19.61	19.38	19.68	19.39	18.91	19.16	20.49	17.01	17.37	19.65
<i>LPNet (T-NNLS 2019) [77]</i>											
Test-1	<i>PSNR</i> ↑	-	19.64	33.63	24.62	26.40	27.39	37.08	21.55	16.78	32.74
	<i>SSIM</i> ↑	-	0.805	0.959	0.859	0.905	0.901	0.968	0.900	0.781	0.960
Test-2	<i>SSEQ</i> ↓	30.61	37.54	31.10	37.36	35.54	35.59	32.76	30.28	28.05	31.01
	<i>NIEQ</i> ↓	3.959	4.733	3.690	4.886	4.313	4.696	3.850	4.070	4.731	3.820
	<i>BRIS</i> ↓	25.69	37.42	24.61	34.23	29.90	32.76	24.58	28.36	39.72	23.98
	<i>BLIIN</i> ↓	80.46	78.15	81.45	72.94	78.47	75.31	78.48	77.86	84.48	83.68
Test-3	<i>Clear</i> ↑	-	0.09	0.12	0.17	0.17	0.17	0.13	0.02	0.02	0.12
	<i>Auth</i> ↑	-	0.05	0.13	0.21	0.15	0.17	0.15	0.00	0.02	0.12
Test-4	<i>FCNN</i> ↑	18.84	14.15	17.89	16.32	17.19	17.75	17.90	16.63	16.16	17.99
	<i>YOLO</i> ↑	19.61	15.48	19.40	17.67	18.76	17.88	19.51	14.47	14.20	19.35

Fig. 10: Illustration of some results in Fu et al. [18]. From the top row to bottom row denote the deraining examples of Test-1, Test-2 and Test-4, respectively. From (a) to (i) denote the results of methods pretrained on the 9 public paired datasets, i.e., Rain100H [19], Rain100L [19], Rain800 [22], Rain14000 [18], Rain12000 [23], SPA-Data [27], RainCityscapes [20], Outdoor-Rain [21], and MPID [32], respectively.

Fig. 11: The numerical evaluation results of the four tests (i.e., Test-1 to Test-4) in the data ranking experiment.

6.3 Experiment Results

We will analyze the deraining results of each test and finally calculate the EOD score in this subsection.

6.3.1 Test-1: Objective evaluation on paired datasets

We evaluate the deraining results on 9 paired datasets, and then calculate the improvement compared with the original performance. For PSNR, the more significant the improvement, the less challenging the dataset will be. For SSIM, the closer the value is to 1, the less challenging the dataset will be. Taking Fu *et al.* [18] as an example, the original PSNR/SSIM of Rain100H [19] and Rain100L [19] testing set are 13.56/0.379 and 26.90/0.838; the derained PSNR/SSIM are 26.78/0.810 and 34.54/0.956. If we consider PSNR, the improvement ratios of Rain100H and Rain100L are 97% and 28%. We can infer Rain100L is more challenging than Rain100H. If we consider SSIM, the SSIM value after deraining are 0.810 and 0.956, i.e., Rain100H still has some space to improve since the maximum value of a clean image is 1. In such case, we can infer that Rain100H is more challenging than Rain100L. As such, we consider both PSNR and SSIM with the same weight. We tested three methods to reduce the error caused by one method, and the results are shown in Table 11. By calculating and normalizing the results, we can obtain the image for each dataset in Test-1, as shown in Fig. 11. We see that RainCityscapes [20] obtains the highest score, while Rain100L obtains the lowest score. This result is because RainCityscapes is synthesized by Eqn. (3), which has a more complex rain streak and atmospheric light circumstance. We also illustrate some deraining results of DetailNet in the first row of Fig. 10, from which we see that Fu *et al.* [18] can remove most rain streak information in Rain100L and MPID, but hardly remove the rain streak in the datasets like Rain800, Rain14000 and RainCityscapes. Note that these visual results are consistent with the test scores.

6.3.2 Test-2: Objective evaluation on real dataset

In this experiment, we evaluate the pre-trained model on a real scenario dataset (i.e., Real-Internet), and then evaluate the improvement compared with the original result. Since these authentic rain images have no corresponding clean ground-truth images, we choose four no-reference IQA as the evaluation criteria. The results are shown in Table 11, from which we see that most of the numerical indices decline after rain removal (the lower the value, the better the effect), which is contrary to what we expected. We also illustrate some deraining results of Fu *et al.* [18] in the second row of Fig. 10, from which we see that the

Fig. 12: The numerical values of EOD based on the four tests.

pre-trained model on Rain100H can remove most rain streaks, while the pre-trained model on MPID can hardly remove the rain. According to the improvement of the four no-reference IQA, we calculate each dataset score of Test-2, as shown in Fig. 11. We see that Rain100H gets 0.06 while MPID gets 0.17, which is the complete opposite of visualization. There are two reasons for this phenomenon: 1) current mainstream no-reference IQA criteria are not suitable for the SID task [32, 34]; 2) The difference between the synthetic and real data results in weak generalization power. Considering this fact, we also use a subjective Test-3 to balance out the disadvantage of the visual perception in Test-2.

6.3.3 Test-3: Subjective evaluation on real dataset

Since Test-2 neglects the visual perception of real images, we also explore the subjective evaluation based on the results in Test-3. Like [113], we decompose the perceptual quality of deraining into two dimensions: **Clearness** and **Authenticity**. Specifically, the clearness metric indicates how thoroughly the rain has been removed, while the authenticity metric indicates how realistic the derained image looks. We first manually selected 180 representative images from the derained results in Test-2 as a random library, and then 50 images will be randomly selected for each evaluator. Since we perform the pairwise comparisons rather than individual ratings, each selected image will restul in $9 \times 9 \times 3 = 108$ comparisons (9 datasets, 3 methods), and 50 images will be 5,400 comparisons. In this study, 20 volunteers aging from 20 to 35 take part in the test, half of whom are professionals with an academic background. After all the pairwise comparisons are ranked, we fit the Bradley-Terry [112] model to estimate the raw subjective scores for each dataset, as shown in Table 11. The final scores of Test-3 after regularization are shown in Fig. 11. We can see that Rain100H has 0.19 while MPID has 0.10, which is consistent with the second row in Fig. 10.

6.3.4 Test-4: High-level task evaluation on real dataset

Finally, we would like to investigate the effect of the deraining results on subsequent high-level tasks. In this study, object detection task is performed as an example. The faster R-CNN [2] pre-trained on PASCAL_VOC_2007 [114] and the YOLO-V4 [31] pre-trained on MS COCO [115] are used for evaluating the performance. The *mAP* score is used as the quantitative evaluation metric. The object detection results are shown in Table 11, from which we see that only a few detection results are higher than the original detection results. That is because SID methods are not optimized towards the task of object detection, and the deraining

process might lose some discriminative and meaningful semantic information [116]. We also calculate the scores of the datasets according to the improvement proportion of the detection results after rain removal, as illustrated in Fig. 11. We can see that the scores of the dataset are fairly average, except for Raincity and Outdoor due to the remnants of rain streaks. We also show some visualization results of the object detection task in the third row of Fig.10, from which we see that in most cases there is a decrease in the detection performance after the rain removal in terms of detection category and detection accuracy.

6.3.5 Calculation of EOD

To ensure the diversity of data rank experiments and the reliability of *EOD*, we decided to retain the result of Test-2. After obtaining the results of the four tests, we can calculate the *EOD* score as

$$EOD = \lambda_1 S_1 + \lambda_2 S_2 + \lambda_3 S_3 + \lambda_4 S_4, \quad (28)$$

where λ_1 , λ_2 , λ_3 and λ_4 are the weights of each test. S_i is the deraining performance of Test- i ($i = 1, 2, 3, 4$). Note that we conducted extensive experiments determine the appropriate weight and finally determined each test's weight as 0.3, 0.1, 0.3 and 0.3, respectively. We set the weight λ_2 of Test-2 to 0.1 to reduce the unreliability of no-reference IQA. The calculated *EOD* scores on each dataset are shown in Fig. 12, from which we see that Rain800 obtains the highest score while Outdoor-Rain obtains the lowest score. Note that the *EOD* shown in Fig. 12 are comprehensive evaluated scores based on both synthetic and real data, objective and subjective metrics, low-level and high-level tasks, which can help the researchers of interest to choose the right and appropriate data when dealing with SID task. The result on each test shown in Fig. 11 can also be used as reference. From the results and analysis, we can draw some conclusions: 1) to obtain state-of-the-art deraining performance, datasets like RainCityscapes, Rain800 and Outdoor-Rain will be more challenging; 2) to obtain better deraining performance in real scenario, datasets like Rain100H, Rain800 and Rain14000 will be more effective; 3) to obtain better performance on the object detection task, datasets like Rain100L, SPA-DATA and MPID will be more effective.

7 CONCLUSIONS AND FUTURE WORK

In this paper, we have rethought the data-driven SID problem by providing a very comprehensive analysis and new division criteria from new perspectives. Specifically, we firstly re-examined the three factors (i.e., *data*, *rain model* and *network architecture*) and divides them under new and more reasonable criteria (i.e., *general vs. specific*, *synthetical vs. mathematical*, *black-box vs. white-box*). After that, we carefully analyze the relationships of the three factors from an effectiveness perspective of data, and reveal two different solving paradigms (*explicit vs. implicit*) for the SID task. Besides, we conduct extensive data ranking experiments to evaluate and calculate the effectiveness of recently proposed SID data. The following remarks and hypotheses can be drawn:

- 1) Most of existing SID methods do not strictly follow Eqn. (16), making the solving paradigm of many existing SID methods not solid. From this viewpoint, Li *et al.* [21] is a good example, in which the three factors are combined to make the solution process more explainable.
- 2) The current mainstream SID datasets cannot enhance the high-level task due to the large gap between the synthetic

and real rain. As such, it is recommended to build the datasets with similar distributions as the real rain. Note that SPA-Data [27] is a good example with a satisfactory score from our data ranking experiment.

- 3) The existing no-reference IQA metrics are not suitable for evaluating the SID tasks. However, subjective evaluation consumes a lot of time and labor. Therefore, it is very urgent to develop a specific no-reference IQA metric for evaluating the SID tasks in real scenario.
- 4) Data play an important role in all data-driven SID tasks. As such, how to produce more effective data is also an important topic in future. It is recommended to design more effective rain models to generate the rain images close to real ones, and we also suggest exploring new and effective automatic rain generation modes, such as DerainCycleGAN [39] or Wang *et al.* [117].

Despite the results and analysis of this study, there is still much space to be explored for the SID task in future. We believe that improving the interpretability of the solving process will be beneficial to improve the deraining performance and applications of SID methods, which will be important directions for future research. We have verified the effectiveness of the data by extensive experiments, and we will design more experiments to study the effectiveness of the rain model and network architecture to explore more interpretabilities in near future.

ACKNOWLEDGMENTS

This work is partially supported by the National Natural Science Foundation of China (62072151, 61672365), Anhui Provincial Natural Science Fund for Distinguished Young Scholars (2008085J30), and the Fundamental Research Funds for the Central Universities of China (JZ2019HGPA0102). Zhao Zhang is the corresponding author of this paper.

REFERENCES

- [1] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.
- [2] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.
- [3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.
- [4] B. Zoph, V. Vasudevan, J. Shlens, and Q. V. Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.
- [5] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 10, pp. 1915–1926, 2011.
- [6] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2013, pp. 1155–1162.

- [7] T.-X. Jiang, T.-Z. Huang, X.-L. Zhao, L.-J. Deng, and Y. Wang, "A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4057–4066.
- [8] P. C. Barnum, S. Narasimhan, and T. Kanade, "Analysis of rain and snow in frequency space," *International journal of computer vision*, vol. 86, no. 2-3, p. 256, 2010.
- [9] J. Liu, W. Yang, S. Yang, and Z. Guo, "D3r-net: Dynamic routing residue recurrent network for video rain removal," *IEEE Transactions on Image Processing*, vol. 28, no. 2, pp. 699–712, 2018.
- [10] —, "Erase or fill? deep joint recurrent rain removal and reconstruction in videos," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3233–3242.
- [11] J. Xu, W. Zhao, P. Liu, and X. Tang, "An improved guidance image based method to remove rain and snow in a single image," *Computer and Information Science*, vol. 5, no. 3, p. 49, 2012.
- [12] —, "Removing rain and snow in a single image using guided filter," in *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, vol. 2. IEEE, 2012, pp. 304–307.
- [13] L.-J. Deng, T.-Z. Huang, X.-L. Zhao, and T.-X. Jiang, "A directional global sparse model for single image rain removal," *Applied Mathematical Modelling*, vol. 59, pp. 662–679, 2018.
- [14] Y. Wang, S. Liu, C. Chen, and B. Zeng, "A hierarchical approach for rain or snow removing in a single color image," *IEEE Transactions on Image Processing*, vol. 26, no. 8, pp. 3936–3950, 2017.
- [15] Y. Chang, L. Yan, and S. Zhong, "Transformed low-rank model for line pattern noise removal," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 1726–1734.
- [16] Y.-L. Chen and C.-T. Hsu, "A generalized low-rank appearance model for spatio-temporally correlated rain streaks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 1968–1975.
- [17] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown, "Rain streak removal using layer priors," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2736–2744.
- [18] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley, "Removing rain from single images via a deep detail network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 3855–3863.
- [19] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1357–1366.
- [20] X. Hu, C.-W. Fu, L. Zhu, and P.-A. Heng, "Depth-attentional features for single-image rain removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8022–8031.
- [21] R. Li, L.-F. Cheong, and R. T. Tan, "Heavy rain image restoration: Integrating physics model and conditional adversarial learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1633–1642.
- [22] H. Zhang, V. Sindagi, and V. M. Patel, "Image de-raining using a conditional generative adversarial network," *IEEE transactions on circuits and systems for video technology*, 2019.
- [23] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 695–704.
- [24] X. Li, J. Wu, Z. Lin, H. Liu, and H. Zha, "Recurrent squeeze-and-excitation context aggregation net for single image deraining," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 254–269.
- [25] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng, "Progressive image deraining networks: a better and simpler baseline," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3937–3946.
- [26] R. Qian, R. T. Tan, W. Yang, J. Su, and 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.
- [27] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. Lau, "Spatial attentive single-image deraining with a high quality real rain dataset," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12 270–12 279.
- [28] W. Wei, D. Meng, Q. Zhao, Z. Xu, and Y. Wu, "Semi-supervised transfer learning for image rain removal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3877–3886.
- [29] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.
- [30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [31] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [32] S. Li, I. B. Araujo, W. Ren, Z. Wang, E. K. Tokuda, R. H. Junior, R. Cesar-Junior, J. Zhang, X. Guo, and 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.
- [33] H. Wang, M. Li, Y. Wu, Q. Zhao, and D. Meng, "A survey on rain removal from video and single image," *arXiv preprint arXiv:1909.08326*, 2019.
- [34] W. Yang, R. T. Tan, S. Wang, Y. Fang, and J. Liu, "Single image deraining: From model-based to data-driven and beyond," *IEEE Transactions on pattern analysis and machine intelligence*, 2020.
- [35] H. Wang, Q. Xie, Y. Wu, Q. Zhao, and D. Meng, "Single image rain streaks removal: a review and an exploration," *International Journal of Machine Learning and Cybernetics*, vol. 11, no. 4, pp. 853–872, 2020.
- [36] Y. Wang, C. Ma, and B. Zeng, "Multi-decoding deraining network and quasi-sparsity based training," in *Proceedings of the IEEE/CVF Conference on Computer Vision and*

- Pattern Recognition*, 2021, pp. 13 375–13 384.
- [37] S. Ni, X. Cao, T. Yue, and X. Hu, “Controlling the rain: From removal to rendering,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6328–6337.
- [38] R. Quan, X. Yu, Y. Liang, and Y. Yang, “Removing rain-drops and rain streaks in one go,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1–10.
- [39] Y. Wei, Z. Zhang, Y. Wang, M. Xu, Y. Yang, S. Yan, and M. Wang, “Deraincyclegan: Rain attentive cyclegan for single image deraining and rainmaking,” *IEEE Transactions on Image Processing*, 2021.
- [40] Y. Wang, Y. Song, C. Ma, and B. Zeng, “Rethinking image deraining via rain streaks and vapors,” in *European Conference on Computer Vision*. Springer, 2020, pp. 367–382.
- [41] S. Li, W. Ren, J. Zhang, J. Yu, and X. Guo, “Single image rain removal via a deep decomposition-composition network,” *Computer Vision and Image Understanding*, vol. 186, pp. 48–57, 2019.
- [42] G. Schaefer and M. Stich, “Ucid: An uncompressed color image database,” in *Storage and Retrieval Methods and Applications for Multimedia 2004*, vol. 5307. International Society for Optics and Photonics, 2003, pp. 472–480.
- [43] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 5, pp. 898–916, 2010.
- [44] S. Patterson, “Adding rain to a photo with photoshop,” Website, <http://www.photoshopessentials.com/photo-effects/rain/>.
- [45] W. Wei, L. Yi, Q. Xie, Q. Zhao, D. Meng, and Z. Xu, “Should we encode rain streaks in video as deterministic or stochastic?” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2516–2525.
- [46] K. Garg and S. K. Nayar, “Photorealistic rendering of rain streaks,” *ACM Transactions on Graphics (TOG)*, vol. 25, no. 3, pp. 996–1002, 2006.
- [47] S. Patterson, “How to add rain to a photo,” Website, <https://www.photoshopessentials.com/photo-effects/photoshop-weather-effects-rain/>.
- [48] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, “The cityscapes dataset for semantic urban scene understanding,” 2016.
- [49] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus, “Indoor segmentation and support inference from rgb-d images,” in *European conference on computer vision*. Springer, 2012, pp. 746–760.
- [50] K. Jiang, Z. Wang, P. Yi, C. Chen, B. Huang, Y. Luo, J. Ma, and J. Jiang, “Multi-scale progressive fusion network for single image deraining,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8346–8355.
- [51] C. Wang, Y. Wu, Z. Su, and J. Chen, “Joint self-attention and scale-aggregation for self-calibrated deraining network,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 2517–2525.
- [52] W. Yu, Z. Huang, W. Zhang, L. Feng, and N. Xiao, “Gradual network for single image de-raining,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1795–1804.
- [53] K. Garg and S. K. Nayar, “Vision and rain,” *International Journal of Computer Vision*, vol. 75, no. 1, pp. 3–27, 2007.
- [54] C. Sakaridis, D. Dai, and L. Van Gool, “Semantic foggy scene understanding with synthetic data,” *International Journal of Computer Vision*, vol. 126, no. 9, pp. 973–992, 2018.
- [55] X. Fu, Q. Qi, Z.-J. Zha, Y. Zhu, and X. Ding, “Rain streak removal via dual graph convolutional network,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, 2021.
- [56] S. W. Zamir, A. Arora, S. Khan, M. Hayat, F. S. Khan, M.-H. Yang, and L. Shao, “Multi-stage progressive image restoration,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2021.
- [57] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley, “Clearing the skies: A deep network architecture for single-image rain removal,” *IEEE Transactions on Image Processing*, vol. 26, no. 6, pp. 2944–2956, 2017.
- [58] Y. Quan, S. Deng, Y. Chen, and H. Ji, “Deep learning for seeing through window with raindrops,” in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 2463–2471.
- [59] H. Wang, Q. Xie, Q. Zhao, and D. Meng, “A model-driven deep neural network for single image rain removal,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3103–3112.
- [60] W. Yang, R. T. Tan, J. Feng, J. Liu, S. Yan, and Z. Guo, “Joint rain detection and removal from a single image with contextualized deep networks,” *IEEE transactions on pattern analysis and machine intelligence*, 2019.
- [61] Y. Ye, Y. Chang, H. Zhou, and L. Yan, “Closing the loop: Joint rain generation and removal via disentangled image translation,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2053–2062.
- [62] C. Chen and H. Li, “Robust representation learning with feedback for single image deraining,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7742–7751.
- [63] H. Huang, A. Yu, and R. He, “Memory oriented transfer learning for semi-supervised image deraining,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 7732–7741.
- [64] Y. Wei, Z. Zhang, Y. Wang, H. Zhang, M. Zhao, M. Xu, and M. Wang, “Semi-deraingan: A new semi-supervised single image deraining network,” in *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2021, pp. 1–6.
- [65] Y. Du, J. Xu, X. Zhen, M.-M. Cheng, and L. Shao, “Conditional variational image deraining,” *IEEE Transactions on Image Processing*, vol. 29, pp. 6288–6301, 2020.
- [66] J. Pan, J. Dong, Y. Liu, J. Zhang, J. Ren, J. Tang, Y. W. Tai, and M.-H. Yang, “Physics-based generative adversarial models for image restoration and beyond,” *IEEE transactions on pattern analysis and machine intelligence*, 2020.
- [67] Y. Yang, W. Ran, and H. Lu, “Rddan: A residual dense dilated aggregated network for single image deraining,” in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.
- [68] R. Yasarla and V. M. Patel, “Confidence measure guided

- single image de-raining,” *IEEE Transactions on Image Processing*, vol. 29, pp. 4544–4555, 2020.
- [69] R. Yasarla, V. A. Sindagi, and V. M. Patel, “Syn2real transfer learning for image deraining using gaussian processes,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2726–2736.
- [70] C. Wang, X. Xing, Y. Wu, Z. Su, and J. Chen, “Dcsfn: Deep cross-scale fusion network for single image rain removal,” in *Proceedings of the 28th ACM International Conference on Multimedia*, 2020, pp. 1643–1651.
- [71] H. Zhu, C. Wang, Y. Zhang, Z. Su, and G. Zhao, “Physical model guided deep image deraining,” in *2020 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2020, pp. 1–6.
- [72] S. Deng, M. Wei, J. Wang, Y. Feng, L. Liang, H. Xie, F. L. Wang, and M. Wang, “Detail-recovery image deraining via context aggregation networks,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 14 560–14 569.
- [73] C.-Y. Lin, Z. Tao, A.-S. Xu, L.-W. Kang, and F. Akhyar, “Sequential dual attention network for rain streak removal in a single image,” *IEEE Transactions on Image Processing*, vol. 29, pp. 9250–9265, 2020.
- [74] Z. Wang, J. Li, and G. Song, “Dtdn: Dual-task de-raining network,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1833–1841.
- [75] X. Liu, M. Suganuma, Z. Sun, and 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.
- [76] G. Wang, C. Sun, and A. Sowmya, “Erl-net: Entangled representation learning for single image de-raining,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 5644–5652.
- [77] X. Fu, B. Liang, Y. Huang, X. Ding, and J. Paisley, “Lightweight pyramid networks for image deraining,” *IEEE transactions on neural networks and learning systems*, 2019.
- [78] Y. Yang and H. Lu, “Single image deraining via recurrent hierarchy enhancement network,” in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 1814–1822.
- [79] H. Zhu, X. Peng, J. T. Zhou, S. Yang, V. Chandrasekh, L. Li, and J.-H. Lim, “Single image rain removal with unpaired information: A differentiable programming perspective,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 9332–9339.
- [80] Y. Wei, Z. Zhang, H. Zhang, R. Hong, and M. Wang, “A coarse-to-fine multi-stream hybrid deraining network for single image deraining,” in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 628–637.
- [81] Y. Yang and H. Lu, “Single image deraining using a recurrent multi-scale aggregation and enhancement network,” in *2019 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2019, pp. 1378–1383.
- [82] X. Jin, Z. Chen, J. Lin, Z. Chen, and W. Zhou, “Unsupervised single image deraining with self-supervised constraints,” in *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2019, pp. 2761–2765.
- [83] R. Yasarla and V. M. Patel, “Uncertainty guided multi-scale residual learning-using a cycle spinning cnn for single image de-raining,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8405–8414.
- [84] W. Yang, J. Liu, S. Yang, and Z. Guo, “Scale-free single image deraining via visibility-enhanced recurrent wavelet learning,” *IEEE Transactions on Image Processing*, vol. 28, no. 6, pp. 2948–2961, 2019.
- [85] J. Pan, S. Liu, D. Sun, J. Zhang, Y. Liu, J. Ren, Z. Li, J. Tang, H. Lu, Y.-W. Tai *et al.*, “Learning dual convolutional neural networks for low-level vision,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 3070–3079.
- [86] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin, “Non-locally enhanced encoder-decoder network for single image de-raining,” in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1056–1064.
- [87] Z. Fan, H. Wu, X. Fu, Y. Huang, and X. Ding, “Residual-guide network for single image deraining,” in *Proceedings of the 26th ACM international conference on Multimedia*, 2018, pp. 1751–1759.
- [88] R. Li, L.-F. Cheong, and R. T. Tan, “Single image deraining using scale-aware multi-stage recurrent network,” *arXiv preprint arXiv:1712.06830*, 2017.
- [89] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, “Generative adversarial nets,” *Advances in neural information processing systems*, vol. 27, 2014.
- [90] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, “Unpaired image-to-image translation using cycle-consistent adversarial networks,” in *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2017.
- [91] K. He, J. Sun, and X. Tang, “Guided image filtering,” in *European conference on computer vision*. Springer, 2010, pp. 1–14.
- [92] A. Kraskov, H. Stögbauer, and P. Grassberger, “Estimating mutual information,” *Physical review E*, vol. 69, no. 6, p. 066138, 2004.
- [93] J. R. Hershey and P. A. Olsen, “Approximating the kullback leibler divergence between gaussian mixture models,” in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP’07*, vol. 4. IEEE, 2007, pp. IV–317.
- [94] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, “Long-term recurrent convolutional networks for visual recognition and description,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 2625–2634.
- [95] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in *European conference on computer vision*. Springer, 2016, pp. 694–711.
- [96] O. Ronneberger, P. Fischer, and 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.
- [97] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “Imagenet: A large-scale hierarchical image database,”

- in *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009, pp. 248–255.
- [98] F. Yu and V. Koltun, “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint arXiv:1511.07122*, 2015.
- [99] J. Hu, L. Shen, and G. Sun, “Squeeze-and-excitation networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132–7141.
- [100] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6848–6856.
- [101] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010, pp. 249–256.
- [102] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in neural information processing systems*, 2017, pp. 5998–6008.
- [103] V. Badrinarayanan, A. Kendall, and R. Cipolla, “Segnet: A deep convolutional encoder-decoder architecture for image segmentation,” *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 12, pp. 2481–2495, 2017.
- [104] B. Zoph and Q. V. Le, “Neural architecture search with reinforcement learning,” *arXiv preprint arXiv:1611.01578*, 2016.
- [105] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *arXiv preprint arXiv:1609.02907*, 2016.
- [106] Q. Huynh-Thu and M. Ghanbari, “Scope of validity of psnr in image/video quality assessment,” *Electronics letters*, vol. 44, no. 13, pp. 800–801, 2008.
- [107] A. C. Brooks, X. Zhao, and T. N. Pappas, “Structural similarity quality metrics in a coding context: exploring the space of realistic distortions,” *IEEE Transactions on image processing*, vol. 17, no. 8, pp. 1261–1273, 2008.
- [108] L. Liu, B. Liu, H. Huang, and A. C. Bovik, “No-reference image quality assessment based on spatial and spectral entropies,” *Signal Processing: Image Communication*, vol. 29, no. 8, pp. 856–863, 2014.
- [109] A. Mittal, R. Soundararajan, and A. C. Bovik, “Making a “completely blind” image quality analyzer,” *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2012.
- [110] A. Mittal, A. K. Moorthy, and A. C. Bovik, “Blind/referenceless image spatial quality evaluator,” in *2011 conference record of the forty fifth asilomar conference on signals, systems and computers (ASILOMAR)*. IEEE, 2011, pp. 723–727.
- [111] M. A. Saad, A. C. Bovik, and C. Charrier, “A dct statistics-based blind image quality index,” *IEEE Signal Processing Letters*, vol. 17, no. 6, pp. 583–586, 2010.
- [112] Bradley, R. Allan, and M. E. Terry, “Rank analysis of incomplete block designs: I. the method of paired comparisons,” *Biometrika*, vol. 39.3/4, pp. 324–345, 1952.
- [113] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang, “Benchmarking single-image dehazing and beyond,” *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 492–505, 2018.
- [114] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
- [115] H. Caesar, J. Uijlings, and V. Ferrari, “Coco-stuff: Thing and stuff classes in context,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1209–1218.
- [116] Y. Pei, Y. Huang, Q. Zou, Y. Lu, and S. Wang, “Does haze removal help cnn-based image classification?” in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 682–697.
- [117] H. Wang, Z. Yue, Q. Xie, Q. Zhao, and D. Meng, “From rain removal to rain generation,” *arXiv preprint arXiv:2008.03580*, 2020.