

Article

Rain Rendering and Construction of *Rain Vehicle Color-24* Dataset

Mingdi Hu ^{1,*} , Chenrui Wang ¹, Jingbing Yang ¹, Yi Wu ¹, Jiulun Fan ¹ and Bingyi Jing ^{2,*}

¹ School of Communications and Information Engineering, Xi'an University of Posts and Telecommunications, Chang'an West St., Chang'an District, Xi'an 710121, China

² Department of Statistics & Data Science, Southern University of Science and Technology, 1088 Xueyuan Avenue, Shenzhen 518055, China

* Correspondence: humingdi2005@xupt.edu.cn (M.H.); jingby@sustech.edu.cn (B.J.)

Abstract: The fine identification of vehicle color can assist in criminal investigation or intelligent traffic management law enforcement. Since almost all vehicle-color datasets that are used to train models are collected in good weather, the existing vehicle-color recognition algorithms typically show poor performance for outdoor visual tasks. In this paper we construct a new *Rain Vehicle Color-24* dataset by rain-image rendering using *PS* technology and a *SyRaGAN* algorithm based on the *Vehicle Color-24* dataset. The dataset contains a total of 40,300 rain images with 125 different rain patterns, which can be used to train deep neural networks for specific vehicle-color recognition tasks. Experiments show that the vehicle-color recognition algorithms trained on the new dataset *Rain Vehicle Color-24* improve accuracy to around 72% and 90% on rainy and sunny days, respectively. The code is available at humingdi2005@github.com.

Keywords: rain rendering; deep convolutional neural network; rain datasets; identification of vehicle color; single-image deraining algorithm



Citation: Hu, M.; Wang, C.; Yang, J.; Wu, Y.; Fan, J.; Jing, B. Rain Rendering and Construction of *Rain Vehicle Color-24* Dataset. *Mathematics* **2022**, *10*, 3210. <https://doi.org/10.3390/math10173210>

Academic Editors: Wen-Yu Chung and Sebastian Iwaszenko

Received: 28 July 2022

Accepted: 1 September 2022

Published: 5 September 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

MSC: 54H30; 68U10; 94A08

1. Introduction

With the development of computer vision technology and hardware, vision algorithms based on deep learning have achieved unprecedented performances, and are increasingly applied to practical scenarios. For example, color recognition is applied in vehicle tracking [1–11]. In order to train vehicle-color recognition algorithms, scholars have constructed multiple vehicle-color datasets. For example, the vehicle-color dataset by Chen et al. [1] included eight color categories, and each image contained one vehicle. The dataset collected by Jeong et al. [3] contains seven colors, and Tilakaratna et al. [12] expanded their dataset to 13 categories. Hu et al. [4] constructed a new benchmark vehicle-color dataset, *Vehicle Color-24*, with 24 colors, and proposed a novel vision color recognition (*VCR*) method based on a Smooth Modulation Neural Network with Multi-Scale Feature Fusion (*SMNN-MSFF*). The *Vehicle Color-24* includes 10,091 images with a total of 31,232 vehicles, and each image contains up to nine vehicles. These datasets and algorithms have been very conducive to vehicle recognition tasks. However, the images of the above datasets are mostly collected in good weather. On the other hand, criminal investigation or intelligent traffic management law enforcement often encounter bad conditions, especially rainy weather [5].

The raindrops are likely to turn into rain streaks due to their high density and fast speed. Rain streaks will typically produce reflection or refraction, and often blur and deform the images captured by cameras, which poses challenges for subsequent visual tasks. Studying low- and high-level tasks has become a hot research direction [5]. Many scholars have paid attention to the joint processing of low- and high-level tasks. The

generalization in object detection is improved by embedding domain adaptation, image restoration, style transfer, or other modules into object detection methods or few-shot learning mechanisms [6–9,13–16]. However, these works all require additional modules, which will undoubtedly increase the burden of outdoor equipment. To solve this problem, a natural solution is to construct image datasets with rich diversity and in various environments for specific tasks, which can be used to train models facing subsequent high-level tasks without adding modules, while still improving generalization. However, it is very expensive to collect such datasets in practice, so this paper constructs the *Rain Vehicle Color-24* dataset by rain-image-rendering technology toward this end. It aims to address the specific vehicle-color recognition task.

There exists much literature on the construction of rain image datasets. For example, Garg and Nayar (2006) used a particle simulator to synthesize rain patterns, and then superimposed the rain patterns with clean backgrounds to synthesize the rain image [17,18]. Hu et al. [19] and Tremblay et al. [20] rendered rain images based on the complex fusion of background and rain layers. Data-driven synthetic rain images based on a generative adversarial network (GAN) have recently received increasing attention [20–22]. Wang et al. [23] constructed a large-scale real-rain image dataset, i.e., *SPA-Data* (spatial attentive data). However, rain vehicle-color image datasets are rare.

Inspired by the above works, this paper constructs a *Rain Vehicle Color-24* dataset by rendering rain images using photoshop (*PS*) technology and the *SyRaGAN* algorithm [24]. The *Rain Vehicle Color-24* dataset has a total of 40,300 rain images with 125 kinds of rain-streak patterns, which is beneficial in improving the generalization of the deep neural network models for the fine identification of vehicle colors. Using the benchmark datasets, *Rain100L* and *Rain Vehicle Color-24*, we trained the current state-of-the-art (SOTA) algorithm *PReNet* network [25] and the lightly weighted *LDVS* deraining network [26] to obtain the models *PReNet1*, *LDVS1*, *PReNet2* and *LDVS2*. After testing on both synthetic and real data, these above models showed obvious advantages for the deraining task. Vehicle-color recognition methods trained on the new datasets showed improved performance for vehicle-color classifying in both sunny and rainy conditions.

The main contributions of this paper are as follows:

- (1) This paper constructs the *Rain Vehicle Color-24* dataset by rain-image-rendering technology, in order to address the specific task of the vehicle-color fine recognition. Both model-based and data-driven-based rendering are used: the former synthesizes 300 images by *PS* to form one subset in which clean background images are from the *Vehicle Color-24*; the latter, i.e., the *SyRaGAN* network, synthesizes 8000 vehicle images to form another subset in which clean background images are also from the *Vehicle Color-24*;
- (2) This dataset helps to increase the performance of vehicle-color recognition methods on rainy days since the *Rain Vehicle Color-24* dataset consists paired vehicle-color rain images with various rain patterns;
- (3) We improve the performances of existing algorithms for vehicle-color identification in rainy conditions. Vehicle-color identification plays a key role in intelligent traffic management and criminal investigation. However, existing algorithms are typically trained on the datasets collected in good weather conditions, which suffer from poor performance in poor weather conditions, such as rainy weather. In this paper, we show that our newly constructed dataset is critically beneficial to the performances of existing algorithms for vehicle-color identification in rainy conditions.

The rest of this paper is structured as follows: Section 2 reviews the related work; Section 3 introduces construction of the *Rain Vehicle Color-24*; Section 4 compares complex experimental results in detail; Section 5 concludes the paper.

2. Related Work

2.1. Photoshop (PS) Technology

At present, *PS* is the main synthetic rain image technology. Garg and Nayar (2006) synthesized various types of rain patterns, and then directly added them to the corresponding clean background images to obtain paired rain images. There has been some work on the simple stacking of background and rain layers. For example, Li et al. [17] proposed a paired rain-image test set (*Rain12*) composed of one type of rain pattern and 12 background images to synthesize 12 rain images. Yang et al. [18] constructed a dataset (*Rain100H*) containing 1900 rain/clean image pairs, with 1800 image pairs for training and 100 image pairs for testing. These datasets are often used as comparative datasets; however, the types of rain streaks are relatively simple. There has also been research on the complex fusion of background and rain layers. For example, Li et al. [27] used *PS* to add noise to form rain patterns of different intensities and directions, and then synthesized the rain images based on the screen blend model (*SBM*), producing the dataset *DDC-Data*. Wang et al. [28] constructed a dataset *QSMD-Data* using synthetic rain images based on a screen-shrouded model (*SBM*).

The process of *PS* technology is as follows (as shown in Figure 1):

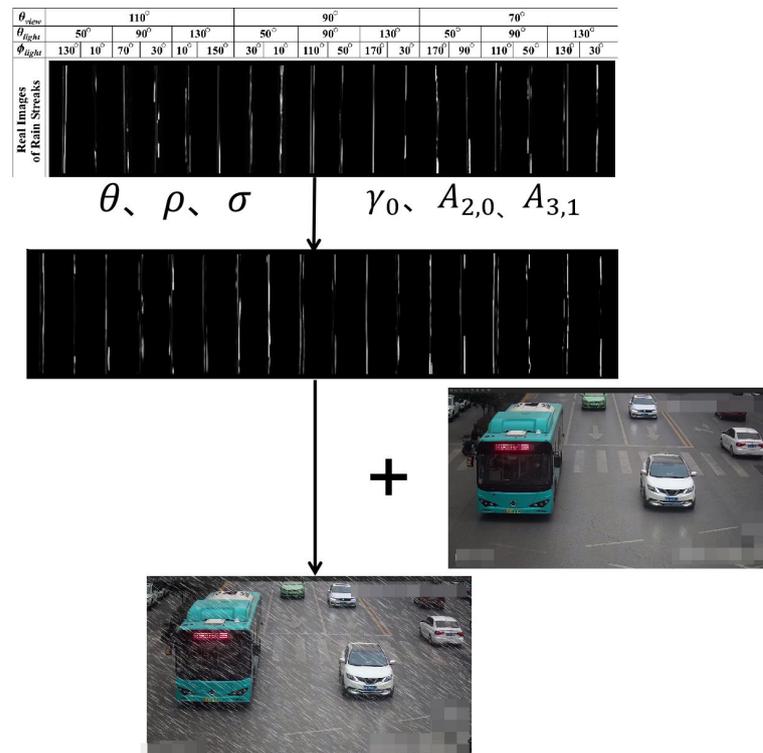


Figure 1. Synthesis process of rain image using photoshop software [5].

- (1) First, the rain-streak patterns under the conditions of two light sources with different angles of illumination and different camera directions are constructed.
- (2) Then, the rain images are synthesized according to the raindrop-modeling equation:

$$\omega_n = [n(n - 1)(n + 2)\sigma / (\rho r_0^3)]^{1/2}, \tag{1}$$

$$r(t, \theta, \phi) = r_0[1 + A_{2,0}\sin(\omega_2 t)P_{2,0}(\theta) + A_{3,1}\sin(\omega_3 t)\cos(\phi)P_{3,1}(\theta)], \tag{2}$$

where r is the surface tension, ρ is the density of water, θ is the angle, ϕ is the azimuth, r_0 is the size of the raindrop, $A_{2,0}$ and $A_{3,1}$ are the amplitudes, and ω_n is the frequency, $P_{n,m}(\theta)$ is the Legendre function that describes the dependence of the shape on the angle

- θ for the mode (n,m). The parameters are usually set by empirical knowledge (Garg and Nayar, 2006 [29]), and the rain pattern is synthesized by formulas (1) and (2).
- (3) Finally, the synthesized rain pattern is directly added to the clean background image to get the rain image.

2.2. Data-Driven Rain-Image-Rendering Technology

There are also semi-automatic methods to collect rain images. For example, Qian et al. [30] collected a paired raindrop image dataset. Jin et al. [31] constructed the rain image dataset, *RaidaR*, under a wide range of circumstances, using cameras on the roof. However, due to the high cost of data collection, in practice, real paired rain images are not sufficient to train models; therefore, most methods are trained with the help of rendering data. Wei et al. [21] synthesized a rain image dataset by an unsupervised learning mechanism with a constrained *CycleGAN* network, resulting in a dataset *Rain200L* with more types of rain-streak patterns and more realistic visual effects. Wang et al. [22] used a Bayesian model to construct a rain-image generation network to generate more than 120 rain-streak patterns. Further exploration of more reasonable and accurate rain-image synthesis models and the generation of more realistically rendered rain images are also important research directions. In addition, the construction of a specific vehicle rain-image dataset for specific vehicle target detection will be very much meaningful and badly needed.

2.3. Single-Image Rain-Removal Algorithm

Algorithms for single-image rain removal are mainly divided into two categories: traditional model-driven method and data-driven deep neural network [5]. Model-driven algorithms rely on the statistical analysis of rain streaks and background scenes, and use priors on rain streaks and background layers to build a rain-removing model to iteratively give explicit solutions. Chen et al. [32] constructed a generalized low-rank model, and they distinguished the low-rank rain pattern as a separate layer from the background layer. Lou et al. [33] proposed a highly discriminative sparse coding method to separate the rain pattern from the background layer.

On the other hand, data-driven deep neural network algorithm for rain removal was first proposed by Fu et al. [34]. The algorithm reconstructs a clean background layer after removing the rain pattern from the high-frequency layer. Li et al. [27] divided the rain-removal network into two stages: decomposition and combination. They then used the residuals of the synthetic and original rain images to train the network to improve the rain-removal performance. Ren et al. [25] proposed a simple and effective rain-removal network, i.e., the Rain-Removal Network Baseline Progressive Recurrent Network (*PReNet*). Since then, a variety of algorithms have been proposed, e.g., lightweight pyramid network (*LPNet*) algorithm [35], local binary pattern conditional generative adversarial network (*LBP-CGAN*) [36], lightweight single-image deraining algorithm incorporating visual saliency (*LDVS*) [26]

Since *PReNet* is one of the simple and effective single-image rain-removal algorithms, and *LDVS* is one of the lightweight algorithm, they will be trained on the *Rain100L* and *Rain Vehicle Color-24* benchmark rain datasets. Moreover, we combine the subsequent target recognition algorithms to examine whether the dataset *Rain Vehicle Color-24* can improve the performance of the algorithm's modeling of the low- and high-level joint task and the low-level rain-removal task. The details are given in Sections 4.2 and 4.3.

2.4. Vehicle-Color Recognition Algorithms

Due to its practical significance, vehicle-color recognition has attracted much attention in computer vision. The literature mainly falls into two categories: manual feature-based methods and emerging data-driven deep learning methods [1–11]. Among others, Hu et al. [5] proposed a novel *VCR* method based on a Smooth Modulation Neural Network with Multi-Scale Feature Fusion (*SMNN-MSFF*), which is then trained and evaluated on

the dataset “*Vehicle Color-24*” with 24 vehicle-color classes. *Vehicle Color-24* consists of 10,091 vehicle images from a 100 h urban road surveillance video.

In this paper, we perform rain-image rendering using the *Vehicle Color-24* dataset to construct a specific task dataset for vehicle-color recognition. The objective is to improve the performance of the low- and high-level joint tasks in vehicle-color fine recognition or improve the generalization of vehicle-color fine recognition in bad weather.

3. Construction of *Rain Vehicle Color-24*

3.1. *Vehicle Color-24*

Firstly, 8000 vehicle images are selected from the existing *Vehicle Color-24* as a clean background image with resolution 1747×982 . The dataset consists of 10,091 vehicle images captured from urban road surveillance videos, with a total of 31,232 vehicles and 24 colors. The authors preprocessed the dataset, including lighting adjustment, dehazing, etc. Samples from *Vehicle Color-24* are shown in Figure 2.



Figure 2. Samples from *Vehicle Color-24*.

3.2. *Rendering by PS*

We randomly selected 300 images from *Vehicle Color-24*, and used *PS* software to generate rain patterns of different directions, sizes, and thicknesses by adjusting parameters such as motion blur and color level. We then superimposed 120 kinds of rain streaks with 300 clean background images from the *Vehicle Color-24* dataset to construct a subset of *Rain Vehicle Color-24*. As a result, we have more types of rain patterns to ensure the

diversity of rain images in the new dataset. The partial rain image samples of the rain image subset are shown in Figure 3.

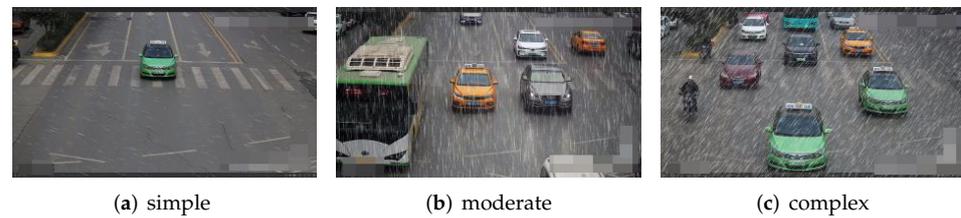


Figure 3. Examples of subset by *PS*. The three images are rendered by *PS* from simple, moderately complex, and complex scenes, respectively. We set the parameters' noise, angle, distance, and Gaussian blur as $(40\%, -20, 50, 0.5)$, $(106\%, -87, 48, 0.3)$, $(146\%, 53, 33, 0.5)$, respectively, for rendering the three rain images.

Figure 3 shows sample instances of this subset, which includes three kinds of images. Every image is formed by a different rain-streak pattern imposed onto a clean vehicle image from a different scene. Three scenes are classified into simple scenes (a single vehicle in an image taken under a clear sky), moderately complex scenes (many vehicles in an image taken under a clear sky), and complex scenes (many vehicles in an image taken under gray skies), respectively.

3.3. Rendering by the *SyRaGAN* Algorithm

In order to further enrich the types of rain streaks, this paper uses the *SyRaGAN* algorithm [24] to render rain patterns on the *Vehicle Color-24* dataset to construct another subset. *SyRaGAN* is inspired by the mapping network used in the latest *I2I* (image-to-Image) translation method, which maps the random noise space to the rain-pattern representation space to generate diverse rain patterns. *SyRaGAN* consists of the feature map network M , encoder network E , two generators G , and two discriminators D . The network input consists of the clean background image x_c and the rain image x_r to produce the synthesized rain images x_{sr1} and x_{sr2} as output, while S_z and S_r are the extracted rain patterns from the network (see Figure 4).

The rain-image rendering process is as follows. First, the clean background image and the rain image are respectively input into the *SyRaGAN* network, and the rain noise is extracted by the mapping network M to produce various rain styles. Second, rain streak is added to the clean background image to generate the rain image. Finally, the generated images are discriminated using the discriminator D . By this way, the network is optimized to generate rain images with various styles. For each image, five kinds of rain images with different directions, sizes, and thicknesses can be correspondingly generated.

To construct a dataset with diversity as wide as possible, we sampled one kind of clean background image with a single vehicle in a image named as "simple scene", and rendered it into five rain images with five rain-streak patterns (Figure 5). Second, we sampled one kind of clean background sample with many vehicles in a image named as "medium scene", and rendered it into five different rain images with five rain-streak patterns (Figure 6). Third, we sampled one kind of many vehicles under gray skies, which we named as "complex scene", and rendered it into five different rain images with five rain-streak patterns (Figure 7). In processes such as those from the clean vehicle dataset *Vehicle Color-24*, 8000 image samples are fed into *SyRaGAN* to obtain 40,000 vehicle rain-image samples. Combined with the rain data subset generated by *PS* technology, 40,300 rain images are finally obtained, with a resolution of 512×384 .

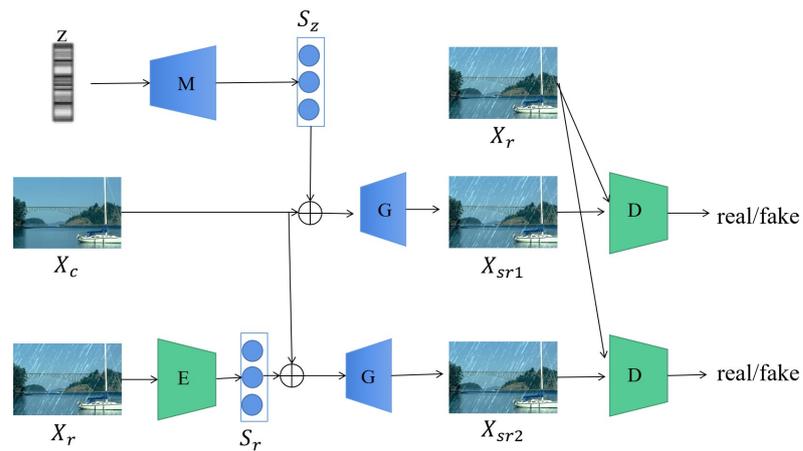


Figure 4. SyRaGAN network structure overview [24]. *M* is the feature extracting module, *E* is the encoder, and *D* are the discriminators. The inputs are clean background image x_c and rain image x_r , and the outputs are the synthesizing rain images x_{sr1}, \dots, x_{sr2} . Further, S_z, S_r are the extracted rain patterns from the network.

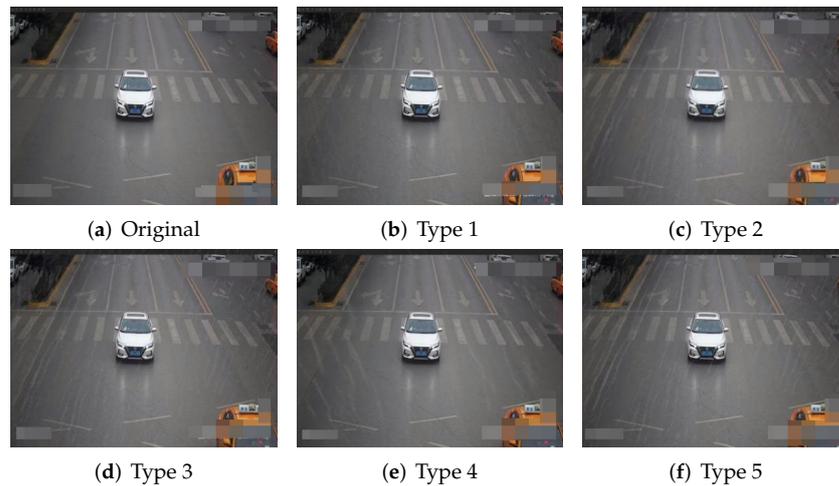


Figure 5. Rendered rain images with single vehicle. (a) Original clean image; (b–f) rendered rain images with 5 kinds of different rain-streak patterns.

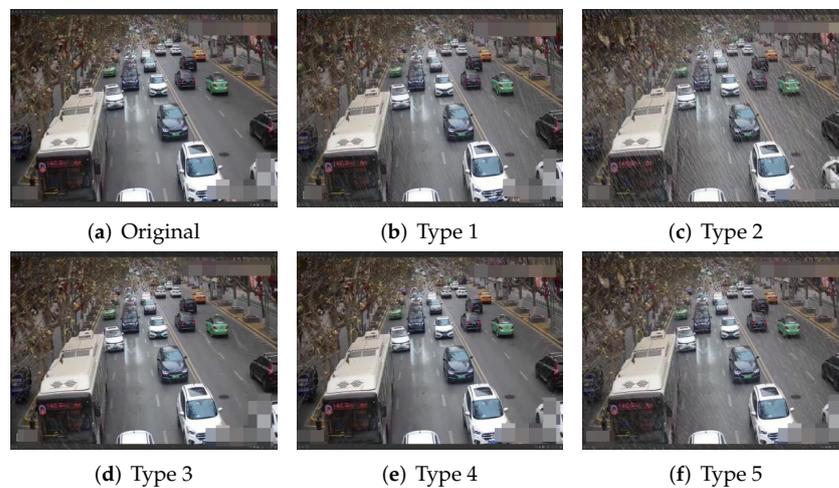


Figure 6. Rendered rain images with many vehicles. (a) Original clean image; (b–f) rendered rain images with 5 kinds of different rain-streak patterns.

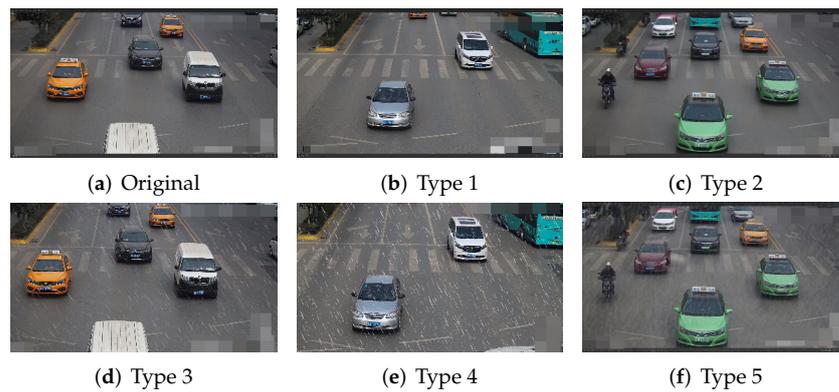


Figure 7. Rendered rain vehicle images under gray skies (the first image is original clean background image, and the rest are rendered rain images with five different types).

To summarize, the previous dataset *Vehicle Color-24* was labeled according to 24 standard vehicle colors, but lacked corresponding rain images. This paper leverages *SyRaGAN* and *PS* technologies to constitute *Rain Vehicle Color-24*. Some samples are illustrated in Figure 8.



Figure 8. Illustrations of some samples from *Rain Vehicle Color-24*.

4. Experimental Results

In this paper we used the metrics *PSNR* and *SSIM* to evaluate the quality of the recovered images. The formulas [37] are

$$MSE = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W (X(i, j) - Y(i, j))^2, \tag{3}$$

$$PSNR = 10 \log_{10} \frac{(2^n - 1)^2}{MSE}, \tag{4}$$

$$SSIM(X, Y) = \left(\frac{2u_X u_Y + C_1}{u_X^2 + u_Y^2 + C_1} \right) * \left(\frac{2\sigma_X \sigma_Y + C_2}{\sigma_X^2 + \sigma_Y^2 + C_2} \right) * \left(\frac{\sigma_{XY} + C_3}{\sigma_X \sigma_Y + C_3} \right). \tag{5}$$

4.1. Experimental Setup

In this paper, we first used *Rain100L* and *Rain Vehicle Color-24* as benchmark datasets to train the two deraining networks *PReNet* [25] and *LDVS* [26], to obtain the deraining network models *PReNet1*, *LDVS1*, *PReNet2*, and *LDVS2*. We then tested them on the synthetic and real rain images. In addition, we used *Faster-RCNN* [38] to detect objects for vehicle-color classification after deraining by *PReNet1*, *LDVS1*, *PReNet2*, and *LDVS2*. The experimental results showed that deraining performance is improved after being trained by the *Rain Vehicle Color-24*, and the performance of *Faster-RCNN* in subsequent vehicle-color fine recognition processes is also improved. All mean average precisions (*mAPs*) of vehicle-color classification are improved when corresponding specific vehicle recognition deep neural networks are trained on *Rain Vehicle Color-24*.

4.2. PReNet Model Trained on Rain100L and Rain Vehicle Color-24

4.2.1. PReNet Network

Ren et al. [25] proposed a better and simpler baseline deraining network with six recurrent modules. Specifically, by repeatedly unfolding a shallow *ResNet*, progressive *ResNet* (*PRN*) was proposed to take advantage of recursive computation. In this paper, we introduce a recurrent-layer *LSTM* module to extract the dependencies of deep features across stages, forming the final framework, which we refer to as progressive recurrent network (*PReNet*). As for loss functions, single-*MSE* or negative-*SSIM* losses are sufficient for training *PRN* and *PReNet*. The illustration of *PReNet* is shown in Figure 9. As shown in this paper, the *PReNet* is one of representative SOTA methods due to its simplicity, efficiency and effectiveness.

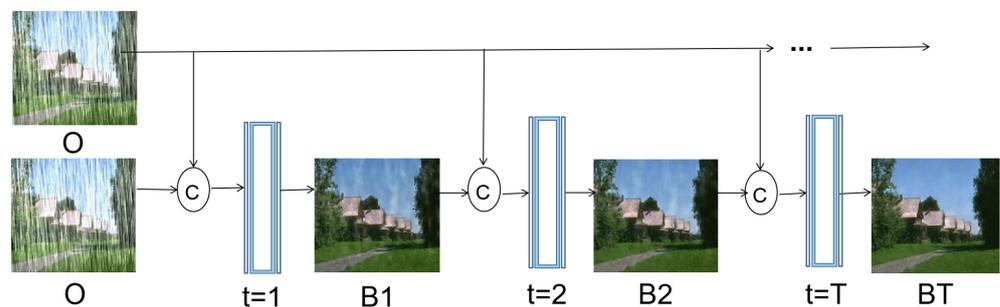


Figure 9. Illustration of *PReNet* [25]. Input *O* is the rain image, output *BT* is the clean background image of the *T* stage output.

4.2.2. Comparison of Synthetic Rain Images

We used the training subset of the *Rain 100L* and *Rain Vehicle Color-24* datasets to train the *PReNet* network, and then we obtained the deraining network models *PReNet1* and *PReNet2*. Finally, we tested the rain-removal performance of the two models on the *Rain100L* test set. The results are given in Figure 10 and Table 1.

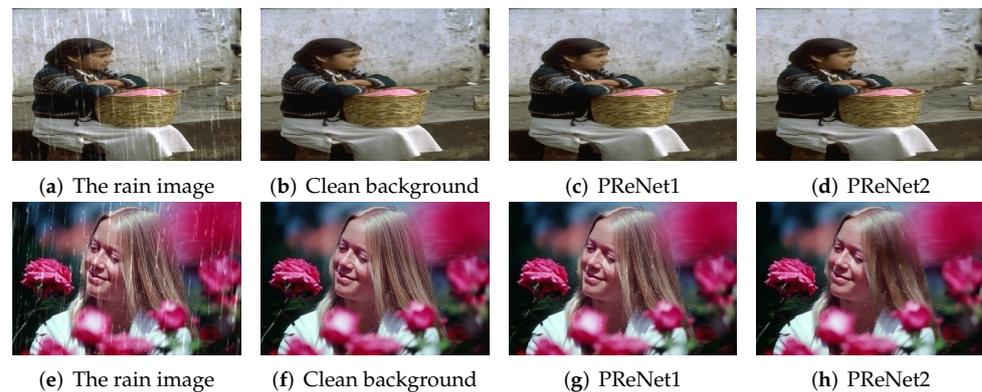


Figure 10. Test results of *PReNet1* and *PReNet2* on two synthetic rain images from the *Rain100L*. *PReNet1* and *PReNet2* represent the different models that are trained on *Rain100L* and *Rain Vehicle Color-24* training subsets, respectively.

We tested the images sampled from the synthetic *Rain100L* test subset with *PReNet1* trained on *Rain100L* and *PReNet2* trained on the *Rain Vehicle Color-24*. The test results are shown in Figure 10. As can be seen from Figure 10, the *PReNet1* model has a better visual effect than the *PReNet2* model with further supporting results given in Table 1. The *PReNet1* outperforms *PReNet2* on the *PSNR* and the *SSIM* have been increased by margins 0.23 and 0.02, respectively. The reason why *PReNet1* performs better is because the test and training images used by *PReNet1* are identically distributed, with the two coming from the same dataset *Rain100L*. On the other hand, the rain-removal effect of *PReNet2* trained with the *Rain Vehicle color-24* is slightly worse than the former, since the test images derained by *PReNet2* are not identically distributed with the training images. Most of the rain streaks have been removed and the background image can be restored. This means that when there is no domain gap, the test effect is better; furthermore, the latter faces the gap between the test and training data and still maintains the performance effect. In other words, the experiments show the better quality of *Rain Vehicle Color-24*.

Figure 11 shows the visual effects of the test images from *Rain Vehicle color-24* and the testing models *PReNet1* and *PReNet2*. Figure 11 shows that they are competitive enough with *PReNet1*, since the distributions of the training and test sets are the same, and *Rain Vehicle Color-24* has more varying rain patterns. The results are given in Table 1.

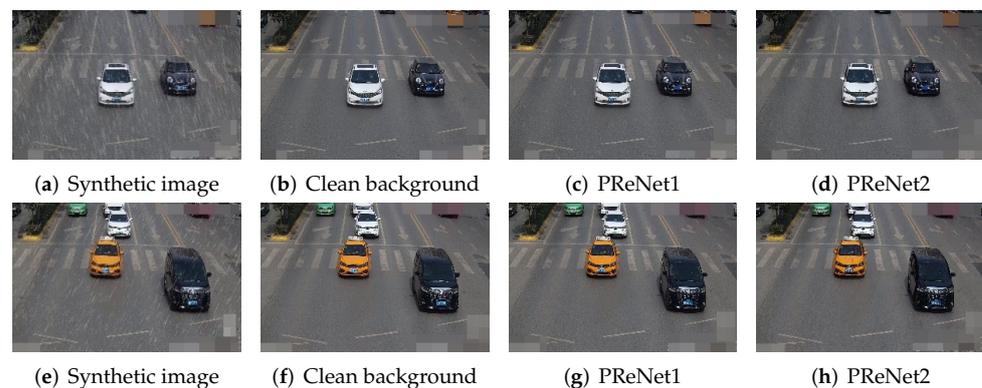


Figure 11. Test results on synthetic rain images from *Rain Vehicle Color-24*. *PReNet1* and *PReNet2* represent the different models, which are trained on the training subsets of *Rain100L* and *Rain Vehicle Color-24*, respectively.

Table 1. Comparing PSNR, SSIM of the PReNet1 and the PReNet2 on Rain100L, Rain Vehicle Color-24, respectively.

| Metrics \ Datasets | Models | | | |
|-----------------------|--------|-------|---------|---------|
| | | | PReNet1 | PReNet2 |
| Rain100L | PSNR | SSIM | PSNR | SSIM |
| | 32.67 | 0.965 | 32.44 | 0.945 |
| Rain Vehicle Color-24 | PSNR | SSIM | PSNR | SSIM |
| | 31.62 | 0.955 | 33.51 | 0.973 |

4.2.3. Comparison of Real Rain Images

To compare the generalization of the rain-removal methods, the PReNet models, which are trained on Rain Vehicle Color-24 and Rain100L, are tested on the real-rain images from the real-world rain image dataset. Figures 12 and 13 show the two rain-removal models do not work; however, PReNet2 is still better than PReNet1.

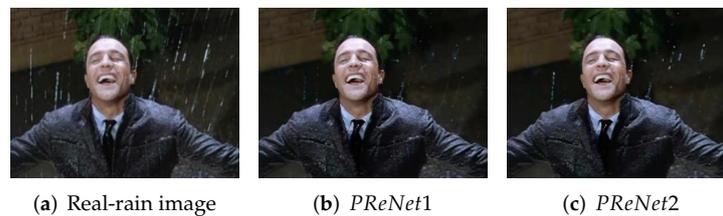


Figure 12. Test result on real-rain image from Real Data. PReNet1 and PReNet2 represent the different models trained on the Rain100L and the Rain Vehicle Color-24 training subsets, respectively.



Figure 13. Test results on two real-rain images containing vehicles from RIS. PReNet1 and PReNet2 represent the different models trained on Rain100L and Rain Vehicle Color-24, respectively.

4.2.4. Comparison of Recognition Effects of Low- and High-Level Joint Tasks

In this section, we divided Vehicle Color-24 into training, verification, and test sets at a ratio of 8:1:1. We fixed the confidence threshold at 0.5. We calculated the results using the relevant confidence code under Python.

To investigate the detection results of the vehicle-color fine recognition of Faster RCNN, PReNet1+Faster RCNN, and PReNet2+Faster RCNN [38], we first used Faster RCNN for target detection without deraining preprocessing; then, we used Faster RCNN for target detection after deraining by PReNet1 or PReNet2. Figure 14 shows that Faster RCNN recognizes the vehicle color on rain images with lower confidence; meanwhile, Faster RCNN recognizes the vehicle color on derained rain images with higher confidence. Furthermore,

PReNet2+F R CNN performs better than *PReNet1+F R CNN*, since the former is trained on *Rain Vehicle Color-24* with more various rain streaks and a much bigger data size.

From Figure 14, *Faster R CNN*, *PReNet1+Faster R CNN*, and *PReNet2+Faster R CNN* achieve around 70%, 70%, and 90% accuracy, respectively, for white-color-vehicle target detection. Therefore, the *Rain Vehicle Color-24* dataset we proposed in this paper provides a better guarantee for visual tasks, such as vehicle-color target detection.

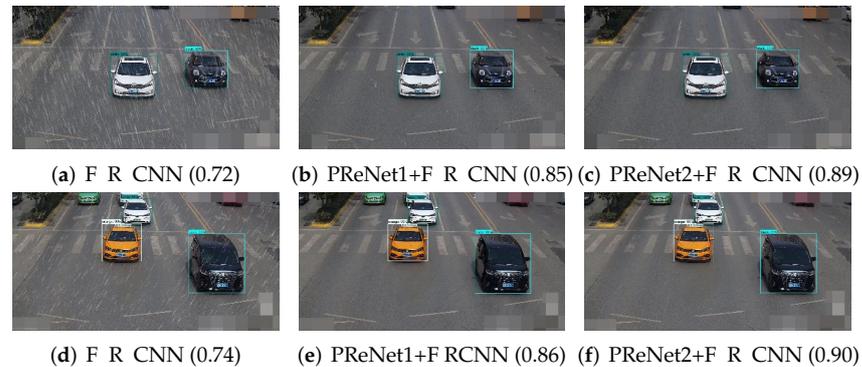


Figure 14. Objective detection test results of *Faster R CNN*, *PReNet1/2+Faster R CNN* on two synthetic rain images with vehicles from the *Rain Vehicle Color-24* dataset. Each subtitle is corresponding object detection method and the corresponding confidence value in the parentheses.

4.3. LDVS Model Tested on Rain100L and Rain Vehicle Color-24

4.3.1. LDVS Network

The network framework of *LDVS* [26] is shown in Figure 15, which is mainly composed of dilated convolution and lightweight attention modules. In the main network, there is an encoder with five feature extraction modules and a convolution operation and a decoder, where each feature extraction module concatenates a dilated convolution with a lightweight attention module *CBAM*. The rain image O is inputted into the network to extract the feature maps, and the output is the rain pattern R and the clean background image B . The clean image is equal to the input rain image O minus the feature map R . The loss function L is defined as:

$$L = SSIM(\hat{B}, B) + \alpha ||\hat{B} - B||^2. \tag{6}$$

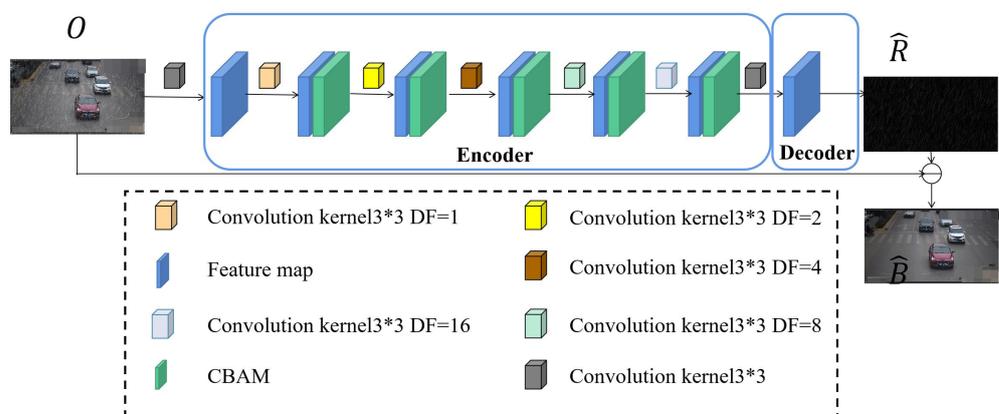


Figure 15. Illustration of *LDVS* [26].

4.3.2. Comparison on Synthetic Rain Image

Without loss of generality, we trained the *LDVS1* and *LDVS2* models obtained through the *LDVS* method on *Rain100L* and *Rain Vehicle Color-24*, respectively.

Figures 16 and 17 show the tested synthetic rain images and the comparison of rain-removal results between the *LDVS1* and *LDVS2* models.

Figure 17 shows that *LDVS* performs better in rain-removal when the test images are taken from the *Rain Vehicle Color-24* dataset. There is almost no obvious residual rain streaks on images *c* or *f*, and the background is relatively clear. Table 2 gives the testing results performed on *Rain100L*, showing that the performance of *LDVS2* is similar to *LDVS1*. When testing on the *Rain Vehicle Color-24* dataset, the performance of *LDVS2* is better than *LDVS1*.

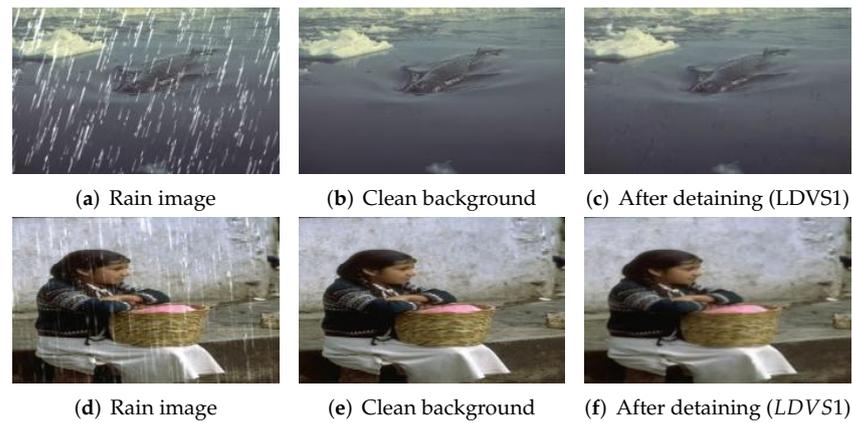


Figure 16. Test results of *LDVS1* on synthetic rain images from *Rain100L*.

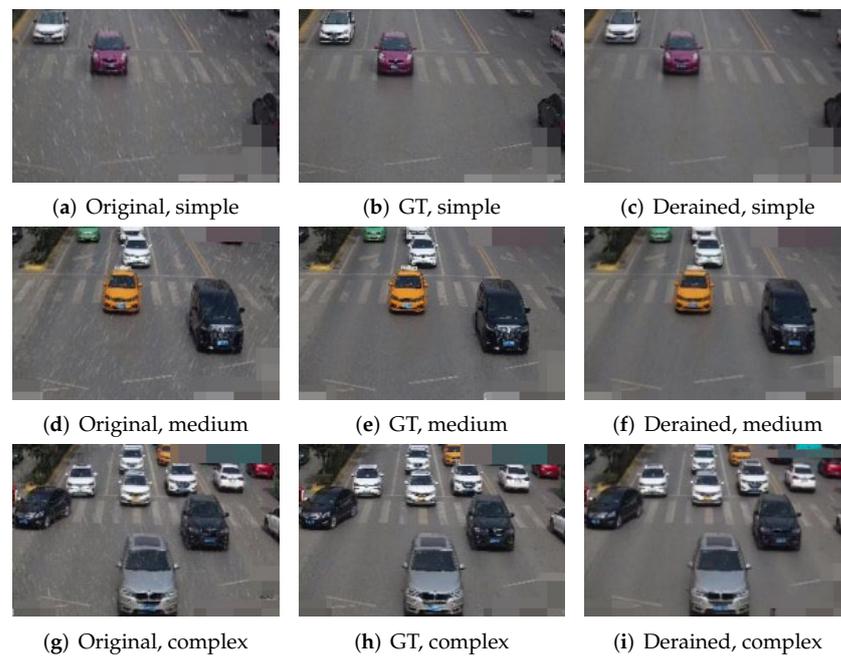


Figure 17. Test results of *LDVS2* on rain image from *Rain Vehicle Color-24* test subset. The first column is the original rainy vehicle image, the second is the clean background image (GT), and the third is the one after *LDVS2*. The image in the first row is in a simple scene, the one in the second row is in a medium complexity scene, and the one in the last row is in a complex scene.

Table 2. Comparing *PSNR* and *SSIM* of test effect on *Rain Vehicle Color-24* from the *LDVS1* and the *LDVS2*.

| Metrics | Models | <i>LDVS1</i> | | <i>LDVS2</i> | |
|------------------------------|--------|--------------|-------------|--------------|-------------|
| | | <i>PSNR</i> | <i>SSIM</i> | <i>PSNR</i> | <i>SSIM</i> |
| <i>Rain100L</i> | | <i>PSNR</i> | <i>SSIM</i> | <i>PSNR</i> | <i>SSIM</i> |
| | | 33.56 | 0.959 | 33.12 | 0.960 |
| <i>Rain Vehicle Color-24</i> | | <i>PSNR</i> | <i>SSIM</i> | <i>PSNR</i> | <i>SSIM</i> |
| | | 31.23 | 0.951 | 34.34 | 0.960 |

4.3.3. Comparison of Real Rain Images

To further test the generalization of *LDVS*, the *LDVS1* model pre-trained on the *Rain 100L* dataset and the *LDVS2* model pre-trained on the *Rain Vehicle Color-24* dataset were used to test rain removal on real-data images, and the results are shown in Figure 18. The qualitative results show that the *LDVS2* trained on *Rain Vehicle Color-24* generalization performs better than the *LDVS1* trained on *Rain100L*.

**Figure 18.** Test results of *LDVS* on real images from *Real Data*. The first column is the original real rainy image, the second is after *LDVS1*, and the third is after *LDVS2*.

Figure 18 shows that *LDVS1* or *LDVS2* works well for the rain removal effect of the synthetic rain images. On the other hand, when working in the real world, the effect is less ideal, and there are a large number of rain streak residues.

4.4. Object Detection Models Trained by *Vehicle Color-24* and *Rain Vehicle Color-24*

Four models, *SSD1*, *Faster R CNN1*, *SSD2*, and *Faster R CNN2* are trained on *Vehicle Color-24* and *Rain Vehicle Color-24*, and all models are then tested on the *Vehicle Color-24* and *Rain Vehicle Color-24* test sets. The test results are shown in Figure 19. *SSD1* and *Faster R CNN1* hardly recognize the color of vehicles on rainy days, while *SSD2* and *Faster R CNN2* can recognize any vehicle color in the rain images. For example, in Figure 19e, white vehicle detection achieves 94% accuracy after *SSD2*, and in Figure 19f, white vehicle detection achieves 100% accuracy after *Faster R CNN2*. Testing on the *Vehicle Color-24* test subset, the results of the models are almost identical.

Table 3 shows the average accuracy of each category of the object detection algorithms when they test on the *Rain Vehicle Color-24* dataset and are trained on *Vehicle Color-24*. Table 4 shows the average accuracy of each category of object detection algorithms when they test on the *Rain Vehicle Color-24* test set and are trained on *Rain Vehicle Color-24*. It is noted that almost all color is classified more accurately by the model trained on the *Rain Vehicle Color-24*.

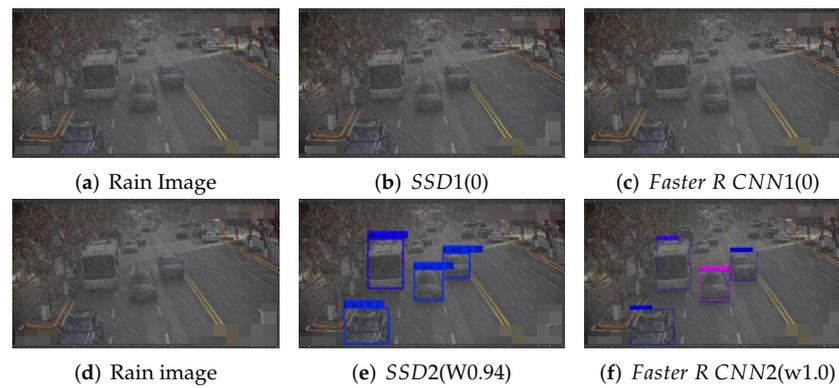


Figure 19. Object detection test results on rain images from *Rain Vehicle Color-24*. (a,d) Rain image; (b,c) object detection test result with SSD1 and *Faster R CNN1*, which are trained on *Vehicle Color-24*; (e,f) object detection test result with SSD2 and *Faster R CNN2*, which are trained on the *Rain Vehicle Color-24*.

Table 3. Comparison of the average accuracy of each category of object detection algorithms on *Vehicle Color-24* and *Rain Vehicle Color-24* test sets. All object detection algorithms are trained on *Vehicle Color-24*.

| Category | SMNN-MSFF1 | | Faster R CNN1 | | SSD1 | |
|---------------|------------|--------|---------------|--------|--------|--------|
| | VC-24 | RVC-24 | VC-24 | RVC-24 | VC-24 | RVC-24 |
| White | 0.98 | 0.64 | 0.84 | 0.80 | 0.96 | 0.74 |
| Black | 0.97 | 0.52 | 0.82 | 0.31 | 0.95 | 0.38 |
| Orange | 0.98 | 0.85 | 0.81 | 0.71 | 0.96 | 0.81 |
| Silver gray | 0.96 | 0.30 | 0.77 | 0.44 | 0.91 | 0.86 |
| Grass green | 0.98 | 0.82 | 0.70 | 0.61 | 0.96 | 0.96 |
| Dark gray | 0.94 | 0.30 | 0.66 | 0.17 | 0.84 | 0.29 |
| Dark red | 0.98 | 0.63 | 0.78 | 0.24 | 0.93 | 0.44 |
| Gray | 0.89 | 0.06 | 0.18 | 0.13 | 0.54 | 0.13 |
| Red | 0.96 | 0.65 | 0.60 | 0.20 | 0.88 | 0.41 |
| Cyan | 0.97 | 0.82 | 0.75 | 0.33 | 0.92 | 0.46 |
| Champagne | 0.97 | 0.17 | 0.63 | 0.29 | 0.81 | 0.25 |
| Dark blue | 0.96 | 0.39 | 0.66 | 0.12 | 0.86 | 0.36 |
| Blue | 0.97 | 0.59 | 0.73 | 0.10 | 0.87 | 0.69 |
| Dark brown | 0.97 | 0.09 | 0.45 | 0.02 | 0.71 | 0.11 |
| Brown | 0.88 | 0.36 | 0.30 | 0.13 | 0.58 | 0.27 |
| Yellow | 0.97 | 0.66 | 0.51 | 0.13 | 0.79 | 0.18 |
| Lemon yellow | 0.99 | 0.88 | 0.87 | 0.84 | 0.93 | 0.70 |
| Dark orange | 0.96 | 0.67 | 0.65 | 0.18 | 0.78 | 0.13 |
| Dark green | 0.94 | 0.28 | 0.38 | 0.08 | 0.58 | 0.00 |
| Red orange | 0.99 | 0.33 | 0.24 | 0.00 | 0.61 | 0.00 |
| Earthy yellow | 0.97 | 0.50 | 0.62 | 0.50 | 0.74 | 0.10 |
| Green | 0.93 | 0.13 | 0.61 | 0.33 | 0.74 | 0.00 |
| Pink | 0.94 | 0.66 | 0.50 | 0.33 | 0.71 | 0.17 |
| Purple | 0.80 | 0.00 | 0.00 | 0.00 | 0.19 | 0.00 |
| mAP | 94.96% | 47.22% | 58.59% | 29.19% | 78.13% | 30.23% |

Table 4. Comparison of the average accuracy of each category of object detection algorithms on the *Vehicle Color-24* and *Rain Vehicle Color-24* test sets. All object detection algorithms are trained on *Rain Vehicle Color-24*.

| Category | SMNN-MSFF2 | | Faster R CNN2 | | SSD2 | |
|---------------|------------|--------|---------------|--------|--------|--------|
| | VC-24 | RVC-24 | VC-24 | RVC-24 | VC-24 | RVC-24 |
| White | 0.62 | 0.60 | 0.94 | 0.96 | 0.94 | 0.95 |
| Black | 0.61 | 0.69 | 0.82 | 0.69 | 0.92 | 0.93 |
| Orange | 0.69 | 0.77 | 0.92 | 0.90 | 0.95 | 0.95 |
| Silver gray | 0.48 | 0.31 | 0.44 | 0.81 | 0.85 | 0.86 |
| Grass green | 0.60 | 0.82 | 0.88 | 0.93 | 0.94 | 0.96 |
| Dark gray | 0.47 | 0.43 | 0.57 | 0.69 | 0.71 | 0.67 |
| Dark red | 0.36 | 0.48 | 0.73 | 0.79 | 0.83 | 0.88 |
| Gray | 0.18 | 0.31 | 0.12 | 0.41 | 0.35 | 0.31 |
| Red | 0.37 | 0.44 | 0.62 | 0.56 | 0.79 | 0.76 |
| Cyan | 0.42 | 0.62 | 0.71 | 0.82 | 0.87 | 0.87 |
| Champagne | 0.33 | 0.28 | 0.46 | 0.74 | 0.65 | 0.73 |
| Dark blue | 0.60 | 0.52 | 0.66 | 0.79 | 0.78 | 0.75 |
| Blue | 0.29 | 0.56 | 0.44 | 0.69 | 0.87 | 0.69 |
| Dark brown | 0.38 | 0.35 | 0.45 | 0.18 | 0.60 | 0.47 |
| Brown | 0.47 | 0.35 | 0.34 | 0.10 | 0.33 | 0.34 |
| Yellow | 0.51 | 0.35 | 0.94 | 0.83 | 0.72 | 0.92 |
| Lemon yellow | 0.32 | 0.57 | 0.95 | 0.99 | 1.00 | 0.75 |
| Dark orange | 0.41 | 0.32 | 0.52 | 0.28 | 0.11 | 0.47 |
| Dark green | 0.62 | 0.59 | 0.10 | 0.18 | 0.36 | 0.34 |
| Red orange | 0.52 | 0.38 | 0.66 | 0.07 | 0.29 | 0.52 |
| Earthy yellow | 1.00 | 0.68 | 0.23 | 0.45 | 0.78 | 0.28 |
| Green | 0.59 | 0.18 | 0.47 | 0.85 | 0.55 | 0.97 |
| Pink | 0.03 | 0.84 | 0.02 | 0.54 | 1.00 | 0.52 |
| Purple | 0.99 | 0.22 | 0.07 | 0.03 | 0.00 | 0.06 |
| mAP | 49.14% | 48.58% | 55.13% | 60.65% | 70.84% | 66.33% |

5. Conclusions

In this paper, the *Rain Vehicle Color-24* dataset is constructed by rendering rain images based on *PS* technology and the *SyRaGAN* algorithm. The dataset has a total of 40,300 rain images, including 125 rain patterns. The aim of constructing *Rain Vehicle Color-24* is to train data-driven deep learning neural networks for specific vehicle color recognition tasks. Specifically, *Rain Vehicle Color-24* consists of two subsets: one is 300 rain images rendered by Photoshop from the *Rain Vehicle Color-24* database, and the other is 40,000 rain vehicle images rendered by the *SyRaGAN* network from another 8000 vehicle images in *Vehicle Color-24*. Extensive experiments show that, when *PReNet* and *LDVS* are trained on the new dataset, *Rain Vehicle Color-24*, both deraining task and subsequent target recognition algorithms after deraining are improved. More specifically, when the model is designed for the task of fine recognition of vehicle color, the corresponding recognition accuracy is improved for good or rainy weather conditions, after the model is fine-tuned on *Rain Vehicle Color-24*,

For future work, we will study low- and high-level joint tasks, based on the above work. We will focus on vehicle object detection and recognition in various adverse conditions such as bad weather. We will also consider fusing fuzzy sets, rough sets, overlap functions (see [39–42]) to expand the method of this paper. These studies will be critically beneficial in fields such as criminal investigation or traffic management law enforcement.

Author Contributions: Writing—original draft preparation, M.H.; experiments in Section 4.3, J.Y.; experiments in Section 4.2, C.W.; experiments in Section 4.4, Y.W.; review, J.F.; writing—review and editing, B.J. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the National Natural Science Foundation of China (No. 62071378), the Shaanxi Province International Science and Technology Cooperation Program (No. 2022KW-04), and the Xi'an Science and Technology Plan Project (No. 21XJZZ0072).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data that support the findings of this study are openly available at [humingdi2005@github.com].

Conflicts of Interest: The authors declare that they have no conflict of interest.

References

1. Chen, P.; Bai, X.; Liu, W. Vehicle color recognition on urban road by feature context. *IEEE Trans. Intell. Transp. Syst.* **2014**, *15*, 2340–2346. [CrossRef]
2. Tariq, A.; Khan, M.Z.; Khan, M.U.G. Real Time Vehicle Detection and Colour Recognition using tuned Features of Faster-RCNN. In Proceedings of the 2021 1st International Conference on Artificial Intelligence and Data Analytics (CAIDA), Riyadh, Saudi Arabia, 6–7 April 2021; pp. 262–267.
3. Jeong, Y.; Park, K.H.; Park, D. Homogeneity patch search method for voting-based efficient vehicle color classification using front-of-vehicle image. *Multimed. Tools Appl.* **2019**, *78*, 28633–28648. [CrossRef]
4. Hu, M.; Bai, L.; Li, Y.; Zhao, S.R.; Chen, E.H. Vehicle 24-Color Long Tail Recognition Based on Smooth Modulation Neural Network with Multi-layer Feature Representation. *arXiv* **2021**, arXiv:2107.09944.
5. Hu, M.; Wu, Y.; Song, Y.; Yang, J.; Zhang, R.; Wang, H.; Meng, D. The integrated evaluation and review of single image rain removal based. *J. Image Graph.* **2022**, *10*, 11834.
6. Xu, C.D.; Zhao, X.R.; Jin, X.; Wei, X.S. Exploring categorical regularization for domain adaptive object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 11724–11733.
7. Vs, V.; Gupta, V.; Oza, P.; Sindagi, V.A.; Patel, V.M. Mega-cda: Memory guided attention for category-aware unsupervised domain adaptive object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 4516–4526.
8. Xu, M.; Wang, H.; Ni, B.; Tian, Q.; Zhang, W. Cross-domain detection via graph-induced prototype alignment. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 12355–12364.
9. Zhang, Y.; Wang, Z.; Mao, Y. Rpn prototype alignment for domain adaptive object detector. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 12425–12434.
10. Li, W.; Liu, X.; Yuan, Y. SIGMA: Semantic-complete Graph Matching for Domain Adaptive Object Detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 19–23 June 2022; pp. 5291–5300.
11. Shan, Y.; Lu, W.F.; Chew, C.M. Pixel and feature level based domain adaptation for object detection in autonomous driving. *Neurocomputing* **2019**, *367*, 31–38.
12. Tilakaratna, D.S.; Watchareeruetai, U.; Siddhichai, S.; Natcharapinchai, N. Image analysis algorithms for vehicle color recognition. In Proceedings of the 2017 International Electrical Engineering Congress (iEECON), Pattaya, Thailand, 8–10 March 2017; pp. 1–4.
13. Kim, T.; Jeong, M.; Kim, S.; Choi, S.; Kim, C. Diversify and match: A domain adaptive representation learning paradigm for object detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12456–12465.
14. Dong, H.; Pan, J.; Xiang, L.; Hu, Z.; Zhang, X.; Wang, F.; Yang, M.H. Multi-scale boosted dehazing network with dense feature fusion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 2157–2167.
15. Sindagi, V.A.; Oza, P.; Yasarla, R.; Patel, V.M. Prior-based domain adaptive object detection for hazy and rainy conditions. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 763–780.
16. Wang, T.; Zhang, X.; Yuan, L.; Feng, J. Few-shot adaptive faster r-cnn. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 7173–7182.
17. Li, Y.; Tan, R.T.; Guo, X.; Lu, J.; Brown, M.S. Rain streak removal using layer priors. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2736–2744.
18. Yang, W.; Tan, R.T.; Feng, J.; Liu, J.; Guo, Z.; Yan, S. Deep joint rain detection and removal from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 1357–1366.
19. Hu, X.; Fu, C.W.; Zhu, L.; Heng, P.A. Depth-attentional features for single-image rain removal. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 8022–8031.
20. Tremblay, M.; Halder, S.S.; De Charette, R.; Lalonde, J.F. Rain rendering for evaluating and improving robustness to bad weather. *Int. J. Comput. Vis.* **2021**, *129*, 341–360.
21. Wei, Y.; Zhang, Z.; Wang, Y.; Xu, M.; Yang, Y.; Yan, S.; Wang, M. Deraincyclegan: Rain attentive cyclegan for single image deraining and rainmaking. *IEEE Trans. Image Process.* **2021**, *30*, 4788–4801.

22. Wang, H.; Yue, Z.; Xie, Q.; Zhao, Q.; Zheng, Y.; Meng, D. From rain generation to rain removal. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 14791–14801.
23. Wang, T.; Yang, X.; Xu, K.; Chen, S.; Zhang, Q.; Lau, R.W. Spatial attentive single-image deraining with a high quality real rain dataset. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 12270–12279.
24. Choi, J.; Kim, D.H.; Lee, S.; Lee, S.H.; Song, B.C. Synthesized rain images for deraining algorithms. *Neurocomputing* **2022**, *492*, 421–439.
25. Ren, D.; Zuo, W.; Hu, Q.; Zhu, P.; Meng, D. Progressive image deraining networks: A better and simpler baseline. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3937–3946.
26. Hu, M.; Yang, J.; Ling, N.; Liu, Y.; Fan, J. Lightweight single image deraining algorithm incorporating visual saliency. *IET Image Process.* **2022**, *16*, 3190–3200.
27. Li, S.; Ren, W.; Zhang, J.; Yu, J.; Guo, X. Single image rain removal via a deep decomposition–composition network. *Comput. Vis. Image Underst.* **2019**, *186*, 48–57. [[CrossRef](#)]
28. Wang, Y.; Ma, C.; Zeng, B. Multi-decoding deraining network and quasi-sparsity based training. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13375–13384.
29. Garg, K.; Nayar, S.K. Photorealistic rendering of rain streaks. *ACM Trans. Graph.* **2006**, *26*, 996–1002. [[CrossRef](#)]
30. Qian, R.; Tan, R.T.; Yang, W.; Su, J.; Liu, J. Attentive generative adversarial network for raindrop removal from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2482–2491.
31. Jin, J.; Fatemi, A.; Lira, W.M.P.; Yu, F.; Leng, B.; Ma, R.; Mahdavi-Amiri, A.; Zhang, H. Raidar: A rich annotated image dataset of rainy street scenes. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2951–2961.
32. Chen, D.Y.; Chen, C.C.; Kang, L.W. Visual depth guided color image rain streaks removal using sparse coding. *IEEE Trans. Circuits Syst. Video Technol.* **2014**, *24*, 1430–1455. [[CrossRef](#)]
33. Luo, Y.; Xu, Y.; Ji, H. Removing rain from a single image via discriminative sparse coding. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 3397–3405.
34. Fu, X.; Huang, J.; Ding, X.; Liao, Y.; Paisley, J. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Trans. Image Process.* **2017**, *26*, 2944–2956. [[CrossRef](#)] [[PubMed](#)]
35. Fu, X.; Liang, B.; Huang, Y.; Ding, X.; Paisley, J. Lightweight pyramid networks for image deraining. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 1794–1807.
36. Xue, P.; He, H. Research of Single Image Rain Removal Algorithm Based on LBP-CGAN Rain Generation Method. *Math. Probl. Eng.* **2021**, *2021*, 8865843. [[CrossRef](#)]
37. Wang, H.; Xie, Q.; Wu, Y.; Zhao, Q.; Meng, D. Single image rain streaks removal: A review and an exploration. *Int. J. Mach. Learn. Cybern.* **2020**, *11*, 853–872.
38. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [[CrossRef](#)]
39. Zhang, X.; Wang, J.; Zhan, J.; and Dai, J. Fuzzy measures and Choquet integrals based on fuzzy covering rough sets. *IEEE Trans. Fuzzy Syst.* **2021**, *30*, 2360–2374. [[CrossRef](#)]
40. Sheng, N.; Zhang, X. Regular partial residuated lattices and their filters. *Mathematics* **2022**, *10*, 2429. [[CrossRef](#)]
41. Wang, J.; Zhang, X. A novel multi-criteria decision-making method based on rough sets and fuzzy measures. *Axioms* **2022**, *11*, 275. [[CrossRef](#)]
42. Liang, R.; Zhang, X. Interval-valued pseudo overlap functions and application. *Axioms* **2022**, *11*, 216. [[CrossRef](#)]