Image De-raining via Continual Learning

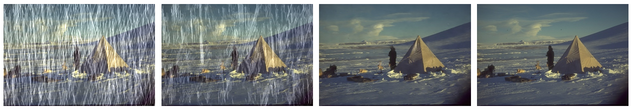
Man Zhou[1†], Jie Xiao[1†], Yifan Chang[1], Xueyang Fu[1*], Aiping Liu[1], Jinshan Pan[2], Zheng-Jun Zha[1]
[1]University of Science and Technology of China, China
[2]Nanjing University of Science and Technology, China
{manman,ustchbxj,cyflz}@mail.ustc.edu.cn, {xyfu,aipingl,zhazj}@ustc.edu.cn, sdluran@gmail.com

Abstract

While deep convolutional neural networks (CNNs) have achieved great success on image de-raining task, most existing methods can only learn fixed mapping rules between paired rainy/clean images on a single dataset. This limits their applications in practical situations with multiple and incremental datasets where the mapping rules may change for different types of rain streaks. However, the catastrophic forgetting of traditional deep CNN model challenges the design of generalized framework for multiple and incremental datasets. A strategy of sharing the network structure but independently updating and storing the network parameters on each dataset has been developed as a potential solution. Nevertheless, this strategy is not applicable to compact systems as it dramatically increases the overall training time and parameter space. To alleviate such limitation, in this study, we propose a parameter importance guided weights modification approach, named PIGWM. Specifically, with new dataset (e.g. new rain dataset), the well-trained network weights are updated according to their importance evaluated on previous training dataset. With extensive experimental validation, we demonstrate that a single network with a single parameter set of our proposed method can process multiple rain datasets almost without performance degradation. The proposed model is capable of achieving superior performance on both inhomogeneous and incremental datasets, and is promising for highly compact systems to gradually learn myriad regularities of the different types of rain streaks. The results indicate that our proposed method has great potential for other computer vision tasks with dynamic learning environments.

1. Introduction

In recent years, remarkable progress has been achieved on single image de-raining and other low-level vision tasks due to the rapid growth of deep learning [32, 9, 5, 35, 45, 18, 50, 43, 27, 8, 44, 41, 33, 23, 13, 37, 3, 6, 30].



(a) Input (b) SI (c) PIGWM (d) Clean

Figure 1: Deraining results of PreNet [29] trained sequentially on task sequence Rain100H [40]-Rain100L [40] from test dataset of Rain100H. (a) Input image with rain streaks. (b) Result of the model trained *sequentially and independently* (SI). (c) Result of the model trained sequentially with our PIGWM. (d) The clean image. This figure presents the de-raining network sufffers from catastrophic forgetting and PIGWM is capable of maintaining the performance on previous task.

Single image rain removal aims to recover the clean image from its rain-polluted version, which is the basis for other downstream computer vision tasks, *e.g.* object detection, image classification, person identification, and more [20, 12, 4, 24, 19, 31, 34]. Despite increasing attentions and great improvements on image de-raining tasks, existing deep CNN-based rain removal methods can only learn fixed mapping rules between paired rainy/clean images on a single type of dataset due to catastrophic forgetting problem. In addition, with sequential training on multiple datasets, current deep neural network leads to almost complete forgetting of former knowledge and largely degrades the model's performance on previous tasks, after the model being trained on the new task.

Aforementioned problems limit their applications in real dynamic situations where the mapping rules do not remain the same but change according to different types of rain streaks. Although haven't explored for de-raining task, a strategy of sharing network structure but independently updating and storing network parameters on each dataset can be adopted. However, this strategy is not suitable for compact systems design, since it dramatically increases the overall training time and parameter space. This hinders the practical implementation of de-raining algorithms on edge devices, *e.g.* mobile phone with limited storage.

In this paper, we aim to solve the catastrophic forgetting

[*†] Co-first authors contributed equally, * corresponding author.

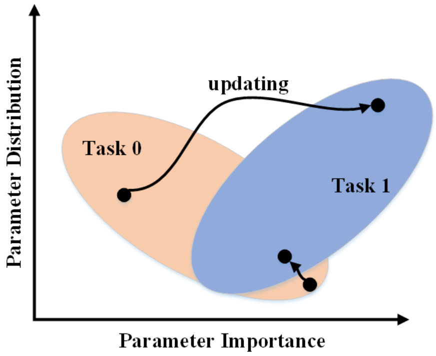


Figure 2: Schematic of our proposed parameter importance guided weights modification. The ellipse represents parameter distribution sub-space. The abscissa is the parameter importance. The arrow indicates the parameter updating process. In the top of figure, the black dot in Task 0 space evaluated as relatively unimportant is much more updated to the black dot in Task 1 space with relatively greater importance. On the contrary, in the bottom of figure, the black dot in Task 0 space evaluated as relatively more important is much less updated. Broadly speaking, the more important the weight is, the less it is updated. This scheme is capable of obtaining favorable performance on new task while maintaining that on previous task.

problem for multiple and incremental datasets that is applicable to compact systems. To this end, one dynamic learning condition is examined: sequentially training the multiple rain datasets with a single model. As shown in Figure 1, training the multiple rain datasets sequentially and independently will largely degrade the model's performance on previous task.

Specifically, we first introduce the continual learning scheme to handle different types of rain streaks using a single model. A parameter importance guided weights modification approach, named PIGWM is proposed to overcome catastrophic forgetting for image de-raining. For convenience, a dataset is considered as a Task and a list of datasets can be denoted as Task 0, Task $1\ldots$, Task $n-1$ where n is the length of the list. When training on new task (*e.g.* new rain dataset), the network weights obtained on previous rain dataset are updated depending on parameter importance evaluated on previous rain task. The more important the weight is, the less it is updated. The proposed PIGWM enables obtaining favorable performance on new task while maintaining the performance on previous task. For instance, as shown in Figure 2, if the weight obtained from Task 0 is evaluated as relatively unimportant, it is more frequently updated with increasing attention to improve the performance on Task 1. On the contrary, the important weight obtained from Task 0 is less possible to be updated in order to maintain the performance on previous task. Generally speaking, the more important the weight is, the less it is updated in subsequent training. This scheme is capable of obtaining superior performance on new dataset while maintaining that on previous one. To the best of our knowledge, it is the first attempt to solve the catastrophic forgetting problem on rain removal task.

The major contributions of this paper are as follows:

1) It is the first attempt to deal with the catastrophic forgetting problem on image rain removal. We introduce the continual learning scheme to handle different types of rain streaks with a single model.
2) A parameter importance guided weights modification approach, named PIGWM is proposed to overcome catastrophic forgetting for image de-raining. This may be easily extended to other computer vision tasks in a plug-and-play manner.
3) Extensive experiments on multiple type of rain streak benchmarks validate the superior performance of our proposed method.

2. Related Work

Image Rain Removal. Recently, CNN-based methods have achieved reliable progress in single image de-raining. Fu *et al.* [11, 10] first introduce deep learning mapping scheme to the de-raining problem. They map high-frequency part of rain image to the rain streak layer by utilizing a deep residual network. However, the method still cannot handle large and sharp rain streaks. Yang *et al.* [40] construct a joint rain detection and removal network. However, it might falsely remove vertical textures and generate underexposed illumination [41]. Further, more complicated CNN-based architectures are designed to improve the performance on rain removal, including non-local operation based encoder-decoder network [22], multi-stages network [47], conditional generative adversarial network [48], deep convolutional and recurrent neural network that removes rain streak stage by stage [25, 29] and so on. Besides, there is a tendency to integrate model-driven approaches with data-driven approaches for taking advantage of image prior and powerful feature mapping [23, 35]. However, due to current deep learning based methods mostly suffering from the catastrophic forgetting problem, existing deep CNN-based promising rain removal methods can only learn fixed mapping rules between paired rainy/clean images on a single type of dataset. When dealing with different types of rain datasets, these models cannot maintain their performance well simultaneously on multiple datasets as they do on single dataset. To this end, we introduce the continual learning scheme into image rain removal.

Continual Learning. The methods of overcoming catastrophic forgetting can be mainly divided into three categories: transfer learning approaches, rehearsal mechanisms, and parameter regularization methods [2, 15, 7, 28, 49, 21, 26, 1]. In detail, transfer learning approaches and rehearsal

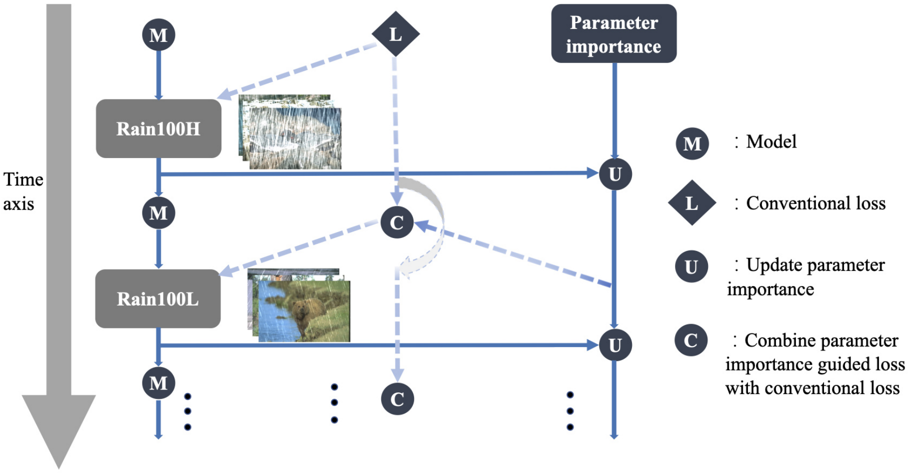

Time
axis
M
L
Parameter importance
Rain100H
U
M
C
Rain100L
U
M
C
M : Model
L : Conventional loss
U : Update parameter importance
C : Combine parameter importance guided loss with conventional loss

Figure 3: The detail illustration of our proposed parameter importance guided weight modification in different types of rain dataset(recorded as Task 0, Task 1 and more). The parameter importance module, which is updated after training the model on one task, is responsible for saving the importance of each weight for the previous tasks. When facing the new task, the network is updated under the restriction of the importance in order to obtain favorable performance on new task while maintaining that on previous tasks.

mechanisms are computationally expensive since they require to record the old tasks' targets and compute old tasks' forward pass process for each novel data sample. In contrast, parameter regularization methods are cost-effective. They only attempt to hinder forgetting by imposing constraints on the updating of the weights in CNNs by regularization such as freezing or consolidating the weights of CNNs well trained on previous tasks. The representative method of classic parameter regularization is attributed to EWC [21]. EWC quantifies how much essential each parameter is for a task with the diagonal of the Fisher information matrix [26] and protecting critical weights with an additional regularization to restrict their movement when updating for the new task. Further, memory aware synapses (MAS) [1] compute the parameter importance based on how sensitive the predicted output function is to a change in this parameter, and penalize changed important parameters. In this paper, we propose the first and second-order parameter importance to jointly estimate the status of one parameter in determining the performance of the whole model more accurately.

3. Continual Learning for Single Image Rain Removal

Recent years have gained promising progress in single image rain removal with designing complicated deep neural networks. However, existing state-of-the-art approaches almost suffer from the catastrophic forgetting problem. That is, when training on the continual learning tasks, the network always forgets the previous knowledge and results in the model's performance degrading abruptly on previous task. To solve this problem, we propose a parameter importance guided weights modification approach, named PIGWM whose detail illustration is in Figure 3 to overcome catastrophic forgetting for the image de-raining community.

3.1. Parameter Importance Calculation

Most of deep learning based de-raining methods may be capable of obtaining off-the-shell results by heuristically constructing a complicated neural network architecture in an end-to-end fashion. These methods always consider the CNN as an encapsulated end-to-end mapping module for mapping the input image to its clean version. Specifically, for the rain imaging process, it can be formulated as:

$$O = R + B, \tag{1}$$

where the O and B represent the rain-polluted image and the clean image respectively. The CNN based methods treat the rain removal as mapping the input to the output. The loss function is employed to evaluate the difference between the output and the ground truth, recorded as:

$$G = F(O), \tag{2}$$

$$\text{minimize} \quad l(B, G), \tag{3}$$

where F indicates the CNN equipped de-raining architecture, l indicates the conventional loss (*e.g.* MSE [39]) to train de-raining network.

We consider the situation where the neural network is trained on Task n and Task $n+1$ sequentially, in which the mapping rules do not remain the same. Suppose the parameter set of de-raining architecture F is denoted $\theta^n = \{\theta_1^n, \theta_2^n, \ldots, \theta_m^n\}$ when a network is trained on Task n, where m is depth of the network, and the continual two rainy image sets are denoted as X^n, X^{n+1} and their clean counterparts are remarked as Y^n, Y^{n+1} respectively. For $x^n \in X^n, y^n \in Y^n$, suppose x^n is random variable, which is independently and identically distributed to $\mathbb{P}^n$, and (x^n, y^n) is the rainy/clean image pair. If x_n is fed into network, the degradation of performance on Task n introduced by the training of network on Task $n+1$ is evaluated by:

$$\begin{aligned}\Delta F(\theta^{n+1}, \theta^n; x^n, y^n) &= \mathrm{Dist}(F(x^n; \theta^{n+1}), F(x^n; \theta^n)) \\ &\triangleq |l(F(x^n; \theta^{n+1}), y^n) - l(F(x^n; \theta^n), y^n)|,\end{aligned} \tag{4}$$

where $|\cdot|$ denotes absolute value operator, $\triangleq$ means definition, Dist measures distance between $F(x^n; \theta^{n+1})$ and $F(x^n; \theta^n))$, l represents conventional loss used by training de-raining network. For simplicity, (4) is rewritten as:

$$\Delta F(\theta^{n+1}, \theta^n; x^n, y^n) = |l(\theta^{n+1}; x^n, y^n) - l(\theta^n; x^n, y^n)|. \tag{5}$$

In the following, we will give the expression of parameter importance and the detailed process of obtaining it, in which we consider the computational complexity and storage space in actual implementation. Taking the element of parameter θ_k^n (k-th depth) for example, the change of parameter θ_k^n when model is trained on the new Task $n+1$ is denoted as $\delta\theta_k^n$ whose mathematical form is

$$\delta\theta_k^n = \theta_k^{n+1} - \theta_k^n. \tag{6}$$

To evaluate $\Delta F(\theta^{n+1}, \theta^n; x^n, y^n)$, we take the Taylor expansion of $l(\theta; x^n, y^n)$ at point θ_k^n, which is an infinite sum of terms that are expressed in the form of target function's derivatives at a single point:

$$\begin{aligned}l(\theta_k^n + \delta\theta_k^n; x^n, y^n) &= l(\theta_k^n; x^n, y^n) + \left(\nabla_{\theta_k^n} l\right)^T \cdot \delta\theta_k^n \\ &+ \frac{1}{2}(\delta\theta_k^n)^T \cdot H \cdot \delta\theta_k^n + O(\|\delta\theta_k^n\|^3),\end{aligned} \tag{7}$$

where H denotes Hessian matrix:

$$H = \nabla^2_{\theta_k^n} l(\theta_k^n; x^n). \tag{8}$$

Then maintaining the performance on previous Task n corresponds to minimize:

$$\begin{aligned}&\mathop{\mathbb{E}}_{x_n \sim \mathbb{P}^n}\left[\Delta F(\theta_k^{n+1}, \theta_k^n; x^n, y^n)\right] \\ \approx &\mathop{\mathbb{E}}_{x_n \sim \mathbb{P}^n}\left[\left|\nabla_{\theta_k^n} l^T \cdot \delta\theta_k^n + \frac{1}{2}(\delta\theta_k^n)^T \cdot H \cdot \delta\theta_k^n\right|\right].\end{aligned} \tag{9}$$

The right of (9) is denoted by $\mu(\theta_k^n, \theta_k^{n+1})$. From this motivation, when training de-raining model on Task $n+1$, we add a regularization term based on conventional loss to keep the knowledge of Task n. Unfortunately, $\mu(\theta_k^n, \theta_k^{n+1})$ does not meet practical requirements for the term with the reason that in order to calculate the expectation, it is necessary to continuously iterate over the training data of the previous Task n for single forward propagation on Task $n+1$, which will undoubtedly greatly increase computational complexity on Task $n+1$. What's worse, the storage requirement brought by Hessian matrices is very tremendous. Specifically speaking, we consider the weight parameter of one convolutional layer, whose dimension is represented by $o \times i \times k \times k$ where o, i, k denotes output channel number, input channel number, kernel size respectively. The need of storage of its Hessian matrix is $(o \times i \times k \times k) \times (o \times i \times k \times k)$. With the consideration that image de-raining network is generally of the architecture of deep stack of convolutional layers, Hessian matrices of the whole neural network will introduce intolerable storage consumption.

In this paper, we introduce a resource-friendly regularization term $g(\theta_k^n, \theta_k^{n+1})$ from which we derive first and second-order parameter importance. $g(\theta_k^n, \theta_k^{n+1})$ is computationally simpler than $\mu(\theta_k^n, \theta_k^{n+1})$ and needs reasonably additional storage space but still theoretically and practically effective. Specifically, $g(\theta_k^n, \theta_k^{n+1})$ relieves the entanglement of the previous Task n in the regularization term on Task $n+1$. When calculating the loss on Task $n+1$, there is no need to iterate over the training data of the previous task, which greatly reduces the computational burden. The following is the process by which we obtain the modified regularization term. At the same time, this process theoretically proves the effectiveness of the regularization term, because it provides higher bound than $\mu(\theta_k^n, \theta_k^{n+1})$ so that if the value of $g(\theta_k^n, \theta_k^{n+1})$ is small enough, that of $\mu(\theta_k^n, \theta_k^{n+1})$ is smaller automatically which ensures the performance on the previous Task n when the model is trained on Task $n+1$.

$$\begin{aligned}&\mu(\theta_k^n, \theta_k^{n+1}) = \mathop{\mathbb{E}}_{x_n \sim \mathbb{P}^n}\left[\left|\nabla_{\theta_k^n} l^T \cdot \delta\theta_k^n + \frac{1}{2}(\delta\theta_k^n)^T \cdot H \cdot \delta\theta_k^n\right|\right] \\ &\leq \mathop{\mathbb{E}}_{x_n \sim \mathbb{P}^n}\left[\left|\nabla_{\theta_k^n} l^T \cdot \delta\theta_k^n\right| + \frac{1}{2}\left|(\delta\theta_k^n)^T \cdot H \cdot \delta\theta_k^n\right|\right] \\ &\leq \mathop{\mathbb{E}}_{x_n \sim \mathbb{P}^n}\left[\left|\nabla_{\theta_k^n} l\right|^T \cdot |\delta\theta_k^n| + \frac{1}{2}|\delta\theta_k^n|^T \cdot |H| \cdot |\delta\theta_k^n|\right] \\ &= \mathop{\mathbb{E}}_{x_n \sim \mathbb{P}^n}\left[\left|\nabla_{\theta_k^n} l\right|^T\right] \cdot |\delta\theta_k^n| + \frac{1}{2}|\delta\theta_k^n|^T \cdot \mathbb{E}\left[|H|\right] \cdot |\delta\theta_k^n|,\end{aligned} \tag{10}$$

where $|\cdot|$ denotes element-wise absolute value. It is apparent that the last term of (10) greatly relieves the entanglement of the previous Task n in the regularization term of Task $n+1$ greatly accelerating the calculation of total loss. Then the problem we have to tackle with is the huge storage requirement of Hessian matrices. Motivated by Gauss-Newton method [38], when conventional loss is SSE (Sum

Square Error), we take an approximation $H \approx 2J^TJ$ for Hessian matrix, where J is Jacobian matrix. Meanwhile, we take the approximation that:

$$\mathop{\mathbb{E}}_{x_n\sim\mathbb{P}^n}\left[\left|J^TJ\right|\right] \approx \mathop{\mathbb{E}}_{x_n\sim\mathbb{P}^n}\left[|J|\right]^T \mathop{\mathbb{E}}_{x_n\sim\mathbb{P}^n}\left[|J|\right]. \tag{11}$$

In implementation, J, whose storage requirement is $(h \times w \times c) \times (o \times i \times k \times k)$ for a weight parameter of one convolutional layer , where h, w, c denotes the height, width and channel number of output image of network, is further reduced to $\nabla_{\theta_k^n} l$ whose storage requirement is $1 \times (o \times i \times k \times k)$ for the same parameter. So, instead of calculating and saving $\mathbb{E}_{x_n\sim\mathbb{P}^n}\left[|H|\right]$, We need only to calculate and store $\mathbb{E}_{x_n\sim\mathbb{P}^n}\left[\left|\nabla_{\theta_k^n} l\right|\right]$, which provides us a storage-saving approach to get approximate Hessian matrices. So, we get the ultimate regularization term $g(\theta_k^n, \theta_k^{n+1})$ whose form is:

$$\begin{aligned} g(\theta_k^n, \theta_k^{n+1}) &= \mathop{\mathbb{E}}_{x_n\sim\mathbb{P}^n}\left[\left|\nabla_{\theta_k^n} l\right|\right]^T |\delta\theta_k^n| \\ &\quad + |\delta\theta_k^n|^T \mathop{\mathbb{E}}_{x_n\sim\mathbb{P}^n}\left[|J|\right]^T \mathop{\mathbb{E}}_{x_n\sim\mathbb{P}^n}\left[|J|\right] |\delta\theta_k^n|. \end{aligned} \tag{12}$$

From (12), first and second-order parameter importance denoted as $I_1(\theta_k^n), I_2(\theta_k^n)$ are defined by

$$I_1(\theta_k^n) = \mathop{\mathbb{E}}_{x_n\sim\mathbb{P}^n}\left[\left|\nabla_{\theta_k^n} l\right|\right], \tag{13}$$

$$I_2(\theta_k^n) = \mathop{\mathbb{E}}_{x_n\sim\mathbb{P}^n}\left[|J|\right]^T \mathop{\mathbb{E}}_{x_n\sim\mathbb{P}^n}\left[|J|\right]. \tag{14}$$

In summary, the total loss on Task $n+1$ is a composite loss, which is of the form:

$$\begin{aligned} l' &= l + \frac{\lambda}{2} \sum_{(\theta_k^n, \theta_k^{n+1})} g(\theta_k^n, \theta_k^{n+1}) \\ &= l + \frac{\lambda}{2} \sum_{(\theta_k^n, \theta_k^{n+1})} \left[I_1(\theta_k^n)^T |\delta\theta_k^n| + |\delta\theta_k^n|^k I_2(\theta_k^n) |\delta\theta_k^n|\right]. \end{aligned} \tag{15}$$

3.2. Parameter Importance Guided Modification

To balance the continual tasks, the latter is trained under the constrict of maintaining the performance on previous task. The proposed PIGWM is capable of obtaining favorable performance on new task while maintaining that on previous task. The pseudo of our proposed parameter importance guided continual image rain removal is illustrated in Algorithm 1.

4. Experiments

To verify the effectiveness of proposed continual learning scheme, we integrate the proposed parameter importance guided modification algorithm with six state-of-the-art single image rain removal methods: ID-CGAN [48], PreNet [29], PRN [29], NLEDN [22], REHEN [42] as well as SASI [36]. With the consideration that it is very time-consuming [22] to train NLEDN, we abandon non-local operations in NLEDN resulting in the architecture mainly consisting of dense blocks [16] and skip connections [14].

Algorithm 1 Continual Learning for Image De-raining with PIGWM

```
Input: continual learning task $T_n, T_{n+1}$; conventional
  train loss $l$
Output: single tight model for continual task.
  for Task $T_n$ do
      Training Task $T_n$ using conventional loss $l$
      if last training epoch then
          for Parameter $\theta_k^n$ do
              $I(\theta_k^n) \leftarrow \overline{\left|\nabla_{\theta_k^n} l\right|}$
              $\theta_k^n \leftarrow \theta_k^n$
          end for
      end if
  end for
  for Task $T_{n+1}$ do
      get conventional loss $l_s$ through forward propagation
  using $l$
      for Parameter $\theta_k^{n+1}$ do
          $\delta\theta_k^n \leftarrow \theta_k^{n+1} - \theta_k^n$
          $J \leftarrow I(\theta_k^n)$
          $l_s \leftarrow l_s + \frac{\lambda}{2}(I(\theta_k^n)^T|\delta\theta_k^n| + |\delta\theta_k^n|^T J^T J|\delta\theta_k^n|)$
      end for
      Back propagation using the composite loss $l$ and up-
  date network
  end for
  Return continually trained model.
```

4.1. Dataset and Performance Metrics

We evaluate our proposed continual learning scheme on three widely-used rain removal datasets, including Rain100H [40], Rain100L [40] and Rain800 [48] in this work. In detail, PreNet, PRN, NLEDN, REHEN and SASI are trained on Rain100H (Task 0) and Rain100L (Task 1) sequentially. In addition to continual task sequence Rain100H-Rain100L, we further experiment on continual task sequences Rain800-Rain100L, Rain800-Rain100H using ID-CGAN [48], which first introduces Rain800 dataset. Both Rain100H and Rain100L consist of 1800 rainy/clean image pairs for training and 100 pairs for testing while Rain800 possesses 600 training samples and 200 testing images. Moreover, peak-signal-to-noise ratio (PSNR) [17] and structure similarity (SSIM) [46] are employed for evaluating the model performance.

Model	Methods	Rain100H		Rain100L		Promotion on Rain100H	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
NLEDN [22]	Baseline	15.84	0.532	34.53	0.958	**5.12**	**0.204**
	PIGWM	20.96	0.736	34.93	0.961		
	Reference	27.11	0.835	35.26	0.963		
PreNet [29]	Baseline	18.97	0.639	38.29	0.981	**9.11**	**0.251**
	PIGWM	28.08	0.890	36.95	0.975		
	Reference	29.46	0.899	37.48	0.979		
PRN [29]	Baseline	18.29	0.619	37.34	0.978	**9.59**	**0.261**
	PIGWM	27.88	0.880	35.64	0.967		
	Reference	28.07	0.884	36.99	0.977		
SASI [36]	Baseline	19.42	0.673	37.40	0.980	**10.34**	**0.206**
	PIGWM	29.76	0.879	36.73	0.968		
	Reference	30.33	0.909	38.80	0.984		
REHEN [42]	Baseline	14.31	0.423	37.34	0.974	**12.45**	**0.433**
	PIGWM	26.76	0.856	35.68	0.961		
	Reference	27.97	0.864	37.41	0.980		

Table 1: Comparison of quantitative results in terms of PSNR and SSIM. The models are trained sequentially on task sequence Rain100H-Rain100L using schemes of baseline and PIGWM respectively. Reference rows refer to the results of the model trained on each dataset individually from scratch. The results shown in this table are the performance of the ultimate model on test datasets of all trained tasks. It clearly indicates that our proposed PIGWM can greatly mitigate catastrophic forgetting.

Model	Methods	Rain800		Rain100L		Promotion on Rain800	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ID-cgan [48]	Baseline	20.57	0.645	25.56	0.876	**2.79**	**0.177**
	PIGWM	23.36	0.822	24.13	0.856		
	Reference	24.34	0.843	25.88	0.891		

Table 2: Comparison of quantitative results in terms of PSNR and SSIM. The model is trained sequentially on task sequence Rain800-Rain100L using schemes of baseline and PIGWM respectively.

Model	Methods	Rain800		Rain100H		Promotion on Rain800	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
ID-cgan [48]	Baseline	19.89	0.641	13.25	0.598	**3.19**	**0.174**
	PIGWM	23.08	0.815	11.16	0.532		
	Reference	24.34	0.843	14.16	0.607		

Table 3: Comparison of quantitative results in terms of PSNR and SSIM. The model is trained sequentially on task sequence Rain800-Rain100H using schemes of baseline and PIGWM respectively.

4.2. Training Details

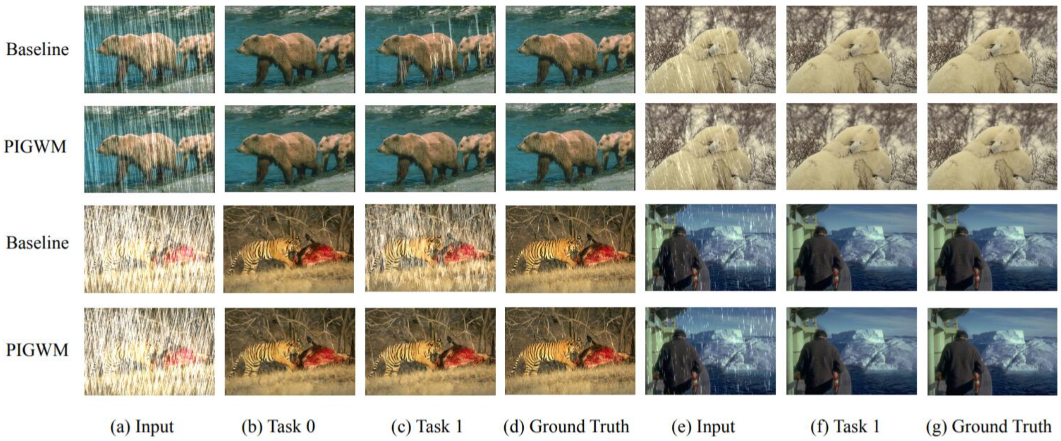
For fair comparison, all the parameters setting and training techniques keep consistent with experiments in original papers. The coefficient λ varying with model is key to keep trade-off between learning new task and ensuring performance on previous task, which will be verified in ablation studies. The important thing is significant improvements of performance on previous task can be achieved with only slight sacrifice of that on new task. Further, all the experiments are implemented on NVIDIA GTX 1080Ti GPUs.

4.3. Results on Benchmark Datasets

To revisit the catastrophic forgetting problem on image de-raining and testify the effectiveness of proposed continual learning algorithm, we conduct both qualitative and quantitative experiments on above datasets and performance metrics.

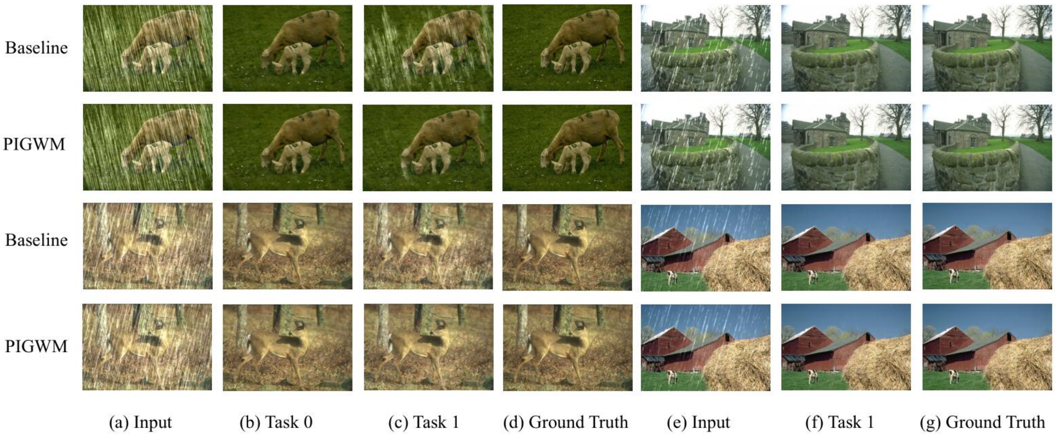
Baseline Setup. The baseline is organized as sequentially and independently feeding rain datasets into a model for training. In the setting of baseline, due to the catastrophic forgetting, the weights well-trained on the previous dataset are covered and updated by the new rain dataset neglecting the previous dataset. After training on the new rain dataset, we evaluate the ultimate model on all the test datasets.

Quantitative Comparison. Tables 1, 2 and 3 report the comprehensive comparison between the baselines and the parameter importance equipped versions which indicate that our method can greatly mitigate catastrophic forgetting



(a) Input (b) Task 0 (c) Task 1 (d) Ground Truth (e) Input (f) Task 1 (g) Ground Truth

Figure 4: Visual comparison of rain-streaks removal results generated from the continual learning process using model PreNet. (a) Input: rainy image from Rain100H; (b) Task 0: train and test on Rain100H; (c) Task 1: train model (b) on Rain100L and test on Rain100H; (d) Ground Truth: clean image of (a); (e) Input: rainy image from Rain100L; (f) Task 1: train model (b) on Rain100L and test on Rain100L; (g) Ground Truth: clean image of (e).

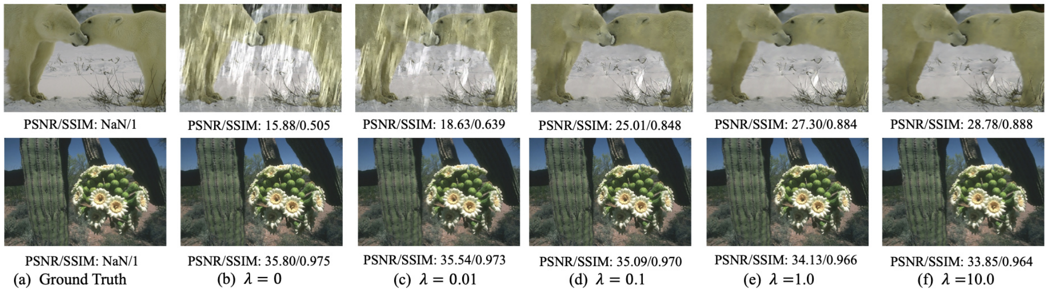


(a) Input (b) Task 0 (c) Task 1 (d) Ground Truth (e) Input (f) Task 1 (g) Ground Truth

Figure 5: Visual comparison of rain-streaks removal results generated from the continual learning process using model NLEDN.

λ	0	0.01	0.1	1	10
Rain100H	18.97 / 0.639	20.68 / 0.707	25.38 / 0.847	28.29 / 0.892	29.08 / 0.896
Rain100L	38.29 / 0.981	38.02 / 0.980	37.49 / 0.977	36.83 / 0.974	36.23 / 0.972

Table 4: Quantitative results about coefficient λ to keep trade-off between learning on new task and maintaining the performance on previous task.



(a) Ground Truth (b) $\lambda = 0$ (c) $\lambda = 0.01$ (d) $\lambda = 0.1$ (e) $\lambda = 1.0$ (f) $\lambda = 10.0$

Figure 6: Visual effect about coefficient λ to keep trade-off between learning on new task and maintaining the performance on previous task. The first row is from test dataset of Rain100H, and the second one is from test dataset of Rain100L.

on all models and multiple task sequences. Reference rows refer to the results of the model trained on each dataset individually from scratch. We adopt the results that reported by the authors expect NLEDN. Since non-local operations are abandoned in NLEDN, we re-train the modified de-raining network and evaluate the well-trained model. As shown in these tables, great improvements are achieved across all the models and task sequences we have experimented with. These tables verify the effectiveness of our proposed continual learning scheme. The proposed parameter importance guided modification is capable of obtaining satisfactory results on new task while maintaining outstanding performance on the previous one.

Qualitative Comparison. To clearly illustrate the catastrophic forgetting of continual de-raining and the effectiveness of proposed scheme, we delve into the process of continual learning on the task sequence. We train the selected models sequentially with scheme of PIGWM and baseline respectively. *Each time a task is completed, we evaluate the performance of the model on the test dataset of the first task in the task sequence*. Furthermore, we also present results of the ultimate model on Rain100L to testify that our PIGWM can learn new task while keeping previous knowledge. Specifically, we take the promising PreNet (Figure 4) and NLEDN (Figure 5) for example. The first and third row represent the results of the baseline of continually learning rain datasets. The second and last row indicate results of the model equipped with the proposed PIGWM. As shown in both figures, when Task 0 is completed, both baseline and PIGWM are capable of obtaining visually pleasing images on test dataset of Rain100H. However, after Task 1, baseline suffers from the catastrophic forgetting so that obvious rain streaks can be found in its output images while high-quality de-raining results can be maintained after Task 1 by our PIGWM. It can be clearly found that our proposed algorithm is able to relieve the catastrophic forgetting problem of previous task significantly while achieving outstanding performance on new task. In addition, since the use of our method can make the deep model remember the knowledge of multiple data sets, it can help improve the generalization ability of the model when facing complex and changeable real-world rainy scenes.

4.4. Ablation Studies

In this section, we first conduct ablation studies to verify the coefficient λ of balancing the two loss terms: the regular loss for well training the new dataset and the regularization term for maintaining the performance on the previous dataset. We take the state-of-the-art PreNet for example and the detail experiments are illustrated in Table 4 and Figure 6. It can be seen clearly that the coefficient is key for overcoming catastrophic forgetting. When it is larger, the model will pay more attention on the previous task performance and make the new task training unsteadily with worse results. On the contrary, the model will forget the knowledge of previous task and be trapped into catastrophic forgetting which degrades the model performance on previous task abruptly.

4.5. Extension to Multiple Datasets

Based on the research of 2 tasks, we can easily extend our method to multiple tasks using steps similar with Dynamic Programming. For the sequence of n tasks, the first $n-1$ tasks are regarded as a task and continue training task n, which is similar with 2 tasks. Taking 3 tasks as an example, the first 2 tasks (task$_1$ and task$_2$) is sequentially trained by using our proposed scheme. Then, this trained model covering task$_1$ and task$_2$ can be regraded as a single model (task$_{1-2}$). Later, the task$_3$ continue to be trained based on the model (task$_{1-2}$). In Table 5, we show one result on Rain100H-Rain100L-Rain1400 based on PreNet to validate the effectivess of our method.

Test set	Rain100H	Rain100L	Rain1400
Baseline	15.31 / 0.424	28.88 / 0.892	31.90 / 0.927
PIGWM	28.18 / 0.891	36.85 / 0.975	28.06 / 0.864
Reference	29.46 / 0.899	37.48 / 0.979	32.60 / 0.946

Table 5: PSNR/SSIM results of PreNet trained on the task list Rain100H-Rain100L-Rain1400.

5. Conclusion

In this work, we first pay attention to the catastrophic forgetting problem of image rain removal and attempt to introduce the continual learning scheme to handle different types of rain streaks in a single model. Specifically, we propose a parameter importance guided weights modification approach, named PIGWM to overcome catastrophic forgetting for the image de-raining community. This scheme is capable of obtaining satisfactory performance while maintaining that on the previous rain dataset. Extensive experiments on multiple type of rain streak benchmarks demonstrate the superior performance of our proposed scheme of overcoming catastrophic forgetting. Moreover, this may be easily extended to other computer vision tasks in a plug-and-play manner.

Acknowledgement

This work was supported by the National Key R&D Program of China under Grant 2020AAA0105702, the National Natural Science Foundation of China (NSFC) under Grants U19B2038, 61620106009 and 61901433, the USTC Research Funds of the Double First-Class Initiative under Grants YD2100002003, YD2100002004.

References

[1] R. Aljundi, F. Babiloni, M. Elhoseiny, M. Rohrbach, and T. Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018. 2, 3

[2] A. Chaudhry, P. K. Dokania, T. Ajanthan, and P. H. S. Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[3] C. Chen, Z. Xiong, X. Tian, Z. Zha, and F. Wu. Camera lens super-resolution. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1652–1660, 2019. 1

[4] D. Comaniciu, V. Ramesh, and P. Meer. Kernel-based object tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(5):564–577, 2003. 1

[5] 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 (CVPR)*, June 2020. 1

[6] Jiangxin Dong and Jinshan Pan. Deep outlier handling for image deblurring. *IEEE Trans. Image Process.*, 30:1799–1811, 2021. 1

[7] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *ECCV*, 2020. 2

[8] Y. Du, J. Xu, Q. Qiu, X. Zhen, and L. Zhang. Variational image deraining. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, March 2020. 1

[9] Y. Du, J. Xu, X. Zhen, M. Cheng, and L. Shao. Conditional variational image deraining. *IEEE Transactions on Image Processing*, 29:6288–6301, 2020. 1

[10] 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*, 26(6):2944–2956, 2017. 2

[11] 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 (CVPR)*, July 2017. 2

[12] 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*, 31(6):1794–1807, 2020. 1

[13] X. Fu, Q. Qi, Z. J. Zha, X. Ding, and J. Paisley. Successive graph convolutional network for image de-raining. *International Journal of Computer Vision*, (5):1–21, 2021. 1

[14] 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*, pages 770–778, 2016. 5

[15] G. Hinton, O. Vinyals, and J. Dean. Distilling the knowledge in a neural network. In *NIPS Deep Learning and Representation Learning Workshop*, 2015. 2

[16] 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 (CVPR)*, July 2017. 5

[17] Q. Huynh-Thu and M. Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electronics Letters*, 44(13):800–801, 2008. 5

[18] 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 (CVPR)*, June 2020. 1

[19] O. L. Junior, D. Delgado, V. Goncalves, and U. Nunes. Trainable classifier-fusion schemes: An application to pedestrian detection. In *2009 12th International IEEE Conference on Intelligent Transportation Systems*, pages 1–6, 2009. 1

[20] L. Kang, C. Lin, and Y. Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE Transactions on Image Processing*, 21(4):1742–1755, 2012. 1

[21] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2, 3

[22] G. Li, X. He, W. Zhang, H. Chang, L. Dong, and L. Lin. Non-locally enhanced encoder-decoder network for single image de-raining. *acm multimedia*, 2018. 2, 5, 6

[23] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1, 2

[24] 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/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 1

[25] 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)*, September 2018. 2

[26] R. Pascanu and Y. Bengio. Revisiting natural gradient for deep networks. *arXiv preprint arXiv:1301.3584*, 2013. 2, 3

[27] W. Ran, Y. Yang, and H. Lu. Single image rain removal boosting via directional gradient. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6, 2020. 1

[28] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 2

[29] D. Ren, W. Zuo, Q. Hu, P. Zhu, and D. Meng. Progressive image deraining networks: A better and simpler baseline. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3932–3941, 2019. 1, 2, 5, 6

[30] Wenqi Ren, Jinshan Pan, Hua Zhang, Xiaochun Cao, and Ming-Hsuan Yang. Single image dehazing via multi-scale

convolutional neural networks with holistic edges. *Int. J. Comput. Vis.*, 128(1):240–259, 2020. 1

[31] M. S. Shehata, J. Cai, W. M. Badawy, T. W. Burr, M. S. Pervez, R. J. Johannesson, and A. Radmanesh. Video-based automatic incident detection for smart roads: The outdoor environmental challenges regarding false alarms. *IEEE Transactions on Intelligent Transportation Systems*, 9(2):349–360, 2008. 1

[32] 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*, MM '20, page 1643–1651, New York, NY, USA, 2020. Association for Computing Machinery. 1

[33] G. Wang, C. Sun, and A. Sowmya. Erl-net: Entangled representation learning for single image de-raining. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5643–5651, 2019. 1

[34] H. Wang, Y. Wu, M. Li, Q. Zhao, and D. Meng. A survey on rain removal from video and single image. *arXiv preprint arXiv:1909.08326*, 2019. 1

[35] 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 (CVPR)*, June 2020. 1, 2

[36] T. Wang, X. Yang, K. Xu, S. Chen, Q. Zhang, and R. W. H. Lau. Spatial attentive single-image deraining with a high quality real rain dataset. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12262–12271, 2019. 5, 6

[37] Y. Wang, Y. Cao, Z. J. Zha, J. Zhang, Z. Xiong, W. Zhang, and F. Wu. Progressive retinex: Mutually reinforced illumination-noise perception network for low light image enhancement. In *the 27th ACM International Conference*, 2019. 1

[38] B. M. Wilamowski and H. Yu. Improved computation for levenberg–marquardt training. *IEEE Transactions on Neural Networks*, 21(6):930–937, 2010. 4

[39] J. Xu, W. Zhao, P. Liu, and X. Tang. Removing rain and snow in a single image using guided filter. In *2012 IEEE International Conference on Computer Science and Automation Engineering (CSAE)*, volume 2, pages 304–307, 2012. 3

[40] 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 (CVPR)*, July 2017. 1, 2, 5

[41] 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*, pages 1–1, 2020. 1, 2

[42] Y. Yang and H. Lu. Single image deraining via recurrent hierarchy enhancement network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1814–1822, 2019. 5, 6

[43] 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)*, pages 1–6, 2020. 1

[44] R. Yasarla and V. M. Patel. Confidence measure guided single image de-raining. *IEEE Transactions on Image Processing*, 29:4544–4555, 2020. 1

[45] R. Yasarla, V. A. Sindagi, and V. M. Patel. Syn2real transfer learning for image deraining using gaussian processes. In *The IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020. 1

[46] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, 2004. 5

[47] H. Zhang and V. M. Patel. Density-aware single image deraining using a multi-stream dense network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 2

[48] 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*, 30(11):3943–3956, 2020. 2, 5, 6

[49] H. Zhao, H. Wang, Y. Fu, F. Wu, and X. Li. Memory efficient class-incremental learning for image classification. *arXiv preprint arXiv:2008.01411*, 2020. 2

[50] 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)*, pages 1–6, 2020. 1