Image Enhancement via Deep Spatial and Temporal Networks

Kaihao Zhang

A thesis submitted for the degree of
Doctor of Philosophy of
The Australian National University

September 2022

Publications

1. **Kaihao Zhang**, Dongxu Li, Wenhan Luo, Wenqi Ren, Björn Stenger, Wei Liu, Hongdong Li, Ming-Hsuan Yang. Benchmarking Ultra-High-Definition Image Super-resolution, In *The IEEE International Conference on Computer Vision (**ICCV**)*, 2021.

2. **Kaihao Zhang**, Rongqing Li, Yanjiang Yu, Wenhan Luo, Changsheng Li. Deep Dense Multi-scale Network for Snow Removal Using Semantic and Geometric Priors, In *IEEE Trans. on Image Processing (**TIP**)*, 2021.

3. **Kaihao Zhang**, Dongxu Li, Wenhan Luo, Wenqi Ren. Dual Attention-in-Attention Model for Joint Rain Streak and Raindrop Removal. In *IEEE Trans. on Image Processing (**TIP**)*, 2021.

4. **Kaihao Zhang**, Wenhan Luo, Yiran Zhong, Lin Ma, Björn Stenger, Wei Liu, Hongdong Li. Deblurring via Realistic Blurring, In *The IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**)*, 2020. (**Oral**)

5. **Kaihao Zhang**, Wenhan Luo, Wenqi Ren, Jingwen Wang, Fang Zhao, Lin Ma, Hongdong Li. Beyond Monocular Deraining: Stereo Image Deraining via Semantic Understanding, In *European Conference on Computer Vision (**ECCV**)*, 2020.

6. **Kaihao Zhang**, Wenhan Luo, Wenqi Ren, Björn Stenger, Lin Ma, Hongdong Li. Every Moment Matters: Detail-Aware Networks to Bring a Blurry Image Alive, In *The ACM International Conference on Multimedia (**ACM MM**)*, 2020. (**Oral**)

7. **Kaihao Zhang**, Wenhan Luo, Lin Ma, Wenqi Ren, Hongdong Li. Disentangled Feature Networks for Facial Portraits Generation, In *IEEE Trans. on Multimedia (**TMM**)*,2020.

8. **Kaihao Zhang**, Wenhan Luo, Lin Ma, Wei Liu, Hongdong Li. Learning Joint Gait Representation via Quintuplet Loss Minimization, In *The IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**)*, 2019. (**Oral**)

9. **Kaihao Zhang**, Wenhan Luo, Lin Ma, Hongdong Li. Cousin Network Guided Sketch Recognition via Latent Attribute Warehouse, In *Proc. of the Association for the Advancement of Artificial Intelligence (**AAAI**)*, 2019. (**Spotlight**)

10. **Kaihao Zhang**, Wenhan Luo, Yiran Zhong, Lin Ma, Wei Liu, Hongdong Li. Adversarial Spatio-Temporal Learning for Video Deblurring, In *IEEE Trans. on Image Processing (**TIP**)*, 2019.

11. **Kaihao Zhang**, Dongxu Li, Wenhan Luo, Wenqi Ren, Wei Liu. Enhanced Spatio-Temporal Interaction Learning for Video Deraining: A Faster and Better Framework. In *arXiv preprint arXiv:2103.12318*.

12. **Kaihao Zhang**, Wenhan Luo, Yanjiang Yu, Wenqi Ren, Fang Zhao, Changsheng Li, Lin Ma, Wei Liu, Hongdong Li. Beyond Monocular Deraining: Parallel Stereo Deraining Network Via Semantic Prior. In *arXiv preprint arXiv:2105.03830*.

13. **Kaihao Zhang**, Dongxu Li, Wenhan Luo, Jingyu Liu, Jiankang Deng, Wei Liu, Stefanos Zafeiriou. EDFace-Celeb-1M: Benchmarking Face Hallucination with a Million-scale Dataset. In *arXiv preprint arXiv:2110.05031*.

14. **Kaihao Zhang**, Wenhan Luo, Boheng Chen, Wenqi Ren, Björn Stenger, Wei Liu, Hongdong Li, Ming-Hsuan Yang. Benchmarking Deep Deblurring Algorithms: A New Large-Scale Multi-Cause Dataset and A New Baseline Model. In *arXiv preprint arXiv:2112.00234*.

15. Dongxu Li*, Chenchen Xu*, **Kaihao Zhang***, Xin Yu, Yiran Zhong, Wenqi Ren, Hanna Suominen, Hongdong Li. Arvo: Learning all-range volumetric correspondence for video deblurring, In *The IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**)*, 2021.

16. Jia Wan, **Kaihao Zhang**, Hongdong Li, Antoni B Chan. Angular-Driven Feedback Restoration Networks for Imperfect Sketch Recognition, In *IEEE Trans. on Image Processing (**TIP**)*, 2021.

17. Wenjia Niu, **Kaihao Zhang**, Wenhan Luo, Yiran Zhong. Blind Motion Deblurring Super-Resolution: When Dynamic Spatio-Temporal Learning Meets Static Image Understanding, In *IEEE Trans. on Image Processing (**TIP**)*, 2021.

18. Xiaobin Hu, Wenqi Ren, Kaicheng Yu, **Kaihao Zhang**, Xiaochun Cao, Wei Liu, Bjoern Menze. Pyramid Architecture Search for Real-Time Image Deblurring, In *The IEEE International Conference on Computer Vision (**ICCV**)*, 2021.

19. Jianyuan Wang, Yiran Zhong, Yuchao Dai, Stan Birchfield, **Kaihao Zhang**, Nikolai Smolyanskiy, Hongdong Li. Displacement-Invariant Cost Computation for Efficient Stereo Matching, In *The IEEE Conference on Computer Vision and Pattern Recognition (**CVPR**)*, 2021.

20. Wenjia Niu, **Kaihao Zhang**, Wenhan Luo, Yiran Zhong, Hongdong Li. Deep Robust Image Deblurring via Blur Distilling and Information Comparison in Latent Space, In *Neurocomputing*, 2021.

21. Dongxu Li, Chenchen Xu, Xin Yu, **Kaihao Zhang**, Benjamin Swift, Hanna Suominen, Hongdong Li. TSPNet: Hierarchical Feature Learning via Temporal Semantic Pyramid for Sign Language Translation, In *Conference on Neural Information Processing Systems, (**NeurIPS**)*, 2020.

22. Jianyuan Wang, Yiran Zhong, Yuchao Dai, **Kaihao Zhang**, Pan Ji, Hongdong Li. Displacement-Invariant Matching Cost Learning for Accurate Optical Flow Estimation, In *Conference on Neural Information Processing Systems, (**NeurIPS**)*, 2020.

23. Fang Zhao, Shengcai Liao, **Kaihao Zhang**, Ling Shao. Human Parsing Based Texture Transfer from Single Image to 3D Human via Cross-View Consistency, In *Conference on Neural Information Processing Systems, (**NeurIPS**)*, 2020.

24. Fang Zhao, Shengcai Liao, Guosen Xie, Jian Zhao, **Kaihao Zhang**, Ling Shao. Unsupervised Domain Adaptation with Noise Resistible Mutual-Training for Person Re-identification, In *European Conference on Computer Vision, (**ECCV**)*, 2020.

25. Ben Niu, Weilei Wen, Wenqi Ren, Xiangde Zhang, Lianping Yang, Shuzhen Wang, **Kaihao Zhang**, Xiaochun Cao, and Haifeng Shen. Single Image Super-Resolution via a Holistic Attention Network, In *European Conference on Computer Vision, (**ECCV**)*, 2020.

26. Lirong Zheng, Yanshan Li, **Kaihao Zhang**, Wenhan Luo. T-Net: Deep Stacked Scale-Iteration Network for Image Dehazing, In *arXiv preprint arXiv:2106.02809*.

27. Yiran Zhong, Charles Loop, Wonmin Byeon, Stan Birchfield, Yuchao Dai, **Kaihao Zhang**, Alexey Kamenev, Thomas Breuel, Hongdong Li, Jan Kautz. Displacement Invariant Cost Computation for Efficient Stereo Matching, In *arXiv preprint arXiv:2012.00899*.

28. Jing Zhang, Yuchao Dai, Mochu Xiang, Dengping Fan, Peyman Moghadam, Mingyi He, Christian Walder, **Kaihao Zhang**, Mehrtash Harandi, Nick Barnes. Dense Uncertainty Estimation, In *arXiv preprint arXiv:2110.06427*.

Kaihao Zhang
16 September 2022

Dedicated to my parents.

Acknowledgments

I would first like to thank my supervisor, Professor Hongdong Li, for guiding me with patience and wisdom throughout this PhD journey. Over the past four years, his insightful feedback pushed me to sharpen my thinking and brought my work to a higher level.

I am also highly grateful to Dr. Wenhan Luo, Dr. Wei Liu, Dr. Lin Ma, Dr. Björn Stenger and Ming-Hsuan Yang, for their constructive suggestion which helps shape me as a researcher.

I thank the Australian National University for providing me with financial security through ANU International PhD Scholarship and the ANU HDR Merit Scholarship, without which this work would not have been possible.

I thank all the members, past and present, of the computer vision groups at ANU, for their companionship. Discussions with them also inspired some of the findings in this thesis.

Finally, I thank my family for their support and encouragement throughout my life.

Abstract

Image enhancement is a classic problem in computer vision and has been studied for decades. It includes various subtasks such as super-resolution, image deblurring, rain removal and denoise. Among these tasks, image deblurring and rain removal have become increasingly active, as they play an important role in many areas such as autonomous driving, video surveillance and mobile applications. In addition, there exists connection between them. For example, blur and rain often degrade images simultaneously, and the performance of their removal rely on the spatial and temporal learning. To help generate sharp images and videos, in this thesis, we propose efficient algorithms based on deep neural networks for solving the problems of image deblurring and rain removal.

In the first part of this thesis, we study the problem of image **deblurring**. Four deep learning based image deblurring methods are proposed. First, for single image deblurring, a new framework is presented which firstly learns how to transfer sharp images to realistic blurry images via a learning-to-blur Generative Adversarial Network (GAN) module, and then trains a learning-to-deblur GAN module to learn how to generate sharp images from blurry versions. In contrast to prior work which solely focuses on learning to deblur, the proposed method learns to realistically synthesize blurring effects using unpaired sharp and blurry images. Second, for video deblurring, spatio-temporal learning and adversarial training methods are used to recover sharp and realistic video frames from input blurry versions. 3D convolutional kernels on the basis of deep residual neural networks are employed to capture better spatio-temporal features, and train the proposed network with both the content loss and adversarial loss to drive the model to generate realistic frames. Third, the problem of extracting sharp image sequences from a single motion-blurred image is tackled. A detail-aware network is presented, which is a cascaded generator to handle the problems of ambiguity, subtle motion and loss of details. Finally, this thesis proposes a level-attention deblurring network, and constructs a new large-scale dataset including images with blur caused by various factors. We use this dataset to evaluate current deep deblurring methods and our proposed method.

In the second part of this thesis, we study the problem of image **deraining**. Three deep learning based image deraining methods are proposed. First, for single image deraining, the problem of joint removal of raindrops and rain streaks is tackled. In contrast to most of prior works which solely focus on the raindrops or rain streaks removal, a dual attention-in-attention model is presented, which removes raindrops and rain streaks simultaneously. Second, for video deraining, a novel end-to-end framework is proposed to obtain the spatial representation, and temporal correlations based on ResNet-based and LSTM-based architectures, respectively. The proposed method can generate multiple deraining frames at a time, which outperforms the

state-of-the-art methods in terms of quality and speed. Finally, for stereo image deraining, a deep stereo semantic-aware deraining network is proposed for the first time in computer vision. Different from the previous methods which only learn from pixel-level loss function or monocular information, the proposed network advances image deraining by leveraging semantic information and visual deviation between two views.

Key Words: Deep Learning, Single Image Deblurring, Video Deblurring, Make a Blurry Image Alive, Single Image Deraining, Video Deraining, Stereo Deraining, Benchmarking.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview

With the development of mobile cameras and social media, a large number of photos and videos are captured and uploaded to the Internet. However, these photos often suffer from artifacts, such as blurriness, noise and low resolution. Many of them are often degraded due to bad weather, such as rain, haze and snow. To improve the visual quality of images and videos, various technologies are proposed in the field of computer vision, such as image super-resolution, deblurring, dehazing and deraining. Traditional algorithms typically rely on a variety of priors or assumptions. In recent years, deep learning has demonstrated great performance in many low-level image restoration problems. In this thesis, several effective deep learning based methods are proposed to study the image enhancing problem in the aspects of image deblurring and deraining.

In the first part of this thesis, we will study the problem of image deblurring. Image deblurring is a classic task in low-level computer vision. Its objective goal is to recover a sharp image from a blurred input image, where the blur can be caused by various factors such as out of focus, camera shake, or fast target motion [Abuolaim and Brown, 2020a; Chen and Shen, 2015; Kang, 2007; Sun et al., 2015]. Some examples are given in Figure 1.1. Recently, image deblurring has attracted the attention from the image processing and computer vision community, as its applications are found in many important fields. For example, a social networking service providing sharper images is more attractive than its competitors. In addition, the deblurring images are also beneficial for the current Intelligent system to detect and recognize objects.

Conventional image deblurring methods often formulate the task as an inverse filtering problem, where a blurred image is modeled as the result of the convolution with blur kernels. Some early approaches assume that the blur kernel is known, and adopt classical image deconvolution algorithms such as Lucy-Richardson, or Wiener deconvolution, with or without Tikhonov regularization, to restore sharp images [Schmidt et al., 2013; Szeliski, 2010; Xu et al., 2014b]. On the other hand, blind image deblurring methods assume the blur kernel is unknown and aim to simultaneously recover both the sharp image and the blur kernel itself. Since this task is ill-posed, various additional constraints are used to regularize the solution [Bahat et al., 2017; Cho and Lee, 2009; Fergus et al., 2006; Xu and Jia, 2010]. While

(a) Camera shake blur (b) Out-of-focus blur (c) Moving object blur (d) Mixed blur

Figure 1.1: **Examples of different blurry images**. Images suffer from blur due to (a) camera shake, (b) out-of-focus scene, (c) moving object, and (d) multiple causes, respectively.

these conventional methods show good performance in certain cases, they typically do not perform well under more complicated yet common scenarios such as images with strong motion blur.

Recent advances of deep learning techniques have revolutionized the field of computer vision; significant progress has been made in all computer vision domains, including image classification [He et al., 2016; Simonyan and Zisserman, 2015a] and object detection [He et al., 2017; Isola et al., 2017; Ren et al., 2015; Zhu et al., 2017a]. Image deblurring is no exception: a large number of deep learning methods have been developed for image deblurring, and have advanced the state of the art. In the first part of thesis thesis, we will introduce our three recently published deep deblurring methods, which correspond to the above tasks, and one benchmark.

- **Single image deblurring (Chapter 3)** . Existing deep learning methods for single image deblurring typically train models using pairs of sharp images and their blurred counterparts. However, synthetically blurring images does not necessarily model the blurring process in real-world scenarios with sufficient accuracy. To address this problem, a new method is proposed which combines two GAN models, *i.e.*, a learning-to-Blur GAN (BGAN) and learning-to-DeBlur GAN (DBGAN), in order to learn a better model for image deblurring by primarily learning how to blur images. The first model, BGAN, learns how to blur sharp images with unpaired sharp and blurry image sets, and then guides the second model, DBGAN, to learn how to correctly deblur such images. In order to reduce the discrepancy between real blur and synthesized blur, a relativistic blur loss is leveraged. As an additional contribution, this part also introduces a Real-World Blurred Image (RWBI) dataset including diverse blurry images. The experiments show that the proposed method achieves consistently superior quantitative performance as well as higher perceptual quality on both the newly proposed dataset and the public GOPRO dataset.

  Even though the above method achieves satisfactory performance for single image deblurring, the DBGAN model cannot extract the spatio-temporal information from continuing blurry frames. To address this problem, we further

introduce a video deblurring method in the next Chapter.

- **Video Deblurring (Chapter 4).** Camera shake or target movement often leads to undesired blur effects in videos captured by a hand-held camera. Despite significant efforts having been devoted to video-deblur research, two major challenges remain: 1) how to model the spatio-temporal characteristics across both the spatial domain (i.e., image plane) and temporal domain (i.e., neighboring frames), and 2) how to restore sharp image details with regard to the conventionally adopted metric of pixel-wise errors. To address the first challenge, a *DeBLuRring Network (DBLRNet)* is proposed for spatio-temporal learning by applying a 3D convolution to both spatial and temporal domains. Our DBLRNet is able to capture jointly spatial and temporal information encoded in neighboring frames, which directly contributes to improved video deblur performance. To tackle the second challenge, the developed DBLRNet is leveraged as a generator in the GAN (generative adversarial network) architecture, and a content loss is employed in addition to an adversarial loss for efficient adversarial training. The developed network, named as *DeBLuRring Generative Adversarial Network (DBLRGAN)*, is tested on two standard benchmarks and achieves the state-of-the-art performance.

  Even the two above methods achieve satisfactory performance for single image deblurring and video deblurring, they cannot generate multiple neighbouring sharp frames from a single motion-blurred image. To address this problem, we further propose a deblurring method in the next Chapter.

- **Make a Blurred Image Alive (Chapter 5).** Motion-blurred images are the result of light accumulation over the period of camera exposure time, during which the camera and objects in the scene are in relative motion to each other. The inverse process of extracting an image sequence from a single motion-blurred image is an ill-posed vision problem. One key challenge is that the motions across frames are subtle, which makes the generating networks difficult to capture them and thus the recovery sequences lack motion details. In order to alleviate this problem, a detail-aware network is proposed with three consecutive stages to improve the reconstruction quality by addressing specific aspects in the recovery process. The detail-aware network firstly models the dynamics using a cycle flow loss, resolving the temporal ambiguity of the reconstruction in the first stage. Then, a GramNet is proposed in the second stage to refine subtle motion between continuous frames using Gram matrices as motion representation. Finally, a HeptaGAN is introduced in the third stage to bridge the continuous and discrete nature of exposure time and recovered frames, respectively, in order to maintain rich detail. Experiments show that the proposed detail-aware networks produce sharp image sequences with rich details and subtle motion, outperforming the state-of-the-art methods.

  Even achieving satisfactory performance, the above methods focus on the task of motion deblurring. To explore the performance of current methods on dif-

(a) Rain streak (b) Raindrop (c) Rain streak and raindrop

Figure 1.2: **Examples of different rainy images** . Images suffer from rain due to (a) rain streak, (b) raindrop, and (c) rain streak and raindrop, respectively.

ferent kinds of blurry images, this thesis further introduces a benchmark.

- **Benchmark (Chapter 6).** In this part, we address how different deblurring methods perform on general types of blur. For in-depth performance evaluation, we construct a new large-scale multi-cause image deblurring dataset called (MC-Blur) including real-world and synthesized blurry images with mixed factors of blurs. The images in the proposed MC-Blur dataset are collected using different techniques: convolving Ultra-High-Definition (UHD) sharp images with large kernels, averaging sharp images captured by a 1000 fps high-speed camera, adding defocus to images, and real-world blurred images captured by various camera models. These results provide a comprehensive overview of the advantages and limitations of current deblurring methods. Further, we propose a level-attention deblurring network to adapt to multiple causes of blurs. By including different weights of attention to the different levels of features, the proposed network derives more powerful features with larger weights assigned to more important levels, thereby enhancing the feature representation. Extensive experimental results on the new dataset demonstrate the effectiveness of the proposed model for the multi-cause blur scenarios.

In the second part of this thesis, we will study the problem of image deraining. Images and videos captured by cameras in outdoor scenarios often suffer from bad weather conditions. As one of the commonest weather phenomena, rain causes visibility degradation and destroys the performance of many computer vision systems. This is because most current computer vision algorithms assume clear weather, with no interference from rain. The goal of deraining is to remove those undesired rain streaks and restore clean images based on the input rainy versions. This is an important problem in the computer vision field as it is beneficial for the robustness of modern intelligent systems [Zheng et al., 2015; Han and Bhanu, 2005; Yang et al., 2019a; Li et al., 2019a].

Based on different inputs, the deraining methods can be divided into two groups: single image deraining and video deraining. Before 2017, the typical deraining methods are driven by image decomposition, sparse coding, and priors based Gaussian mixture models. For example, [Kang et al., 2011] use a bilateral filter to decompose

an image into the high-frequency part to help remove rain. [Chen et al., 2014] and [Luo et al., 2015] use the sparse coding framework to separate rain and backgrounds via classified dictionary atoms and discriminate sparse coding, respectively. Considering that rain streaks often exhibit similar patterns, [Chen and Hsu, 2013] generalize a low-rank model to capture the spatio-temporally correlated rain streaks. [Kim et al., 2013] detect rain streak regions and then use non-local means filter to remove rain from the detected regions via selecting non-local neighbour pixels.

Since 2017, with the development of deep learning methods, deraining enters into a new era. The major development of deep deraining models are indicated by the ideas of CNN, GAN and unsupervised learning. For example, [Fu et al., 2017b] propose a deep detail network to remove rain streaks from individual images via directly reducing the mapping range. [Yang et al., 2017] introduce a deep joint rain detection and removal model to first detect rainy space via a contextualized dilated network and then recover the clean image based on the detected rainy space. [Qian et al., 2018] apply attention mechanism into deep neural network to first detect raindrops and then use a GAN-based architecture to restore realistic clean images.

In recent years, the performance of deep deraining methods has overtaken non-deep deraining methods. In the second part of these thesis, we aim to introduce our three recently published deep deraning methods.

- **Image Deraining (Chapter 7).** Rain streaks and rain drops are two natural phenomena, which degrade image capturing in different ways. Currently, most existing deep deraining networks take them as two distinct problems and individually address one, and thus cannot deal adequately with both simultaneously. To address this, a Dual Attention-in-Attention Model (DAiAM) which includes two DAMs for removing both rain streaks and raindrops is proposed. Inside the DAM, there are two attentive maps - each of which attends to the heavy and light rainy regions, respectively, to guide the deraining process differently for applicable regions. In addition, to further refine the result, a Differential-driven Dual Attention-in-Attention Model (D-DAiAM) is proposed with a “heavy-to-light" scheme to remove rain via addressing the unsatisfying deraining regions. Extensive experiments on one public raindrop dataset, one public rain streak and a synthesized joint rain streak and raindrop (JRSRD) dataset have demonstrated that the proposed method is not only capable of removing rain streaks and raindrops simultaneously, but also achieves the state-of-the-art performance on both tasks.

  Even the above method achieves promising performance for single image deraining, the proposed model cannot extract the spatio-temporal information from continuing rainy frames. To address this problem, we further introduce a video deraining method in the next Chapter.

- **Video Deraining (Chapter 8).** Video deraining is an important task in computer vision as the unwanted rain hampers the visibility of videos and deteriorates the robustness of most outdoor vision systems. Despite the significant success which has been achieved for video deraining recently, two major

challenges remain: 1) how to exploit the vast information among continuous frames to extract powerful spatio-temporal features across both the spatial and temporal domains, and 2) how to restore high-quality derained videos with a high-speed approach. In this thesis, a new end-to-end video deraining framework is presented, named as Enhanced Spatio-Temporal Interaction Network (ESTINet), which considerably boosts current state-of-the-art video deraining quality and speed. The ESTINet takes the advantage of deep residual networks and convolutional long short-term memory, which can capture the spatial features and temporal correlations among continuing frames at the cost of very few computational sources. Extensive experiments on three public datasets show that the proposed ESTINet can achieve faster speed than the competitors, while maintaining better performance than the state-of-the-art methods.

Even the two above methods achieve satisfactory performance for single image deraining and video deraining, the input of them should be monocular rainy images. To address the problem of stereo deraining, we further propose a stereo deraining method in the next Chapter.

- **Stereo Deraining (Chapter 9).** Nowadays, state-of-the-art models adopted in autonomous driving rely on stereo cameras. However, there are few studies on deraining for stereo images. Meanwhile, even for monocular deraining, most of current methods fail to understand and remove rain because these methods consider only pixel-level loss functions during training. In this thesis, a Paired Rain Removal Networks (PRRNet) is present, the first stereo semantic-aware deraining networks, which can be trained without pairs of rainy image and its segmentation annotation. Within PRRNet, there is a Semantic-Aware Deraining Module (SADM) considering both tasks of semantic understanding and deraining of scene, a Semantic-Fusion Network (SFNet) combining semantic segmentation and deraining images, and a View-Fusion Network (VFNet) fusing information from multiple views. Two stereo rainy datasets are also synthesized to evaluate different deraining methods. Experimental results on one public monocular and two developed stereo rainy datasets demonstrate that the PRRNet achieves the state-of-the-art performance on both monocular and stereo image deraining.

In summary, this thesis aims to develop deep learning based methods to extract spatial information from a single image, or spatio-temporal information from a video, to help recover high-quality images and videos based on their corresponding low-quality versions. Specifically, for image deblurring, the objective is to remove blur kernels and generate sharp images or videos. For image deraining, the objective is to remove rain streaks or raindrops from rainy images, and generate clean images and videos.

1.2 Problem Formulation

1.2.1 Blur models

Image blur can be caused by various factors during image capture: camera shake, in-scene motion, or out-of-focus blur. We denote a blurred image I_b as

$$I_b = \Phi(I_s; \theta_\eta), \tag{1.1}$$

where Φ is the image blur function, and θ_η is a parameter vector. Deblurring methods can be categorized into non-blind and blind methods, depending on whether or not the blur function is known. The goal of image deblurring is to recover a sharp image, i.e., , finding the inverse of the blur process, as

$$I_{db} = \Phi^{-1}(I_b; \theta_\nu), \tag{1.2}$$

where Φ^{-1} is the deblurring model, θ_ν represents its parameters, and I_{db} is the deblurred image, which is the estimate of the latent sharp image I_s.

Motion Blur. An image is captured by measuring photons over the time period of camera exposure. Under bright illumination the exposure time is sufficiently short for the image to capture an instantaneous moment. However, a longer exposure time may result in motion blur and degrade a sharp image to a blurry version. Numerous methods directly model the degradation process as a convolution process by assuming that the blur is uniform across the entire image:

$$I_b = K * I_s + \theta_\mu\,, \tag{1.3}$$

where K is the blur kernel and θ_μ represents additive Gaussian noise.

In such an image, any object moving with respect to the camera will look blurred along the direction of relative motion. For camera shake, motion blur often occurs in the static background, while fast moving objects (without camera shake) will cause these objects to be blurred while the background remains sharp. A blurred image can naturally contain blur caused by both factors. Early methods model blur using shift-invariant kernels [Fergus et al., 2006; Xu et al., 2014a], while more recent studies address the case of non-uniform blur [Gao et al., 2019; Kupyn et al., 2018, 2019; Nah et al., 2017a; Tao et al., 2018].

Out-of-focus Blur. Aside from motion blur, image sharpness is also affected by the distance between the scene and the camera's focal plane. When objects are in focus, they are exactly on the focal plane, otherwise blur will appear. If objects are at different distance to the camera, parts of the scene may be in focus, while others appear blurry. The Point Spread Function (PSF) for out-of-focus blur is modeled as:

$$K(x,y) = \begin{cases} \dfrac{1}{\pi r^2}, & \text{if } (x-k)^2 + (y-l)^2 \leq r^2, \\ 0, & \text{elsewhere,} \end{cases} \tag{1.4}$$

where (k, l) is the center of the PSF and r the radius of the blur. Out-of-focus deblurring has applications in salient detection [Jiang et al., 2013], defocus magnification [Bae and Durand, 2007] and image refocusing [Zhang and Cham, 2009]. To address the problem of out-of-focus blur, classic methods remove blurry artifacts via coded apertures [Masia et al., 2011] or blur detection [Shi et al., 2014]. Deep neural networks have recently been used to detect blur regions [Tang et al., 2019; Zhao et al., 2019] and predict depth [Lee et al., 2019] to guide the deblurring process.

Gaussian Blur. Gaussian convolution is a common simple blur model used in image processing, defined as

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}}, \tag{1.5}$$

where x and y are the distance from the origin in the horizontal and vertical axis, respectively, σ is the standard deviation. Several classic methods have been developed to remove the Gaussian blur [Chen and Ma, 2009; Hummel et al., 1987; Vairy and Venkatesh, 1995].

Mixed Blur. In many real-world scenes multiple factors contribute to blur, such as camera shake, object motion, and depth variation. For example, when a fast-moving object is captured at an out-of-focus distance, the image may include both motion blur and out-of-focus blur as shown in Figure 7.1(d). To synthesize this type of blurry image, one option is to firstly transform sharp images to their motion-blurred versions (*e.g.*, by averaging neighboring sharp frames taken in sequence) and then apply an out-of-focus blur kernel based on Eq. 1.4. Alternatively, one can train a blurring network to directly generate realistically blurred images.

1.2.2 Rain models

Many important factors (including surface tension, hydrostatic pressure, ambient illumination, and aerodynamic pressure [Beard and Chuang, 1987; Tripathi and Mukhopadhyay, 2014]) produce rapid shape distortions in a falling raindrop. Rain streaks with different brightness, directions and distort background object of images will appear as a result of these aberrations [Tripathi and Mukhopadhyay, 2011; Garg and Nayar, 2005]. In the following, we will introduce three popular rain models used in existing studies.

Additive composite model. The most simple and popular rain model used in existing studies is the additive composite model, whose mathematical formulation is expressed as

$$O = B + R\,, \tag{1.6}$$

where O, B and R are the observed rainy image, the latent clean image and the rain-streak component, respectively. The goal of image deraining is to recover a clean image as

$$I_{dr} = \Phi^{-1}(B; \theta_v), \tag{1.7}$$

where Φ^{-1} is the deraining model, θ_ν represents its parameters, and B is the derained image, which is the estimate of the latent clean image I_c.

Screen Blend Model. Considering the background and rain layers can influence the appearance of each other, [Luo et al., 2015] propose a screen blend model to model rainy images, as:

$$O = 1 - (1 - B) \circ (1 - R), \tag{1.8}$$

where $\circ$ is the point-wise multiplication. The screen blend model is able to model visual properties of real rainy images including the effect of internal reflections and thus generate more authentic ran images.

Heavy rain model. [Yang et al., 2017] offer a rain model that considers rain streaks as well as rain accumulation. This is the first deraining model that reflects two rain phenomena. Rain accumulation, also known as rain veiling, is caused by water particles in the atmosphere and distant rain-streaks that are hard to be seen separately. Rain accumulation has a similar visual effect to mist. Considering the rain streaks and rain accumulation, the heavy rain can be modeled as:

$$O = \alpha \circ (B + \sum_{t=1}^{s} S_t) + (1 - \alpha)A, \tag{1.9}$$

where S_t is the rain-streak layer that has the same streak. t indexes the rain-streak layer and s is the number of rain-streak layers. A, α and $\circ$ are the global atmospheric light, asmospheric transmission and operation of point-wise multiplication, respectively.

1.3 Thesis Structure

The thesis consists of basically two parts. The first part is about image deblurring and the second part is about image deraining. In the first part, we have four chapters, which are Chapter-3, Chapter-4, Chapter-5 and Chapter-6. In the second part, we have four chapters, which are Chapter-7, Chapter-8 and Chapter-9. In addition, this thesis introduces the recently published deep learning based image deblurring and deraining methods in Chapter-2, and provide the future research directions in Chapter-10.

Chapter 2

Related Work

This chapter presents a brief review to existing methods for image deblurring. Focuses are given to recently published work especially those that are based on deep-learning.

2.1 Deblurring

The first part of this thesis studies the problem of single image deblurring, video deblurring, and making a blurry image alive. Therefore, the following is a brief review of related methods.

2.1.1 Single Image Deblurring

The input of single image deblurring methods is one blurry image, and single image deblurring methods aim to generate its corresponding sharp version [Zhang et al., 2020c]. As one early deep deblurring method, [Sun et al., 2015] design a CNN-based architecture for non-uniform image deblurring, where CNN is utilized to predict the probabilistic distribution of blur kernels from local patches under the help of a Markov random field model. Another early CNN-based image deblurring model is introduced by [Chakrabarti, 2016], to predict the Fourier coefficients of a deconvolution filter, which can be applied to recover latent sharp images from blurry ones.

Apart from them, to facilitate the spatially variant blur removal, [Zhang et al., 2018a] propose a RNN-based model to help remove spatially variant blurs, under the help of a CNN model by updating the weights of RNNs. The CNN and RNN work together to fuse information from different directions and cover a large receptive field. On the other hand, recurrent layer can be also used to extract features across images in multiple scales with a coarse-to-fine scheme. Two representative methods are SRN [Tao et al., 2018] and PSS-SRN [Gao et al., 2019]. PSS-SRN uses a selective sharing scheme to share more parameters and improve the performance of SRN.

In addition, numerous image deblurring models have been developed to exploit residual learning with local and global residual layers. The first group addresses deep deblurring via the local residual layer, which is similar to the residual layers in ResNet and widely used in image deblurring models [Nimisha et al., 2017; Tao et al.,

2018; Gao et al., 2019; Zhang et al., 2019a]. To learn the complicated transformation from a blurry image to a sharp one, residual learning is employed. As such, it can better restore the missing details in a blurry image. The second group employs the global residual layer in their methods [Nah et al., 2017a; Kupyn et al., 2018; Zhang et al., 2018d]. It is well known that blurry images and their corresponding sharp versions have high correlation. To better model the translation between two different images, some models directly calculate the residuals between them to further improve the translation ability.

To alleviate the vanishing-gradient problem, strengthen feature propagation, reuse features and reduce the number of parameters. [Purohit and Rajagopalan, 2019] propose a region-adaptive dense network to remove motion blur. It is composed of region adaptive modules to learn the spatially varying shifts in the input blurry image. These region adaptive modules are incorporated into a densely connected Autoencoder architecture to further improve the learning ability.

The attention models are able to focus on the most relevant parts considering the context. Together with RNN and CNN, attention-based methods have benefited various tasks, including image deblurring. The attention scheme helps model to focus on the parts important for deblurring. [Shen et al., 2019] propose an attention-based deep deblurring method, which consists of three branches to remove blurs from the foreground, background, and global parts, respectively. Considering that human is usually the most important object in blurry images, an attention module is built to detect the position of human and then recover the sharp images under the guidance of a so-called human-aware map.

To generate more realistic sharp images, [Kupyn et al., 2018] develop a DeblurGAN, which is an end-to-end conditional GAN model for motion deblurring. The generator of DeblurGAN contains two strided convolution blocks, nine residual blocks, and two transposed convolution blocks. The generator transfers a blurry image to its corresponding sharp version. Meanwhile, the discriminator distinguishes whether its inputs are sharp or blurry. The goal of the generator is to generate high-quality deblurred images to fool the discriminator. They evaluate the performance with different metrics, such as PSNR, SSIM, and the performance of object detection on deblurred images to demonstrate the effectiveness of DeblurGAN. It is further extended to the DeblurGan-v2 [Kupyn et al., 2019] scheme based on a relativistic conditional GAN and a double-scale discriminator. The core block of the generator is a feature pyramid network, which achieves better performance and improves the efficiency.

2.1.2 Video Deblurring

The input of video deblurring methods is one blurry video, and video deblurring methods aim to generate its corresponding sharp version [Zhang et al., 2018b]. [Su et al., 2017b] and [Wang et al., 2019b] design two DAE architectures to remove blurs from videos. Several continuing blurry images are put into the encoder together and then the decoder recovers the central sharp frame of them. Different from the stan-

dard autoencoder architecture, [Su et al., 2017b] connect the corresponding layers in the encoder and decoder via symmetric skip connections [Mao et al., 2016]. Features extracted from different layers of the encoder are element-wisely added to the corresponding layers of the decoder. This method can accelerate the convergence and generate much sharper videos. While [Wang et al., 2019b] employ an upsampling layer, which can address the super-resolution task simultaneously.

In addition, [Hyun Kim et al., 2017] and [Nah et al., 2019b] propose spatio-temporal recurrent networks to enforce temporal consistency between consecutive frames by dynamic temporal blending or exploiting information from past frames in the form of hidden state. Some methods generate the deblurred frames via directly inputting the deblurred frames from the last time. For example, [Zhou et al., 2019a] propose a frame-recurrent method to remove motion blurs. Extracted features from blurry frames are fed into an STFAN layer to learn the static of information. During this stage, the deblurred images from the last time are also set as input of their STFAN to guide the deblurring process. Finally, the output of the STFAN layer is put into a decoder to recover sharp frames.

Similar to single image deblurring, GANs are also utilized in video deblurring. The main difference comes from the generator, which usually has to consider the modeling of temporal information implied in neighbouring frames. For example, [Kupyn et al., 2019] propose DeblurGAN-v2 to restore sharp videos via modifying the single image deblurring method of DeblurGAN [Kupyn et al., 2018].

2.1.3 Making a Blurred Image Alive

The aim of making a blurry image alive [Zhang et al., 2020b] is to generate several sharp images based on one blurry image. This task is related to single image deblurring, video deblurring and video generation. Generating videos from texts, images or videos poses challenges to existing generative models [Isola et al., 2017; Zhu et al., 2017a; Wang et al., 2018b]. For motion prediction, recent methods focus on training transform networks to compress the current information and generate a sequence of future frames [Mathieu et al., 2016; Villegas et al., 2017; Xiong et al., 2018; Zhao et al., 2018]. Using a GAN, [Mathieu et al., 2016] predicted future frames based on adversarial loss and image gradient difference loss. [Villegas et al., 2017] built a model based on an Encoder-Decoder CNN and a Conv-LSTM to capture the spatial-temporal dynamics. Their model effectively handles complex variations in pixel space. [Zhao et al., 2018] proposed a two-stage framework to generate frames and then refine by temporal signals.

The closest work for producing a video sequence from a blurry image is the pioneering work in [Jin et al., 2018]. It first estimates the middle frame of the temporal sequence and then sequentially reconstructs pairs of frames, one forward and one backward in time, in each step. Following this work, [Pan et al., 2019] proposed an EDI model to reconstruct a sharp video from a single blurry frame based on event camera, while [Purohit et al., 2019] try to learn a motion encoder for blurred images based on a pre-trained convolutional recurrent video autoencoder network.

2.2 Deraining

The second part of this thesis studies the problem of singe image deraining, video deraning and stereo deraining. Therefore, the following is a brief review of current related deraining methods. Considering rain streak and raindrops removal are two main tasks for single image deraining, this chapter introduces them separately. The aim of image deraining is to generate clean images/videos based on given rainy versions [Yang et al., 2020a].

2.2.1 Rain Streak Removal

Traditional methods design hand-crafted priors to remove rain streaks [Barnum et al., 2010; Kang et al., 2011; Huang et al., 2013; Luo et al., 2015; Li et al., 2016; Chang et al., 2017; Zhu et al., 2017c, 2020; Hu et al., 2021; Wang et al., 2020b]. [Kang et al., 2011] use a bilateral filter to decompose an image into the low- and high-frequency parts, which are then decomposed into different components by performing dictionary learning and sparse coding. Similarly, [Huang et al., 2013] present a method to first learn an over-complete dictionary from the image high spatial frequency parts and then perform unsupervised clustering on the dictionary atoms. [Zhu et al., 2017c] use a joint optimization process with three image priors to remove rain-streak details.

Recently, deep learning achieves significant success in rain streak removal [Fu et al., 2017a,b; Yang et al., 2017; Zhang and Patel, 2018b; Li et al., 2018d; Zhang et al., 2019c; Zhu et al., 2020; Zhou et al., 2021; Hu et al., 2021]. [Fu et al., 2017b] propose a deep network to remove background interference and focus on the structure of rain based on prior knowledge. [Zhang and Patel, 2018b] introduce a DID-MDN model to jointly estimate rain density and remove rain. [Li et al., 2018d] propose a deep convolutional and recurrent neural network for deraining. To make the derained images more realistic, [Zhang et al., 2019c] introduce a CGAN-based model with additional regularization. [Wang et al., 2020a] explore the intrinsic prior structure of rain streaks and then propose a novel interpretable network to remove the rain streaks from rainy images. [Li et al., 2021] propose a comprehensive benchmark analysis of several single image deraining networks. [Zhu et al., 2020] and [Hu et al., 2021] introduce two non-local networks to improve the performance of image deraining. [Wang et al., 2020b] rethink about the image deraining and reformulate rain streaks as transmission medium together with vapors to address the problem of image deraining.

2.2.2 Raindrop Removal

Most methods for rain streak removal are not directly applicable for raindrop removal. Therefore, many methods are proposed like raindrop detection and removal [Kurihata et al., 2005; Roser and Geiger, 2009; Roser et al., 2010; Yamashita et al., 2005, 2009; You et al., 2015; Eigen et al., 2013; Quan et al., 2019; Alletto et al., 2019; Hao et al., 2019]. Specifically, [Kurihata et al., 2005] use PCA to learn the shape of

raindrops, which are then utilized to match rainy regions. [Yamashita et al., 2005] introduce a method based on the stereo measurement and disparities between stereo image pair. Position of raindrops can be detected. Finally, sharp image can be obtained by replacing raindrop regions. [Roser and Geiger, 2009] propose a method to perform monocular raindrop detection. [You et al., 2015] introduces a method to exploit local spatio-temporal cue for video raindrop removal. They first model and detect adherent raindrops, then remove them and restore the images. More recently, there are many methods using deep learning methods for single image raindrop removal [Eigen et al., 2013; Qian et al., 2018], which are trained with pairs of raindrops and corresponding sharp images. [Quan et al., 2019] propose a CNN-based method to restore an image taken through glass window in rainy weather via using shape-driven attention and channel re-calibration.

Almost all existing single image deraining methods dissever the two tasks and focus on either rain streaks or raindrops [Li et al., 2019c]. Meanwhile, most datasets typically contain only one kind of rain.

2.2.3 Video Deraining

The aim of video deraining methods is to generate a clean video based on its corresponding rainy version [Zhang et al., 2021a]. In order to make use of the temporal corrections among video sequence frames, several video-based deraining methods are proposed and show huge advantage for removing rain [Garg and Nayar, 2004; Barnum et al., 2010; Santhaseelan and Asari, 2012, 2015; You et al., 2015]. The early works focus on capturing the temporal context and motion information via prior-based methods [Garg and Nayar, 2004, 2006]. These kinds of methods model the rain streaks based on the photo-metric appearance of rain [Zhang et al., 2006; Liu et al., 2009; Santhaseelan and Asari, 2015; Brewer and Liu, 2008; Jiang et al., 2017] and propose learn-based models to address the problem of video deraining [Chen and Chau, 2013; Tripathi and Mukhopadhyay, 2012; Kim et al., 2015; Wei et al., 2017; Ren et al., 2017]. For example, [Zhang et al., 2006] combine temporal and chromatic properties to remove rain from video. [Santhaseelan and Asari, 2015] and [Barnum et al., 2010] remove rain streaks via extracting phase congruence features and Fourier domain features, respectively. [Kim et al., 2015] propose a temporal correlation and low-rank matrix completion method to remove rain based on the observation that rain streaks cannot affect the optical flow estimation between frame.

Recently, many deep learning based methods are proposed and bring significant changes to the video deraining [Li et al., 2018b; Liu et al., 2018a,b; Chen et al., 2018b; Yang et al., 2019b, 2020b; Yue et al., 2021]. [Chen et al., 2018b] firstly use a super-pixel segmentation scheme to decompose the image into depth consistent unites, and then restore clean video via a robust deep CNN. Liu *et al.* present a recurrent neural network to classify all pixels in rain frames, remove rain and reconstruct background details in , and introduce a dynamic routing residue recurrent network to integrate their proposed hybrid rain model in [Liu et al., 2018a]. In order to make use of the additional degradation factors in the real world, [Yang et al., 2019b] build a two-stage

recurrent network to firstly capture motion information and then keep the motion consistency between frames to remove rain. There also exists self-learning deep video deraining method [Yang et al., 2020b], which can learn how to remove rain without pairs of training samples. [Yue et al., 2021] design a dynamical rain generator for semi-supervised video deraining. Specifically, this method represents the sequence of rain layers in rain videos using the dynamical rain generator, which is able to facilitate the rain removal task. A semi-supervise learning manner is proposed to handle the generalization issue for real cases.

Although the above deep deraining methods achieve great success in video deraining, most of them focus on the performance and ignore the computational time.

2.2.4 Stereo Deraining

Stereo images provide more information from cross views and thus have been utilized to improve the performance of various computer vision tasks, including traditional problems [Godard et al., 2017; Eslami et al., 2016; Luo et al., 2016] and novel tasks [Jeon et al., 2018; Li et al., 2018a; Chen et al., 2018a; Zhou et al., 2019b].

However, there are few methods that leverage the stereo images to remove rain so far. The aim of stereo deraining methods is to generate clean stereo images based on their corresponding rainy versions [Zhang et al., 2020a]. [Tanaka et al., 2006] remove the rain via utilizing disparities between stereo images to detect positions of noises and estimate true disparities of images regions hidden by rain. In order to obtain the deraining left-view images, [Kim et al., 2014] warp the spatially adjacent right-view frames and subtract warped frames from the original frames. However, these traditional methods do not consider the importance of semantic information. Meanwhile, the strong capability of learning features implied in deep neural networks is also ignored by them.

2.3 Quality Assessment

Methods for image quality assessment (IQA) can be classified into subjective and objective metrics. Subjective approaches are based on human judgment, which may not require a reference image. One representative metric is the Mean Opinion Score (MOS) [Hoßfeld et al., 2016], where people rate the quality of images on a scale of 1-5. MOS values vary based on different opinions, and methods relying on these scores typically take the statistics of opinion scores into account. For image deblurring and deraining, most existing methods are evaluated on objective assessment scores, which can be further split into two categories: full-reference and no-reference IQA metrics.

Full-Reference Metrics. Full-reference metrics assess the image quality by comparing the restored image with the ground-truth (GT). Such metrics include PSNR [Hore and Ziou, 2010], SSIM [Wang et al., 2004], WSNR [Mitsa and Varkur, 1993], MS-SSIM [Wang et al., 2003b], IFC [Sheikh et al., 2005], NQM [Damera-Venkata et al.,

2000], UIQI [Wang and Bovik, 2002] and VIF [Sheikh and Bovik, 2006]. Among these, PSNR and SSIM are the most commonly used metrics in image restoration tasks [Nah et al., 2017a; Kupyn et al., 2018; Zhang et al., 2018a; Tao et al., 2018; Gao et al., 2019; Kupyn et al., 2019; Shen et al., 2019; Zhang et al., 2020c; Suin et al., 2020; Zhang et al., 2020a, 2021c,b]. On the other hand, LPIPS and E-LPIPS Zhang et al. [2018c] are able to accurately predict the human judgment of image quality.

No-Reference Metrics. While the full-reference metrics require a ground-truth image for evaluation, no-reference metrics use only the deblurred images to measure the quality. To evaluate the performance of deblurring and deraining methods on real-world images, several no-reference metrics have been used, such as BIQI [Moorthy and Bovik, 2010], BLINDS2 [Saad et al., 2012], BRISQUE [Mittal et al., 2012a], CORNIA [Ye et al., 2012], DIIVINE [Moorthy and Bovik, 2011], NIQE [Mittal et al., 2012b], and SSEQ [Liu et al., 2014]. Further, a number of metrics have been developed to evaluate the performance of image deblurring and deraining algorithms by comparing the accuracy on different vision tasks such as object detection and recognition [Li et al., 2018c; Yasarla et al., 2019].

Chapter 3

Deblurring: Deblurring via Realistic Blurring

This chapter is about single image deblurring. Existing deep learning methods for image deblurring typically train models using pairs of sharp images and their blurred counterparts. However, synthetically blurring images does not necessarily model the blurring process in real-world scenarios with sufficient accuracy. To address this problem, we propose a new method which combines two GAN models, *i.e.*, a learning-to-Blur GAN (BGAN) and learning-to-DeBlur GAN (DBGAN), in order to learn a better model for image deblurring by primarily learning how to blur images. The first model, BGAN, learns how to blur sharp images with unpaired sharp and blurry image sets, and then guides the second model, DBGAN, to learn how to correctly deblur such images. In order to reduce the discrepancy between real blur and synthesized blur, a relativistic blur loss is leveraged. As an additional contribution, this chapter also introduces a Real-World Blurred Image (RWBI) dataset including diverse blurry images. Our experiments show that the proposed method achieves consistently superior quantitative performance as well as higher perceptual quality on both the newly proposed dataset and the public GOPRO dataset.

3.1 Introduction

Given a blurred image, which is corrupted by some unknown blur kernel or a spatially variant kernel, the task of (blind) single image deblurring is to recover the sharp version of the original image, by reducing or removing the undesirable blur in the image. Traditional deblurring methods handle this problem via estimating a blur kernel, through which a sharp version of the blurred input image can be recovered. Often, special characteristics of the blur kernel are assumed, and natural image priors are exploited in the deblurring process [Cho and Lee, 2009; Goldstein and Fattal, 2012; Pan et al., 2014; Xu and Jia, 2010; Xu et al., 2013]. However, estimating the optimal blur kernel is a difficult task and can therefore impair the overall performance.

Recently, deep learning methods, particularly convolutional neural networks (CNNs), have been applied to tackle this task and obtained a remarkable success, *e.g.*, [Nah

et al., 2017a; Su et al., 2017b; Tao et al., 2018; Zhang et al., 2018a]. Existing deep learning methods focus on training deblurring models using *paired* blurry and sharp images. For example, [Nah et al., 2017a] propose a multi-scale loss function to implement a coarse-to-fine processing pipeline. [Tao et al., 2018] and [Gao et al., 2019] improve the work by using shared network weights among different scales, achieving state-of-the-art performance.

However, many common effects are not adequately captured by the current deep learning models in the following sense.

First, since in real-world scenarios, an image is captured during a time window (*i.e.*, the exposure duration), the blurred image is in fact the integration of multi-frame instant and sharp snapshots [Hirsch et al., 2011]. This can be formulated as

$$I_B = g\left(\frac{1}{T}\int_{t=0}^{T} I_{S(t)}\mathrm{d}t\right), \tag{3.1}$$

where I_S is an instant sharp frame and I_B is the blurry image. T is the exposure time period and $g(\cdot)$ is the Camera Response Function (CRF). In contrast, in conventional deblurring methods, blurry images used in the training set are often artificially synthesized by approximating the integration step with a simple averaging operation, as shown in Eq. (3.2), where M is the number of frames:

$$I_B \simeq g\left(\frac{1}{M}\sum_{t=0}^{M-1} I_{S[t]}\right). \tag{3.2}$$

Prior methods use M sharp frames $I_{S[t]}$ to replace the continuous sequence $I_{S(t)}$ and generate paired training data, avoiding the complexity of obtaining pairs of real blurry and sharp images. However, there is a clear gap between real blurry images and those artificially blurred images. Based on the Eq. 3.1 and 3.2, a real blurry image can be regarded as the averaging of infinite frames captured in the exposure time period, while the artificially blurry images are synthesized by M frames, whose number is fewer.

Second, in real situations there are multi-fold factors (not limiting to a single linear integration or summation) which can cause image blurs, for instance, camera shake, fast object motion, and small aperture with a wide depth of field. Many of these factors are very difficult to model accurately because current researchers still do not find some mathematical formulas which can model the same blurring process as the real world. To design a better deblurring algorithm, all these factors should be taken into consideration. If the real blurred images are different from the samples in the training set, the trained model may not perform well on the testing data. This observation inspires us to develop a new deblurring method which does not assume any particular blur type; rather such a method will be able to learn blurring process in order to achieve better deblurring quality.

Specifically, we propose a method which contains a leaning-to-Blur GAN (BGAN) module and a learning-to-DeBlur GAN (DBGAN) module. BGAN and DBGAN are two complementary processes, in the sense that BGAN learns to mimic properties of

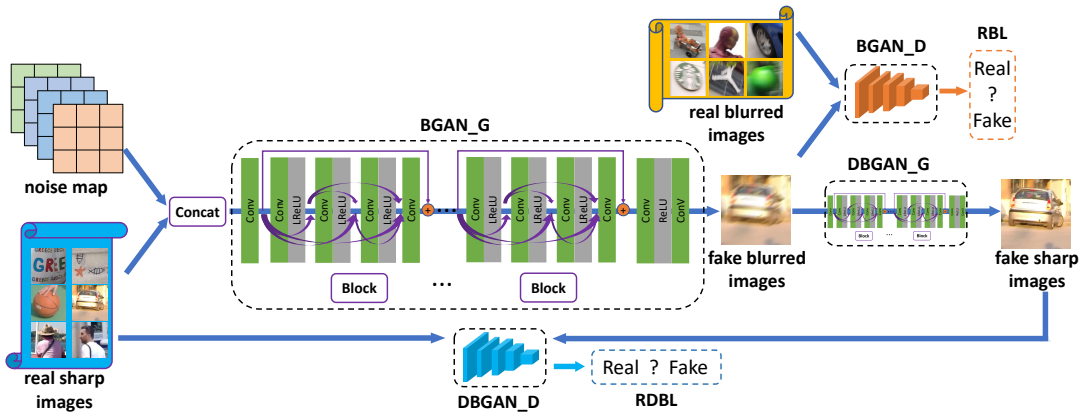


Figure 3.1: **The proposed framework and training process.** This framework contains two main modules, a BGAN and a DBGAN. *D* and *G* denote discriminator and generator networks, respectively. The BGAN takes sharp images as input and outputs realistic blurry images, which are then fed into the DBGAN in order to learn to deblur. During the testing stage, only the DBGAN is applied.

real-world blurs by generating photo-realistic blurry images. This module is trained using unpaired sharp and blurry images, thus relaxing the requirement of needing paired data. Recently, [Shaham et al., 2019] propose SinGAN to produce different images based on random noises, which inspires us to generate various blurry images given different noises. During the generation, sharp images are also fed into BGAN to make the generated blurry images have the same content as the input images. The DBGAN module learns to recover sharp images from blurry images with real sharp and generated blurry images. We further employ a relativistic blur loss, which helps predict the probability that a real blurry image is relatively more realistic than a synthesized one. Finally, a Real-World Blurry Image (RWBI) dataset is created to help train the BGAN model and evaluate the performance of our proposed image deblurring model.

3.2 Deblurring by Blurring

3.2.1 Overall Architecture

Our framework contains two primary modules. Similar to prior image deblurring works, our framework includes a learning-to-DeBlur GAN (DBGAN) module, which is trained on paired sharp and blurry images to recover sharp images from blurry images. The paired sharp-blurry images are obtained from the BGAN module. The BGAN is trained on unpaired data, where sharp images come from a public dataset, while the blurry images come from a new real-world blurry dataset. Fig. 3.1 shows the overall architecture of the proposed framework.

We further enhance the standard GAN model with a relativistic blur loss. In traditional GAN-based models for image deblurring, the discriminator *D* estimates

the probability that the input data is real, and the generator G is trained to increase the probability that the generated data looks real. The developed relativistic blur loss estimates the probability that the given real-world blurry images are more realistic than the generated blurry images.

In the training stage, sharp images are input into the BGAN generator and its output is fed into the DBGAN to learn how to deblur. The generators in the DBGAN and BGAN modules generate corresponding images, and the discriminators conduct discrimination to create more realistic synthetic images. During the testing stage, only the DBGAN generator network is required for the image deblurring task.

3.2.2 BGAN: Learning to Blur

The BGAN module is the primary difference from other neural network based methods for image deblurring. Similar to other GAN based models, the BGAN consists of a generator network and a discriminator network. In this section, we first discuss its architecture and loss functions.

BGAN Generator. The input to the BGAN generator is a sharp image from a public dataset. Given the numerous possible factors that can cause undesired blurring artifacts, we concatenate the input image with a noise map to model the different conditions. To obtain the noise map, we sample a noise vector of length 4 from a normal distribution and duplicate it 128×128 times in the spatial dimension to obtain a $4 \times 128 \times 128$ noise map as in [Zhu et al., 2017b]. In this way, we can generate various blurry images based on one sharp image. The network architecture consists of one convolutional layer, 9 residual blocks (ResBlocks) [He et al., 2016] and another two convolutional layers. Each ResBlock consists of 5 convolutional layers ($64 \times 3 \times 3$) and 4 ReLU activations. There is also a skip connection in each ResBlock, connecting the input and output features (refer to Fig. A.6). The output of our BGAN generator is a blurry image of the same size as the sharp input image.

BGAN Discriminator. The input to the BGAN discriminator is the output of the BGAN generator. Its architecture is same as the VGG19 network [Simonyan and Zisserman, 2015a], and its output is the probability of the blurry image being classified as real.

BGAN Loss. The generator and discriminator of the BGAN are trained with a perceptual loss and an adversarial loss. Specifically, the perceptual loss is calculated based on the synthesized blurry images and real sharp images taken from a public dataset. The adversarial loss is calculated between the synthesized and real blurry images. The real blurry images are taken from our newly created dataset.

3.2.3 DBGAN: Learning to Deblur

The BGAN module aims to mimic the real-world blurry images and cover as many blur cases as possible. Its goal is to drive the DBGAN module to be more effective in recovering sharp images from blurry images. In the following, we present the architecture and loss of the DBGAN.

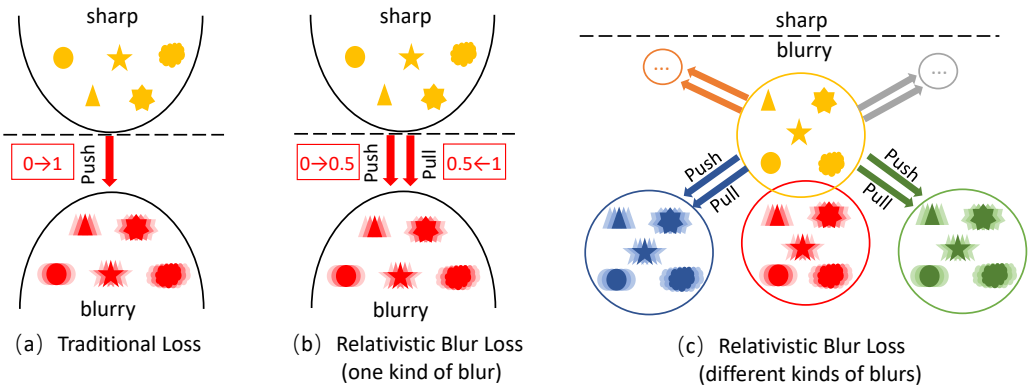


Figure 3.2: **An illustration of the Relativistic Blur Loss (RBL).** Real and synthesized images are labeled as 1 and 0, respectively. (a) A traditional loss function is used to update the generator to create blurry images (label=0) which are similar to real ones (label=1). (b) The RBL not only increases the probability that generated images look real ($0 \rightarrow 0.5$, which is labeled as "Push"), but also simultaneously decreases the output probability that real images are real ($1 \rightarrow 0.5$, which is labeled as "Pull"). (c) In order to increase the variations of blurry images, different blurry images are used to model the different types of blurs in the real world.

DBGAN Generator. The input to the DBGAN generator is a blurry image. Many approaches have been proposed for this task [Chakrabarti, 2016; Nah et al., 2017a; Sun et al., 2015; Tao et al., 2018]. When we design the DBGAN generator, we adopt their advantages. Specifically, we remove the batch normalization layers, which have been shown to increase the computational complexity and decrease the performance on different tasks [Nah et al., 2017a]. Secondly, we use additive residual layers in each block, which combine multi-level residual networks and dense connections [Huang et al., 2017]. The BGAN consists of one convolutional layer, 16 residual blocks (ResBlocks) [He et al., 2016] and another two convolutional layers. The kernel size in ResBlocks is $63 \times 3 \times 3$. The details can be referred to Fig. A.6. The output of the DBGAN generator is the desired sharp image.

DBGAN Discriminator. Similar to the BGAN discriminator, the DBGAN also adopts the VGG19 network [Simonyan and Zisserman, 2015a] as its discriminator. The output of this model is the probability of the given sharp images looking realistic.

DBGAN Loss. Like the BGAN module, the proposed DBGAN model is trained using a perceptual loss and an adversarial loss. We also use an L_1 loss to update the DBGAN. All the three types of loss functions are calculated based on the generated and real sharp images, so the DBGAN is trained on paired images.

3.2.4 Relativistic Blur Loss

In this section, we describe a Relativistic Blur Loss (RBL) and other loss functions which are used to train our framework.

Perceptual Loss. In contrast to previous image deblurring methods [Nah et al.,

2017a; Tao et al., 2018], the proposed framework applies a perceptual loss $\mathcal{L}_{perceptual}$ to update models. Note that [Johnson et al., 2016] use a similar loss. However, in contrast to their work, we calculate the perceptual loss based on features before rather than after the ReLU activation layer.

Content Loss. The Mean Squared Error (MSE) is widely used as a loss function for image restoration methods. Based on the MSE, the content loss between ground-truth and generated images is calculated.

Relativistic Blur Loss. In order to drive the BGAN generator to produce blurry images similar to the real-world images, we develop a relativistic blur loss based on [Jolicoeur-Martineau, 2019] to update the model. The BGAN generator parameters are updated in order to fool the BGAN discriminator. The adversarial loss D is formulated as:

$$\begin{aligned} D(I^{real}_{blurry}) &= \sigma(C(I^{real}_{blurry})) \rightarrow 1, \\ D(I^{fake}_{blurry}) &= D(G(I^{real}_{sharp})) = \sigma(C(G(I^{real}_{sharp}))) \rightarrow 0\,, \end{aligned} \tag{3.3}$$

where $D(\cdot)$ is the probability that the input is a real image. $C(\cdot)$ is the feature representation before activation and $\sigma(\cdot)$ is the sigmoid function. The generator G is trained to increase the probability that synthesized images are real. Real and synthesized images are labeled as 0 or 1 by D, respectively. As Fig. 3.2 (a) shows, the effect of G is to transfer real sharp images to blurry images and "push" these generated images (label=0) closer to real blurry images (label=1). However, during the training stage, only the second part of Eq. (3.3), i.e., , $D(I^{fake}_{blurry}) = D(G(I^{real}_{sharp})) \rightarrow 0$, updates the parameters of generator G, while the first part is used to update the discriminator D model rather than generator G [Nah et al., 2017a]. In fact, a powerful generator G should also decrease the probability that real blurry images are real. This is because a realistic synthesized image labeled as fake is similar to real one, and will thus fool the D model to learn to distinguish real or fake in the training stage. Based on this idea, we add $D(I^{real}_{blurry})$ into the process of learning G in BGAN. Specially, a Relativistic Blur Loss (RBL) is developed to help calculate whether a real blurry image is more realistic than the synthesized blurry image. The formulation of Eq. 3.3 is modified to

$$\begin{aligned} D(I^{real}_{blurry}) - E(C(G(I^{fake}_{blurry}))) &\rightarrow 1\,, \\ D(G(I^{fake}_{blurry})) - E(C(I^{real}_{blurry})) &\rightarrow 0\,, \end{aligned} \tag{3.4}$$

where $E(\cdot)$ denotes the averaging operation over images in one batch. Fig. 3.2 (b) shows the aim of RBL. Although the goal is still to generate realistic blurry images which are similar to real-world ones, the optimization objective is different. RBL aims to update G to generate synthetic images which are near to 0.5, and meanwhile to fool the D model, making it difficult to distinguish real images from fake ones. In this way, the probability of real blurry images predicted by D is also near to 0.5. We

term the effects as "push" and "pull", respectively, which can complement each other to update the generator G. As Fig. 3.2 shows, the sharp and blurry images can be regarded as two different domains. In order to rapidly generate blurry images and utilize prior research results of generating blurry images, we first train our BGAN model with artificial blurry images as Fig. 3.2(b) shows. We then add other types of blurry images to increase the variations of the produced blurry images based on Eq. 3.4 to cover different conditions in the real world, which is shown in Fig. 3.2(c).

Based on Eq. 3.4 and Fig. 3.2, our RBL, which is used in the BGAN generator, can be represented as

$$\mathcal{L}_{RBL} = -[\log(D(I^{real}_{blurry}) - E(C(G(I^{input})))) + \log(1 - (D(G(I^{input})) - E(C(I^{real}_{blurry}))))]. \tag{3.5}$$

Based on the RBL, we apply a Relativistic Deblur loss (RDBL) in DBGAN generator as

$$\mathcal{L}_{RDBL} = -[\log(D(I^{real}_{sharp}) - E(C(G(I^{input})))) + \log(1 - (D(G(I^{input})) - E(C(I^{real}_{sharp}))))]. \tag{3.6}$$

Balance of Different Loss Functions. During the training stage, the loss functions for DBGAN and BGAN are combinations of different terms using a weighted fusion,

$$\mathcal{L}_{BGAN} = \mathcal{L}_{perceptual} + \beta \cdot \mathcal{L}_{RBL}, \tag{3.7}$$

$$\mathcal{L}_{DBGAN} = \mathcal{L}_{perceptual} + \alpha \cdot \mathcal{L}_{content} + \beta \cdot \mathcal{L}_{RDBL}\,. \tag{3.8}$$

In order to balance the different kinds of losses, we use two hyper-parameters α and β to yield the final loss $\mathcal{L}$ for BGAN and DBGAN.

3.3 Experiments

We test our approach on the widely used public GOPRO dataset [Nah et al., 2017a] and our developed Real-World Blurry Image (RWBI) dataset, which are introduced firstly. Then the implementation details of our work are presented. Comparison with the state-of-the-art methods is reported in the following subsection, and the application in real-world scenarios is demonstrated finally.

3.3.1 Datasets

GOPRO Dataset. We evaluate the performance of our model on the public GOPRO dataset [Nah et al., 2017a], which contains 3,214 image pairs. The training and testing sets include 2,103 and 1,111 pairs, respectively. Existing methods convolve sharp images with a blur kernel [Chakrabarti, 2016; Schuler et al., 2016; Sun et al., 2015] to synthesize blurry images. These synthetic blurry images are different from real ones captured by camera. In order to model more realistic blurry conditions, in the GOPRO dataset, sharp images with a high-speed camera and synthesize blurry images were collected by averaging these sharp images from videos.

Figure 3.3: **Synthesized blurry images.** Examples of different blurry images created by the proposed BGAN with different random noise maps. The first column shows input sharp images, and the following three columns are the produced blurred images used to train the DBGAN.

RWBI Dataset. In order to train our BGAN model and evaluate the performance of deblurring models, we collect a Real-World Blurry Image dataset. The blurry images are captured with different hand-held devices, including an iPhone XS, a Samsung S9 Plus, a Huawei P30 Pro and a GoPro Hero 5 Black. Multiple devices are used to reduce bias towards one specific device which may capture blurry images with unique characteristics. The dataset contains 22 different sequences of $3,112$ diverse blurry images.

We compare the performance of the proposed method with the state-of-the-art methods on the public GOPRO dataset quantitatively and qualitatively. As there is not ground truth of the developed RWBI dataset, we only conduct a qualitative comparison.

3.3.2 Implementation Details

When training BGAN and DBGAN, we use a Gaussian distribution with zero mean and a standard deviation of 0.01 to initialize the weights. In each iteration, we update all the weights after learning a mini-batch of size 4. To augment the training set, we crop a 128×128 patch at any location of an image. To further increase the number of training samples, we also randomly flip frames. The two modules are trained without an adversarial loss at first. We use a learning rate annealing scheme, starting with a value of 10^{-4} and reducing it to 10^{-6} after the training loss gets converged. We train for $2,000$ epochs, and subsequently add the adversarial loss functions to fine-tune the modules using a learning rate of 10^{-6} for 500 further epochs. The hyper-parameters α and β are set as 0.005 and 0.01, respectively.

Figure 3.4: **Qualitative ablation results.** Examples of deblurred images generated by the proposed framework with different model structures. The first column shows input blurred images, and the following three columns are the deblurred images produced by DBGAN(-), DBGAN and DBGAN(+), respectively.

Table 3.1: **Performance for different model structures on the *GOPRO_Large* dataset.**

Methods	DBGAN (-)	DBGAN	DBGAN(+)
PSNR	30.23	30.43	31.10
SSIM	0.9346	0.9372	0.9424

3.3.3 Ablation Study

In this section, we conduct experiments to investigate the effectiveness of different components of our model. The proposed model has three variants:

(1) **DBGAN** is the model for learning to deblur. Its input is a blurry image and the output is a deblurred image. Similar to previous GAN-based deblurred methods [Nah et al., 2017a; Kupyn et al., 2018], this model contains generator and discriminator networks. Thus its loss function is a combination of $\mathcal{L}_{percetpual}$, $\mathcal{L}_{content}$ and $\mathcal{L}_{RDBL}$ with weights α and β. The final loss function is shown in Eq. (3.8).

Table 3.2: **Performance comparison on the *GOPRO_Large* dataset.**

Method	Kim *et al.*	Sun *et al.*	Nah *et al.*	Tao *et al.*	Zhang *et al.*	Gao *et al.*	**DBGAN**	**DBGAN(+)**
PSNR	23.64	24.64	29.08	30.10	30.90	30.92	**30.43**	**31.10**
SSIM	0.8239	0.8429	0.9135	0.9323	0.9419	0.9421	**0.9372**	**0.9424**

Figure 3.5: **Comparison with state-of-the-art deblurring methods.** From left to right: blurry images, results of [Nah et al., 2017a], [Tao et al., 2018] and the proposed DBGAN(+) method. The improvement is clearly visible in the magnified patches. More qualitative comparison results can be accessed in the supplementary material.

(2) **DBGAN(-)** has the same architecture as DBGAN. Differently, we replace the $\mathcal{L}_{RDBL}$ with a traditional adversarial loss as [Nah et al., 2017a]. Namely, the training process does not contain the relativistic loss functions. It is trained based on $\mathcal{L}_{percetpual}$, $\mathcal{L}_{content}$ and the traditional adversarial loss.

(3) **DBGAN(+)** is our full method. It has a similar architecture to DBGAN with the main difference of additionally employing the BGAN module during the training stage. The BGAN module generates more realistic blurry images to enhance the learning performance of DBGAN. Fig. 3.3 shows the examples of different blurry images produced by the proposed BGAN with different input noises.

Table 3.1 shows results of the quantitative comparison. The proposed DBGAN outperforms the DBGAN(-) in terms of both PSNR and SSIM values, which shows the effectiveness of the relativistic loss function for image deblurring. With the learning-to-blur module, DBGAN(+) achieves a further improvement over DBGAN, suggesting the benefits of *learning to deblur by learning to blur*. Fig. 9.6 presents exemplar qualitative results from different model structures, also revealing the advantage of DBGAN(+) over its counterparts.

Figure 3.6: **Performance comparison on real-world blurry images.** From left to right: blurry images, results of [Nah et al., 2017a], [Tao et al., 2018] and the proposed DBGAN(+) method. The improvement is clearly visible in the magnified patches.

3.3.4 Comparison with Existing Methods

To verify the effectiveness of our model, we compare its performance with several state-of-the-art approaches on the GOPRO dataset quantitatively and qualitatively. [Hyun Kim et al., 2013] by Kim *et al.* is a traditional method to handle complex dynamic blurring images. For deep learning methods, [Sun et al., 2015] use a CNN network to estimate blur kernels and apply traditional deconvolution methods to synthesize sharp images. [Nah et al., 2017a] propose a multi-scale function to model the coarse-to-fine approach. Similar to [Nah et al., 2017a], [Tao et al., 2018] propose a multi-scale network via sharing network weights between different scales to recover sharp images. In addition, [Zhang et al., 2019a] introduce a VMPHN model and [Gao et al., 2019] propose a nested skip connection structure and achieve state-of-the-art performance. Table 3.2 shows the results of the quantitative comparison. DBGAN outperforms most of previous methods, while DBGAN(+) achieves the state-of-the-art performance due to the framework of learning to deblur by learning to blur. For fair comparison, all values refer to the performance achieved by single model trained on the GOPRO dataset. Qualitative comparisons with some state-of-the-art methods are shown in Fig. 3.5, demonstrating that our method consistently achieves

better visual quality results. Please also refer to our supplementary material for more qualitative comparison results.

3.3.5 Performance in Real-World Scenarios

To validate the effectiveness of our method, we compare the performance of our approach with several state-of-the-art methods on the RWBI dataset of real-world blurry images. Fig. 3.6 shows qualitative results of different models. The blurry images in the first column are from the RWBI dataset, and the images in the following columns are the results of [Nah et al., 2017a], [Tao et al., 2018] and the proposed DBGAN(+). Fig. 3.6 shows that our method achieves better performance on real-world blurry images.

3.4 Conclusion

The main contribution of this chapter is that we present a new framework which firstly learns how to transfer sharp images to realistic blurry images via a learning-to-blur GAN (BGAN) module. The framework trains a learning-to-deblur GAN (DBGAN) module to learn how to recover a sharp image from a blurry image. In contrast to prior work which solely focuses on learning to deblur, our method learns to realistically synthesize blurring effects using unpaired sharp and blurry images. In order to generated more realistic blurred images, a relativistic blur loss is employed to help the BGAN module reduce the gap between synthesized blur and real blur. In addition, a RWBI dataset is built to help train and test deblurring models. Experimental results have demonstrated that our method not only produces results of consistently higher perceptual quality, but also outperforms state-of-the-art methods quantitatively.

Chapter 4

Deblurring: Adversarial Spatio-Temporal Learning for Video Deblurring

This chapter is about video deblurring. Camera shake or target movement often leads to undesired blur effects in videos captured by a hand-held camera. Despite significant efforts having been devoted to video-deblur research, two major challenges remain: 1) how to model the spatio-temporal characteristics across both the spatial domain (i.e., image plane) and temporal domain (i.e., neighboring frames), and 2) how to restore sharp image details w.r.t. the conventionally adopted metric of pixel-wise errors. In this chapter, to address the first challenge, we propose a *DeBLuRring Network (DBLRNet)* for spatial-temporal learning by applying a 3D convolution to both spatial and temporal domains. Our DBLRNet is able to capture jointly spatial and temporal information encoded in neighboring frames, which directly contributes to improved video deblur performance. To tackle the second challenge, we leverage the developed DBLRNet as a generator in the GAN (generative adversarial network) architecture, and employ a content loss in addition to an adversarial loss for efficient adversarial training. The developed network, which we name as *DeBLuRring Generative Adversarial Network (DBLRGAN)*, is tested on two standard benchmarks and achieves the state-of-the-art performance.

4.1 Introduction

Videos captured by hand-held cameras often suffer from unwanted blurs either caused by camera shake [Kang, 2007], or object movement in the scene [Sun et al., 2015; Shi et al., 2015]. The task of video deblurring aims at removing those undesired blurs and recovering sharp frames from the input video. This is an active research topic in the applied fields of computer vision and image processing. Applications of video deblurring are found in many important fields such as 3D reconstruction [Seok Lee and Mu Lee, 2013], SLAM [Lee et al., 2011] and tracking [Jin et al., 2005].

In contrast to single image deblurring, video deblurring is a relatively less tapped

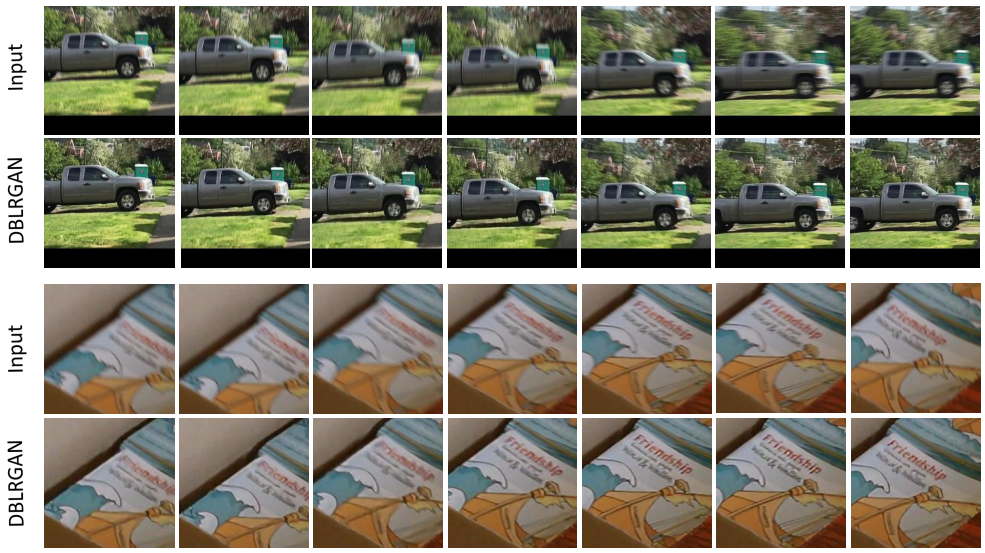


Figure 4.1: **Deblurring results of the proposed DBLRGAN on real-world video frames**. The first and third rows show crops of consecutive frames from the VideoDeblurring dataset. The second and fourth rows show corresponding deblurring results of DBLRGAN.

task until recently. And video deblurring is more challenging, partly because it is not entirely clear about how to model and exploit the inherent temporal dynamics exhibited among continuous video frames. Moreover, the commonly adopted performance metric, namely, pixel-wise residual error, often measured by PSNR, is questionable, as it fails to capture human visual intuitions of how sharp or how realistic a restored image is [Wang et al., 2003b, 2004]. We plan to leverage the recent advance of the adversarial learning technique to improve the performance of video deblurring.

One key challenge for video deblurring is to find an effective way to capture spatio-temporal information existing in neighboring image frames. Deep learning based methods have recently witnessed a remarkable success in many applications including image and video denoising and deblurring. Previous deep learning methods are however primarily based on 2D convolutions, mainly for computational sake. Yet, it is not natural to use 2D convolutions to capture spatial and temporal joint information, which is essentially in a 3-D feature space. We propose a deep neural network called DeBLuRing Network (DBLRNet), which uses 3D (volumetric) convolutional layers, as well as deep residual learning, aims to learn feature representations both across temporal frames and across image plane.

As noted above, we argue that the conventional pixel-wise PSNR metric is insufficient for the task of image/video deblurring. To address this issue, we resort to adversarial learning, and propose DeBLuRring Generative Adversarial Network (DBLRGAN). DBLRGAN consists of a generative network and a discriminate network, where the ***generative*** network is the aforementioned DBLRNet which restores sharp images, and the ***discriminate*** network is a binary classification network, which

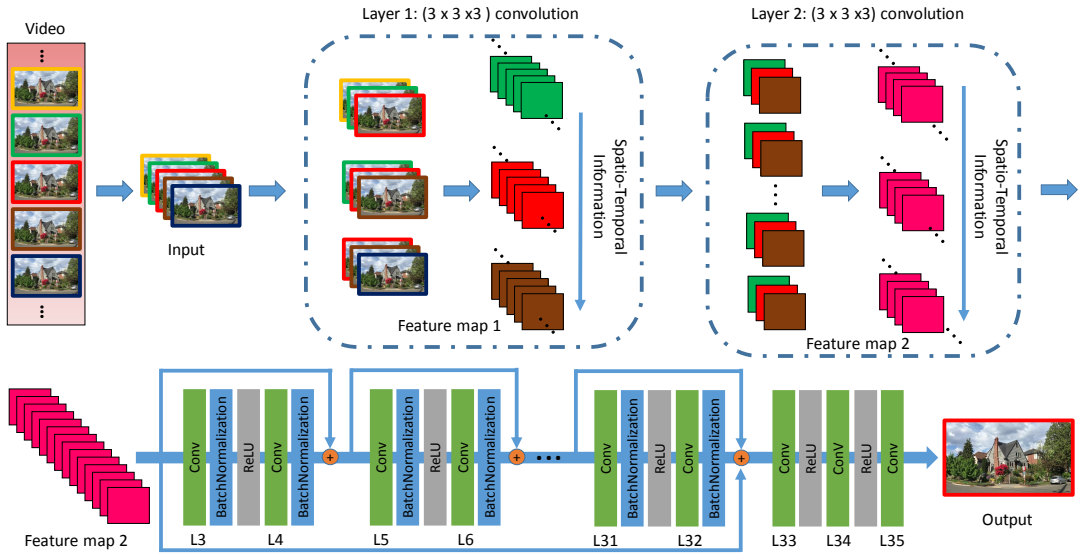


Figure 4.2: **The proposed DBLRNet framework.** The input to our network is five time-consecutive blurry frames. The output is the central deblurred frame. By performing 3D convolutions, this model learns joint spatial-temporal feature representations.

tells a restored image apart from a real-world sharp image.

We introduce a training loss which consists of two terms: content loss and adversarial loss. The content loss is used to respect the pixel-wise measurement, while the adversarial loss promotes a more realistically looking (hence sharper) image. Training DBLRGAN in an end-to-end manner, we recover sharp video frames from a blurred input video sequence, with some examples shown in Figure 4.1.

4.2 Our Model

Overview. In this section, we first introduce our DBLRNet, and then present the proposed network DBLRGAN which is on the basis of DBLRNet. Finally we detail the two loss functions (content and adversarial losses) which are used in the training stage. Both the DBLRNet and DBLRGAN are end-to-end systems for video deblurring. Note that, blurry frames can be put into our proposed models without alignment.

4.2.1 DBLRNet

In 2D CNN, convolutions are applied on 2D images or feature maps to learn features in spatial dimensions only. In case of video analysis problems, it is desirable to consider the motion variation encoded in the temporal dimension, such as multiple neighboring frames. We propose to perform 3D convolutions [Ji et al., 2013] the convolution stages of deep residual networks to learn feature representations from

both spatial and temporal dimensions for video deblurring. We operate the 3D convolution via convolving 3D kernels/filters with the cube constructed from multiple neighboring frames. By doing so, the feature maps in the convolution layers can capture the dynamic variations, which is helpful to model the blur evolution and further recover sharp frames.

Formally, the 3D convolution operation is formulated as:

$$V_{ij}^{xyz} = \sigma(\sum_{m} \sum_{p=0}^{P_i-1} \sum_{q=0}^{Q_i-1} \sum_{r=0}^{R_i-1} V_{(i-1)m}^{(x+p)(y+q)(z+r)} \cdot g_{ijm}^{pqr} + b_{ij}), \tag{4.1}$$

where V_{ij}^{xyz} is the value at position (x, y, z) in the j-th feature map of the i-th layer, (P_i, Q_i, R_i) is the size of 3D convolution kernel. Q_i responds to the temporal dimension. g_{ijm}^{pqr} is the (p, q, r)-th value of the kernel connected to the m-th feature map from the $(i-1)$-th layer. $\sigma(\cdot)$ is the ReLU nonlinearity activation function, which is shown to lead to better performance in various computer vision tasks than other activation functions, e.g. Sigmoid and Tanh.

Defining 3D convolution, we propose a model called DBLRNet, which is shown in Figure 4.2. DBLRNet is composed of two $3 \times 3 \times 3$ convolutional layers, several residual blocks [He et al., 2016], each containing two convolution layers, and another five convolutional layers. This architecture is designed inspired by the Fully Convolutional Neural Network (FCNN) [Long et al., 2015], which is originally proposed for semantic segmentation. Different from FCNN and DBN [Su et al., 2017a], spatial size of feature maps in our model keeps constant. Namely, there is not any down-sampling operation nor up-sampling operation in our DBLRNet. The detailed configurations of DBLRNet is given in Table 4.1.

As Figure 4.2 shows, the input to DBLRNet is five consecutive frames. Note that we does not conduct deblurring in the original RGB space. Alternatively, we conduct deblurring on basis of gray-scale images. Specifically, the RGB space is transformed to the YCbCr space, and the Y channel is adopted as input since the illumination is the most salient one. We perform 3D convolutions with kernel size of $3 \times 3 \times 3$ (3×3 is the spatial size and the last 3 is for the temporal dimension) in the first and second convolutional layers. To be more specific, in layer 1, three groups of consecutive frames are convolved with a set of 3D kernels respectively, resulting in three groups of feature maps. These three groups of feature maps are convolved with 3D filters again to obtain higher-level feature maps. In the following layers, the size of convolution kernels is $3 \times 3 \times 1$ due to the decrease of temporal dimensions. The stride and padding are set to 1 in every layer. The output of DBLRNet is the deblurred central frame. We transform the gray-scale output back to colorful images with the original Cb and Cr channels.

4.2.2 DBLRGAN

GAN is proposed to train generative parametric models by [Goodfellow et al., 2014]. It consists of two networks: a generator network G and a discriminator network D. The goal of G is to generate samples, trying to fool D, while D is trained to dis-

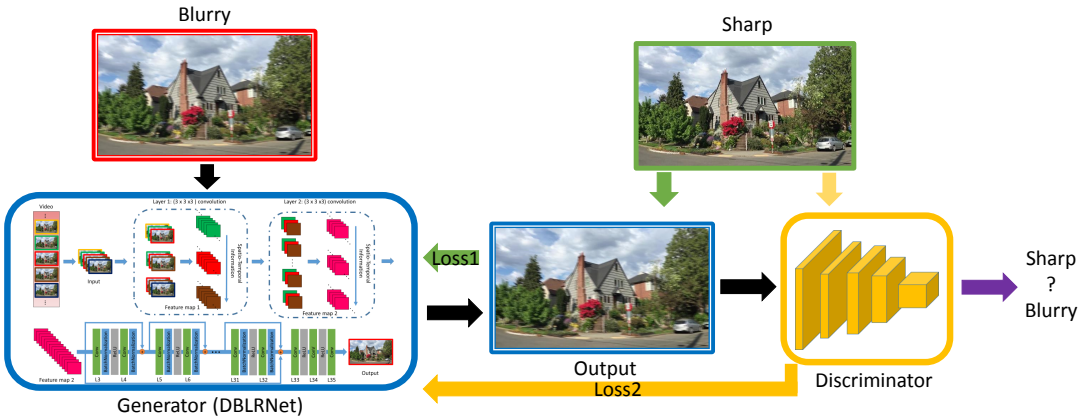


Figure 4.3: **The DBLRGAN framework for video deblurring.** The architecture consists of a Generator and a Discriminator. The Generator is our proposed DBLRNet, while the Discriminator is a VGG-like CNN net.

tinguish generated samples from real samples. Inspired by the adversarial training strategy, we propose a model called DeBLuRring Generative Adversarial Network (DBLRGAN), which utilizes G to deblur images and D to discriminate deblurred images and real-world sharp images. Ideally, the discriminator can be fooled if the generator outputs sharp enough image.

Following the formulation in [Goodfellow et al., 2014], solving the deblurring problem in the generative adversarial framework leads to the following min-max optimization problem:

$$\min_G \max_D V(G,D) = \mathrm{E}_{h\sim p_{train(h)}}[\log(D(h))] + \mathrm{E}_{\hat{h}\sim p_{G(\hat{h})}}[\log(1-D(G(\hat{h})))], \tag{4.2}$$

where h indicates a sample from real-world sharp frames and $\hat{h}$ represents a blurry sample. G is trained to fool D into misclassifying the generated frames, while D is trained to distinguish deblurred frames from real-world sharp frames. G and D models are trained alternately, and our ultimate goal is to train a model G that recovers sharp frames given blurry frames.

As shown in Figure 4.3, we use the proposed DBLRNet (Figure 4.2 and Table 4.1) as our G model, and build a CNN model as our D model, following the architectural guidelines proposed by [Radford et al., 2016]. This D model is similar to the VGG network [Simonyan and Zisserman, 2015a]. It contains 14 convolutional layers. From bottom to top, the number of channels of the convolutional kernels increases from 64 to 512. Finally, this network is trained via a two-way soft-max classifier at the top layer to distinguish real sharp frames from deblurred ones. For more detailed configurations, please refer to Table 4.2.

Table 4.1: **Configurations of the proposed DBLRNet.** It is composed of two convolutional layers (L1 and L2), 14 residual blocks, two convolutional layers (L31 and L32) without skip connection, and three additional convolutional layers (L33, L34 and L35).

layers	Kernel size	output channels	operations	skip connection
L1	$3 \times 3 \times 3$	16	ReLU	-
L2	$3 \times 3 \times 3$	64	ReLU	L4, L32
L3	$3 \times 3 \times 1$	64	BN + ReLU	-
L4	$3 \times 3 \times 1$	64	BN	L6
L5	$3 \times 3 \times 1$	64	BN + ReLU	-
L6	$3 \times 3 \times 1$	64	BN	L8
L7	$3 \times 3 \times 1$	64	BN + ReLU	-
L8	$3 \times 3 \times 1$	64	BN	L10
L9	$3 \times 3 \times 1$	64	BN + ReLU	-
L10	$3 \times 3 \times 1$	64	BN	L12
L11	$3 \times 3 \times 1$	64	BN + ReLU	-
L12	$3 \times 3 \times 1$	64	BN	L14
L13	$3 \times 3 \times 1$	64	BN + ReLU	-
L14	$3 \times 3 \times 1$	64	BN	L16
L15	$3 \times 3 \times 1$	64	BN + ReLU	-
L16	$3 \times 3 \times 1$	64	BN	L18
L17	$3 \times 3 \times 1$	64	BN + ReLU	-
L18	$3 \times 3 \times 1$	64	BN	L20
L19	$3 \times 3 \times 1$	64	BN + ReLU	-
L20	$3 \times 3 \times 1$	64	BN	L22
L21	$3 \times 3 \times 1$	64	BN + ReLU	-
L22	$3 \times 3 \times 1$	64	BN	L24
L23	$3 \times 3 \times 1$	64	BN + ReLU	-
L24	$3 \times 3 \times 1$	64	BN	L26
L25	$3 \times 3 \times 1$	64	BN + ReLU	-
L26	$3 \times 3 \times 1$	64	BN	L28
L29	$3 \times 3 \times 1$	64	BN + ReLU	-
L30	$3 \times 3 \times 1$	64	BN	L32
L31	$3 \times 3 \times 1$	64	BN + ReLU	-
L32	$3 \times 3 \times 1$	64	BN	-
L33	$3 \times 3 \times 1$	256	ReLU	-
L34	$3 \times 3 \times 1$	256	ReLU	-
L35	$3 \times 3 \times 1$	1	-	-

4.2.3 Loss Functions

In our work, we use two types of loss functions to train DBLRGAN.

Content Loss. The Mean Square Error (MSE) loss is widely used in optimization objective for video deblurring in many existing methods. Based on MSE, our content loss function is defined as:

$$\mathcal{L}_{content} = \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} \left(I_{x,y}^{sharp} - G(I^{blurry})_{x,y} \right)^2, \tag{4.3}$$

where W and H are the width and height of a frame, $I_{x,y}^{sharp}$ is the value of sharp frames at location (x, y), and $G(I^{blurry})_{x,y}$ corresponds to the value of deblurred frames which are generated from DBLRNet.

Adversarial Loss. In order to drive G to generate sharp frames similar to the real-

Table 4.2: **Configurations of our D model in DBLRGAN.** BN means batch normalization and ReLU represents the activation function.

Layers	1-2	3-5	6-9	10-14	15-16	17
kernel	3 x 3	3 x 3	3 x 3	3 x 3	FC	FC
channels	64	128	256	512	4096	2
BN	BN	BN	BN	BN	-	-
ReLU	ReLU	ReLU	ReLU	ReLU	-	-

world frames, we introduce an adversarial loss function to update models. During the training stage, parameters of DBLRNet are updated in order to fool the discriminator D. The adversarial loss function can be represented as:

$$\mathcal{L}_{adversarial} = \log(1 - D(G(I^{blurry}))), \tag{4.4}$$

where $D(G(I^{blurry})$ is the probability that the recovered frame is a real sharp frame.

Balance of Different Loss Functions. In the training stage, the loss functions are combined in a weight fusion fashion:

$$\mathcal{L} = \mathcal{L}_{content} + \alpha \cdot \mathcal{L}_{adversarial}. \tag{4.5}$$

In order to balance the content and adversarial losses, we use a hyper-parameter α to yield the final loss $\mathcal{L}$. We investigate different values of α from 0 to 0.1. When $\alpha = 0$, only the content loss works. In this case, DBLRGAN degrades to DBLRNet. With the increase of α, the adversarial loss plays a more and more important role. The value of α should be relative small, because large values of α can degrades the performance of our model.

4.3 Experimental Results

In this section, we conduct experiments to demonstrate the effectiveness of the proposed DBLRNet and DBLRGAN on the task of video deblurring.

4.3.1 Datasets

VideoDeblurring Dataset. Su *et al.* build a benchmark which contains videos captured by various kinds of devices such as iPhone 6s, GoPro Hero 4 and Nexus 5x, and each video includes about 100 frames of size 1280×720 [Su et al., 2017a]. This benchmark consists of two sub datasets: quantitative and qualitative ones. The quantitative subset contains 6708 blurry frames and their corresponding ground-truth sharp frames from 71 videos. The qualitative subset includes 22 scenes, most of which contain more than 100 images. Note that there is not ground truth for the qualitative subset, thus we can only conduct qualitative experiments on this subset. We split the quantitative subset into 61 training videos and 10 testing videos, which is the same

setting as the previous method [Su et al., 2017a]. Besides quantitative experiments on the 10 testing videos, we additionally test our models on the qualitative subset.

Blurred KITTI Dataset. Geiger *et al.* develop a dataset called KITTI by using their autonomous driving platform [Geiger et al., 2013]. The KITTI dataset consists of several subsets for various kinds of tasks, such as stereo matching, optical flow estimation, visual odometry, 3D object detection and tracking. Based on the stereo 2015 dataset in the KITTI dataset, Pan *et al.* create a synthetic Blurred KITTI dataset [Pan et al., 2017], which contains 199 scenes. Each of the scenes includes 3 images captured by a left camera and 3 images captured by a right camera. It is worthy noting that, the KITTI data set is not used when training our models. Namely, this dataset is utilized only for testing.

4.3.2 Implementation Details and Parameters

When training DBLRNet, we use Gaussian distribution with zero mean and a standard deviation of 0.01 to initialize weights. In each iteration, we update all the weights after learning a mini-batch of size 4. To augment the training set, we crop a 128×128 patch at any location of an image (1280×720). In this way, there are at least 712193 possible samples per one frame on the dataset [Su et al., 2017a], which greatly increases the number of training samples. In addition, we also randomly flip frames in the training stage. The DBLRNet is trained with a learning rate of 10^{-4}, based on the content loss only. We also decrease the learning rate to 10^{-5} when the training loss does not decrease (usually after about 1.5 x 10^5 iterations), for the sake of additional performance improvement.

In DBLRGAN, we set the hyper parameter α as 0.0002 when we conduct experiments as empirically this value achieves the best performance. It has a better PSNR value due to three reasons. Firstly, when training DBLRGAN, we directly place DBLRNet as our generator and fine-tune our DBLRGAN. Thus, the DBLRGAN has a high PSNR value like DBLRNet at the beginning. Secondly, the loss functions of DBLRGAN are combined in a weight fusion fashion. We set the hyper parameter α as 0.0002 when we conduct experiments. This is a very small value, which forces the content loss to have an overwhelming superiority over the adversarial loss on PSNR value during the training stage. Thirdly, the learning rate is set as 10^{-5}, so the PSNR value does not have severe changes. We early stop training our DBLRGAN before the PSNR start to drop.

4.3.3 Effectiveness of DBLRNet

The proposed DBLRNet has the advantage of learning spatio-temporal feature representations. In order to verify the effectiveness of DBLRNet, we develop another two similar neural networks: DBLRNet (single) and DBLRNet (multi). These two models have the same network architectures as the original DBLRNet while there are two differences between them and the original DBLRNet. The first difference is the input. The input of DBLRNet (single) is one single frame, while the input of DBLRNet

Table 4.3: **Performance comparisons in terms PSNR with PSDEBLUR, WFA [Delbracio and Sapiro, 2015], DBN (single), DBN (noalign), DBN(flow) [Su et al., 2017a], DBLRNet (single) and DBLRNet (multi) on the VideoDeblurring dataset.** The best results are shown in bold, and the second best are underlined. All results of DBLRNet and DBLRGAN are obtained without aligning.

Methods	1	2	3	4	5	6	7	8	9	10	Average (PSNR)
INPUT	24.14	30.52	28.38	27.31	22.60	29.31	27.74	23.86	30.59	26.98	27.14
PSDEBLUR	24.42	28.77	25.15	27.77	22.02	25.74	26.11	19.71	26.48	24.62	25.08
WFA	25.89	32.33	28.97	28.36	23.99	31.09	28.58	24.78	31.30	28.20	28.35
DBN (single)	25.75	31.15	29.30	28.38	23.63	30.70	29.23	25.62	31.92	28.06	28.37
DBN (noalign)	27.83	33.11	31.29	29.73	25.12	32.52	30.80	27.28	33.32	29.51	30.05
DBN (flow)	28.31	33.14	30.92	29.99	25.58	32.39	30.56	27.15	32.95	29.53	30.05
DBLRNet (single)	28.68	29.40	35.11	32.25	24.94	30.77	29.81	25.67	33.14	30.06	29.98
DBLRNet (multi)	30.40	32.17	36.68	33.38	26.20	32.20	30.71	26.71	36.50	30.65	31.56
DBLRNet	31.96	34.31	**37.86**	**35.21**	27.23	33.63	32.32	27.84	38.23	31.83	33.04
DBLRGAN	**32.32**	**34.51**	37.63	35.18	**27.42**	**33.81**	**32.43**	**28.18**	**38.32**	**32.06**	**33.19**

Table 4.4: **Performance comparisons with [Hyun Kim and Mu Lee, 2015], [Sellent et al., 2016] and [Pan et al., 2017] on the Blurred KITTI dataset in terms of the PSNR criterion.** The best results are shown in bold, and the second best are underlined.

Methods	PSNR-LEFT	PSNR-RIGHT
Kim et al.	28.25	29.00
Sellent et al.	27.75	28.52
Pan et al.	30.24	30.71
DBLRNet (single)	28.97	29.55
DBLRNet (multi)	29.94	30.33
DBLRNet	30.10	30.54
DBLRGAN	**30.42**	**30.87**

(multi) and DBLRNet is a stack of five neighboring frames. The second difference is that, in both DBLRNet (single) and DBLRNet (multi), all the convolution operations are 2D convolution operations.

Table 4.3 and 4.4 show the PSNR values of DBLRNet (single), DBLRNet (multi) and DBLRNet on the VideoDeblurring dataset and the Blurred KITTI dataset, respectively. Compared with DBLRNet (single), DBLRNet (multi) achieves approximately $3\% \sim 5\%$ improvement of PSNR values, which shows that stacking multiple neighboring frames is useful to learn temporal features for video deblurring even in case of 2D convolution. Comparing DBLRNet with DBLRNet (multi), there are additionally $1\% \sim 5\%$ improvement in terms of PSNR. We suspect that the improvement results from the power of spatio-temporal feature representations learned by 3D convolution. Conducting these two kinds of comparisons, the effectiveness of DBLRNet has been verified.

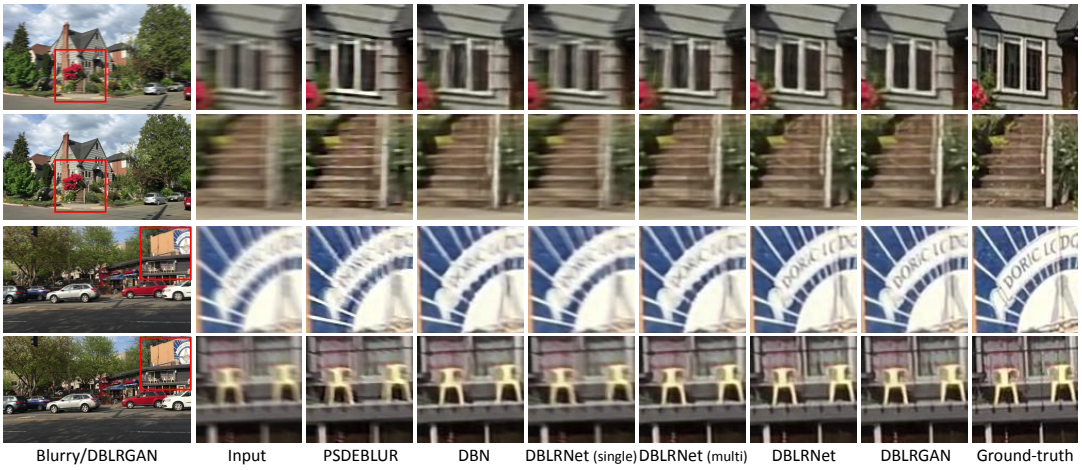


Figure 4.4: **Exemplar results on the VideoDeblurring dataset (quantitative subset).** From left to right: real blurry frame/ Output of DBLRGAN, input, PSDEBLUR, DBN [Su et al., 2017a], DBLRNet (single), DBLRNet (multi), DBLRNet, DBLRGAN and ground-truth. All results are obtained without alignment. Best viewed in color.

4.3.4 Effectiveness of DBLRGAN

In this section, we investigate the performance of the proposed DBLRGAN. Table 4.3 and 4.4 show the quantitative results on the VideoDeblurring and Blurred KITTI dataset, respectively. Quantitatively, DBLRGAN outperforms DBLRNet with slight advance (about 1% improvement). As have mentioned above, the generator model in DBLRGAN aims to generate frames with similar pixel values as the sharp frames while the discriminator model along with the adversarial loss drives the generator to recover realistic images like real-word images. These two models complement each other and achieve better results. The results in Table 4.3 and 4.4 show that the improvement achieved by DBLRNet is more obvious than GAN model. While according to Figure 4.5, the deblurred frames generated by DBLRGAN are sharper than DBLRNett, *e.g.*, the word "Bill" in the top row. α should be set as a little value because a bigger α will break the balance of content and adversarial loss, which causes worse performance of video deblurring.

Figure 4.4 and 4.5 provide exemplar results on the quantitative and qualitative subsets of the VideoDeblurring dataset, respectively. Please notice the two columns corresponding to DBLRNet and DBLRGAN in Figure 4.4, especially the letters in the third row, where results of DBLRGAN are more photo-realistic than those of DBLR-Net. The same case is observed in Figure 4.5. Letters in results of DBLRGAN are sharper than those of DBLRNet, which consistently shows that, DBLRGAN generates more realistic frames with finer textural details compared with DBLRNet.

All results of DBLRNet and DBLRGAN are obtained without aligning. Aligning images is computationally expensive and fragile [Su et al., 2017a]. [Kim et al., 2017] evaluate DBN model and find that the speed of DBN model without aligning is

Figure 4.5: **Exemplar results on the VideoDeblurring dataset (qualitative subset).** From left to right: real blurry frame/Output of GBLRGAN, input, PSDEBLUR, DBN [Su et al., 2017a], [Hyun Kim and Mu Lee, 2015], DTBNet [Kim et al., 2017], DBLRNet (single), DBLRNet (multi), DBLRNet and DBLRGAN. All results are attained without alignment. Best viewed in color.

almost more than 20 times faster than it with aligning because aligning procedure is very time-consuming. Our proposed models enable the generation of high quality results without computing any alignment, which makes it highly efficient to scene types.

4.3.5 Comparison with Existing Methods

To further verify the effectiveness of our models, we additionally compare the performance of DBLRNet and DBLRGAN with that of several state-of-the-art approaches on both the VideoDeblurring dataset and the KITTI dataset.

On the VideoDeblurring dataset, we compare our models with PSDEBLUR, WFA [Delbracio and Sapiro, 2015], DBN [Su et al., 2017a] and DBN (single). PSDEBLUR is the deblurred results of PHOTOSHOP. WFA is a method based on multiple frames as input. DBN achieves the state-of-the-art performance on the VideoDeblurring data set before this work. DBN (single) is a variant of DBN which stacks 5 copies of one single frame as input. Table 4.3 shows quantitative comparisons between our methods and these methods. Specially, the results indicate that our method significantly outperforms the DBN model by 3.14 db. Figure 4.4 and 4.5 also represent visual comparison between our models and these methods on both the quantitative (Figure 4.4) and qualitative (Figure 4.5) sub-datasets, respectively. Evidently our models achieves sharper results.

(a)

(b)

(c)

Figure 4.6: **Performance of our method on blurry videos caused by bokeh.** The figure shows a sample frame from the Blurred KITTI dataset, which is captured from a car moving at a high speed. The blurs take place in the side area, while the center part is clear. We show a few pairs of zoomed-in patches from the frame before and after applying our method. The sharper edge demonstrates that our method can generalize well to other types of blurry videos.

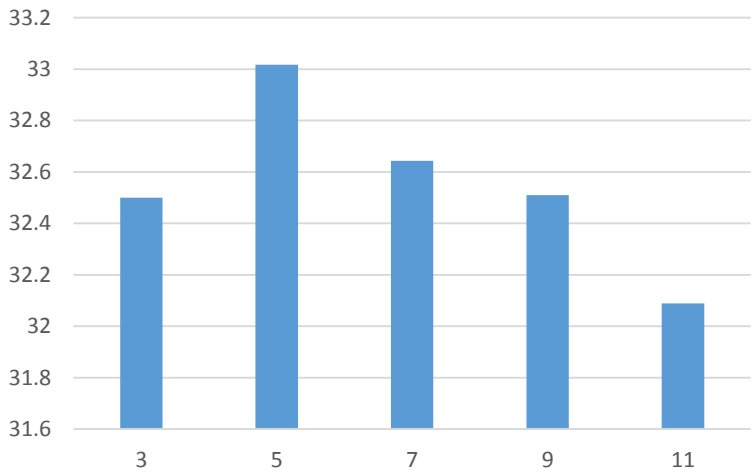


Figure 4.7: **Performance comparisons of our method in terms of PSNR by varying the number of input frames.**

On the dataset of Blurred KITTI, we conduct comparison with [Hyun Kim and Mu Lee, 2015], [Sellent et al., 2016] and [Pan et al., 2017]. [Pan et al., 2017] is a geometry based method and utilizes additional stereo information from image pairs. It is the current state-of-the-art on the Blurred KITTI dataset. We simply apply the DBLRNet trained on the VideoDeblurring dataset to the Blurred KITTI dataset and still achieve comparable results with [Pan et al., 2017]. With the additional adversarial loss, DBLRGAN slightly outperforms [Pan et al., 2017]. Please note that, our models are not specialized for the stereo setting.

4.3.6 Different Frames & Other Types of Blur

Different Frames. We are curious about how the number of consecutive frames influences the performance of our DBLRGAN model. Thus we compare the PSNR values of the model by varying the number of input blurry frames. Making it more specific, on the VideoBlurring dataset, five kinds of settings, three, five, seven, nine and eleven continuous frames are taken as input to our model. Fig. 4.7 shows that our model with five frames as input achieves the best performance. With the increase of input frames, the PSNR values become lower. We suspect that, as our 3D convolution based network can extract powerful representations to describe short-term fast-varying motions occurring in continuous input frames, it is suitable to set the temporal span relatively small to capture the rapid dynamics across local adjacent frames.

Generalize to Other Types of Blurry Videos. Though our model is trained on the VideoDeblurring dataset, which includes only blurry frames caused by camera shakes, we are also curious about how it generalize to blurry videos of other blur types. To this end, we test it on videos from the Blurred KITTI dataset. Fig. 4.6 shows exemplar frames, which is captured by a camera mounted on a high-speed

car. The dominated blur is cause by bokeh (see the comparison between the center area and the border area in the image), rather than camera shakes. As shown in the comparison of the enlarged patches, by applying our DBLRGAN model, the edges in the image become sharper. As discussed above, this verifies the advantage of our method capturing short-term fast-varying motions.

Limitation. Removing jumping artifacts is a challenge of video deblurring. As shown in Fig. 4.1 (col. 4&5, row 2), there are also some jumping artifacts in the deblurred frames. Thus our method cannot solve it completely. However, the proposed model contributes to alleviate the unexpected temporal artifacts because it captures jointly spatial and temporal information encoded in neighboring frames. Even without post-processing and aligning, our proposed model can also achieve satisfied performance. Please refer to Fig. 4.4 and 4.5. Comparing with prior methods, when frames are severely blurred, our methods can generate better deblurred frames.

4.4 Conclusions

In this chapter, we have resorted to spatio-temporal learning and adversarial training to recover sharp and realistic video frames for video deblurring. Specifically, we propose two novel network models. The first one is our DBLRNet, which uses 3D convolutional kernels on the basis of deep residual neural networks. We demonstrate that DBLRNet is able to capture better spatio-temporal features, leading to improved blur removal. Our second contribution is DBLRGAN equipped with both the content loss and adversarial loss, which are complementary to each other, driving the model to generate visually realistic images. The experimental results on two standard benchmarks show that our proposed DBLRNet and DBLRGAN outperform the existing state-of-the-art methods in video deblurring.

Chapter 5

Deblurring: Detail-Aware Networks to Bring a Blurry Image Alive

This chapter is about making a blurry image alive. Motion-blurred images are the result of light accumulation over the period of camera exposure time, during which the camera and objects in the scene are in relative motion to each other. The inverse process of extracting an image sequence from a single motion-blurred image is an ill-posed vision problem. One key challenge is that the motions across frames are subtle, which makes the generating networks difficult to capture them and thus the recovery sequences lack motion details. In order to alleviate this problem, we propose a detail-aware network with three consecutive stages to improve the reconstruction quality by addressing specific aspects in the recovery process. The detail-aware network firstly models the dynamics using a cycle flow loss, resolving the temporal ambiguity of the reconstruction in the first stage. Then, a GramNet is proposed in the second stage to refine subtle motion between continuous frames using Gram matrices as motion representation. Finally, we introduce a HeptaGAN in the third stage to bridge the continuous and discrete nature of exposure time and recovered frames, respectively, in order to maintain rich detail. Experiments show that the proposed detail-aware networks produces sharp image sequences with rich details and subtle motion, outperforming the state-of-the-art methods.

5.1 Introduction

Motion blur is a common artifact when taking photos and is caused by either camera shake [Bahat et al., 2017; Zhang and Wipf, 2013] or object motion [Michaeli and Irani, 2014; Shi et al., 2015; Pan et al., 2016] during the exposure period within which light from the scene is accumulated [Gupta et al., 2010; Harmeling et al., 2010]. Observing a motion-blurred image, humans seem to be able to infer a plausible explanation of both the scene appearance and the underlying motion.

This chapter aims to recover a temporal sequence of clean and sharp image frames from a single motion-blurred image, to mimic the above human ability. This task

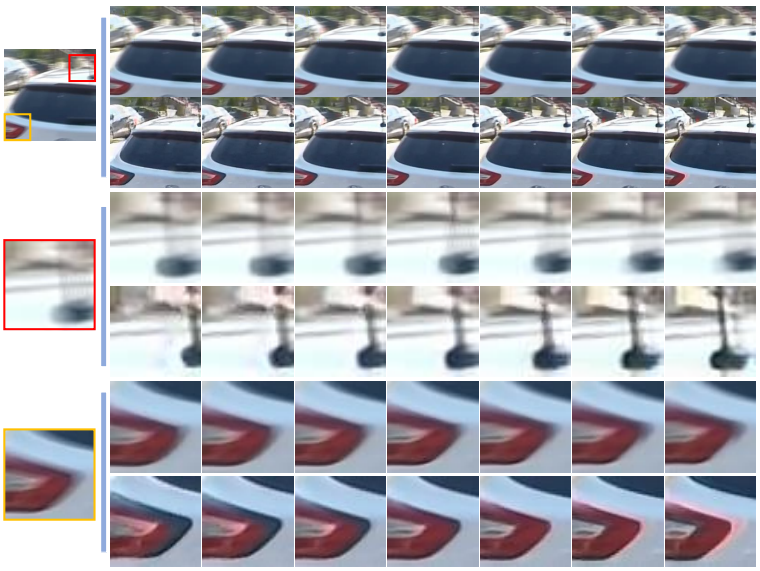


Figure 5.1: **Video generation example**. The left column shows a blurry input image (top), and two zoomed-in regions. Rows to the right show frames extracted by the model released by [Jin et al., 2018] (top) and the proposed method (bottom), respectively. Our method recovers sharper detail (car antenna) and better preserves small motion (rear light).

involves solving a severely under-constrained inverse problem, i.e., , to recovering multiple images from a single image which is the integration of the former. To some extent, the task is related to single image deblurring [Cho et al., 2011; Hirsch et al., 2011; Whyte et al., 2012]. However, our task contains additional complexity, as we also want to get a set of temporally ordered sharp images that gave rise the single blurred version. This is particular challenging, since the image integration operator is temporal-order invariant therefore multiple valid solutions exist. Moreover, besides multiple sharp frames, we also aim to recover the underlying motion across neighboring frames, yet often the motion is small between time-consecutive frames. For example, without modeling the subtle motion across frames, as shown in the fifth row of Fig. 5.1, the frames recovered by [Jin et al., 2018] look identical to each other. Finally, a motion-blurred image is generated during a *continuous* exposure period, yet one has to approximate this process by discretizing the time axis, leading to loss of information in image details.

To address the above challenges, we propose a generative model trained in three stages for video sequence extraction from a motion-blurred image. The first stage, called **BaseGAN**, learns to recover sharp video frames with a cycle flow loss to constrain that the motions across frames before and after the recovery are identical, thus resolving ambiguity in the recovery process. The second stage, **GramGAN**, is designed to recover subtle motions, employing a Gram matrix for motion feature representations. By minimizing the difference of Gram matrix description between the recovered frames and sharp frames, subtle motions are recovered. In the third stage,

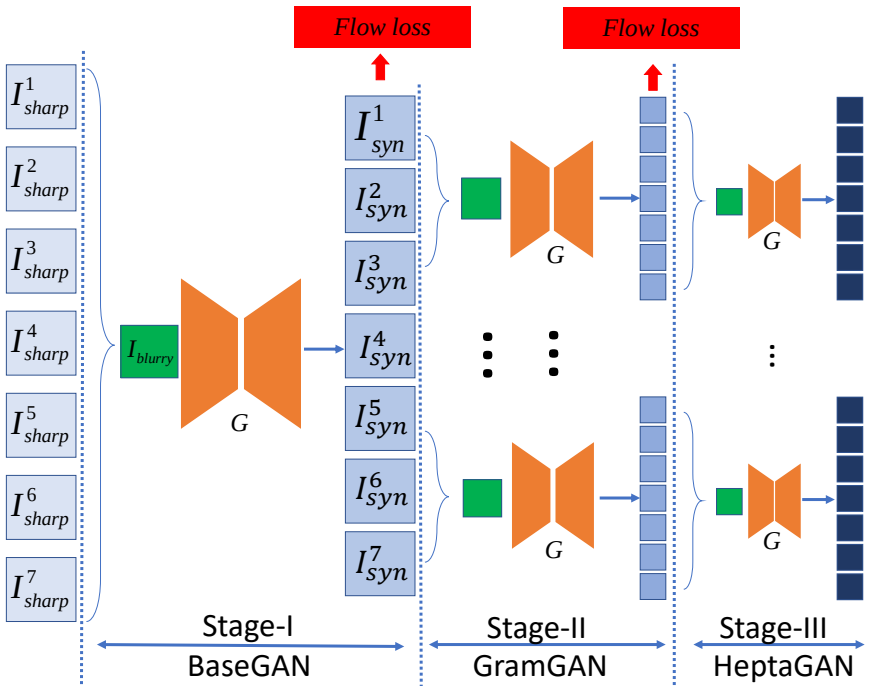


Figure 5.2: **Cascaded structure for generator training**. Consecutive input frames are averaged and input to BaseGAN to recover sharp frames. At Stage-2, frames are averaged to be new blurry images and sent into GramGAN to recover images with more appreciable subtle movements. HeptaGAN at Stage-3 guides to recovers disparity information.

HeptaGAN training is carried out, taking multiple images, in our case seven, as input and generating the same number of output images. Specifically, we synthesize a motion-blurred image I_{blurry} from the seven input images $\{I_{in}\}$ and learn to recover a sequence of sharp images $\{I_{out}\}$ from the blurry image. The recovered frames $\{I_{out}\}$ are again used to produce a blurry image I'_{blurry}. The **HeptaGAN** model is optimized by not only forcing the recovered frames $\{I_{out}\}$ to be identical to the input sharp frames $\{I_{in}\}$, but also by minimizing the distance of the blurry images before and after the recovery procedure, I_{blurry} and I'_{blurry}, respectively. With this bi-cycle consistency, we minimize the disparity during the continuous-to-discrete transform.

Trained with this three-stage architecture, our generative model produces visually pleasing video frames given a motion-blurred image, as shown in Fig. 5.1. Different from existing approaches, which estimate frame sequences with multiple models, e.g., [Jin et al., 2018], our method is able to extract multiple frames with a single model, which is more efficient and is better capable of exploiting spatio-temporal information. It is notable that the generators in different stages share weights.

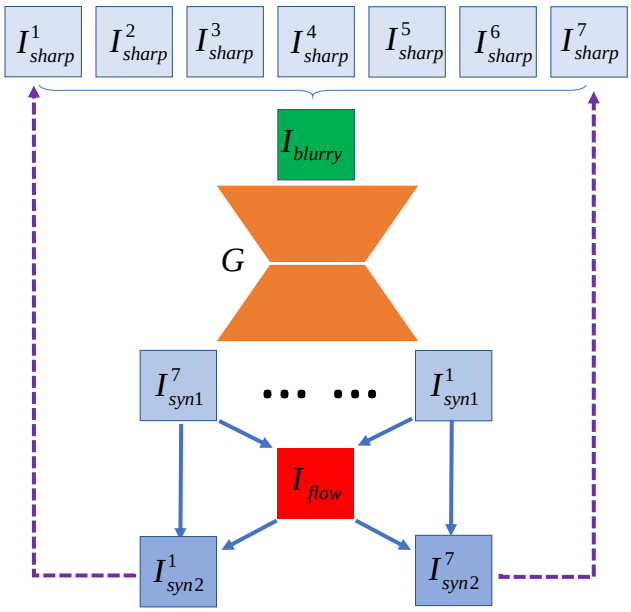


Figure 5.3: **BaseGAN architecture with optical flow**. Seven continuous sharp frames are averaged into a blurry image as input to a generator to recover seven sharp frames $\{I^i_{syn1}, i = 1, ..., 7\}$. Flow images are calculated based on the 1st and 7th synthesized images. The flow is applied to warp the synthesized image I^7_{syn1} and results in I^1_{syn2}. Likewise, I^1_{syn1} is warped with the flow to produce I^7_{syn2}. These two images, I^1_{syn2} and I^7_{syn2}, are constrained to be close to their sharp counterparts I^1_{sharp} and I^7_{sharp}, to make sure the recovered motion across frames is identical to that before recovery.

5.2 Approach

To approach the task of extracting multiple frames from a motion-blurred image, we propose to train a generator G in a cascaded structure with three stages (Fig. 5.2): (1) In the first stage, a BaseGAN module with a flow loss function generates a sharp and realistic video without ambiguity. Seven continuous frames are averaged to simulate a motion-blurred image which is input to the BaseGAN, and the output is seven sharp frames. (2) Subtle movements are addressed by a GramGAN module in the second stage, which takes the output of the first stage as input and outputs seven sharp frames. (3) The third stage employs HeptaGAN training to recover the information of the discrete predicted frames regarding the continuous exposure process. G is an encoder-decoder model with 30 convolutional layers and a 21-channel output, corresponding to seven consecutive frames (three channels per frame). The generator structure remains unchanged and weights are shared in the three stages. During inference, G predicts seven output frames from a motion-blurred image with a single forward pass.

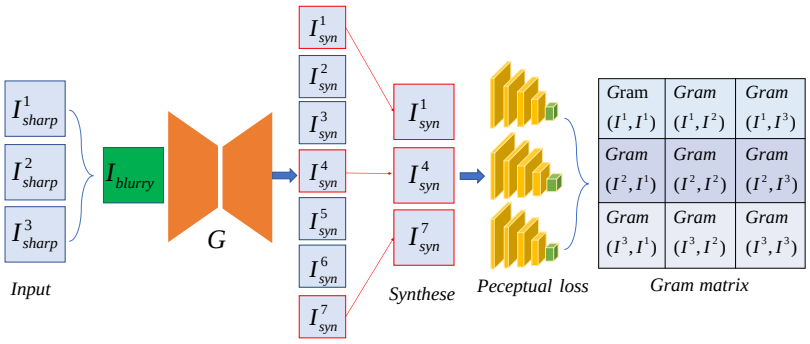


Figure 5.4: **Gram matrix components for three sequential frames**. Blocks on the diagonal are the individual frames themselves, while the off-diagonal blocks are correlations between the sequential frames.

5.2.1 Ambiguity Resolving with Flow: BaseGAN

The BaseGAN module in the first stage is a generative adversarial network. The generator produces seven frames recovering as much information as possible and the discriminator aims to discriminate the predicted frames against real frames to ensure the predicted frames are realistic. The pixel-wise MSE loss is widely used for generating deblurred images, which may have high PSNR values but are unsatisfying due to over-smoothed textures. Thus the content loss G for the central frame includes both MSE and perceptual loss [Johnson et al., 2016] as

$$\mathcal{L}^{central}_{content} = \left\| I_{sharp} - G(I_{blurry}) \right\| + \left\| \Phi(I_{sharp}) - \Phi(G(I_{blurry})) \right\| , \tag{5.1}$$

where $G(I_{blurry})$ is the deblurred image, and I_{sharp} corresponds to the sharp frame. Φ denotes the features obtained from the last convolution layer of VGG19 [Simonyan and Zisserman, 2015b], which is employed to measure the perceptual loss.

The procedure of recovering the other six neighboring frames is unstable if employing the same content loss defined above, because different orders among frames produce the same motion-blurred image. Thus, the content loss for these frames can be represented as [Jin et al., 2018]:

$$\mathcal{L}^{pair}_{content} = \sum_{i=1}^{3} \left([I^i_{sharp}, I^{8-i}_{sharp}]_+ - [G(I^i_{blurry}), G(I^{8-i}_{blurry})]_+ \right) + \left([I^i_{sharp}, I^{8-i}_{sharp}]_- - [G(I^i_{blurry}), G(I^{8-i}_{blurry})]_- \right) , \tag{5.2}$$

where $[x,y]_+ = |sum(x,y)|^2$ and $[x,y]_- = |sub(x,y)|^2$ denote the summation and subtraction operation on corresponding positions of two input images, respectively.

To generate realistic sharp frames, an adversarial loss function is introduced with the goal to fool the discriminator D:

$$\mathcal{L}_{adv} = \log \left(1 - D(G(I_{blurry})) \right) \tag{5.3}$$

where $D(G(I_{blurry}))$ classifies a recovered frame to determine whether or not the

reconstructed frame is a real image.

Since the reconstruction is invariant to the temporal order of frames, we introduce a loss function based on optical flow, shown in Fig. 5.3. Seven sharp frames are averaged to create a blurry image, which is input into a generator to produce seven synthesized sharp frames. The first and seventh synthesized frames, I^1_{syn1} and I^7_{syn1}, are then fed into a PWC-Net [Sun et al., 2018] which computes pair-wise optical flow. This is applied to the first and seventh synthesized frames to obtain new seventh and first frames, respectively. The loss function is calculated based on the input (*sharp*) and output (*syn*2) frames as

$$\mathcal{L}_{flow} = ||I^1_{sharp} - W(I^7_{syn1}, I^{7\to1}_{flow})||^2_2 + ||I^7_{sharp} - W(I^1_{syn1}, I^{1\to7}_{flow})||^2_2, \tag{5.4}$$

where I_{sharp} are real sharp frames, $I^{i\to j}_{flow}$ is the optical flow image from the ith to the jth frame. $W(I^7_{syn1}, I^{7\to1}_{flow})$ means that we generate the new first frame using the seventh synthesized frame and flow images via spatial transformer networks (STN) [Jaderberg et al., 2015]. By constraining the generation process with the flow loss, the unique order among sequential frames is maintained in training and thus recovered in inference. During training in the first stage, the loss functions are combined as

$$\mathcal{L} = \mathcal{L}^{central}_{content} + \mathcal{L}^{pair}_{content} + \alpha\mathcal{L}_{adversarial} + \beta\mathcal{L}_{flow}\,. \tag{5.5}$$

In order to balance the content, adversarial and flow losses, we use hyper-parameters α and β to yield the final loss $\mathcal{L}$.

5.2.2 Learning Subtle Movements: GramGAN

The first stage guides the generator to produce sharp realistic frames, while the content loss is weak in learning motion across frames. The loss is very small in the case of subtle movements, resulting in small pixel variations across neighboring frames. This makes it difficult to learn the motion dynamics in training, reconstructing nearly identical sequence frames. Thus in the second stage, we focus on learning subtle movements, to improve the robustness of the model in the extreme case.

To this end, we introduce the Gram matrix at this stage to process high-level semantic features and incorporate temporal information. Note that the Gram matrix has been employed in recent work to represent motion for dynamic texture synthesis and generation of time-lapse videos [Tesfaldet et al., 2018; Xiong et al., 2018]. However, to the best of our knowledge, this is the first time to introduce it to the task of reconstruction from blurry images. Further, in contrast to prior work, which applies the Gram matrix on the features of a GAN discriminator, our model uses the Gram matrix in the generator.

The second training stage, GramGAN, is illustrated in Fig. 5.4. Three of the seven output frames of the first stage are averaged to create a blurry image, and a Gram matrix is computed by combining the feature maps of three sequential frames

(I, I', I''). The feature map of a synthesized frame is a 3-dimensional tensor, whose axes are width, height, and channel, respectively. Firstly, we concatenate feature maps along an additional axis to produce a 4-d tensor, whose first axis corresponds to the three sequential frames. Then we reshape the 4-d tensor into a 2-d one, $\mathbf{F}$, whose first axis is combined from the first two axes and second axis is from the last two axes. Finally, the product of the new tensor $\mathbf{F}$ and its transpose describes motion by spatio-temporal statistics. Thus the Gram matrix entry for three frames can be be formulated as a perceptual term as

$$Gram(I, I', I'') = \frac{1}{M}\mathbf{F}^{\mathrm{T}}\mathbf{F}\,, \tag{5.6}$$

where $M = CHW$ denotes the product of channel, height and width of feature maps.

Given seven output frames by the first stage, there are nine combinations of three frames with equal distance along the time axis (i.e., ., $I_1I_2I_3$, $I_2I_3I_4$, ... , $I_1I_4I_7$). The additional loss function with regard to the Gram matrix is

$$\mathcal{L}_G = \sum_{i=1}^{9} \left\| Gram_i(G(I_{blurry})) - Gram_i(\{I_{sharp}\}) \right\|\,, \tag{5.7}$$

where I_{blurry} is the blurry image produced by averaging the three frames $\{I_{sharp}\}$ taken from the seven frames output by the BaseGAN. As Fig. 5.4 shows, I^1_{syn}, I^4_{syn}, and I^7_{syn} from the generated seven frames $G(I_{blurry})$ are taken as input to calculate the Gram matrix. We constrain these three frames by referring to the corresponding ground truth $\{I_{sharp}\}$. $Gram_i(\cdot)$ corresponds to the i-th way of taking three images.

Note that in Fig. 5.2 we take three frames of the seven frames from the first stage, rather than all of them to simulate a blurry image. The motivation is that, by doing so, we can interpolate the motion across the three frames into the fine-grained motion dynamics across the output seven frames. That also explains why we use the first synthesized frame (I^1_{syn}), the midterm frame (I^4_{syn}), and the last frame (I^7_{syn}), as the start, intermediate and end state of the motion dynamics, and use the corresponding input three frames to constrain them.

There are several advantages of GramGAN training in this stage. First, we can learn the motion dynamics more efficiently with the Gram matrix as motion representation, avoiding generation of multiple identical frames. Second, with less input and more output images, the model is able to unravel fixed time period into more discrete time steps with fine-grained motion. We verify this in Sec. 5.3.3.

5.2.3 Disparity Recovery: HeptaGAN

Output frames from the trained first and second stages are already realistic and exhibit more appreciable subtle motions across neighboring frames. The exposure process in the real world producing the motion-blurred image is *continuous*. However, our task of recovering multiple frames from a single motion-blurred image is a reverse process and it is actually *discrete*. To address this, we propose a HeptaGAN stage, using a blur function F and a blur-removal function G to encourage the

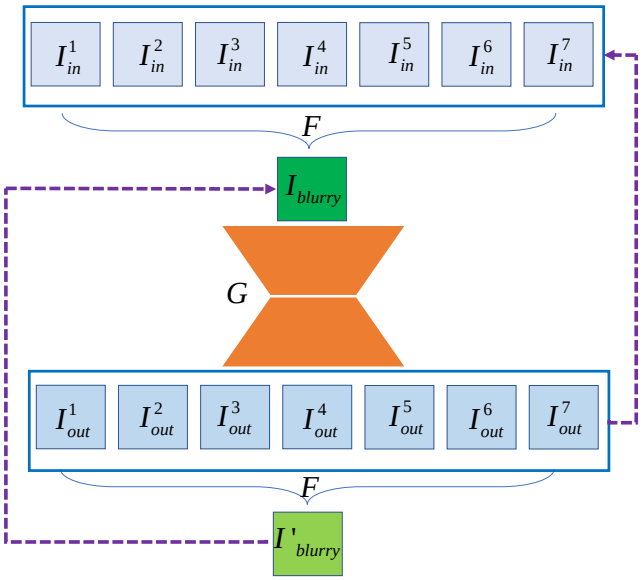


Figure 5.5: **HeptaGAN schematic**. Given seven continuous frames, our system simultaneously creates a corresponding blurry image based on function F and learns a video recovery function G. G outputs seven frames, serving as input to F to produce a new blurry image. The learning constrain can be written as: $G(F(\{I_{in}\})) = \{I_{out}\} \approx \{I_{in}\}$ and $F(G(I_{blurry})) = I'_{blurry} \approx I_{blurry}$.

preservation of original information contained in the single motion-blurred image.

In particular, given frames generated from the GramGAN stage as input, the idea is to produce a blurry image and recover the sharp frames by using the produced blurry image, and the recovered sharp frames are averaged to produce a blurry image again, forming a bi-cycle process. We train the model in an unsupervised manner in this stage. As illustrated in Fig. 5.5, we build two function approximators F and G. F produces blurry images from consecutive sharp frames and G recovers the video sequence from the synthesized motion-blurred image. Because of the assumption that motion-blurred images can be produced by averaging multiple frames, the model F is the average function and we only train G. Given seven sharp frames $\{I_{in}\}$, the motion-blurred images can be produced as $I_{blurry} = F(\{I_{in}\})$. We expect that G can generate continuous sharp frames $\{I_{out}\} = G(I_{blurry})$, whose corresponding averaging motion-blurred frame $I'_{blurry} = F(\{I_{out}\})$ is the same as I_{blurry}. This imposes the bi-cycle consistency. Note that, different from the traditional CycleGAN [Zhu et al., 2017a] which simultaneously trains G and F on paired or unpaired images, the input to our HeptaGAN are seven consecutive frames. We only train G, but with two cycle-like losses which are discussed as follows and shown in Fig. 5.5.

The L1 loss is used to constrain this learning process as

Table 5.1: **Performance comparison with [Nah et al., 2017a], [Jin et al., 2018], [Pan et al., 2019] and [Purohit et al., 2019] on the *GOPRO_Large_all* dataset and ablation study of model G after different stages of training.**

Method	PSNR	SSIM	EPE
Nah et al.	28.98	0.911	-
Jin et al.	26.98	0.881	17.93
Pan et al.	28.49	0.920	-
Purohit et al.	30.58	0.941	-
B	28.14	0.905	13.62
BG	29.65	0.921	11.25
BGH	**30.64**	**0.942**	**10.03**

$$\mathcal{L}_C = \frac{1}{N}\sum^{N} \|F(\{I_{in}\}) - F(G(F(\{I_{in}\})))\|_1 + \frac{1}{7N}\sum^{K}\sum_{j=1}^{7}\left\|I_{in}^{j} - (G(F(I_{in}^{j})))\right\|_1 , \quad (5.8)$$

where N is the number of seven-frame groups.

5.3 Experiments

We test our approach on the widely used public GOPRO dataset [Nah et al., 2017a], which is first introduced along with evaluation metrics. Then implementation details are given and ablation study is conducted, and a comparison with the state of the art is reported. Finally, we test the generalization of our method to blur caused by bokeh.

5.3.1 Dataset & Metrics

In our experiments we use the GOPRO_Large_all frames of the GOPRO dataset, including 22 training and 11 test videos, respectively. We average consecutive frames to produce blurry images. To compute the fidelity of the extracted frames, we use the PSNR as a metric. Additionally we check how accurately the motion across frames is preserved by computing the end-point error (EPE) of flow across the generated frames with respect to the flow from the ground truth frames.

5.3.2 Implementation Details

During training, model weights are initialized from a normal distribution with zero mean and a standard deviation of 0.01. We update all weights with a mini-batch of size 4 in each iteration. To augment the dataset, 128×128 patches are cropped at random locations and horizontally mirrored at random. The model is trained with

Figure 5.6: **Qualitative comparison**. Two input images and zoomed in regions are shown in the first row. The 2nd/6th rows show results of the method by [Jin et al., 2018]. The 3rd/7th to 5th/9th rows show the performance of our model trained after one (**B**), two (**BG**) and three stages (**BGH**).

an annealing learning rate scheme, starting with 10^{-4} and decreasing to 10^{-5} after convergence. The hyper-parameters α and β in Eq. (5.5) are empirically set as 0.0005 and 0.01.

The training procedure is as follows. The generator G is trained using BaseGAN at first, and then we incrementally train G with GramGAN and HeptaGAN to fine-tune the model. The generator G in the BaseGAN, GramGAN and HeptaGAN shares weights, thus a video recovery model G is obtained which is robust to resolve ambiguity (BaseGAN), preserve subtle movements (GramGAN) and recover disparity information (HeptaGAN). The ablation study compares the model performance after different stages of training. During inference, given a motion-blurred image, we generate seven frames in one forward pass of G.

Figure 5.7: **Example of interpolation of subtle motions.** 42 frames (from left to right, top to bottom) are extracted by the proposed method based on the input image shown in Fig. 5.1. Please check the movement of the rear light comparing the first frame with last one. Note that there are no 42 original frames as the input blurry frame is produced by averaging only 7 frames. By iteratively applying the model we are therefore able to create slow-motion videos from blurry images.

5.3.3 Ablation Study

In this section, we conduct experiments to investigate the effect of the different training stages. We show both qualitative results and quantitative results in the form of PSNR and EPE values. We compare the following models:

(1) **B** is the network trained as BaseGAN. The input to this model is a motion-blurred image, which is created from seven real consecutive frames.

(2) **BG** is the generator trained with BaseGAN (Stage-1) and GramGAN (Stage-2) stages. The input to the GramGAN is the output of the BaseGAN.

(3) **BGH** is the model trained after all three stages, adding the HeptaGAN third stage.

Table 5.1 shows the PSNR and EPE values. Performance increases after each training stage, with the fully trained model, **BGH**, achieving the best performance.

Fig. 5.6 shows qualitative results of the different models. Compared to model **B**, the results of **BG** shows more evident subtle movements across neighboring frames, suggesting the effectiveness of learning motion dynamics using GramGAN. The **BGH** model recovers more details and creates sharper images due to the disparity recovery. Please check the area marked with the yellow bounding boxes. The contrast of digit “3" by **BGH** is higher than **BG**. The ear of the man is also recovered

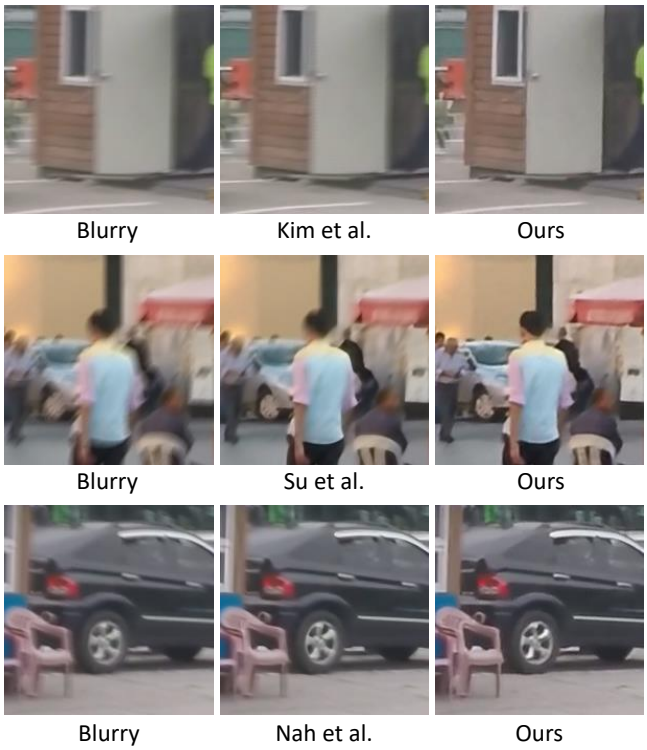


Figure 5.8: **Comparison with deblurring methods.** Methods provided by [Hyun Kim and Mu Lee, 2015], [Su et al., 2017b] and [Nah et al., 2017a] are specialized for recovering a sharp frame from a blurry image.

with more details by **BGH**.

Fine-grained Motion Interpolation. We are able to recover more than seven frames by iteratively applying model G to output frames. Seven output frames form six groups (I_1I_2, I_2I_3, ... , I_6I_7), and each can be averaged to produce another blurry image, which can be fed in our generator to again produce seven frames. By doing so, we recover $6 \times 7 = 42$ frames with extremely subtle motions from one blurry image, as shown in Fig. 5.7. We can even recover arbitrarily many frames by repeating this procedure. This demonstrates our model can be employed to disassemble a single motion-blurred image into multiple frames with interpolated fine-grained motion dynamics across frames.

5.3.4 Comparison with Existing Methods

We compare our method with different methods, including [Jin et al., 2018], [Pan et al., 2019], [Purohit et al., 2019], [Nah et al., 2017a], [Hyun Kim and Mu Lee, 2015] and [Su et al., 2017b]. [Jin et al., 2018], [Pan et al., 2019] and [Purohit et al., 2019] are the state-of-the-art methods for extracting image sequences from a motion-blurred image. [Nah et al., 2017a], [Hyun Kim and Mu Lee, 2015] and [Su et al., 2017b] are popular image deblurring methods. Table 5.1 shows quantitative results. Our method achieves higher PSNR values than [Jin et al., 2018], [Pan et al., 2019] and

Figure 5.9: **Results on the KITTI dataset**. The first column shows details of the two blurry input images in the top row, and the following seven columns show images generated by the proposed model. The subtle motion outlined by boxes with different colors shows that the model generalizes well to blur caused by bokeh.

[Purohit et al., 2019]. The smaller EPE value suggests that our method is better able to learn subtle motion across frames. We suspect the improvement is attributed to our specific handling of the challenges faced by extracting video from a single motion-blurred image. Figs. 5.1 and 5.6 show qualitative comparisons, highlighting the improved ability of our method to recover subtle motion and image details.

We also compare our method with image deblurring methods [Nah et al., 2017a]. Since deblurring methods typical output only a single image, we select the central frame of our reconstruction for comparison. As shown in Table 5.1, our method outperforms the one in [Nah et al., 2017a]. This may be explained by the fact that we use consecutive sharp frames to produce motion-blurred images during training, while [Nah et al., 2017a] only trains with one sharp image per motion-blurred image. Qualitative results comparing with [Nah et al., 2017a], [Hyun Kim and Mu Lee, 2015]

and [Su et al., 2017b] are shown in Fig. 5.8. The proposed method produces sharper and more realistic frames.

5.3.5 Generalization to Other Types of Blur

Our model is trained on the GOPRO dataset, within which the blur artifacts are mainly caused by camera shake. In this section we apply our method to images containing a different type of blur. The KITTI dataset [Geiger et al., 2013] includes images captured by a camera mounted on a moving vehicle, thus the dominant blur is caused by bokeh rather than camera shake. We test our model on this dataset and show example results in Fig. 5.9. The results demonstrate that the proposed method is able to recover sharper frames with evident subtle motion across neighboring frames and rich details for various kinds of blur artifacts.

5.4 Conclusion

The main contribution of this chapter is the proposed detail-aware network, which is a cascaded generator to extract an image sequence from a blurry image. To handle the problems of ambiguity, subtle motion, and loss of details, we train a model using a BaseGAN constrained with optical flow, a GramGAN, using a Gram matrix as motion representation, and a HeptaGAN with a bi-cyclic constraint. Experimental results demonstrate that our generator not only produces compelling results but also outperforms state-of-the-art methods.

Chapter 6

Deblurring: A Large-Scale Multi-Cause Blurry Dataset

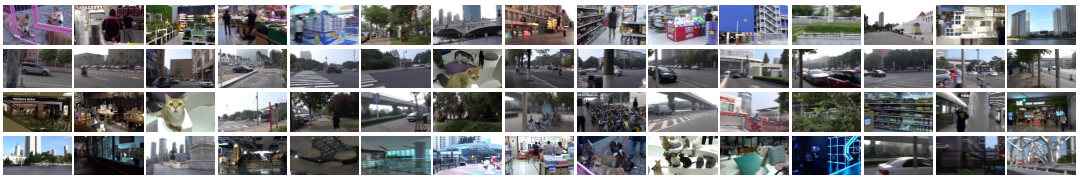
This chapter is about benchmarking current deep deblurring methods. Blur artifacts can seriously degrade the visual quality of images, and numerous deblurring methods have been proposed for specific scenarios. However, in most real-world images, blur is caused by different factors, e.g., motion and defocus. In this chapter, we address how different deblurring methods perform on general types of blur. For in-depth performance evaluation, we construct a new large-scale multi-cause image deblurring dataset including real-world and synthesized blurry images with mixed factors of blurs. The images in the proposed MC-Blur dataset are collected using different techniques: convolving Ultra-High-Definition (UHD) sharp images with large kernels, averaging sharp images captured by a 1000 fps high-speed camera, adding defocus to images, and real-world blurred images captured by various camera models. These results provide a comprehensive overview of the advantages and limitations of current deblurring methods. Further, we propose a new baseline model, level-attention deblurring network, to adapt to multiple causes of blurs. By including different weights of attention to the different levels of features, the proposed network derives more powerful features with larger weights assigned to more important levels, thereby enhancing the feature representation. Extensive experimental results on the new dataset demonstrate the effectiveness of the proposed model for the multi-cause blur scenarios.

6.1 Introduction

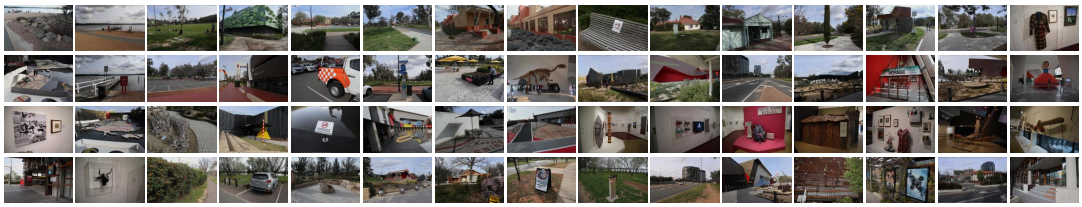
Deblurring has been widely used in applications such as medical image analysis, computational photography, and video enhancement. Traditional methods usually formulate the task as an inverse filtering problem, using the blur model

$$I_B = I_S * K + \sigma_N \,, \tag{6.1}$$

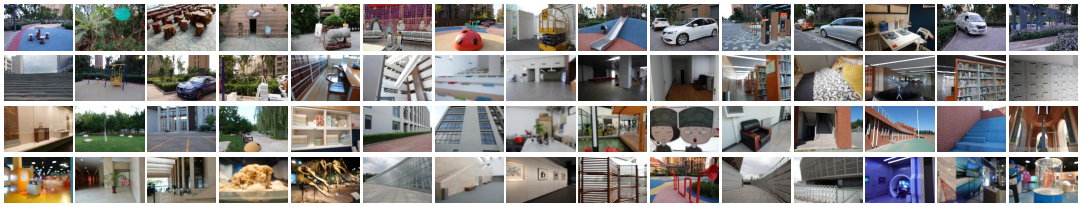
where I_B is the observed blurry image, I_S is the latent sharp image, K is the unknown blur kernel, σ_N is the additive noise, and $*$ is the convolution operation used to model



(a) Exemplar motion-blurred images based on real high-FPS cameras.



(b) Exemplar motion-blurred UHD images based on large blur kernels.



(c) Exemplar defocus blurry images.

Figure 6.1: **Exemplar images from the proposed MCID dataset**. It consists of a large number of real high-FPS motion-blurred images, large blur kernel based UHD motion-blurred images, and defocus blurry images, real-world blurry images, respectively.

the blur. The problem is ill-posed, because we need to estimate I_B, but I_S and K are both unknown. Prior models like nature image statistics have been employed in some work to constrain the solution space. Estimating I_S using this formula typically involves iterative estimation, which is time-consuming.

Recently, deep learning deblurring models have achieved impressive results. These models require a large number of pairs of corresponding sharp and blurry images to train networks in an end-to-end manner. To obtain pairs of images, many existing datasets are created by averaging continuous frames, by convolving with blur kernels, or by directly taking photos with two cameras with different shutter durations. Although these datasets have advanced the state of the art of deep deblurring models, there are several issues with these datasets. (1) As shown in [Nah et al., 2019a], averaging sharp images of low frame rate to synthesize blurry images can cause unnatural blur. For datasets specific to motion blur, the contained images are usually averaged from images captured by devices with relatively slow shutter speed, as in the GoPro dataset (240 FPS), or from images in interpolated high FPS videos rather than physical high FPS videos, e.g., the REDS dataset [Nah et al., 2019a]. (2) For datasets with blur based on convolution with a kernel, e.g. the dataset from Köhler

et al. [Köhler et al., 2012], the number of images is insufficient for training deep networks. At the same time, the kernel size is relatively small and the images are not high-definition images. With an increasing number of devices being able to capture Ultra-High-Definition (UHD) images, previous datasets are unable to handle such images. (3) For datasets of real-world blurry images, it typically requires complex procedure to process the images. This might lead to issues such as the alignment problem [Rim et al., 2020a]. (4) Moreover, defocus is a very popular type of blur. While most of the existing methods do not pay attention to it, and there are fewer datasets of defocus blur and the scale is usually not large [Abuolaim and Brown, 2020b].

To overcome these limitations, we build a large-scale dataset including images of blur caused by multiple factors, named as Multi-Cause Image Deblurring (MCID) dataset (shown in Fig. 6.1). This dataset is composed of four subsets. The first one contains images averaged from sharp images for motion blur. Different from existing ones, the frame rate of the sharp images, carefully captured by us with a ultra-high-speed camera, is as high as 1000 FPS. Meanwhile, this subset contains also blurry images from the sharp images captured by other different types of device, with various frame rates like 250 and 500 FPS. With different types of device and different settings of FPS, this subset mimics various motion blurs in the real world. The second subset contains motion-blur images based on convolving sharp images with blur kernels. Due to the popularity of device supporting high definition, we capture a large amount of UHD images of 4K+ resolution. These UHD images are used to convolve with blur kernel of big size, thus making the evaluation and development of deep deblurring methods convenient. The third subset is specific for the defocus blur. We also capture a set of images with the effect of defocus to assess the performance of current state-of-the-art methods. The last subset is composed of real-world blurry images captured by different kinds of device, like mobile phones (iPhone, Huawei, Samsung, etc). Being short of ground truth, this subset is dedicated to evaluate methods from the qualitative perspective.

Moreover, we also propose a novel network for deblurring based on the attention mechanism, which is called Level-Attentive Deblurring Network (LADN). This network integrates a Level Attention Module (LAM) to learn the dependency among features from different levels. With the different attentions (weights) for the different levels of features, important features are emphasized and redundant features are neglected. Thus the derived more powerful feature representations result in better performance. Experimental study results on the MCID dataset verify the advantage of our method over the existing ones including the DBGAN model.

6.2 The MCID Dataset

The progress of the deblurring problem highly relies on the various datasets in the community. As we have introduced above, there are issues with them. To benchmark the current state-of-the-art image deblurring methods in various conditions,

Table 6.1: **Representative benchmark datasets for evaluating single image deblurring algorithms.**

Dataset	Sharp Images	Blurred Images
Levin *et al.* [Levin et al., 2009]	4	32
Sun *et al.* [Sun and Hays, 2012]	80	640
Köhler *et al.* [Köhler et al., 2012]	4	48
Lai *et al.* [Lai et al., 2016]	108	300
GoPro [Nah et al., 2017a]	3,214	3,214
HIDE [Shen et al., 2019]	8,422	8,422
Blur-DVS [Jiang et al., 2020]	2,178	2,918
Rim *et al.* [Rim et al., 2020b]	4,556	4,556
Abuolaim *et al.* [Abuolaim and Brown, 2020b]	500	500
RHFPSM-250FPS	**25,000**	**25,000**
RHFPSM-500FPS	**25,000**	**25,000**
RHFPSM-1000FPS	**37,500**	**37,500**
LKUHDM	**2,000**	**10,000**
LSD	**22,400**	**22,400**
RMBQ	**-**	**10,000**

we thus correspondingly build a large-scale MCID dataset, including images with blur caused by multiple causes. The MCID is composed of four sets, which respectively correspond to images of motion blur by averaging continuous frames, images of motion blur by convolving with blur kernels, images of defocus blur and images of real-world blur. Table 6.1 compares the MCID dataset with the existing representative ones in details. The four sets are introduced in the following.

6.2.1 The Real High-FPS Based Motion-blurred Set (RHFPSM)

Averaging continuous frames within a time window to generate motion-blurred images is a popular synthesis operation. For example, sharp images of frame rate as 240 FPS are averaged to produce blurry images in the commonly employed GoPro dataset [Nah et al., 2017b]. However, as studied by Nah et al. in [Nah et al., 2019a], if the frame rate of the images to be averaged is not sufficiently high, the synthesized motion blur could be unnatural. They thus record videos of 120 FPS and interpolate them into ones of virtual 1920 FPS. As the interpolation is conducted by engineered CNN networks, the interpolated missing information can never be the same as the information recorded by camera with physical high shutter speed. To remedy this, we contribute the set of motion-blurred images from real high FPS sharp images, termed as the Real High-FPS Based Motion-blurred set.

Specifically, we have three settings in the set. The first setting corresponds to the highest FPS, as high as $1,000$ FPS. The sharp videos are recorded using a Sony RX10 camera. There are $30,000$ and $7,500$ images for training and testing respectively in this setting. The sharp images of the second setting are also captured by the Sony

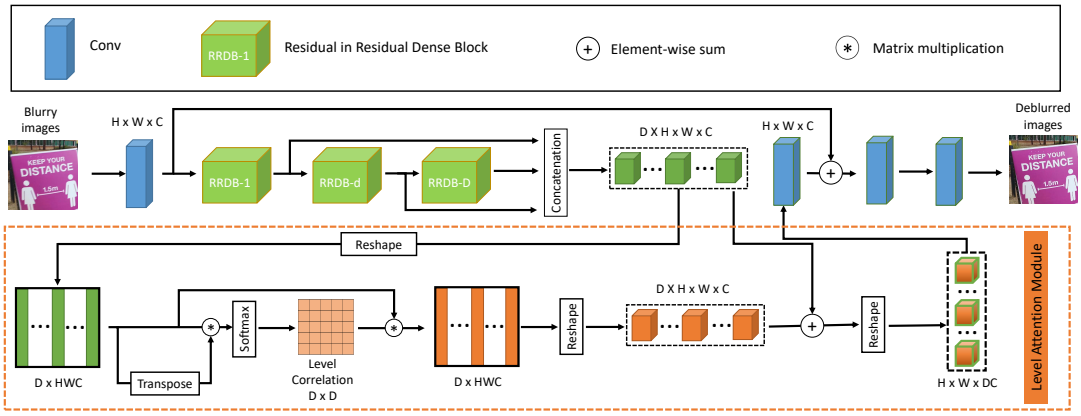


Figure 6.2: **The architecture of the proposed Level Attentive Deblurring Network.** The LADN takes a blurry image as input, and use RRDB and convolution to extract feature maps from different levels, which are further fed into a Level Attention Module to learn the correlations among different levels. Finally, a set of layers are stacked to generate the deblurred image.

RX10 camera, and the FPS is set as 500 FPS when recording. For the second setting, the training and testing sets contain 20,000 and 5,000 images. It is emphasized that though the frame rate of our dataset is not as high as that (1920 FPS) in the REDS dataset [Nah et al., 2019a], one should notice the difference between videos captured by real high-speed camera and ones interpolated from low FPS videos. The third setting is corresponding to images of frame rate as 250 FPS, captured by mobile device like iPhone, Huawei phones and Sony RX10 camera. For training and testing, this setting contains images of 20,000 and 5,000, respectively. All the images are resized via bicubic downsampling to reduce noise. The resolution in this set is either 960×540 or 640×360.

6.2.2 The Large-Kernel Based UHD Motion-blurred Set (LKUHDM)

There is another way to synthesize motion blur in images, which is convolving images with blur kernel. Existing datasets adopting this way either use low resolution images or small-size blur kernels. For example, when the resolution is lower than 2K, the size of employed blur kernel is usually set as $15, 17, 21, 23, 25, 27$, typically smaller than 50. Deblurring images of 4K+ resolution requires restoration of more details, which might not be feasible if models are trained with low-resolution images. On the contrary, we capture sharp images of 4K+ resolution (the resolution of some images is as high as 6K), composing the Large-Kernel Based UHD Motion-blurred (LKUHDM) set. To convolve with the sharp images, we utilize blur kernels of size as $111, 131, 151, 171$ and 191, respectively. The training subset and the testing subset contain individually 8,000 and 2,000 images.

6.2.3 The Large-Scale Defocus Blurred Set (LSD)

Defocus effect is pleasing in artistic photography, while it does not attract attention as much as that for motion blur in the deep learning era. There is a latest dataset proposed for the defocus deblurring in [Abuolaim and Brown, 2020b], which includes 500 images. But the scale of it is not large, which can hardly fulfill the demanding of training satisfactory deep neural networks. Meanwhile, it mainly focuses on the dual-pixel problem.

we build a large-scale defocus blurred set LSD, by capturing $18,000$ image pairs of sharp image and blurry image with the defocus effect as the training set, and $4,400$ image pairs for testing. The resolution is at least 900×600. To obtain the pairs of training and testing samples, we manually control the focus like [Abuolaim and Brown, 2020b] to obtain the defocus blurry images and their corresponding sharp ones.

6.2.4 The Real Mixed Blurry Qualitative Set (RMBQ)

The above three sets aim to simulate the blur with different operations based on sharp images. However, the blur artifacts in the real world are difficult to approximate. For instance, the real-world blur in images could be a mixture of multiple reasons, like the blur caused by both the camera shake and the fast object motion. Thus it is difficult to guarantee the generalization of models trained with images of a specific kind of blur. We thus capture a set of blurry images with various device, including both high-end digital camera and convenient mobile phones (iPhone, Samsung, Huawei, etc). The total number of images in this Real Mixed Blurry Qualitative (RMBQ) set is $10,000$. This set can be used only for qualitative testing, as there is not ground truth for these blurry images.

6.3 The Level-Attentive Deblurring Network

We develop a neural network, called Level-Attentive Deblurring Network, LADN, for the image deblurring task. The network integrates an effective level attention module to enhance the representation power of the features. In the following, we firstly introduce the network architecture of the LADN, and then represent the level attention module.

6.3.1 Network Architecture

The whole architecture of the LADN is shown in Fig. 6.2. Given a blurry image, convolution is applied to extract features. Then a sequence of Residual in Residual Dense Blocks (RRDB) [Wang et al., 2018c] are employed to extract different levels of features. The level attention module is used to derive a 2D matrix to measure the correlation among the different levels of features. By paying different weights of attention to the different levels of features, we thus derive more powerful features

with great weights on the more important feature levels and little weights on the redundant feature levels. The attended features are skip-connected to the primarily extracted features, and processed by several convolutional layers to produce the finally deblurred image.

To be specific, the blurry image I_B is input into the LADN, and after the convolutional layers, the primary feature F_P is extracted. As mentioned above, a set of RRDB extracts different levels of features, which are denoted as $F_{RRDB-1}, F_{RRDB-2}, \cdots, F_{RRDB-D}$. These different levels of features are concatenated and processed by the level attentive module. It can be represented as,

$$F_{LAM} = \Phi(con(F_{RRDB-1}, F_{RRDB-2}, \cdots, F_{RRDB-D})), \tag{6.2}$$

where D is the number of different levels, $con(\cdot)$ means the concatenation operation, Φ is a function of the level-aware attention mechanism, approximated by a network parameterized with $\mathbf{w}_{LAM}$.

The feature F_{LAM} is of level-wise attentions, emphasizing level of features with great attention/weight and neglecting level of features with little attention/weight. F_{LAM} is then added with the primary feature F_P by a skip connection, and further processed by several convolutional layers to produce the sharp image, formulated as,

$$I_S = \Theta(F_P \oplus F_{LAM}), \tag{6.3}$$

where Θ indicates the process of the convolutional layers and $\oplus$ is the element-wise addition operation.

6.3.2 Level Attention Module

Features play roles of different importance in the deblurring task. Without attention mechanism, the features maps in different levels will be treated without discrimination. On the contrary, the LAM aims to learn different levels of attention (weight) for the feature maps in different levels.

To achieve this goal, the feature maps from the sequential levels are firstly concatenated. The concatenation is firstly reshaped as a 2D matrix with size $D \times HWC$, where D, H, W, C are respectively the number of RRDBs, height, width and channel. This matrix is multiplied with its transpose to derive a 2D matrix of size D by D. Each element in this matrix represents the correlation between the two feature levels corresponding to the column and row index. This correlation matrix is multiplied with the reshaped feature concatenation, and the derived features are reshaped into the $D \times H \times W \times C$ feature tensor. The feature tensor can be taken as feature residual, and added to the original feature concatenation in an element-wise manner. The result additive features are reshaped as $H \times W \times DC$ tensor by absorbing the level number D. Subsequently, convolution is applied to decrease the channel number from DC to C. The feature is again added with the primary features F_P in an element-wise manner, which is processed by several convolutional layers to generate the deblurred image I_S.

Table 6.2: **Performance comparison of representative methods for deep image deblurring on the proposed RHFPSM set.**

Method	DeepDeblur	DeblurGAN	SRN	DeblurGAN-v2	DMPHN	DBGAN	**LADN**
250fps	30.38/0.8766	24.89/0.6364	30.57/0.8799	26.99/0.8061	30.42/0.8768	27.89/0.8191	**31.19/0.8918**
500fps	31.08/0.8974	24.66/0.6748	31.54/0.9051	27.67/0.8320	31.43/0.9018	28.36/0.8388	**31.77/0.9104**
1000fps	32.41/0.8966	25.20/0.6535	32.69/0.9016	29.81/0.8461	32.41/0.9096	29.66/0.8318	**32.77/0.9031**

6.4 Experiments

In this section, we benchmark existing deblurring methods on our proposed MCID dataset. Specially, we first introduce the evaluated deblurring methods and protocol in Sec. 6.4.1. Then, we evaluate the performance of them and our proposed LADN on different motion-blurred images including real high-fps based motion-blurred images (Sec. 6.4.2) and large-kernel based motion-blurred UHD images (Sec. 6.4.3), defocus images (Sec. 6.4.4), and real mixed blurry images (Sec. 6.4.5). We further compare our proposed LADN model with current state-of-the-art methods on the GoPro dataset (Sec. 6.4.6). Finally, the efficiency analysis on UHD blurry images is reported in Sec. 8.3.5.

6.4.1 Evaluated Deblurring Methods and Implementation Details

We evaluate six representative state-of-the-art methods on the proposed MCID dataset, including: DeepDeblur [Nah et al., 2017b], DeblurGAN [Kupyn et al., 2018], SRN [Tao et al., 2018], DeblurGAN-v2 [Kupyn et al., 2019], DMPHN [Zhang et al., 2019b], and DBGAN [Zhang et al., 2020c].

Among these methods, DeepDeblur and SRN are Multi-scale networks which first generate a small-size sharp image to help obtain the final sharp version of its original size. DeblurGAN, DeblurGAN-v2 and DBGAN are GAN-based models which use a generator to restore sharp images and apply a discriminator to push the deblurred images to be more realistic. DMPHN is a multi-patch network, which first removes blur from small patches to help the final deblurring operation. we use the same settings as in the original publications to re-train six models on the proposed MCID dataset.

The architecture of LADN is shown in Fig. 6.2, which includes four traditional convolution and eight RRDB modules, except the model on the LKUHDM dataset, which uses three RRDB modules to save memory. The convolution kernel size is 3×3. For the proposed LADN, we initialize its weight using a Gaussian distribution with zero mean and a standard deviation of 0.01. All weights are updated after learning a mini-batch of size 8 in each iteration. During the training stage, we crop a 256×256 patch at any location and randomly flip frames to augment the data. We use ADAM to update our model with a learning rate of 10^{-4}. All the deblurred results are quantitatively assessed using PSNR and SSIM in the RGB space.

Table 6.3: **Performance comparison of representative methods for deep image deblurring on the proposed LKUHDM set.**

Method	PSNR	SSIM
DeepDeblur [Nah et al., 2017b]	22.23	0.6322
DeblurGAN [Kupyn et al., 2018]	20.39	0.5568
SRN [Tao et al., 2018]	22.28	0.6346
DeblurGAN-v2 [Kupyn et al., 2019]	21.03	0.5839
DMPHN [Zhang et al., 2019b]	22.20	0.6378
DBGAN [Zhang et al., 2020c]	21.52	0.6025
LADN	**22.49**	**0.6238**

Table 6.4: **Performance comparison of representative methods for deep image deblurring on the proposed LSD set.**

Method	PSNR	SSIM
DeepDeblur [Nah et al., 2017b]	20.73	0.7218
DeblurGAN [Kupyn et al., 2018]	20.04	0.6335
SRN [Tao et al., 2018]	21.66	0.7664
DeblurGAN-v2 [Kupyn et al., 2019]	21.13	0.6964
DMPHN [Zhang et al., 2019b]	21.23	0.7519
DBGAN [Zhang et al., 2020c]	21.56	0.7536
LADN	**21.83**	**0.7658**

6.4.2 Results on Real High-FPS Based Motion-blurred Images

We first evaluate the state-of-the-art image deblurring methods and our proposed LADN on the RHFPSM dataset to explore their performance on motion-blurred images. Table 6.2 shows the results of the quantitative comparison. We can find that DeepDeblur, SRN and DMPHN achieves better performance in terms of PSNR and SSIM. One reason is that they use pixel-level loss function to update their models, which have the advantage of obtaining high values of full-reference pixel-based metrics. DeblurGAN, DeblurGAN-v2 and DBGAN use a discriminator to push the deblurred images to be more realistic. This forces their networks to not only focus on pixel-wise quality, but also pay attention to whole images. The proposed LADN uses a level attention module to learn the correlation of features from different layers. Therefore, it can make better use of feature maps and achieve better performance than other methods including DBGAN. We also show a visual comparison of different methods on the RHFPSM dataset in Fig. 6.4(a), which also verifies that the proposed LADN is able to remove blur artifact and restore sharp images.

6.4.3 Results on Large-Kernel Based Motion-blurred UHD Images

The above section evaluates the state-of-the-art deblurring methods on motion-blurred images synthesized by averaging real high-fps frames. In this section, these methods

(a) Input (b) [Tao et al., 2018]

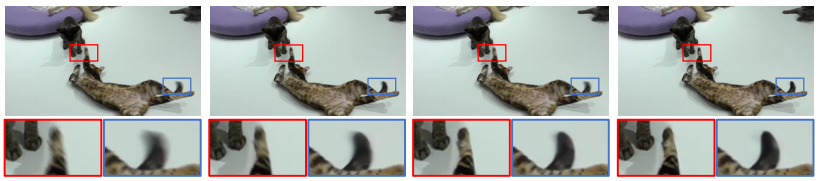
(c) LADN without LAM (d) LADN

Figure 6.3: **Visual results of different models on the GoPro dataset**. From left to right: input, results of [Tao et al., 2018], LADN without LAM, and LADN with LAM. Best viewed in color.

are evaluated on the LKUHDM dataset of images synthesized by convolving with kernels and Table 6.3 shows the results of the quantitative comparison. We can find that the values of PSNR and SSIM of all the methods are significantly lower than those in the Table 6.2. One reason is that we use large-size blur kernels to synthesize blurry images, which makes the deblurring task more difficult. The other reason is that, comparing with HD (2K) image deblurring, deblurring of the UHD (4K+) images requires to recover more details. The proposed LADN demonstrates its effectiveness for this blur cause, as shown by the results in Table 6.3. Again, we show qualitative results corresponding to this blur cause in Fig. 6.4(b).

6.4.4 Results on Large-Scale Defocus Blurred Images

To investigate the performance of the state-of-the-art deblurring methods along with our proposed LADN in the case of defocus blur, we conduct a comparison study on the LSD dataset. The quantitative and the qualitative results are respectively shown in Table 6.4 and Fig. 6.4(c). It is obvious that defocus image deblurring is a more difficult problem compared with deblurring of motion-blurred images. The current deep deblurring methods can restore high-quality motion-deblurred images synthesized by averaging neighbouring frames. However, the performance of defocus deblurring is significantly poor. Compared with the synthesized motion blur, defocus effect ex-



(a) Visual results on the RHFPSM set.

(b) Visual results on the LKUHDM set.

(c) Visual results on the LSD set.

(d) Visual results on the RMBQ set.

Figure 6.4: **Test results on the proposed MCID dataset**. From left to right: input, results of [Zhang et al., 2019b], [Tao et al., 2018], and ours. Best viewed in color.

hibits distinctive properties. Dedicated studies should be carried out for deblurring images of defocus blur.

6.4.5 Results on Real Mixed Blurry Images

In addition, we show the performance of the current state-of-the-art deep deblurring methods in the case of real-world scenarios. Taking a real-world blurry image, we process it by different methods to restore the deblurred one, and the results are shown in Fig. 6.4(d).

Table 6.5: **Ablation study results and comparison with the state-of-the-art deep deblurring methods on the GoPro dataset.**

Method	PSNR	SSIM
DeepDeblur [Nah et al., 2017b]	29.08	0.914
SRN [Tao et al., 2018]	30.26	0.934
DeblurGAN-v2 [Kupyn et al., 2019]	29.55	0.934
DMPHN [Zhang et al., 2019b]	30.25	0.935
DBGAN [Zhang et al., 2020c]	30.43	0.937
RNNDeblur [Zhang et al., 2018a]	29.19	0.931
Shen *et al.* [Shen et al., 2019]	30.26	0.940
AlJadnnay *et al.* [Aljadaany et al., 2019]	30.35	0.961
Gao *et al.* [Gao et al., 2019]	30.92	0.942
Park *et al.* [Park et al., 2020]	31.15	0.945
LADN (w/o LAM)	31.19	0.942
LADN	**31.43**	**0.947**

Table 6.6: **Speed comparison of state-of-the-art deep deblurring methods (in seconds).**

Method	DeepDeblur	DeblurGAN	SRN	DeblurGAN-v2	DMPHN	DBGAN	**LADN**
Speed	26.76	2.46	28.41	3.63	17.63	31.62	**1.67**

6.4.6 Ablation Study on the GoPro Dataset

To further evaluate the propose LADN, we assess it on the GoPro dataset. In addition, to demonstrate the effective of the LAM, we also test a variant of LADN without the LAM. The ablation study results are shown in Table 6.5. The variant without LAM achieves satisfactory performance, compared with the existing ones. Equipping the LAM further boosts the performance. The qualitative results are shown in Fig. 6.3.

6.4.7 Efficiency Analysis on UHD images

Efficiency should be taken into consideration when the image resolution is high, especially for UHD images. We have conducted a study to investigate the performance of deblurring methods in the case of UHD images, as shown in Table 6.3. In this section we report the speed of existing methods along with ours, on the LKUHDM dataset. The study is carried out using an ordinary platform with P40 GPU. Table 6.6 shows the results. Among these methods, DeepDeblur, SRN, DMPHN and DBGAN take more than ten seconds to process an UHD image. The rest ones needs only a few seconds (less than ten) to accomplish the deblurring task of a UHD image. Our proposed LADN runs the fastest among all the methods.

6.5 Conclusion

This chapter has introduced a new large-scale dataset to benchmark current deblurring methods on single images with blur caused by various factors. We also propose a layer-attentive deblurring network, LADN, which achieves high performance on the proposed MCID dataset and the public GoPro dataset, in terms of PSNR and SSIM metrics, as well as in terms of run time. For future work, I will consider building new datasets for benchmarking video deblurring methods.

Chapter 7

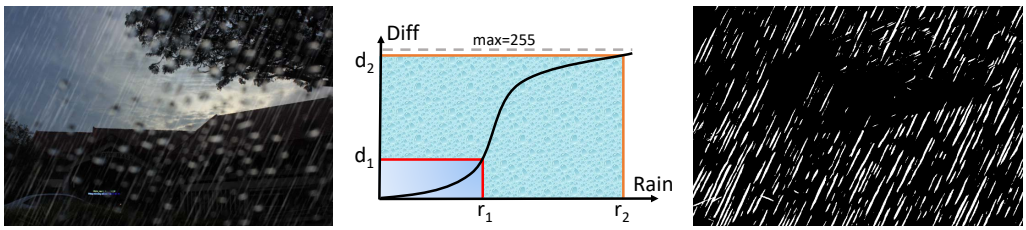
Deraining: Joint Rain Streak and Raindrop Removal

This chapter is about single image deraining. Rain streaks and rain drops are two natural phenomena, which degrade image capture in different ways. Currently, most existing deep deraining networks take them as two distinct problems and individually address one, and thus cannot deal adequately with both simultaneously. To address this, we propose a Dual Attention-in-Attention Model (DAiAM) which includes two DAMs for removing both rain streaks and raindrops. Inside the DAM, there are two attentive maps - each of which attends to the heavy and light rainy regions, respectively, to guide the deraining process differently for applicable regions. In addition, to further refine the result, a Differential-driven Dual Attention-in-Attention Model (D-DAiAM) is proposed with a “heavy-to-light" scheme to remove rain via addressing the unsatisfying deraining regions. Extensive experiments on one public raindrop dataset, one public rain streak and our synthesized joint rain streak and raindrop (JRSRD) dataset have demonstrated that the proposed method not only is capable of removing rain streaks and raindrops simultaneously, but also achieves the state-of-the-art performance on both tasks.

7.1 Introduction

As one of the commonest weather phenomenons, rain causes visibility degradation and destroys the performance of many computer vision systems, *e.g.*, object detection [Girshick, 2015; He et al., 2017], outdoor surveillance [Zheng et al., 2015; Han and Bhanu, 2005] and autonomous driving [Yang et al., 2019a; Li et al., 2019a]. Rain removal is to restore clean images from rainy ones, which is challenging due to its various types (i.e., , rain streaks and raindrops), and different intensities (i.e., , heavy and light rain).

In the last decade, a set of methods have been proposed for rain removal. For rain streak removal, some methods model the physical characteristics of rain and generate sharp version with various image priors [Sun et al., 2014a; Kang et al., 2011; Chen and Hsu, 2013; Zhang et al., 2006]. we have also witnessed significant progress of deep learning based methods [Fu et al., 2017a,b; Li et al., 2018d; Yang et al., 2017;



(a) Input rainy images (b) Rainy regions (c) Attention: rain streaks

(d) Attention: raindrops (e) DAM(odd) (f) DAM(dual)

Figure 7.1: **Analyses and deraining results.** (a) is an input rainy image. (b) describes the relationship of the rain intensity and the difference between rainy and clean images. (c) and (d) are the generated attention maps for rain streaks and raindrops, respectively. (e) and (f) are the deraining results of the proposed DAM with odd attention and dual attention, respectively.

Zhang and Patel, 2018b]. Some others focus on raindrop removal via detecting and removing raindrop using multiple images or single image [Roser and Geiger, 2009; Roser et al., 2010,?; Eigen et al., 2013; Qian et al., 2018]. Despite of the achieved promising performance, there still exist major challenges in rain removal:

- Rain streaks and raindrops are two related but different types. The rain streaks lead to the occlusion of objects and scene, while raindrops can cause change of shape. In the real world, both of them often appear simultaneously. However, most deep learning based deraining methods and datasets typically focus on one of them.

- As Fig. 7.1(b) shows, the pixel difference between clean and rainy images increases as the rain becomes heavier. Previous attention based methods use a fixed threshold d_1 to determine whether a pixel is part of rain regions. These methods focus only on the top-right heavy rainy region and ignore the bottom-left light rainy region. In this case, the efficacy of attention mechanism will be restricted if d_1 is set inappropriately large or small.

- For many cases like heavy rain, the current rain removal methods can remove rain to some extent and generate a derained image with less rain. However, it is difficult to further improve the performance by simply modifying the structure of deep networks like increasing the depth.

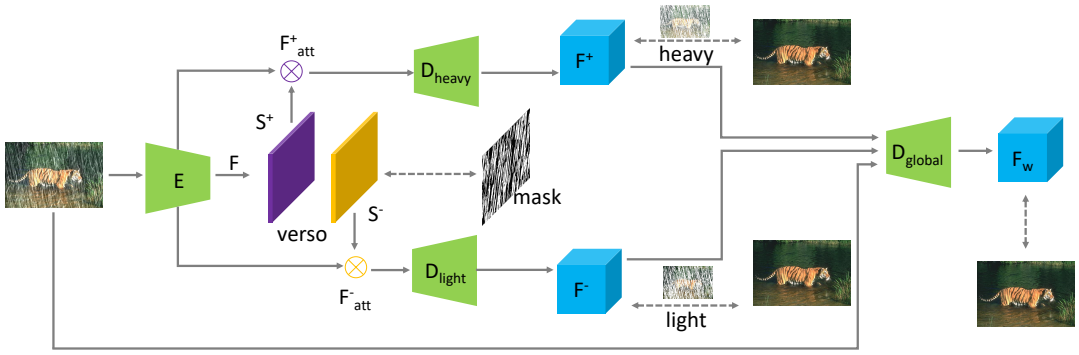


Figure 7.2: **The framework of DAM for image deraining.** It contains three main branches, i.e., , heavy-rain branch, light-rain branch and full-image branch. The dual attention sub-network in the middle is utilized to generate a pair of heavy-rain-aware and light-rain-aware maps to pointedly remove rain from different regions. The original rainy image and the intermediate results are then concatenated to generate the final deraining image.

To address the first and second problems, a new framework which exploits the cues from different types of rain is proposed. Specially, we propose a Dual Attention-in-Attention Model, termed as **DAiAM**, to remove rain streaks and raindrops, simultaneously. It contains two branches, corresponding to two Dual Attention Model (**DAM**). Each DAM removes one type of rain via simultaneously focusing on different rain intensities. Different from previous attention-based deraining methods, which learn only the attention map of heavy rain regions (top-right regions in Fig. 7.1(b)), an advantage of the DAM is that it also pays attention to the light rain regions (bottom-left regions in Fig. 7.1(b)). One pair of heavy-rain-aware and light-rain-aware attention maps is generated to help remove rain from multiple regions. As such, the proposed method avoids the negative effects from unsuitable thresholds. Fig. 7.1(e) and 7.1(f) show the attention maps for rain streaks and raindrops, respectively.

For the third challenge, a Differential-driven Dual Attention-in-Attention Model (**D-DAiAM**), is proposed based on a "*heavy-to-light*" scheme. The input rainy images and output derained images from DAiAM are processed with the proposed differential-driven module, guiding the learning of the following DAiAM to further remove rain with different intensities or different types.

In order to evaluate the performance of the proposed method on rain streak and raindrop removal, a joint rain streak and raindrop dataset (**JRSRD**), is built. The rain streaks and raindrops often happen simultaneously, thus evaluating methods in this scenery is necessary to verify the performance of different methods in the wild.

7.2 Method

We first take rain streak removal as an example to introduce the architecture and learning details of DAM. Then we represent DAiAM (Sec. 7.2.4) to jointly remove rain streaks and raindrops. Finally, a D-DAiAM framework (Sec. 7.2.5) is discussed to overcome the limitation of single model.

7.2.1 Overall Architecture of DAM

The overall architecture of the proposed **DAM** is shown in Fig. 7.2. A rainy image is fed into DAM to learn two attention maps, *i.e.*, heavy-rain-aware and light-rain-aware maps. The heavy-rain-aware map learns the attention which indicates the regions with heavy rain, and the light-rain-aware map represents the regions with light rain (Sec. 7.2.2).

Different from other deraining methods which directly concatenate the attention maps to generate final images, we produce two different kinds of intermediate results by two sub-networks in Sec. 7.2.3. The two attention maps provide not only attention to generate the final global deraining image, but also the reference to evaluate the performance of two sub-networks of DAM. Finally, the intermediate results concatenated with the input rainy image are put into a global decoder to generate the deraining image.

7.2.2 Dual Intensity-Aware Maps

In general, the DAM takes input images and produce weighting maps to focus on different spatial regions of images. By doing so, different sub-networks can exactly focus on different spatial regions that contribute most for differentiated image deraining. Specially, the proposed DAM take rainy images as input to capture the features F from the first-step encoder E. Then the feature maps are fed into two attention sub-networks to generate heavy-rain-aware and light-rain-aware maps, respectively. The heavy-rain-aware map S^+ can be defined as:

$$S^+ = g(W * F + b)\ , \tag{7.1}$$

where $*$, W and b denote respectively convolution, convolution filters and biases. g is the sigmoid function.

Then we can similarly generate the light-rain-aware map based on Eq. (7.1). The heavy and light rain regions are a pair of complementary regions. Thus a constraint of them is set as:

$$S^+ + S^- = 1\,. \tag{7.2}$$

The two attention maps are two weighting maps which denote different region-aware attentions from the input features. Based on them, it is easy for the following sub-networks to pay attention to different regions and obtain different outputs. The operation to obtain the different features based on the two attention maps can be

represented as,

$$F_{att}^{+} = F \otimes S^{+}\,, \tag{7.3}$$

$$F_{att}^{-} = F \otimes S^{-}\,, \tag{7.4}$$

where $\otimes$ denotes the channel-wise Hadamard matrix operation. F_{att}^{+} and F_{att}^{-} have the same size as F but are two re-weighted features by the two attention maps to focus on heavy-rain and light-rain regions, respectively. The S^{-} is the light-rain-aware attention map, where light-rain regions have higher weights and the heavy-rain regions have lower values. In order to guarantee that S^{+} learns the heavy-rain regions, we develop another constraint to make it focus on the heavy-rain regions and thus simultaneously push S^{-} to learn the light-rain regions. The loss function with this constraint is represented as

$$\mathcal{L}_{att} = \sum_{x=1}^{X}\sum_{y=1}^{Y} M_{(x,y)} - S_{(x,y)}^{+}\,, \tag{7.5}$$

where M is the rain-aware mask. X and Y are the width and height of the input features. Different from the previous methods [Yang et al., 2017; Wang et al., 2019a], which use a binary mask to represent the rain and no-rain regions, we apply a “soft" manner. Specially, we calculate the difference of images between the rainy and non-rainy versions and then normalize to the range between 0 and 1. This not only denotes whether the regions are rainy or not, but also represents the intensity of rain. In this way, we can avoid the negative effects caused by inappropriate thresholds and binary masks. Based on the above mechanism, two different attention maps are obtained with focus on heavy-rain and light-rain regions, respectively.

7.2.3 Attentive Deraining from Regional and Global Levels

After the two attention maps are generated, we can improve the performance of deep deraining networks with them as reference. Specially, the attended features with the heavy-rain-aware attention map S^{+} and light-rain-aware attention map S^{-} are sent into two decoder networks to reconstruct two different deraining images with focus on different regions. The learning process can be defined as:

$$\mathcal{L}_{heavy} = I_c - D_{heavy}(F_{att}^{+}, I_i)\,, \tag{7.6}$$

$$\mathcal{L}_{light} = I_c - D_{light}(F_{att}^{-}, I_i)\,, \tag{7.7}$$

where I_c denotes the clean image and I_i is the input rainy image. The encoder networks D_{heavy} and D_{light} generate two deraining images, and the attentions of them are different. $\mathcal{L}_{heavy}$ specially constrains the network D_{heavy} to mainly focus on the heavy-rain regions but consider less the light-rain regions due to the weighting values from S^{+}. The $\mathcal{L}_{light}$ pushes the D_{light} to remove rain from light regions. Finally, both of the intermediate deraining images are concatenated with the original rainy

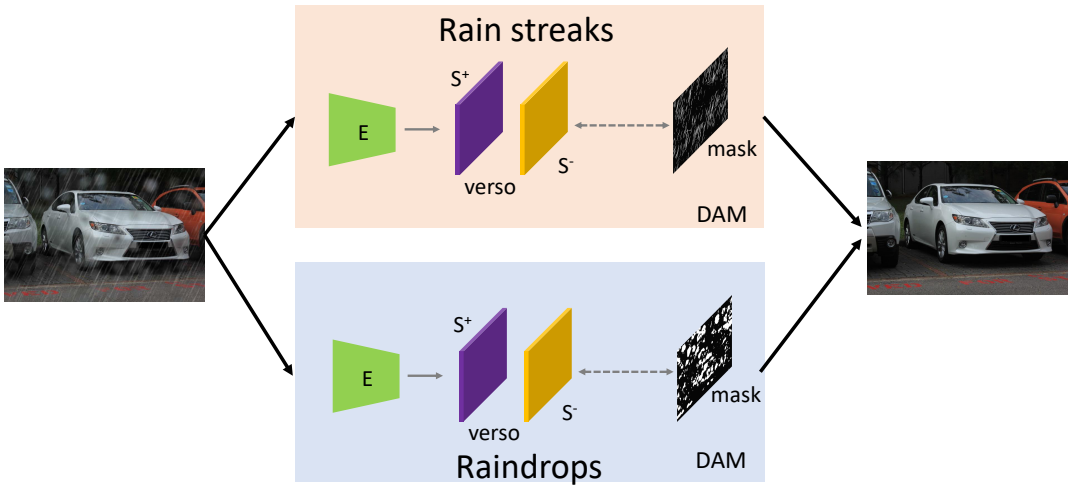


Figure 7.3: **The framework of DAiAM for joint rain streak and raindrop removal.** DAiAM takes a rainy image as input to capture attention maps for rain streaks and raindrops via two DAMs. Then the outputs of them are concatenated to generate final deraining result.

image to generate the final deraining image via a global decoder, denoted as:

$$I_o = D_{global}(F^+, F^-, I_i) \,, \tag{7.8}$$

where I_o is the derained image. We use MSE to update the model as

$$\mathcal{L}_{global} = \sum_{x=1}^{X} \sum_{y=1}^{Y} I_{c(x,y)} - I_{o(x,y)} \,. \tag{7.9}$$

The final loss function of the DAM contains $\mathcal{L}_{att}$, $\mathcal{L}_{heavy}$, $\mathcal{L}_{light}$ and $\mathcal{L}_{global}$, which is defined as,

$$\mathcal{L}_{DAM} = \alpha \cdot \mathcal{L}_{att} + \beta_1 \cdot \mathcal{L}_{heavy} + \beta_2 \cdot \mathcal{L}_{light} + \mathcal{L}_{global} \,, \tag{7.10}$$

where α, β_1 and β_2 are three parameters to balance different loss functions, respectively.

7.2.4 Dual Attention-in-Attention Model

As discussed above, raindrops and rain streaks are two different rain types and usually appear simultaneously in the real world. In this case, rain removal becomes a more challenging problem. Previous methods [Li et al., 2019c] often focus on removing one type of rain from rainy images. To simultaneously remove both of them, a Dual Attention-in-Attention Model, **DAiAM**, is proposed based on DAM.

Fig. 7.3 shows the core idea of DAiAM. Image of raindrops and rain streaks is fed into our proposed DAiAM, which has two branches to pay attention to removal of

raindrops and rain streaks, respectively. The branch for raindrop removal is similar to the method of removing rain streaks, which is represented in the above based on DAM. The main difference is that the attention loss function $\mathcal{L}_{att}$ is calculated based on the mask of raindrops, rather than rain streaks. In this way, the DAiAM first pays attention to two kinds of rain variations, and then focuses on two kinds of rain intensity in different branches. The final loss function of DAiAM is defined as,

$$\mathcal{L}_{DAiAM} = \mathcal{L}_{streak} + \mathcal{L}_{drop} + \mathcal{L}_{global} \ , \tag{7.11}$$

where $\mathcal{L}_{drop}$ and $\mathcal{L}_{streak}$ are two loss functions to remove rain drops and streaks, respectively. The loss functions of them are

$$\mathcal{L}_{streak} = \alpha \cdot \mathcal{L}_{att}^{streak} + (\beta_1 \cdot \mathcal{L}_{heavy}^{streak} + \beta_2 \cdot \mathcal{L}_{light}^{streak}), \tag{7.12}$$

$$\mathcal{L}_{drop} = \alpha \cdot \mathcal{L}_{att}^{drop} + (\beta_1 \cdot \mathcal{L}_{heavy}^{drop} + \beta_2 \cdot \mathcal{L}_{light}^{drop}), \tag{7.13}$$

where α, β_1 and β_2 are parameters to balance different loss terms. The attention loss function L_{att}^{drop} and L_{att}^{streak} are calculated based on the masks of raindrops and rain streaks, respectively.

7.2.5 Differential-Driven DAiAM (D-DAiAM)

Rain has different intensities and various types. Images exhibiting both rain streaks and raindrops also pose increasing difficulty of deraining. Deep deraining methods can remove rain to some extent and transfer the heavy-rain images to light-rain ones [Hu et al., 2019; Zhang and Patel, 2018b]. However, the performance of a single model is often limited. Simply increasing neural network depth is easy to exhaust the potential and difficult to further improve the performance of rain removal, even for some special heavy rain removal methods [Li et al., 2019b].

[Li et al., 2019c] show that light rainy images are easier to derain. Therefore, we propose a differential-driven dual attention-in-attention model, **D-DAiAM**, to remove various kinds of rain. Different from most methods [Li et al., 2019c] which aim to directly derive final deraining images via increasing the depth or width of a single model, we aim to remove heavy rains via transferring heavy rain to light rain and then to no rain in multiple stages. In each stage, we use a DAiAM to generate better visible deraining images and attention information driven by the *differential between the current output and original input*, and the *differential between the current and previous outputs*.

Specifically, this process is conducted via a differential-driven module. As shown in Fig. 7.4, we calculate two types of differential. One is the differential between the current output I_o^t and the original input I_i. By comparing these two items, the differential is able to guide the following stage to focus on the remaining rainy regions in I_o^t. The other is the differential between the current and the previous outputs (I_o^t and I_o^{t-1}). This differential leads the next stage to pay special attention to regions of the current output I_o^t which are not handled well in the current stage.

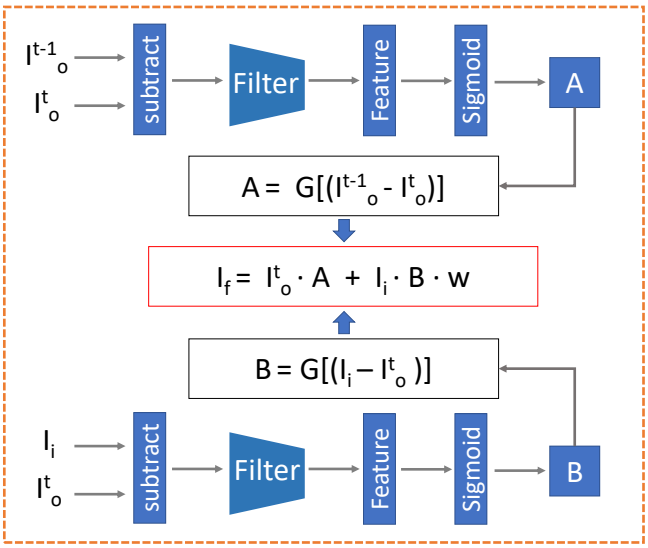


Figure 7.4: **The illustration of the differential-driven module.** It consists of three streams, i.e., , two differential streams and a fusion stream. The FilterNet inside it pointedly selects key regions to help remove rain in the next stage.

Based on these two kinds differential, we employ two *FilterNets* to generate soft maps A and B for our purpose, i.e., , the mark of regions needing special attention in the next stage. The FilterNet includes three convolutional layers with 2×2 kernels to perceive local regions, rather than directly using the input differences. We apply these two soft maps to the original input I_i and the current output I^t_o and fuse them, as defined in

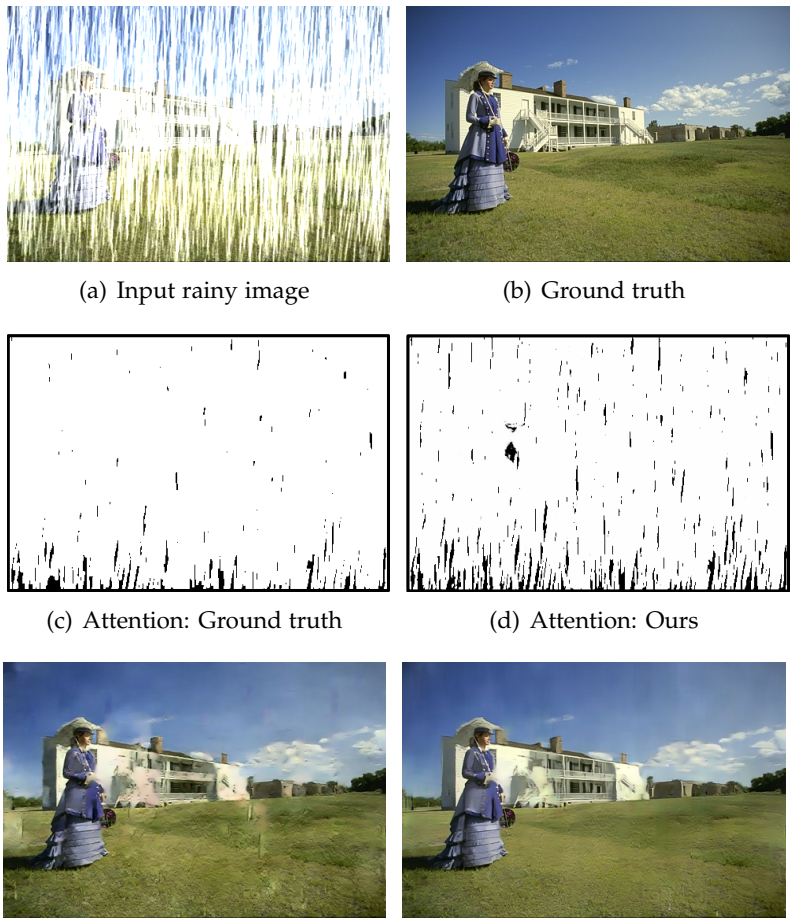
$$I_f = I^t_o \otimes A + I_i \otimes B \cdot w, \tag{7.14}$$

where w balances different types of differential.

The coarsest-level DAiAM locates in the begin of D-DAiAM. A latent deraining image is generated at the end of this stage. Even there still exists rain, the generated deraining image exhibits lighter rain. Then, the information from the coarsest level output is addressed by the differential-driven module, and then fed into finer-level network (which has a similar architecture as DAiAM) with deraining images. The final derained image is the output of the last DAiAM. The objective function to update the D-DAiAM is denoted as:

$$\mathcal{L} = \sum_{t=1}^{N} ||I^t_o - I_c|| \,, \tag{7.15}$$

where I^t_o is the derained image in the t-th stage and I_c is the ground-truth image.



(a) Input rainy image (b) Ground truth

(c) Attention: Ground truth (d) Attention: Ours

(e) Deraining: PReNet [Ren et al., 2019] (f) Deraining: Ours

Figure 7.5: **Heavy rain streak removal results of sample images from Rain Streak dataset [Yang et al., 2017].**

7.3 Experiments

We first introduce the implementation details. Then the performance of rain streak removal and raindrop removal is compared with the state-of-the-art methods on two public datasets. We develop a new dataset of joint rain streaks and raindrops and test different deraining methods on it. Further, ablation study is carried out to verify the components of our proposal. Finally, the application of deraining in real-world scenarios is demonstrated.

7.3.1 Implementation Details

The weights of the proposed networks are initialized with Gaussian distribution with zero mean and a standard deviation of 0.01. The parameters are updated after a mini-batch of size 4 in each iteration. In the training stage, 112×112 patches at random

Table 7.1: **Performance of different model structures on the Rain Streak dataset [Yang et al., 2017] in terms of PSNR and SSIM.**

Methods	PSNR	SSIM
GMM [Li et al., 2016]	15.05	0.425
DDN [Fu et al., 2017b]	21.92	0.764
RGN [Fu et al., 2017a]	25.25	0.841
JORDER [Yang et al., 2017]	26.54	0.835
RESCAN [Li et al., 2018d]	28.88	0.866
PReNet [Ren et al., 2019]	29.46	0.899
DAM	**29.99**	**0.905**
D-DAM	**30.35**	**0.907**

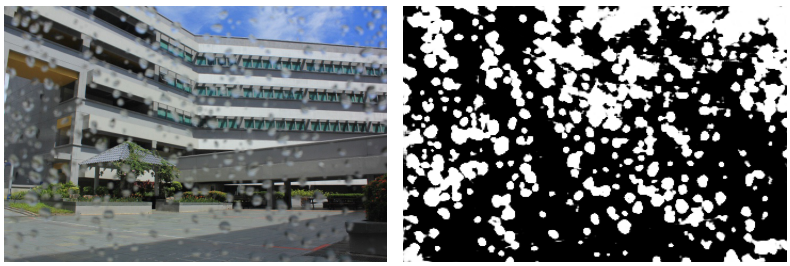
Table 7.2: **Performance of different model structures on the Raindrop dataset [Qian et al., 2018] in terms of PSNR and SSIM.**

Methods	PSNR	SSIM
DID-MDN [Zhang and Patel, 2018b]	24.76	0.7930
DDN [Fu et al., 2017b]	25.23	0.8366
JORDER [Yang et al., 2017]	27.52	0.8239
[Qian et al., 2018]	31.57	0.9023
DAM	**30.26**	**0.9137**
D-DAM	**30.63**	**0.9268**

locations of an image are cropped to increase the number of training samples. We also randomly flip training images (horizontally) to further augment the training set. The models are trained under a learning rate which starts with a value of 10^{-4} and reduces to 10^{-6} after the training has converged. The hyper-parameters α, β_1, β_2 and w are set as 0.8, 1.0, 0.3 and 0.5, respectively. To reduce training time, we apply one differential-driven module in our practice. The encoder E contains three residual blocks [He et al., 2016] and one LSTM layer. D_{heavy} and D_{light} contain one CNN layer, five residual blocks and another CNN layer. D_{global} contains two residual blocks and one CNN layer. The size of all the kernels in this work is set to 3×3. ReLU function is adopted after convolution operation except the last CNN layer in each structure.

Table 7.3: **Performance of different model structures on the JRSRD dataset in terms of PSNR and SSIM.**

Methods	PSNR	SSIM
RESCAN [Li et al., 2018d]	21.05	0.768
PReNet [Ren et al., 2019]	23.29	0.789
[Qian et al., 2018]	22.49	0.772
DAiAM	**24.67**	**0.819**
D-DAiAM	**25.26**	**0.825**



(a) Input rainy image (b) Attention: Ours

(c) [Qian et al., 2018] (d) Deraining: Ours

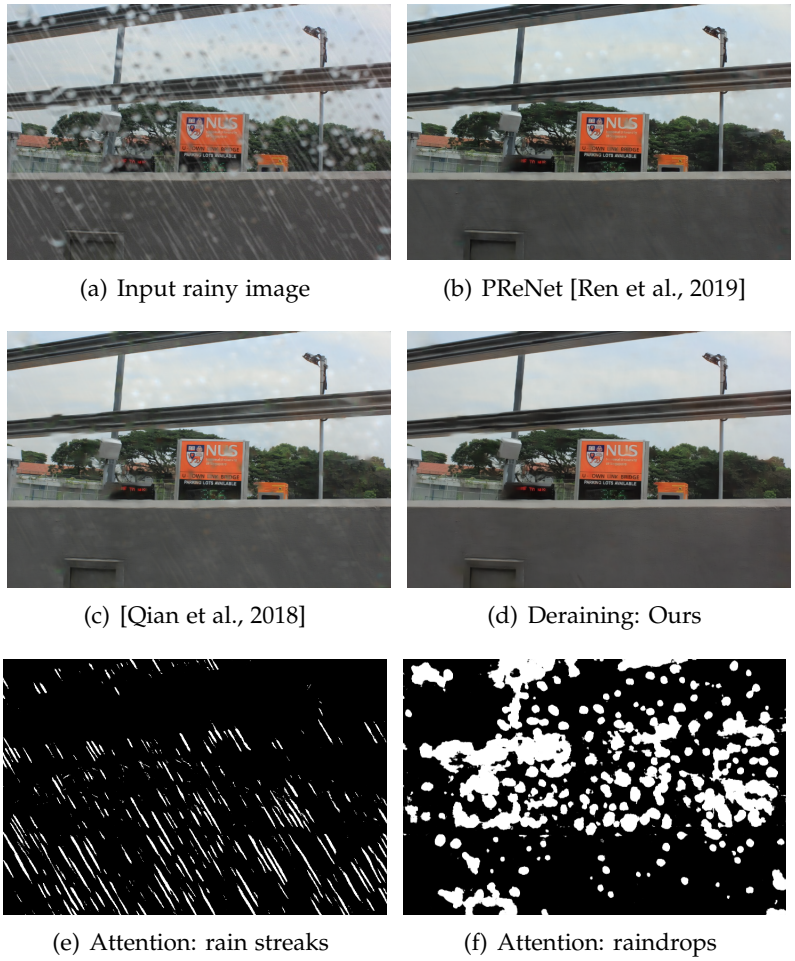
Figure 7.6: **Raindrop removal results on sample images from the [Qian et al., 2018] Raindrop dataset.**

7.3.2 Results on Rain Streak Dataset

[Yang et al., 2017] build a dataset of heavy rain streaks, named as Rain100H. In order to synthesize heavy rain, they apply two different methods, including the photo-realistic rendering techniques proposed by [Garg and Nayar, 2006] and directly adding simulated sharp line streaks to clear images. The Rain100H dataset consists of 1,800 and 100 pairs of images for training and testing, respectively. [Ren et al., 2019] removes some training images with the same background contents as testing images. Table 7.1 reports the comparison results with the state-of-the-art rain streak removal methods, including GMM [Li et al., 2016], DDN [Fu et al., 2017b], RGN [Fu et al., 2017a], JORDER [Yang et al., 2017], RESCAN [Li et al., 2018d] and PReNet [Ren et al., 2019]. Note that, as the rainy images contain only rain streaks,

Table 7.4: **Ablation study on the JRSRD dataset in terms of PSNR and SSIM.**

Methods	PSNR	SSIM
DAM(zero)	21.97	0.729
DAM(odd)	23.41	0.791
DAM(dual)	24.15	0.806
DAiAM	24.67	0.819
DAiAM-DAiAM	24.84	0.823
D-DAiAM	25.26	0.825
D-DAiAM(3)	25.68	0.833



(a) Input rainy image (b) PReNet [Ren et al., 2019]

(c) [Qian et al., 2018] (d) Deraining: Ours

(e) Attention: rain streaks (f) Attention: raindrops

Figure 7.7: **Rain streak and raindrop removal results on sample images from JRSRD dataset.**

our full method D-DAiAM degrades as D-DAM in this scenery. The quantitative results demonstrate the advance of our proposed method over the existing methods. Fig. 7.5 shows the qualitative deraining results and the associated attention maps. Our result is better than that of PReNet [Ren et al., 2019]. The latent attention map is also close to the ground truth.

7.3.3 Results on Raindrop Dataset

[Qian et al., 2018] capture 1,119 pairs of images with different background scenes and raindrops. They use two glasses to model the raindrops. One is clean to capture GT images. The other is sprayed with water to generate corresponding rainy version. The training set and testing set A include 861 and 58 pairs, respectively. In order to verify the performance of the propose method, we compare with state-of-the-art deraining methods. As mentioned before, our method becomes D-DAM in this case.

(a) Input rainy image (b) DAM(odd) (c) DAM(dual)

(d) DAiAM (e) DAiAM-DAiAM (f) D-DAiAM

Figure 7.8: **Ablation study results of rain streak and raindrop removal on sample images from JRSRD dataset.** Zoom-in for details.

Table 7.1 presents the results of DID-MDN [Zhang and Patel, 2018b], DDN [Fu et al., 2017b], JORDER [Yang et al., 2017], [Qian et al., 2018] and ours, respectively. The deraining results and attention maps are provided in Fig. 7.6. Both the quantitative and the qualitative results reveal that our method is more advanced.

7.3.4 Results on the Joint Rain Streak and Raindrop Dataset

There are many rain removal datasets for image deraining [Yang et al., 2017; Qian et al., 2018; Zhang and Patel, 2018b; Wang et al., 2019a; Li et al., 2019c]. However, most of them focus on either rain streaks or raindrops. To this end, we synthesize a new joint rain streak and raindrop (JRSRD) dataset to evaluate the performance of different methods for removing both of them. Specially, the JRSRD training set contains 3,444 synthetic rainy images, generated using images with raindrops from [Qian et al., 2018]. We synthesize four images with different intensity levels of rain streaks for each of them via Photoshop. The noise levels are set between 20% and 60% to model various intensity. The JRSRD testing set contains 232 pairs. The rainy images in our synthesized dataset contain both rain streaks and raindrops. Therefore, we apply DAiAM to remove rain. The performance compared with three current deraining methods is shown in Table 7.3. Our proposed method beats these CNN-based methods on the task of joint rain streak and raindrop removal. Exemplar visual results are given in Fig. 7.7, suggesting that the proposed method is capable of generating cleaner images.

7.3.5 Ablation Study

To demonstrate the effectiveness of DAM, DAiAM and differential-driven module, we compare these structures with several variant structures. Different from previous methods which merely focus on heavy rain, the proposed DAM generates two feature maps paying attention to heavy rain and light rain, respectively. Thus we compare to model without attention, DAM(zero), and the models with one or two attention maps, which are named as DAM(odd) and DAM(dual), respectively. Then, we compare the performance of the proposed dual attention-in-attention model, DAiAM, which can jointly perceive rain streaks and raindrops. The D-DAiAM is the model which removes rain using the differential-driven module. We compare it with the method directly connecting two DAiAM, termed as DAiAM-DAiAM. We also aggressively use two differential-driven modules in D-DAiAM(3). Table 7.4 shows the performance of them in terms of PSNR and SSIM. Apparently, the counterpart without attention performs worst. Using attention of heavy rain improves the performance, as demonstrated by DAM(odd). While dual attention mechanism further improves the results. The DAiAM outperforms these three by simultaneously removing both raindrops and rain streaks. Directly connecting two DAiAM as DAiAM-DAiAM indeed boosts the values, while the improvement is not as significant as that of the proposed D-DAiAM. Fig. 7.8 present exemplar visual deraining results, which also suggest the effectiveness of the proposed method.

7.3.6 Deployment in Real World

The proposed method is also evaluated on real-world images from the Internet. Fig. 7.9 shows the visual deraining results of different methods. DID-MDN [Zhang and Patel, 2018b] and PReNet [Ren et al., 2019] are two state-of-the-art methods for rain streak removal, and [Qian et al., 2018] is one of the best methods to remove raindrops [Li et al., 2019c]. The proposed method achieves better performance on removing both rain streaks and raindrops than them, due to the proposed dual attention-in-attention mechanism. The compared methods can only remove either raindrops (*e.g.* [Qian et al., 2018]) or rain streaks (*e.g.* [Zhang and Patel, 2018b] and [Ren et al., 2019]).

7.4 Conclusion

In this chapter, we tackle the problem of joint removal of raindrops and rain streaks in this chapter. A dual attention-in-attention model, DAiAM, is presented to focus on raindrops and rain streaks simultaneously. Inside DAiAM, we propose a dual attention model, DAM. The proposed DAM learns two intensity-aware maps to remove rain from heavy and light rainy regions. We further introduce a differential-driven module to optimize the deraining process. Experimental results have demonstrated that our method performs best against the state-of-the-art methods and is capable of deraining well in real-world scenarios.

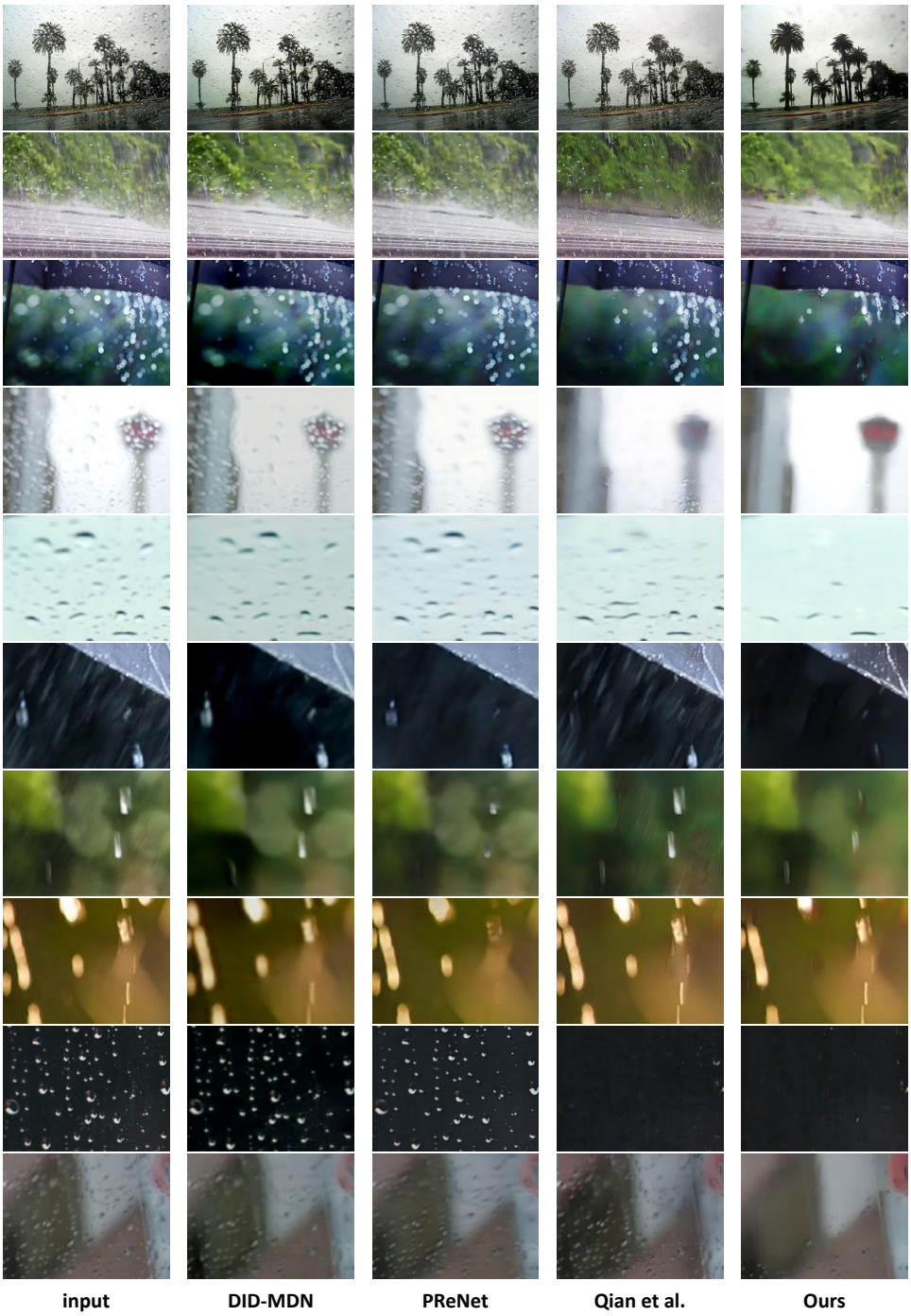


Figure 7.9: The performance of different methods on real-world rainy images. From the left to right are the input, DID-MDN Zhang and Patel [2018b], PReNet Ren et al. [2019], Qian *et al.* Qian et al. [2018] and ours. DID-MDN and PReNet are two rain streak removal methods, which only work on removing rain streaks. Qian *et al.* is a raindrop removal method, which does not work on rain streak removal. Our proposed method achieves better performance by removing rain streaks and raindrops simultaneously on real-world rainy images.

Chapter 8

Deraining: Enhanced Spatio-Temporal Interaction Learning for Video Deraining

This chapter is about video deraining. Video deraining is an important task in computer vision as the unwanted rain hampers the visibility of videos and deteriorates the robustness of most outdoor vision systems. Despite the significant success which has been achieved for video deraining recently, two major challenges remain: 1) how to exploit the vast information among continuous frames to extract powerful spatio-temporal features across both the spatial and temporal domains, and 2) how to restore high-quality derained videos with a high-speed approach. In this chapter, we present a new end-to-end video deraining framework, dubbed Enhanced Spatio-Temporal Interaction Network (ESTINet), which considerably boosts current state-of-the-art video deraining quality and speed. The ESTINet takes the advantage of deep residual networks and convolutional long short-term memory, which can capture the spatial features and temporal correlations among successive frames at the cost of very little computational resource. Extensive experiments on three public datasets show that the proposed ESTINet can achieve faster speed than the competitors, while maintaining superior performance over the state-of-the-art methods.

8.1 Introduction

Images and videos captured by cameras in the outdoor scenarios often suffer from bad weather conditions. As one common condition, rain streaks cause a series of visibility degradation, which seriously deteriorates the performance of outdoor vision-based systems. In contrast to image deraining methods, which rely solely on the texture appearances of the single frame, video deraining is a more challenging task as one has to consider how to model and exploit the inherent temporal correlation among continuing video frames. Moreover, several video deraining methods [Yang et al., 2019b; Liu et al., 2018b; Chen et al., 2018b] achieve state-of-the-art performance but their speed is relatively slow. There also exists method [Jiang et al., 2018] in-

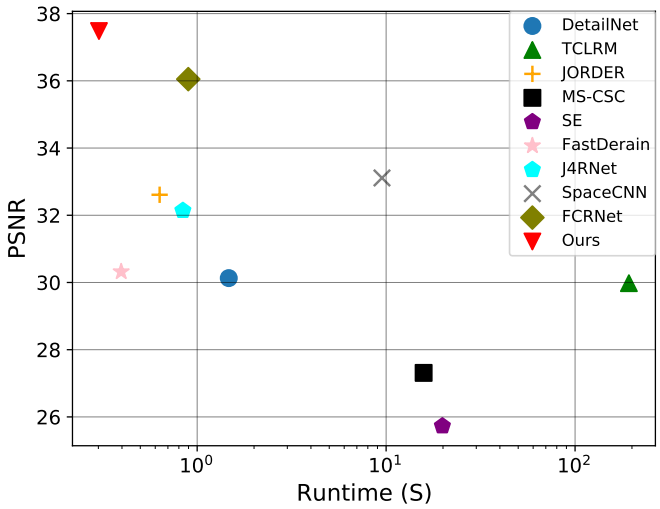


Figure 8.1: **The PSNR versus runtime of the state-of-the-art deep video deraining methods and our method on the NTURain dataset.**

troducing fast video deraining models. However, the performance is far behind the current state-of-the-art methods. Therefore, the ideal approach of video deraining is to find an effective model to learn more powerful spatio-temporal features existing among the continuing frames with higher speed (Fig. 8.1).

We propose an Enhanced Spatio-Temporal Integration Networks (*ESTINet*) to exploit the spatio-temporal information for rain streak removal. Fig. 8.2 illustrates the overall architecture of *ESTINet*. It contains three parts: spatial information collection module (*SICM*), spatio-temporal interaction module (*STIM*), enhanced spatio-temporal module (*ESTM*).

Considering that the spatial information plays an important role in video deraining, we firstly build an architecture called *SICM* to directly extract high-level spatial features from the input rainy frames. The SICM includes ResNet as the backbone because it has a powerful ability to extract spatial information from a single frame. Then the representations are fed into the second part, *STIM*, to recover the coarsely derained frames. *STIM* is a convolutional bidirectional long short-term memory (*CBLSTM*) like architecture, called *Interaction-CBLSTM*, which can directly make use of spatial features captured from the previous module. Therefore, it is a light-weighted module and mainly considers the temporal correlations to help remove rain streaks with very little increase in computational cost. I choose CBLSTM as the backbone because it can capture spatio-temporal information from a video. Meanwhile, the loss calculated based on the output of *STIM* also helps to update the *SICM* to extract more powerful spatial features. Moreover, different from traditional *CBLSTM*, our *Interaction-CBLSTM* (Fig. 8.4) architecture connects the features extracted from the last frame to the input and uses convolutional operation to replace the *tanh* function to adapt to different scales of input frames. Finally, *ESTM* takes the coarse deraining video as input and refines the temporal transformation with a 3D DenseNet-like architecture while preserving the realistic content information.

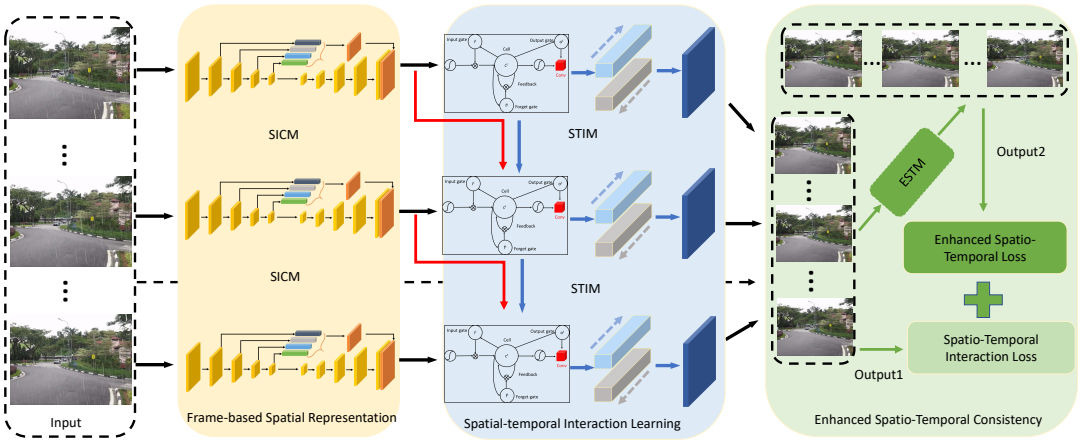


Figure 8.2: **Our proposed Enhanced Spatio-Temporal Interaction Networks (*ESTINet*).** The input rainy frames are fed into SICM to extract spatial cue, which is further forwarded into *STIM* to extract spatio-temporal features. Finally, the proposed *ESTM* takes the extracted features as input to capture the spatio-tempoal consistency and generate the final results.

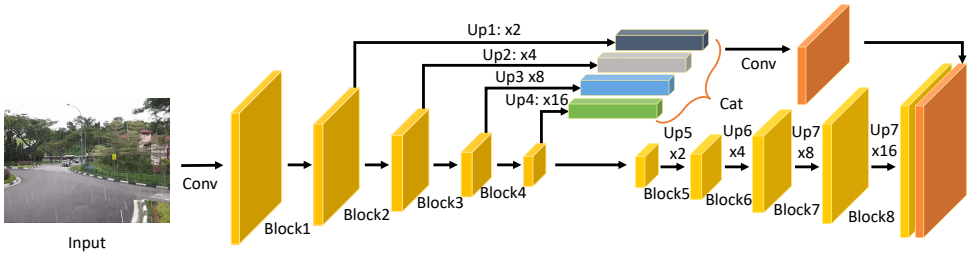
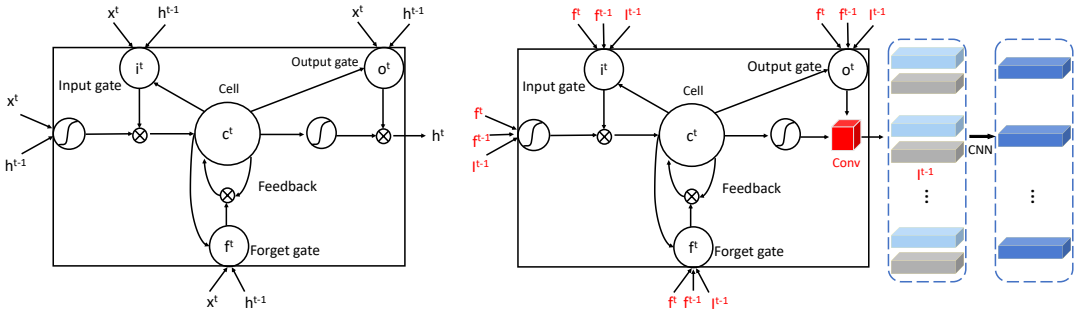


Figure 8.3: **Illustration of the ResNet-based Encoder-Decoder backbone (*SICM*) to extract spatial representations from frames.** The input is a single rainy frame, while the output is its spatial features. “Up" means the upsampling operation.

8.2 ESTINet

8.2.1 Overall Architecture

The ultimate goal of our work is to remove the rain streaks and recover clean videos. In order to extract powerful spatio-temporal information efficiently, an Enhanced Spatio-Temporal Interaction Network, termed as *ESTINet*, is proposed to extract features across both the spatial and temporal domains with less computational cost. In this section, we will first introduce an *SICM* architecture to extract spatial representation from each input rainy frame in Sec. 8.2.2. Then, the spatial representations are fed into our proposed *STIM* to exploit the temporal information among continuing frames (Sec. 8.2.3). Finally, we build a 3D-DenseNet backbone, *ESTM*, to enhance the spatial-temporal consistency in Sec. 8.2.4. Fig. 8.2 shows the overall architecture of our proposed framework.



(a) LSTM (b) Interaction-BCLSTM

Figure 8.4: **Comparison illustration between LSTM and the Interaction-BCLSTM backbone in *STIM* to learn temporal correlations.** The input (f^t) is the spatial features extracted from SICM illustrated in Fig. 8.3. The output of "Conv" is two kinds of spatio-temporal features which mean bidirectional sequences, which are further fed into a CNN to obtain the forward sequence.

8.2.2 Frame-based Spatial Representation

As it is shown in Fig. 8.3, our *SICM* is an Encoder-Decoder architecture. Both the encoder and decoder include one convolutional layer and four ResBlocks. The input is original RGB images. Following the input, the convolutional layer encodes the RGB images into feature maps with the same size as the original input. Then the four ResBlocks in the encoder employ four down-projection operation to decrease the resolution of the feature maps to their 1/16. The decoder reconstructs clean images with original resolution via four up-projection operation. In order to fuse multi-scale features, there exists a multi-scale fusion module between the encoder and decoder.

Spatial features play an important role in the task of image restoration. Different from existing methods, which extract spatial features from a single frame, the proposed architecture directly learns spatial representation from a video sequence for the following processing. In addition, we use a relatively light-weighted encoder-decoder architecture. In this way, the proposed model can process the input frames with a high speed. It can also be replaced with some other state-of-the-art backbones to improve the ability of spatial feature extraction.

8.2.3 Spatial-temporal Interaction Learning

After obtaining the spatial representation from the stack of input frames. We propose an *STIM* to learn the temporal correlation between the continuing frames. The structure of *STIM* is based on LSTM model, which is shown in Fig. 8.4. The traditional LSTM can be formulated as follows:

$$
\begin{aligned}
f^{(t)} &= \sigma(W^{(f)}x^{(t)} + W^{(f)}h^{(t-1)} + b^{(f)}),\\
i^{(t)} &= \sigma(W^{(i)}x^{(t)} + W^{(i)}h^{(t-1)} + b^{(i)}),\\
\widetilde{C}^{(t)} &= \tanh(W^{(C)}x^{(t)} + W^{(C)}h^{(t-1)} + b^{(C)}),\\
C^{(t)} &= f^{(t)} \odot C^{(t-1)} + i^{(t)} \odot \widetilde{C}^{(t)},\\
o^{(t)} &= \sigma(U^{(o)}x^{(t)} + W^{(o)}h^{(t-1)} + b^{(o)}),\\
h^{(t)} &= o^{(t)} \odot \tanh c^{(t)}
\end{aligned}
\tag{8.1}
$$

where $U^{(\cdot)}$ and $W^{(\cdot)}$ are the input-to-hidden and hidden-to-hidden weight matrices, and $b^{(\cdot)}$ are bias vectors. σ and $\odot$ are sigmoid activation function and point-wise multiplication, respectively.

Different from the traditional LSTM, the proposed *STIM* is modified based on the traditional LSTM to deal with video deraining. Firstly, the Hadamard product in LSTM is replaced with the convolution to address the 2D spatial representation extracted by *SICM*. Secondly, we add the spatial representation of the last frame into the calculation of forget gate $f^{(t)}$. Thirdly, we replace the hyperbolic tangent activation function with the convolution operation during the calculation of hidden state $h^{(t)}$ like ConvLSTM, and add bidirectional operation like bidirectional-LSTM. Our *STIM* is formulated as:

$$
\begin{aligned}
f^{(t)} &= \sigma(W^{(f)} * [f_x^{(t)}, f_x^{(t-1)}, h^{(t-1)}] + b^{(f)}),\\
i^{(t)} &= \sigma(W^{(i)} * [f_x^{(t)}, f_x^{(t-1)}, h^{(t-1)}] + b^{(i)}),\\
\widetilde{C}^{(t)} &= \tanh(W^{(C)} * [f_x^{(t)}, f_x^{(t-1)}, h^{(t-1)}] + b^{(C)}),\\
C^{(t)} &= f^{(t)} \odot C^{(t-1)} + i^{(t)} \odot \widetilde{C}^{(t)},\\
o^{(t)} &= \sigma(W^{(o)} * [f_x^{(t)}, f_x^{(t-1)}, h^{(t-1)}] + b^{(o)}),\\
h^{(t)} &= Conv(o^{(t)}, \tanh C^{(t)}),\\
{I_f}^{(t)} &= Conv(h^{(t)}, {h'}^{(t)})
\end{aligned}
\tag{8.2}
$$

where $*$ is the convolution operation. f_x^t is spatial feature maps extracted from the frame t by *SICM*. We concatenate f_x^t with the spatial feature maps f_x^{t-1} extracted from the last frame $t-1$, and then feed them into *STIM* to update the information. Then, the information from the output gate and updated memory cell are concatenated and fed into two convolutional layers to obtain the restoration results $h^{(t)}$. We can also obtain the other results ${h'}^{(t)}$ by reserving the order of frames. The results from two directions are finally put into another two convolutional layers to obtain finer derained results ${I_f}^{(t)}$.

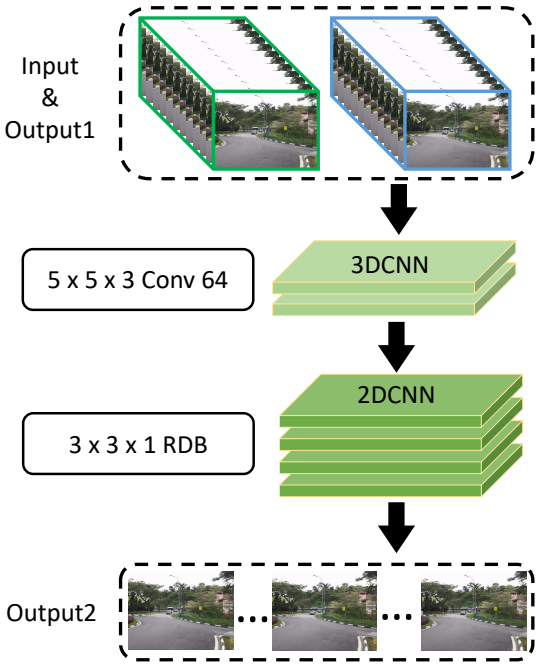


Figure 8.5: **Illustration of the Enhanced Spatio-Temporal Model (ESTM) to refine the deraining videos.**

8.2.4 Enhanced Spatial-Temporal Consistency

The *SICM* and *STIM* work together to restore clean videos from input rainy versions. In order to enhance the spatio-temporal consistency and make full use of the correlations between continuing frames, we input the coarse results from *STIM* into *ESTM* to further improve the quality of the generated videos.

When training the *SICM* and *STIM*, we find it is difficult to remove heavy rain while maintaining realistic content details. In other words, the *SICM* and *STIM* are helpful to remove most of the rain artifacts and restore coarse results, but may not be able to generate a better video and remove heavy rain. Therefore, we build a new architecture, which is illustrated in Fig. 8.5.

Besides the ConvLSTM, which is able to capture the temporal correlation between continuous frames, 3D CNN is another popular architecture. We apply 3D CNN in *ESTM* to cover the shortage of the traditional LSTM and refine the deraining results. The coarse results and the original rainy frames are concatenated and fed into the *ESTM*, which operates 3D convolutions via convolving 3D kernels on these frames. By doing so, the feature maps in convolutional layers can also capture the dynamic variations to help further remove rain and recover the details of images. Specially, we perform 3D convolution with kernel size of 3×3 in the first and second convolutional layers to reduce the temporal dimension from five to one. In the following layers, we use the 2D convolution to replace 3D operation as their temporal dimensions have already been decreased to one.

Table 8.1: **Performance comparison with state-of-the-art methods on the RainSynLight25, RainSynHeavy25 and NTURain dataset.**

Dataset	Metric	DetailNet	TCLRM	JORDER	MS-CSC	SE	FastDerain	J4RNet	SpaceCNN	FCRNet	**Ours**
NTURain	PSNR	30.13	29.98	32.61	27.31	25.73	30.32	32.14	33.11	36.05	**37.48**
	SSIM	0.9220	0.9199	0.9482	0.7870	0.7614	0.9262	0.9480	0.9474	0.9676	**0.9700**
RainSynLight25	PSNR	25.72	28.77	30.37	25.58	26.56	29.42	32.96	32.78	35.80	**36.12**
	SSIM	0.8572	0.8693	0.9235	0.8089	0.8006	0.8683	0.9434	0.9239	0.9622	**0.9581**
RainSynHeavy25	PSNR	16.50	17.31	20.20	16.96	16.76	19.25	24.13	21.21	27.72	**28.48**
	SSIM	0.5441	0.4956	0.6335	0.5049	0.5293	0.5385	0.7163	0.5854	0.8239	**0.8242**

8.2.5 Loss Function

In our work, we use two types of loss functions to train the proposed framework.

Spatio-Temporal Interaction Loss. The *SICM* and *STIM* are able to learn the spatial representations and temporal correlations from input frames. In order to help them interact with each other to recover coarse results, we apply the Mean Square Error (MSE) to calculate the spatio-temporal interaction loss, which is defined as:

$$\mathcal{L}_{STI} = \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} \left(I_{x,y}^{clean} - G(I^{rainy})_{x,y} \right)^2, \tag{8.3}$$

where W and H are the width and height of a frame, $I_{x,y}^{clean}$ and $G(I^{rainy})_{x,y}$ correspond to the value of coarse derained frames and rainy frames at location (x, y). Note that, as this loss measures the results from the SICM and STIM, which are dedicated for spatial and temporal domains, we call this loss spatio-temporal interaction loss.

Enhanced Spatio-Temporal Loss. Our proposed framework is a two-stage architecture. In order to drive our framework to generate finer derained frames, we introduce another loss function to refine the coarse results. During the training stage, parameter of *ESTM* is updated based on the Enhanced Spatio-Temporal loss to further remove rain and recover clean images. The loss function can be represented as:

$$\mathcal{L}_{EST} = \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} \left(I_{x,y}^{clean} - G(I^{rainy}, I^{derained})_{x,y} \right)^2, \tag{8.4}$$

where $I^{derained}$ is the coarse derained frames generated from the *STIM*. This loss is used to assess the enhanced results regarding the ground truth, so we term it as the enhanced spatio-temporal loss.

Balance of Different Loss Functions. In the training stage, the above two loss functions are combined as:

$$\mathcal{L}_{final} = \mathcal{L}_{STI} + \alpha \cdot \mathcal{L}_{EST}, \tag{8.5}$$

where α is a hyper-parameter to balance the two loss functions. We set it as 1.

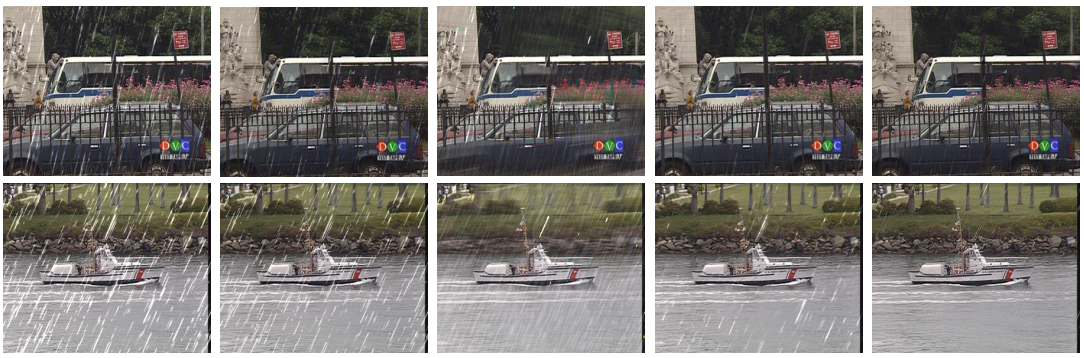


Figure 8.6: **Exemplar results on the RainSynLight25 dataset**. From left to right: input, results of [Jiang et al., 2018], [Wei et al., 2017], [Chen et al., 2018b] and ours. All results are attained without alignment. Best viewed in color.

Figure 8.7: **Exemplar results on the RainSynHeavy25 dataset**. From left to right: input, results of [Jiang et al., 2018], [Wei et al., 2017], [Chen et al., 2018b] and ours. All results are attained without alignment. Best viewed in color.

8.3 Experiments

We test our approach on three widely used public datasets, NTURain [Chen et al., 2018b], RainSynLight25 and RainSynComplex25 [Liu et al., 2018a], which are introduced firstly in Sec. 8.3.1. Then we introduce the implementation details of our framework in Sec. 8.3.2 and compare our method with the state-of-the-art methods in Sec. 8.3.3. An ablation study is conducted to show the effectiveness of its different components in Sec. 8.3.4. Efficiency analysis is reported subsequently in Sec. 8.3.5.

8.3.1 Datasets

NTURain. This dataset is created by [Chen et al., 2018b]. The images are taken by a camera with slow and fast movements. The training contains 24 rainy sequences and their corresponding clean versions, while the testing set contains 8 pairs of sequences. In addition, it also provides seven real-world rainy videos.

RainSynLight25. It contains 190 pairs of RGB rainy and clean sequences for

Figure 8.8: **Exemplar results on the NTURain dataset**. From left to right: input, results of [Jiang et al., 2018], [Wei et al., 2017], [Chen et al., 2018b] and ours. All results are attained without alignment. Best viewed in color.

Figure 8.9: **Deraining results on the real-world rainy sequences**. The top and bottom rows are the input sequences and the output sequences from out proposed model, respectively. Best viewed in color.

training, and 27 pairs for testing. The sharp images are from CIF testing sequences, HDTV sequences and HEVC standard testing sequences. Via adding rain streaks generated by the probabilistic model [Garg and Nayar, 2006], the corresponding rainy images are obtained.

RainSynHeavy25. This dataset is similar to the dataset of RainSynLight25. The main difference is that the rain streaks in rainy images are generated by the probabilistic model, sharp line streaks and sparkle noises. Therefore, they are heavier than those in the RainSynLight25 dataset.

8.3.2 Implementation Details

The weights of networks in our framework are initialized via a Gaussian distribution with zero mean and a standard deviation of 0.01. Models are updated after learning a mini-batch of size 8 in each iteration. We also crop patches of size 224×224 from images, and randomly flip frames horizontally to augment the training set. During

Table 8.2: **Speed comparison with state-of-the-art methods. The numbers are in seconds.**

Dataset	DetailNet	TCLRM	JORDER	MS-CSC	SE	FastDerain	J4RNet	SpaceCNN	FCRNet	**Ours**
NTURain	1.4698	192.7007	0.6329	15.7957	19.8516	0.3962	0.8401	9.5075	0.8974	**0.3122**

the training stage, we first train the *SICM* and *STIM* without the *ESTM* module, and then update all weights of them. The learning rate is set as a value of 10^{-4} and reduces to 10^{-6} after the training loss gets converged. For evaluation, we employ PSNR and SSIM as metrics.

8.3.3 Comparison with Existing Methods

In this section, we compare the performance of our proposed framework with several state-of-the-art deraining methods on the above three widely used datasets. Among these methods, stochastic encoding (SE) [Starik and Werman, 2003], temporal correlation and low-rank matrix completion (TCLRM) [Kim et al., 2015], FastDerain [Jiang et al., 2018], joint recurrent rain removal and reconstruction (J4RNet) [Liu et al., 2018b] and superpixel alignment, compensation CNN (SpacCNN) and frame-consistent recurrent network (FCRNet) [Yang et al., 2019b] are video-based deraining methods, joint rain detection and removal (JORDER) [Yang et al., 2017], deep detail network (DetailNet) [Fu et al., 2017b], J4RNet [Liu et al., 2018b], SpacCNN [Chen et al., 2018b] (FCRNet) [Yang et al., 2019b] are deep deraining methods. Table 8.1 shows the quantitative comparison results between our method and the current deraining methods. Before our work, FCRNet achieves the state-of-the-art performance on three public video deraining datasets. Our method further improves over the FCRNet method and obtains the best performance, in terms of both PSNR and SSIM values. This indicates that our framework achieves better feature representations due to the learned spatio-temporal interactions. To give a intuitive view of how ours and these compared methods perform, Fig. 8.6, Fig. 8.7 and Fig. 8.8 show the exemplar visual results on the datasets of RainSynLight25, RainSynHeavy25, and NTURain. The qualitative comparison results also evidently verify that our method achieves better performance than the existing ones. In addition, we also show the performance of our approach in the case of real-world scenarios. Taking a real-world rainy video from the NTURain dataset, we process this video by our method to remove the rain, and the result frames are shown in Fig. 8.9. The rain is successfully removed to some extent.

8.3.4 Ablation Study

The proposed *STIM* has the advantage of capturing temporal correlations from continuing frames and helping update the *SICM* to learn better spatial representations. The *ESTM* is able to learn enhanced spatio-temporal representations via making use of the coarse derained images and the 3D Convolution to refine the results. In order to verify its effectiveness, we develop five variant networks: *SICM + 2DCNN*, *SICM*

Table 8.3: **Performance comparison of different architectures on the NTURain dataset.**

Methods	PSNR	SSIM
SICM + 2DCNN	35.44	0.9562
SICM + STIM (#2)	36.61	0.9668
SICM + STIM (#3)	36.93	0.9677
SICM + STIM (#4)	37.16	0.9682
SICM + STIM (#5)	37.28	0.9693
SICM + STIM (#5) + ESTM	37.48	0.9700

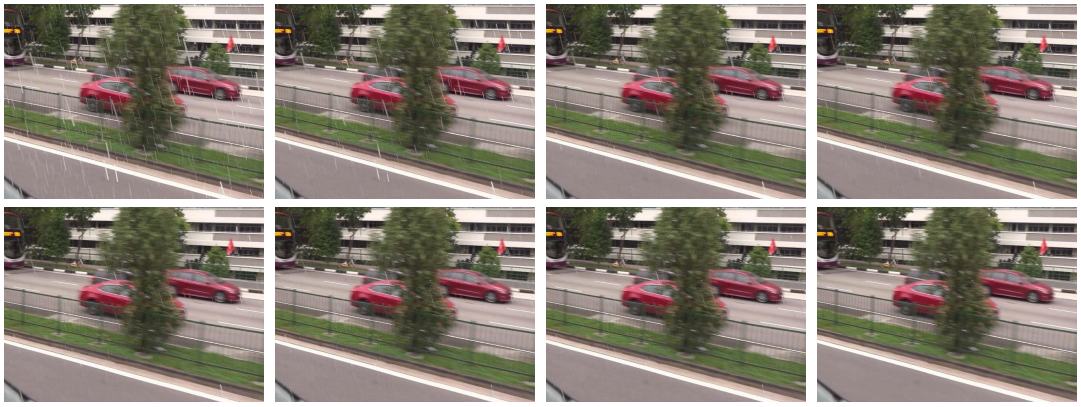


Figure 8.10: **Exemplar results on the NTURain dataset**. From left to right, top to bottom: input, SICM + 2DCNN, SICM + STIM (#2), SICM + STIM (#3), SICM + STIM (#4), SICM + STIM (#5), SICM + STIM (#5) + ESTM, and ground-truth. Best viewed in color.

+ STIM (#2), *SICM + STIM (#3)*, *SICM + STIM (#4)*, *SICM + STIM (#5)* and *SICM + STIM (#5) + ESTM*. *SICM + 2DCNN* is a baseline method, which replaces the STIM with three ordinary convolutional layers. The input to *SICM + 2DCNN* is a single frame, so it does not take into consideration of the temporal information among the consecutive frames. In order to show that how the number of input frames influences the performance of our proposed model, we compare the values of PSNR and SSIM of models with different numbers of input rainy frames. Specially, the number n in *SICM + STIM (#n)* represents the number of consecutive frames.

The quantitative results are shown in Table 8.3. Specifically, by learning temporal information from continuous frames, all the variants of *SICM + STIM (#n)* outperform the plain model *SICM + 2DCNN*, which verifies the usefulness of the temporal information. And with the increase of input frames, better performance is achieved. By considering the spatio-temporal interaction, our full method *SICM + STIM (#5) + ESTM* achieves additional gains. Fig. 8.10 shows the qualitative results of these variants.

8.3.5 Efficiency Analysis

Table 8.2 shows the speed of the state-of-the-art deraining methods. J4RNet, FCR-Net and our proposed methods are based on the PyTorch framework, while other methods are implemented based on Matlab. We evaluate the speed on the NTU-Rain dataset on an ordinary platform. Our proposed method is faster than other state-of-the-art methods, including the FastDeRain method.

8.4 Conclusion

The main contribution of this chapter is that we propose a novel end-to-end framework to address the problem of video deraining by a **faster** scheme with **better** quantitative and qualitative results. To obtain the spatial representation, a ResNet-based architecture, *SICM* is built to directly extract spatial features from a stack of input frames. The representations are then fed into a well-designed Interaction-*BCLSTM* architecture, *STIM*, to capture the temporal correlations. In the training stage, the proposed *SICM* and *STIM* interact with each other to capture the spatial information and temporal correlations between continuing frames to obtain coarse results, which are fed into a 3D-DenseNet based architecture, *ESTM*, to enhance the performance of rain removal and obtain finer results. Extensive experiments have verified that the proposed framework outperforms the state-of-the-art methods in terms of quality and speed.

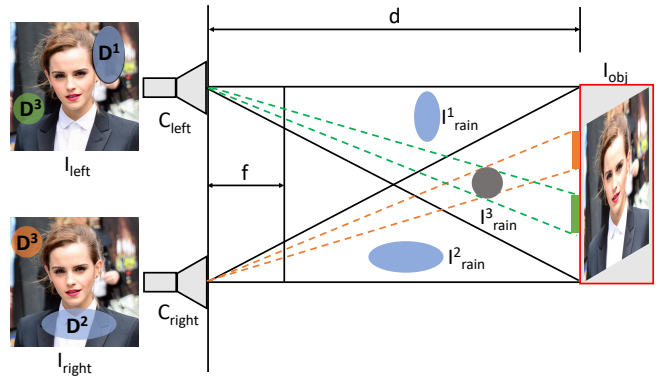
Chapter 9

Deraining: Stereo Image Deraining via Semantic Understanding

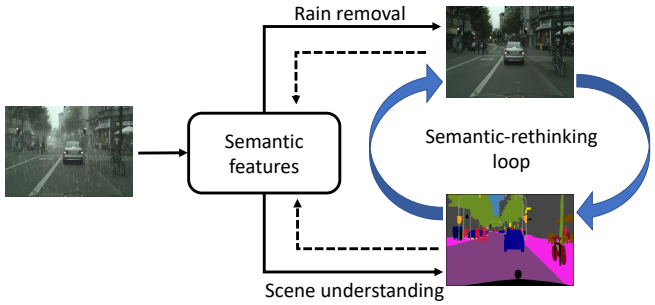
This chapter is about stereo deraining. Rain can hamper the visibility of background scenes and degrade the quality of images, which deteriorates the performance of most existing vision systems. Nowadays, overwhelming state-of-the-art models adopted in autonomous driving rely on stereo cameras. However, there are few studies on deraining for stereo images. Meanwhile, even for monocular deraining, most of current methods fail to understand and remove rain because these methods consider only pixel-level loss functions during training. In this chapter, we present a Paired Rain Removal Networks (**PRRNet**), the first stereo semantic-aware deraining networks, which can be trained without pairs of rainy image and its segmentation annotation. Within PRRNet, there is a Semantic-Aware Deraining Module (SADM) considering both tasks of semantic understanding and deraining of scene, a Semantic-Fusion Network (SFNet) combining semantic segmentation and deraining images, and a View-Fusion Network (VFNet) fusing information from multiple views. We also synthesize two stereo rainy datasets to evaluate different deraining methods. Experimental results on one public monocular and two developed stereo rainy datasets demonstrate that the PRRNet achieves the state-of-the-art performance on both monocular and stereo image deraining.

9.1 Introduction

Autonomous driving has become an increasingly active research field in computer vision with the development of stereoscopic vision [Chen et al., 2015]. Based on stereo images, many key technologies such as depth estimation [Godard et al., 2017; Liu et al., 2015; Riegler et al., 2019], scene understanding [Eslami et al., 2016; Shao et al., 2015; Zhao et al., 2017] and stereo matching [Luo et al., 2016; Chang and Chen, 2018; Pang et al., 2017] have achieved great success. As an inevitable natural phenomenon in the wild, rain causes visual discomfort and degrades the quality of images, which can deteriorate the performance of many core models, thus increasing the latent danger of autonomous driving [Li et al., 2019c]. However, there are few studies for stereo deraining. We address the problem of removing rain from stereo



(a) Two views from stereo cameras



(b) The semantic-aware deraining module

Figure 9.1: **The illustration of stereo cameras and the semantic-aware deraining module.** (a) One pair of images captured by stereo cameras. Same rain can cause different effects on images from two views. (b) Integrating semantic features to jointly remove rain and understand scene semantics.

images.

In fact, stereo deraining has an intrinsic advantage over monocular deraining because the effects of identical rain streaks in corresponding pixels from stereo images are different. As Fig. 9.1(a) shows, the mapping of object I_{obj} on stereo images can be represent as

$$I_{left} = I_{obj} * \frac{d}{f}, \qquad I_{right} = I^{ref}_{obj} * \frac{d}{f}, \tag{9.1}$$

where d and f are the distance between object and camera, and the camera focal length, respectively. I^{ref}_{obj} is the reflection of I_{obj}. Assuming that the object I_{obj} is in the middle of two cameras, the length of identical object, I^{ref}_{obj} and I_{obj}, on stereo views are the same. However, the effects of rain across stereo images are different. For example, the degraded regions by rain I^1_{rain} on the two images can be denoted as

$$D^1_{left} = I^1_{rain} * \frac{d}{f}, \qquad D^1_{right} = 0. \tag{9.2}$$

The I^1_{rain} degrades the quality of object on the left image but does not affect the visual comfort of the right view. There is also rain influencing different regions on both stereo images like I^3_{rain}. The image in Fig. 9.1(a) shows the different effects of identical rain streaks on stereo views.

Moreover, the geometric cue and semantics provide important prior information, serving as a latent advantage for removing rain. Recently, most deep monocular deraining methods achieve great success by reconstructing objects based on pixel-level objective functions like MSE. However, these methods ignore modeling the geometric structure of objects and understanding the semantic information of scene, which in fact benefit deraining. [Hu et al., 2019] try to remove rain via depth estimation, but they also fail to understand the rainy scenes and their method relies on RGB-depth image pairs, which are costly and time-consuming to collect [Godard et al., 2017].

We first propose a semantic-aware deraining module, *SADM*, which removes rain by leveraging scene understanding. Fig. 9.1(b) illustrates the concept of *SADM*. It contains two parts. The first part is an encoder which takes a rainy image as input and encodes it as semantic-aware features. Then the representations are fed into the second part, a conditional generator, to transform them into the deraining image and scene segmentation. Based on a multi-task shared learning mechanism and different input conditions, the single *SADM* is capable of jointly removing rain and understanding scene. With the multi-task shared learning, we thus do not need paired data of rainy image and its semantic segmentation label. To further enhance the understanding of input image, a *Semantic-Rethinking Loop* is proposed to utilize the difference between the outputs of the conditional generators in different stages.

Based on *SADM*, we then present a stereo deraining model, *Paired Rain Removal Networks (PRRNet)*, which consists of *SADM*, *Semantic-Fusion Network (SFNet)* and *View-Fusion Network (VFNet)*. The *SADM* is utilized to learn the semantic information and reconstruct deraining images, while the *SFNet* and *VFNet* are to fuse the semantic information with coarse deraining images, and obtain the final deraining images by fusing stereo views, respectively. Currently, there is no public large-scale stereo rainy datasets. In order to evaluate the performance the proposed method and compare with the state-of-the-art methods, two large stereo rainy image datasets are thus synthesized.

9.2 The Semantic-aware Deraining Module

The ultimate goal of our work is to recover the deraining images from their corresponding rainy versions. In order to improve the capability of our model, a semantic-aware deraining module is proposed to learn semantic features based on clean images, rainy images and semantic labels. In this section, we will first introduce the consolidation of different tasks in Sec. 9.2.1 and how to train the proposed module based on unpaired images and semantic-annotated images in Sec. A.4.3. Then, a semantic-rethinking loop is discussed in Sec. 9.2.3 to further enhance our module and extract powerful features.

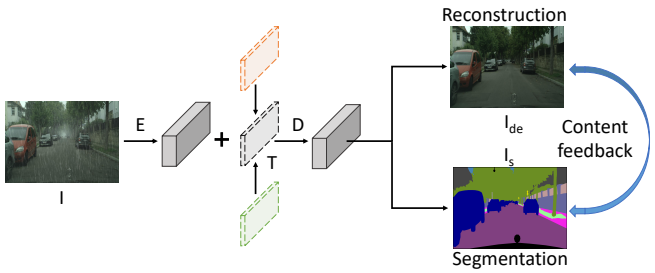


Figure 9.2: **The architecture of the proposed semantic-aware deraining module.** Rainy images are put into the encoder to extract features. Then the decoder generates deraining and segmentation results for different tasks.

9.2.1 The Consolidation of Different Tasks

Currently, most deep deraining methods directly learn the transformation from rainy images to derained ones [Li et al., 2019c]. [Hu et al., 2019] proposes a depth-aware network to jointly learn depth estimation and image deraining via two different sub-networks. Though their encoder networks share weights with the two tasks, the layers in decoder across different branches have non-shared weights, increasing the complexity and training difficulty of the model. Meanwhile, expensive pairs of RGB-depth images are required during the training stage.

To overcome the limitation, a unified autoencoder architecture is employed to merge different tasks in the learning stage. Fig. 9.2 illustrates the architecture of the proposed module. Images are input into the encoder of the proposed module to extract semantic features F. Then the semantic features F combined with a task label T are fed into the following unified decoder architecture to obtain a prediction P corresponding to label T. Based on different task labels like *deraining* or *scene understanding*, different outputs will be obtained. The learning stage can be formulated as

$$P = D(E(I), T)\,, \tag{9.3}$$

where E and D are the encoder and decoder of the *SADM*, respectively. I is the input image. T represents the label of different tasks. Based on the output of encoder and T, different predictions will be derived.

The branch of image deraining can be denoted as

$$I_{de} = \sigma_{de}(P \mid T_{de})\,, \tag{9.4}$$

where T_{de} corresponds to the label of deraining image. σ_{de} is the mapping function.

The branch of understanding scene can be denoted as

$$I_{seg} = \sigma_{seg}(P \mid T_{seg})\,, \tag{9.5}$$

where T_{seg} corresponds to the semantic segmentation label. σ_{seg} is a softmax function.

Based on the conditional architecture, the proposed *SADM* can jointly learn scene

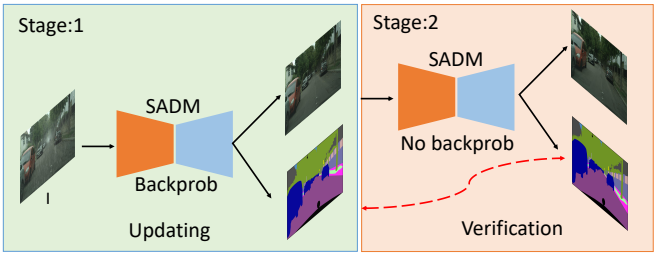


Figure 9.3: **The Semantic-rethinking Loop.** During training, rainy images are put into *SADM* to generate deraining and segmentation results in the stage I. Then the deraining images are utilized to generate segmentation results again in the stage II. Through comparing the two segmentation results from rainy and deraining images, the *SADM* can better understand scene and remove the unwanted rain. The *SADM*s in two stages share weights.

understanding and image deraining, which can extract more powerful semantic-aware features via sharing the information learned from different tasks, and thus is beneficial for different tasks.

9.2.2 Image Deraining and Scene Segmentation

Image Deraining. When T is set to T_{de}, the output of the proposed module is the deraining image. To learn the image deraining model, we compute the image reconstruction loss based on MSE loss function:

$$\mathcal{L}_{de} = ||I_c - \sigma_{de}(D(E(I_{rainy}), T_{de}))||^2 , \tag{9.6}$$

where I_c is the clean image.

Scene Segmentation. Most existing deraining methods focus on pixel-level loss function and thus fail to model the geometric and semantic information. This makes it difficult for models to understand the input image and generate deraining results with favorable details. To address this problem, we remove rain from rainy images by leveraging semantic information. The learning process of scene understanding can be denoted as

$$\mathcal{L}_{seg} = \sigma_h(I^{gt}_{seg}, I_{seg}) , \tag{9.7}$$

where I_{seg} and I^{gt}_{seg} indicate the scene understanding of the model and ground truth labels from auxiliary training sets. The σ_h is the cross-entropy loss function.

9.2.3 Semantic-rethinking Loop

In order to further enhance the semantic understanding of our model and help remove rain, a semantic-rethinking loop is proposed to refine the erroneous semantic understanding. Fig. 9.3 illustrates its scheme. It consists of a “updating" part and a “verification" part, whose core architecture is the semantic-aware deraining module, which has been illustrated in Fig. 9.2.

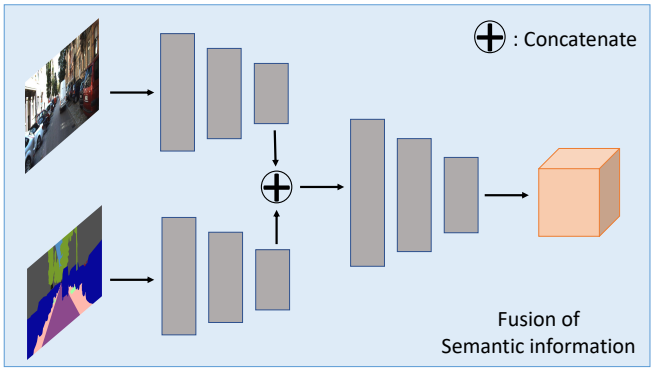


Figure 9.4: **The architecture of *SFNet*.** The coarse deraining images and semantic segmentation results from the *SADM* are put into *SFNet* to generate features volume with semantic information.

In the training stage, the “updating" part takes a rainy image as input, then generates the deraining image and semantic segmentation. Loss functions introduced in above sections are calculated and then update the weights of layers in the semantic-aware deraining module. Then the deraining image obtained in the “updating" part is fed into the “verification" part to obtain new semantic segmentation. The semantic understanding can improve the performance of deraining, which will be demonstrated in the next section. However, rain increases the difficulty of scene understanding. Via comparing segmentation results in different parts and pushing them to be close, the *SADM* can better understand scene and thus better derain. Both “updating" and “verification" parts employ the semantic-aware deraining module. The main difference between the “updating" and “verification" parts is that the weights in semantic-aware deraining module are updated in the “updating" part but fixed in the “verification" part. The semantic-rethinking loop provides the content feedback from the coarse-deraining image and improves the semantic understanding of *SADM*. In the testing stage, only the core semantic-aware deraining model is utilized to remove rain from images. The loss function can be noted as

$$\mathcal{L}_{con} = ||I_{seg}^{ver} - I_{seg}^{up}||, \tag{9.8}$$

where I_{seg}^{ver} and I_{seg}^{up} are the semantic segmentation results from the “verification" and “updating" parts, respectively.

9.3 The Paired Rain Removal Networks

In order to remove rain from stereo images, we further present a *PRRNet* based on *SADM*. The overall of the proposed networks will firstly be introduced in Sec. A.6, then two core sub-networks will be discussed in Sec. 9.3.2 and 9.3.3. Finally, the objective functions to train the proposed model will be presented in Sec. 9.3.4.

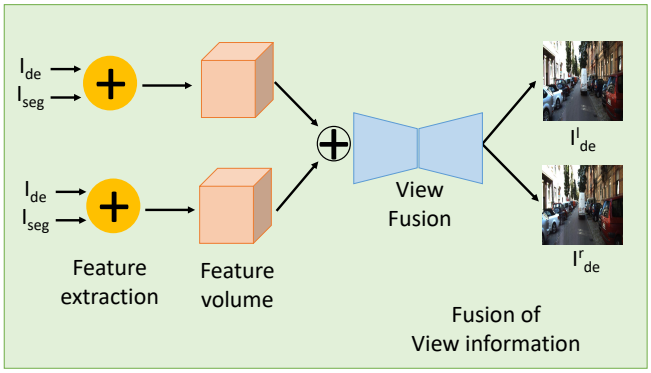


Figure 9.5: **The architecture of *VFNet*.** Features volumes from stereo images are fused to generate final stereo deraining images.

9.3.1 Network Architecture

The *PRRNet* consists of three sub-networks, i.e., , the *SADM*, Semantic-Fusion Net (*SFNet*) and View-Fusion Net (*VFNet*). The *SADM* is introduced in Sec. 9.2 to jointly remove rain and understand semantic information. Semantic-Fusion Net is utilized to combine the semantic information with coarse deraining images, while View-Fusion Net is to combine information from different views to obtain final deraining images. This proposed *PRRNet* has the following benefits: 1) due to the above-mentioned stereo semantic-aware deraining module, the proposed method simultaneously considers cross views and semantic information to help remove rain from images. 2) In the training stage, the *PRRNet* eliminates the requirements of paired stereo images and semantic-annotated images.

9.3.2 SFNet

The architecture of *SFNet* is shown in Fig. 9.4. The input is semantic segmentation and coarse deraining images from *SADM*. Given that the semantic information can help remove rain, we first process them individually and concatenate them, and then forward them into the following layers, to generate feature volume, which is utilized for generating final deraining results.

9.3.3 VFNet

The Fig. 9.5 illustrates the architecture of *VFNet*. The input is extracted fusion features from *SFNet*. The features extracted from the right view is helpful to remove the rain in the left-view image. Similarly, removing the rain from the right-view image also takes the advantage of features captured from the left-view image. Through the *VFNet*, the final finer deraining stereo images are obtained. The loss function in this part can be denoted as

$$\mathcal{L}_{view} = ||I_{de}^{left} - I_{gt}^{left}|| + ||I_{de}^{right} - I_{gt}^{right}|| \tag{9.9}$$

where I_{de}^{left} and I_{de}^{right} are stereo deraining images from *VFNet*, respectively. The I_{gt}^{left} and I_{gt}^{right} are the clean version of the stereo images.

9.3.4 Objective Functions

The loss function consists of two kinds of data terms, which is calculated based on semantic understanding and deraining reconstruction images. The final loss function can be written as

$$\mathcal{L}_f = \mathcal{L}_{de} + \lambda_1 \mathcal{L}_{seg} + \lambda_2 \mathcal{L}_{con} + \lambda_3 \mathcal{L}_{view}, \tag{9.10}$$

where $\mathcal{L}_{de}$ and $\mathcal{L}_{view}$ are utilized to remove the rain from rainy images, $\mathcal{L}_{seg}$ and $\mathcal{L}_{con}$ push the model to understand scene better, which is helpful for stereo deraining. λ_1, λ_2 and λ_3 are three parameters to balance different loss functions, which are set as 1.0, 0.2 and 1.0, respectively.

9.4 Experiments

9.4.1 Datasets

RainKITTI2012 dataset. To the best of our knowledge, there are no benchmark datasets that provide stereo rainy images and their corresponding ground-truth clean version. We first use Photoshop to create a synthetic RainKITTI2012 dataset based on the public KITTI stereo 2012 dataset [Geiger et al., 2013]. The training set contains $4,062$ image pairs from various scenarios, and the testing set contains $4,085$ image pairs. The size of images is 1242×375.

RainKITTI2015 dataset. The KITTI2015 dataset is another set from the KITTI stereo 2015 dataset [Geiger et al., 2013]. Therefore, we also synthesize a RainKITTI2015 dataset, whose training set and testing set contain $4,200$ and $4,189$ pairs of images, respectively.

Cityscapes dataset. Cityscapes dataset is utilized as the semantic segmentation data to train *PRRNet*. This dataset contains various urban street scene and provides images with pixel-wise segmentation labels. It includes $2,975$ images and their corresponding ground truth semantic labels.

RainCityscapes dataset. This dataset is built by [Hu et al., 2019] based on Cityscapes dataset [Cordts et al., 2016]. The training set contains $9,432$ rainy images and the corresponding clean images and depth labels. For evaluation, the testing set contains $1,188$ images with the size 2048×1024. We use this dataset to evaluate the performance of monocular deraining.

9.4.2 Implementation Details

The *SADM* has an encoder network and a decoder network. The encoder networks consists of 13 CNN layers, which is initialized by a VGG16 network pre-trained for object classification. The decoder has also 13 CNN layers. The *SFNet* contains three CNN layers ($32 \times 3 \times 3$) which are utilized to fuse the semantic information. The

Table 9.1: **Ablation study on the RainKITTI2012 dataset.**

Methods	PSNR	SSIM
PRRNet (D)	30.71	0.923
PRRNet (D+S)	31.56	0.928
PRRNet (D+S+L)	31.89	0.930
PRRNet (stereo)	33.01	0.936

(a) Input (b) PRRNet(D) (c) PRRNet(D+S)

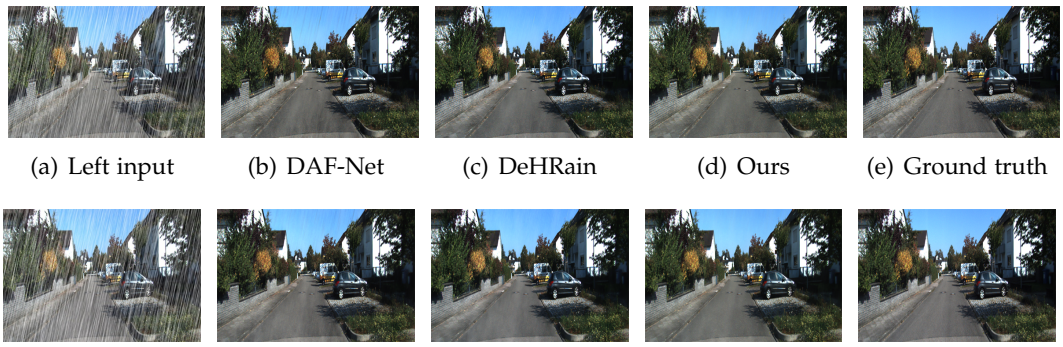
(d) PRRNet(D+S+L) (e) PRRNet(stereo) (f) Ground truth

Figure 9.6: **Deraining evaluation of different baseline models on the RainKITTI2012 dataset.**

VFNet contains five ResBlocks [He et al., 2016] to generate final deraining results. Each ResBlock consists of three CNN layers of $64 \times 3 \times 3$ kernels and two ReLU activation layers. The proposed *PRRNet* is trained with Pytorch library. The base learning rate is set to 10^{-4} and then declined to 10^{-5}. The model is updated with the batch size of 2 during the training stage. The branches of deraining and segmentation in *SADM* are optimized based on the data from RainKITTI2012/2015 and Cityscapes, respectively.

9.4.3 Ablation Study

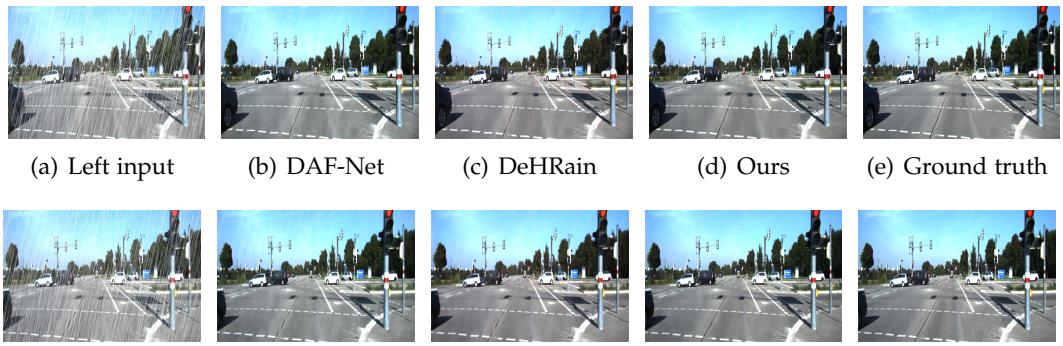
The proposed *PRRNet* takes advantage of semantic information to remove rain from images. In order to show the effectiveness of semantic information, we compare the performance of our model with the one which is trained without semantic information. Another advantage of the *PRRNet* is that it fuses the varying information in corresponding pixels across two stereo views to remove rain. Therefore, we also compare models trained on monocular and stereo images. Table 9.1 and Fig. 9.6 show the quantitative and qualitative comparison results. *PRRNet(D)* is the model



(a) Left input (b) DAF-Net (c) DeHRain (d) Ours (e) Ground truth

(f) Right input (g) DAF-Net (h) DeHRain (i) Ours (j) Ground truth

Figure 9.7: **Qualitative evaluation of current state-of-the-art models on the RainKITTI2012 dataset.**



(a) Left input (b) DAF-Net (c) DeHRain (d) Ours (e) Ground truth

(f) Right input (g) DAF-Net (h) DeHRain (i) Ours (j) Ground truth

Figure 9.8: **Qualitative evaluation of current state-of-the-art models on the RainKITTI2015 dataset.**

trained on monocular images with the single deraining task. *PRRNet(D+S)* is the one trained on monocular images with both deraining and segmentation tasks. *PRRNet(D+S+L)* is the model trained on monocular images with the above two tasks plus the semantic-rethinking loop. *PRRNet(stereo)* is our full model trained based on stereo images.

The results in Table 9.1 suggest that, the plain *PRRNet(D)* accomplishes the task fairly well. Additionally considering the semantic segmentation task, *PRRNet(D+S)* improves the performance. With the semantic-rethinking loop, the results are further improved by *PRRNet(D+S+L)*. However, the improvement is not as significant as that from *PRRNet(D+S+L)* to *PRRNet(stereo)* in the stereo case. This is also verified by the qualitative results in Fig. 9.6. Additional components incrementally improve the visibility of the input image, and the image generated by *PRRNet(stereo)* is the closest to the ground truth.

Table 9.2: **Quantitative evaluation of current state-of-the-art models on the RainKITTI2012 dataset.**

Methods	PSNR	SSIM
DDN [Yang et al., 2017]	29.43	0.904
DID-MDN [Zhang and Patel, 2018b]	29.14	0.901
DAF-Net [Hu et al., 2019]	30.44	0.914
DeHRain [Li et al., 2019b]	31.02	0.923
PRRNet(monocular)	**31.89**	**0.930**
PRRNet(stereo)	**33.01**	**0.936**

Table 9.3: **Quantitative evaluation of current state-of-the-art models on the RainKITTI2015 dataset.**

Methods	PSNR	SSIM
DDN [Yang et al., 2017]	29.23	0.906
DID-MDN [Zhang and Patel, 2018b]	28.97	0.899
DAF-Net [Hu et al., 2019]	30.17	0.915
DeHRain [Li et al., 2019b]	30.84	0.921
PRRNet(monocular)	**31.64**	**0.932**
PRRNet(stereo)	**32.58**	**0.937**

9.4.4 Stereo Deraining

We quantitatively and qualitatively compare our *PRRNet* with current state-of-the-art methods, which includes DDN [Yang et al., 2017], DID-MDN [Zhang and Patel, 2018b], DAF-Net [Hu et al., 2019] and DeHRain [Li et al., 2019b]. Table 9.2 and Table 9.3 show the quantitative results on our synthesized RainKITTI2012 and RainKITTI2015 datasets, respectively. In both tables, our monocular version, *PRRNet(monocular)*, outperforms the existing state-of-the-art methods, with remarkable advance. The model *PRRNet(stereo)* achieves the best performance with additional improvement. This demonstrates the advance of stereo deraining over monocular deraining.

Figs. 9.7 and 9.8 compare the qualitative performance between our method *PRRNet(stereo)* and various state-of-the-art methods. The results produced by our method exhibit the smallest portion of artifact, by referring the ground truth.

9.4.5 Monocular Deraining

The proposed *PRRNet* is not only able to remove rain from stereo images, but also has the advantage of removing rain from a single image with its monocular version. In this section, we also evaluate it on the monocular dataset RainCityscapes. We compare the *PRRNet*'s monocular version, *PRRNet(monocular)*, with state-of-the-art methods, including DID-MDN [Zhang and Patel, 2018b], RESCAN [Li et al., 2018d], JOB [Zhu et al., 2017c], GMMLP [Li et al., 2016], DSC [Luo et al., 2015], DCPDN

Table 9.4: **Quantitative evaluation of current state-of-the-art models on the RainCityscapes dataset.**

Methods	PSNR	SSIM
DID-MDN [Zhang and Patel, 2018b]	28.43	0.9349
RESCAN [Li et al., 2018d]	24.49	0.8852
JOB [Zhu et al., 2017c]	15.10	0.7592
GMMLP [Li et al., 2016]	17.80	0.8169
DSC [Luo et al., 2015]	16.25	0.7746
DCPDN [Zhang and Patel, 2018a]	28.52	0.9277
DAF-Net [Hu et al., 2019]	30.06	0.9530
***PRRNet*(monocular)**	**31.44**	**0.9688**

(a) Input (b) DID-MDN (c) DAF-Net (d) Ours (e) Ground truth

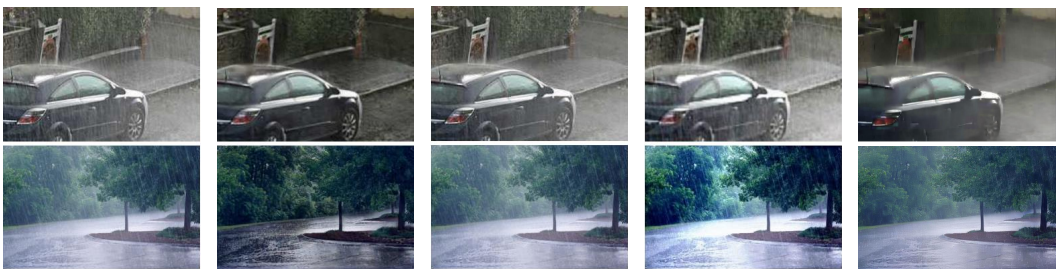
Figure 9.9: **Qualitative evaluation of current state-of-the-art models on the RainCityscapes dataset.**

[Zhang and Patel, 2018a], DAF-Net [Hu et al., 2019], from both quantitative and qualitative aspects.

The quantitative results on the RainCityscapes dataset are shown in Table 9.4. DID-MDN [Zhang and Patel, 2018b] and DCPDN [Zhang and Patel, 2018a] perform well and DAF-Net [Hu et al., 2019] outperforms these two methods. Our monocular version *PRRNet(monocular)* achieves the best performance compared with all the compared methods on this task, revealing the effectiveness of taking semantic segmentation into consideration and the semantic-rethinking loop. Fig. 9.9 compares its qualitative performance with different methods. The results show that the monocular version of our *PRRNet* also achieves the best performance in terms of monocular image deraining.

9.4.6 Evaluation on Real-world Images

To further verify the effectiveness of our method, we show its performance of deraining on the real world rainy images. Fig. 9.10 shows the qualitative results on two exemplar images from the Internet. Compared to other competing methods, the proposed method achieves better performance via understanding the scene structure. For example, DAF-Net seems generate well-derained images, but the produced derained images suffer from color distortion (*e.g.* the colors turn dark in the results).



(a) Input (b) DAF-Net (c) RESCAN (d) RESCAN + DCPDN (e) Ours

Figure 9.10: **Qualitative evaluation on real rainy images.** From left to right are the input images, DAF-Net [Hu et al., 2019], RESCAN [Li et al., 2018d], RESCAN + DCPDN [Zhang and Patel, 2018a] and ours, respectively.

RESCAN and RESCAN+DCPDN perform worse than our method in removing rain.

9.5 Conclusion

The main contribution of this chaper is that we present *PRRNet*, the first stereo semantic-aware deraining networks, for stereo image deraining. Different from previous methods which only learn from pixel-level loss functions or monocular information, the proposed model advances image deraining by leveraging semantic information extracted by a semantic-aware deraining model, and visual deviation between two views fused by two Fusion Nets, i.e., , *SFNet* and *VFNet*. We also synthesize two stereo deraining datasets to evaluate different deraining methods. Experimental results show that our *PRRNet* outperforms the state-of-the-art methods on both monocular and stereo image deraining.

Chapter 10

Conclusion and Future Work

Image enhancement is one of the fundamental problems in computer vision which has broad application. This topic remains an active topic in recent decades but many classic problems are yet to be solved.

10.1 Conclusion

This thesis has been devoted to investigating the problems of image enhancing. It addresses current challenges and pushes the limits of the state-of-the-art in various tasks, including image deblurring and image deraining. We have made several key contributions in three aspects.

Deblurring. For *single image deblurring (Chapter 3)*, we propose a new method which combines two GAN models, *i.e.*, a learning-to-blur GAN and learning-to-deblur GAN, to learn a better model for image deblurring by learning how to blur images. In order to reduce the discrepancy between real blur and synthesized blur, a relativistic blur loss is leveraged. As an dditional contribution, we also introduces a real-world blurred image datast including diverse blurry images.

For *video deblurring (Chapter 4)*, we present a network for spatial-temporal learning by applying a 3D convolution to both spatial and temporal domains. To restore sharp image details, we further employ a content loss and an adversarial loss for efficient adversarial training.

For *making a blurry image alive (Chapter 5)*, we introduce an method to extract an image sequence from a single motion-blurred images. Motion-blurred image are the results of light accumulation over the period of camera exposure times, during which the camera and objects in the scene are in relative motion to each other. The inverse process is an ill-posed vision problem. To alleviate this issue, we propose a detail-aware network with three consecutive stages to improve the reconstruction quality by addressing different challenges in the recovery process.

For *benchmarking current deblurring methods (Chapter 6)*, we construct a new large-scale dataset called the Multi-Cause Image Deblurring (MCID) dataset. The dataset includes blurry images generated by averaging sharp images captured by a 1000fps high-speed camera, images obtained by convolving Ultra-High-Definition (UHD) sharp images with a large kernel size, blurry images with a defocus effect, and real-

world blurred images captured by various cameras. The results provide a comprehensive overview of the advantages and limitations of current deblurring methods. Further, we propose a level-attention deblurring network to adopt to multiple causes of blur.

Deraining. For *single image deraining (Chapter 7)*, we propose a dual attention-in-attention model which includes two attention models for removing both rain streaks and raindrops. To further refine the results, a differential-driven module is proposed to remove rain via addressing the unsatisfying deraining regions.

For *video deraining (Chapter 8)*, we present new end-to-end framework, which takes the advantage of deep residual features and temporal correlations among continuing frames at the cost of very little computational source. The experimental results show that it achieve faster speed than the competitors, while maintaining better performance than the state-of-the-art methods.

For *stereo deraining (Chapter 9)*, we present a paired rain removal method, which is the first stereo semantic-aware deraining network. Experimental results on developed stereo rainy datasets demonstrate that the proposed method achieves the state-of-the-art performance on both monocular and stereo image deraining.

Even four image deblurring and three deraining methods are proposed. All of these methods face a main challenge, *i.e.*, they cannot achieve the same performance on real-world images as on synthesized images. Therefore, how to capture realistic training datasets to help train current data-driven methods, or use machine learning algorithms like domain adaptation to alleviate the gap between real and synthesized samples, remain open research topics.

10.2 Future work

10.2.1 Deep Image Deblurring: A Survey

In chapter 3, 4, 5, 6, we propose four deep image deblurring methods. In future, we will write a survey to give a comprehensive overview of recent advances in image deblurring with deep learning. In this survey, we will review the preliminaries for image deblurring, including problem definitions, causes of blurs, types of deblurring, image quality assessment, and deep learning architectures. In addition, the recent developments of deep learning models for single image deblurring and video deblurring will be discussed. Finally, we will analyze the challenges of image deblurring and discuss the possible future research opportunities for image deblurring.

10.2.2 Blind Face Restoration

Blind face restoration recovers high quality images from low quality images with unknown degradation. It has been widely applied to real-world scenarios like old photo renovation, low quality face recognition and detection. These thesis provides several methods to improve the quality of image, including image deblurrring and

image draining. However, compared with general image restoration, face restoration can make use of abundant facial prior knowledge, such as parsing maps, facial heatmaps and face reference priors, to recover details even if the images are severely degraded. Therefore, how to use various facial prior knowledge is also an important research direction in deep learning. In future, we will consider to use the architecture of Neural Architecture Search to study how to effectively learn multiple facial priors for blind face restoration.

Appendix A

APPENDIX: Learning Joint Gait Representation via Quintuplet Loss Minimization

Gait recognition is an important biometric technique relevant to video surveillance, where the task is to identify people at a distance by their walking patterns captured in the video. Most of the current approaches for gait recognition either use a pair of gait images to form a *cross-gait* representation or rely on a single gait image for *unique-gait* representation. These two types of representations empirically complement one another. In this chapter, we propose a new *Joint Unique-gait and Cross-gait Network (JUCNet)* representation, to combine the advantages of both schemes, leading to significantly improved performance. A second contribution of this work is a tailored *quintuplet* loss function, which simultaneously boosts inter-class differences by pushing different subjects further apart and contracts intra-class variations by pulling same subjects closer. Extensive tests demonstrate that our method achieves the best performance tested on multiple standard benchmarks, compared with other state-of-the-art methods.

A.1 Introduction

Gait recognition is the task of identifying people at a distance using videos of their walking patterns [Wang et al., 2003a]. This is an active research topic in the field of computer vision, due to its importance in real-world applications such as video surveillance, forensic identification, and evidence collection [Bouchrika et al., 2011; Larsen et al., 2008]. As a behavioral biometric, gait exhibits unique advantages over other biometrics like fingerprint, iris and face [Wang et al., 2018a], because gait based methods can identify subjects from low-resolution video sequences [Mori et al., 2010] without subject's cooperation.

In real-world scenarios, variations such as clothing [Rokanujjaman et al., 2015], walking speed [Mansur et al., 2014; Tsuji et al., 2010], carrying condition [Tao et al., 2007], and camera viewpoints [Lu and Tan, 2010] result in remarkable changes in gait appearance, which may further degrade the performance of gait recognition. Previ-

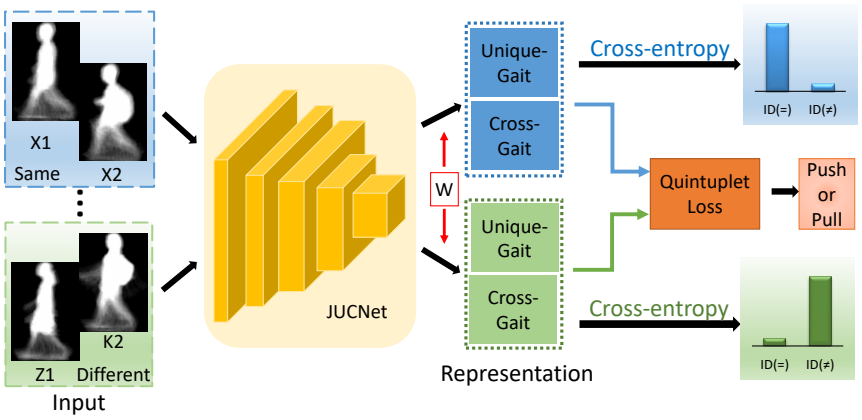


Figure A.1: **An illustration of our feature learning process.** The JUCNet structure synchronously learns unique-gait and cross-gait representations, and the Quintuplet loss is proposed to increase the inter-class differences and meanwhile reduce the intra-class variations.

ous methods [Kusakunniran et al., 2009, 2013; Wu et al., 2017] have been proposed to alleviate these issues. Most of them focus on *cross-gait* representation, which is the concatenation of a pair of gait images and labeled to *"Same"* or *"Different"* like the input of Fig. A.1. While being effective in capturing the relationship between a pair of gaits (gallery and probe [Li and Jain, 2015]), these methods ignore the label (*e.g.*, "$X1$", "$X2$", "$Z1$", and "$K2$" in Fig. A.1) of each single gait image. The potential of *unique-gait/single-gait* representation is ignored, which makes these methods confused in discriminating different subjects with similar clothing, illumination, and carrying conditions. For example, X_1 and Y_1 in Fig. A.2 (a) may be predicted to be an identical subject as they are close in the feature space. Nowadays, some deep learning methods (*e.g.*, [Shiraga et al., 2016]) tackle this problem based on unique-gait representation solely. They extract unique-gait features enclosed in a single image and then match them to predict the relationship. While these methods ignore the cross-gait representation.

To this end, we develop a deep network called *JUCNet* to jointly learn the unique-gait and cross-gait representations. Different from existing gait recognition methods, there are three output branches in our network, of which two branches learn unique-gait representation and one branch learns concatenated cross-gait representation. Fig. A.2 (b) shows the effectiveness of JUCNet. Additionally considering the identity uniqueness, our model can extract discriminative features, which enlarges the inter-class variations due to the uniqueness information. This could improve the performance in the case that gaits are difficult to recognize based on sole cross-gait information.

When conducting recognition, conventional models rank the affinity scores of a given probe against all gallery gaits. To achieve this, these models are usually trained by combining a pair of gaits as a whole, and predicting their relationship via a binary classifier supervised by recognition signals. By doing so, they can obtain correct

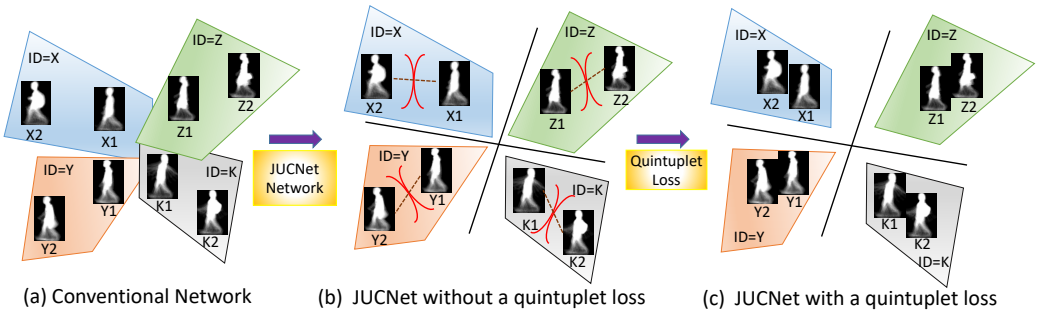


Figure A.2: **A conventional network, our JUCNet without and with the quintuplet loss are shown in (a), (b), and (c), respectively.** From (a) to (b), JUCNet additionally learns the identical unique-gait representation, which enlarges inter-class differences among subjects. From (b) to (c), not only the inter-class variations increases, but also the intra-class discrepancy is decreased, with the help of the proposed quintuplet loss. Red arch lines of each subject domain in (b) indicate the significant intra-class discrepancy, which is reduced as shown by the red circles in (c).

classification on the training set. However, models trained in this way extract features of relatively large intra-class variations and small inter-class differences, leading to inferior performance in the testing stage. Though JUCNet is designed to enlarge inter-class differences to some extent, the intra-class variations are still large. For instance, JUCNet increases the distance between inter-class subjects (*e.g.*, X_1 and Y_1 in Fig. A.2 (b)), while the intra-class subjects (*e.g.*, X_1 and X_2) are not sufficiently tight.

In order to address this issue, we propose a quintuplet loss function which is a joint of both recognition and verification signals as the supervision. The basic JUCNet described above is therefore extended to be Multi-Pair JUCNet. This Multi-Pair JUCNet, trained effectively with the proposed quintuplet loss, learns to enlarge the inter-class differences by separating the cross-gait representation from different classes and reduces the intra-class variations by grouping the representation in the same class together. Fig. A.2 (c) shows the effect. The distance between gait features from different subjects (*e.g.*, X_1 and Y_1) becomes larger, while the discrepancy of gait features from an identical subject (*e.g.*, X_1 and X_2) becomes smaller.

Our main contributions are as follows. 1) We develop a neural network called JUCNet, which jointly learns unique-gait representation and cross-gait representation. The two kinds of representations complement each other and boost the performance of gait recognition. 2) An effective loss function for gait recognition, termed as quintuplet loss, is proposed to guide an extension of JUCNet, named as Multi-Pair JUCNet, to extract powerful features with small intra-class variations and large inter-class differences. 3) Our proposed model outperforms the state-of-the-art models on public challenging gait datasets, showing its superiority.

A.2 Related Work

Model based methods. These methods aim to model the underlying structure of human body and extract motion features for recognition [Ariyanto and Nixon, 2011; Bodor et al., 2009; Kusakunniran et al., 2009]. They have the advantage of recognizing gaits under various situations like different clothing, carrying conditions, *etc*. It is difficult for these methods to model body structures from relatively low-resolution images, so they can merely work under uncontrolled conditions.

Appearance based methods. Appearance based methods [Goffredo et al., 2010; Kusakunniran et al., 2014, 2013; Makihara et al., 2006; Man and Bhanu, 2006; Murase and Sakai, 1996; Wagg and Nixon, 2004; Wang et al., 2012] directly extract gait features from videos without modeling the underlying structure of human body. Therefore these methods can work in low-resolution conditions. They usually consist of three steps: 1) obtaining human silhouettes, 2) computing silhouette based representations such as Gait Energy Images (GEIs) [Man and Bhanu, 2006], chrono-gait images [Kusakunniran, 2014], and gait flow [Lam et al., 2011], and 3) evaluating similarities between gaits.

Deep neural network based methods. Deep learning methods have achieved a great success in the field of computer vision [He et al., 2016; Simonyan and Zisserman, 2015a; Tang et al., 2017, 2018; Zhang et al., 2017; Luo et al., 2019; Feng et al., 2018]. Recent methods for gait recognition have also adopted CNNs [Alotaibi and Mahmood, 2017; Castro et al., 2017; Wu et al., 2017; Yu et al., 2017]. These methods learn features from pair GEIs in low-level [Wu et al., 2017; Yu et al., 2017], middle-level, or high-level layers [Alotaibi and Mahmood, 2017; Castro et al., 2017; Feng et al., 2016; Shakhnarovich and Darrell, 2002; Wu et al., 2017] and then forward features to a binary classifier for prediction. Wu *et al.* [Wu et al., 2017] conducted comprehensive experiments to evaluate these models. However, in these methods, models are trained by merely learning the cross-gait representation, ignoring the identical uniqueness. On the other hand, representative works like [Castro et al., 2017] train models based on unique-gait representations, without considering useful cross-gait representations. On the contrary, the proposed JUCNet learns both unique-gait and cross-gait representations. Meanwhile, we design a quintuplet loss to guide the model to extract features with smaller intra-class variations and larger inter-class differences.

A.3 Joint Learning with a Quintuplet Loss

Our method jointly learns unique-gait and cross-gait representations based on a proposed quintuplet loss. Before introducing JUCNet, we represent the method of joint learning and the quintuplet loss in the following.

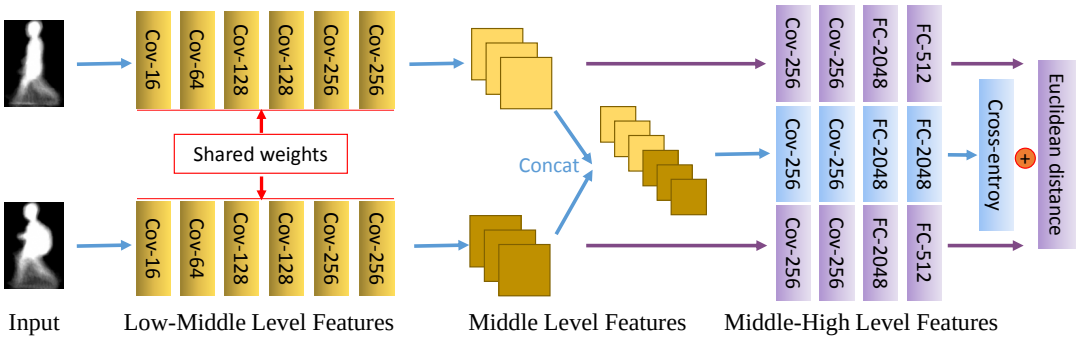


Figure A.3: **The architecture of the basic JUCNet model for gait recognition. Its input is a pair of gaits.** There are three output branches, with two corresponding to unique-gait representations (purple part) and one for cross-gait representation (blue part). The unique-gait and cross-gait representations complement each other to update our model.

A.3.1 Joint Learning

Cross-gait Learning. Methods based on cross-gait representation concatenate probe and gallery gait features and input them to a binary classifier to obtain the correct order via their ranking scores. In this work, we denote an instance of pair gaits as $\{(\mathbf{x}_p, \mathbf{x}_g), \theta_{pg}\}$, where $\mathbf{x}_p$ is the p-th probe, $\mathbf{x}_g$ is the g-th gallery gait, and θ_{pg} is the relationship between them. $\theta_{pg} = 0$ means that $\mathbf{x}_p$ and $\mathbf{x}_g$ come from an identical subject, and $\theta_{pg} = 1$ indicates that they are from different subjects. The cross-gait representation should satisfy the following conditions,

$$\begin{aligned} d(\mathbf{x}_p, \mathbf{x}_g) &\leq \mathrm{b}_c - 1 + \delta_{pg}, \quad if\ \theta_{pg} = 0, \\ d(\mathbf{x}_p, \mathbf{x}_g) &\geq \mathrm{b}_c - 1 + \delta_{pg}, \quad if\ \theta_{pg} = 1, \end{aligned} \tag{A.1}$$

where δ_{pg} is a nonnegative slack variable, b_c is a distance threshold, and $d(\cdot, \cdot)$ is a predefined or learned metric measuring discrepancy between a pair of gaits. We minimize the cross-entropy loss which is formulated as,

$$\mathcal{L}_c(\mathbf{x}_p, \mathbf{x}_g) = -\sum_{p,g} P(\mathbf{x}_{pg}) \log Q(\mathbf{x}_{pg}), \tag{A.2}$$

where $\mathbf{x}_{pg}$ is the cross-gait feature vector, $P(\mathbf{x}_{pg})$ is the true distribution, and $Q(\mathbf{x}_{pg})$ is the predicted distribution.

Unique-gait Learning. Similar to the learning of cross-gait representation, the unique-gait representation should satisfy the constraints as,

$$\begin{aligned} ||U(\mathbf{x}_p) - U(\mathbf{x}_g)||_2^2 &\leq \mathrm{b}_u - 1 + \delta_{pg}, \quad if\ \theta_{pg} = 0, \\ ||U(\mathbf{x}_p) - U(\mathbf{x}_g)||_2^2 &\geq \mathrm{b}_u - 1 + \delta_{pg}, \quad if\ \theta_{pg} = 1, \end{aligned} \tag{A.3}$$

where $U(\mathbf{x}_p)$ and $U(\mathbf{x}_g)$ are unique-gait representations and b_u is a distance threshold

between them. In this formulation, the discrepancy between unique-gait representations from identical subjects in terms of Euclidean distance is expected to be smaller than b_u, while that of unique-gait representations from different subjects is expected to be greater than b_u.

In our model, we consider multiple pairs of gaits as input, so the above constraints should be modified as follows,

$$||U(\mathbf{x}_{\hat{p}}) - U(\mathbf{x}_{g'})||_2^2 - ||U(\mathbf{x}_p) - U(\mathbf{x}_g)||_2^2 \geq 1 - \delta_{pg}, \tag{A.4}$$

where $\{\mathbf{x}_p, \mathbf{x}_g\}$ come from an identical subject, while $\{\mathbf{x}_{\hat{p}}, \mathbf{x}_{g'}\}$ are from different subjects. Our aim is to make the distinction between $\mathbf{x}_{\hat{p}}$ and $\mathbf{x}_{g'}$ greater than the distance between $\mathbf{x}_p$ and $\mathbf{x}_g$. The above constraint should be satisfied no matter $\mathbf{x}_p$ and $\mathbf{x}_{\hat{p}}$ are identical or not. Thus, the loss function of learning unique-gait representation is composed of two terms,

$$\begin{aligned}\mathcal{L}_u(\mathbf{x}_p, \mathbf{x}_g, \mathbf{x}_{p'}, \mathbf{x}_{g'}) = \sum_{p,g,g',p'} \{[1 + ||U(\mathbf{x}_p) - U(\mathbf{x}_g)||_2^2 - ||U(\mathbf{x}_p) - \\ U(\mathbf{x}_{g'})||_2^2]_+ + \eta_i \cdot [1 + ||U(\mathbf{x}_p) - U(\mathbf{x}_g)||_2^2 - ||U(\mathbf{x}_{p'}) - U(\mathbf{x}_{g'})||_2^2]_+\},\end{aligned} \tag{A.5}$$

where $[z]_+ = max(z, 0)$. The first term corresponds to the case that $\mathbf{x}_p$ and $\mathbf{x}_{\hat{p}}$ are identical ($\hat{p} = p$), the second term corresponds to the case that $\mathbf{x}_p$ and $\mathbf{x}_{\hat{p}}$ are different ($\hat{p} \neq p$ thus we employ p' for clarity). We note that, in both cases, $\{\mathbf{x}_p, \mathbf{x}_{g'}\}$ and $\{\mathbf{x}_{p'}, \mathbf{x}_{g'}\}$ are from different subjects, individually.

Joint Learning Function. Finally, JUCNet is updated based on both unique-gait and cross-gait representations, so the overall loss function is the combination of $\mathcal{L}_c$ and $\mathcal{L}_u$,

$$\mathcal{L}_o = \mathcal{L}_c + \eta_u \cdot \mathcal{L}_u, \tag{A.6}$$

where η_u is a hyperparameter to balance cross-gait and unique-gait.

A.3.2 Quintuplet Loss

The popular methods for learning the cross-gait representation summarized in Wu *et al.* [Wu et al., 2017] are based on recognition signals in Eq. (A.2), which aims to classify concatenated cross-gait representation. Namely, one class is "identical subject", and the other class is "different subjects". In order to obtain more powerful cross-gait representation, we adopt both recognition and verification signals as our supervision and propose a quintuplet loss, targeting at simultaneously enlarging the inter-class differences and reducing the intra-class variations. Different from the traditional recognition-verification loss [Mobahi et al., 2009; Sun et al., 2014b; Chen et al., 2017], we define a novel quintuplet loss associated with quintuplet gaits. This loss function considers not only discriminating gait *instances*, but also differentiating gait *pairs*.

The Euclidean distance can be employed to measure the similarity between two gaits in the quintuplet loss. While in this work, we replace the Euclidean distance

with a *learned metric* $C(\cdot,\cdot)$, which represents the distance between two gaits. Specially, the concatenated cross-gait features are forwarded to a fully-connected layer with two neurons. The output value of one neuron is set to be the metric. Considering multiple pairs, constraints in Eq. (A.1) are reformulated as,

$$C(\mathbf{x}_{\hat{p}}, \mathbf{x}_{g'}) - C(\mathbf{x}_p, \mathbf{x}_g) \geq 1 - \delta_{pg\hat{p}g'}, \tag{A.7}$$

where $\{\mathbf{x}_p, \mathbf{x}_g\}$ are from an identical subject, while $\{\mathbf{x}_{\hat{p}}, \mathbf{x}_{g'}\}$ are from different subjects. $\delta_{pg\hat{p}g'}$ is a nonnegative slack variable. Different from the loss function Eq. (A.2) utilized in [Wu et al., 2017], the loss with the learned metric $C(\cdot,\cdot)$ can be denoted as

$$\mathcal{L}_c(\mathbf{x}_p, \mathbf{x}_g, \mathbf{x}_{\hat{p}}, \mathbf{x}_{g'}) = \sum_{p,g,\hat{p},g'} [C(\mathbf{x}_p, \mathbf{x}_g) - C(\mathbf{x}_{\hat{p}}, \mathbf{x}_{g'}) + \delta_1]_+, \tag{A.8}$$

where δ_1 is the value of margin. The last fully-connected layer is followed by a softmax layer, which normalizes the learned metric into the range of [0, 1].

Due to the normalization operation, the parameter δ_1 is set to 1 in our model. The purpose of the above loss can be concluded as two aspects: 1) Gaits from the same subject $\{\mathbf{x}_p, \mathbf{x}_g\}$ are predicted to the class with label 0 and gaits from different subjects $\{\mathbf{x}_{\hat{p}}, \mathbf{x}_{g'}\}$ are predicted to the other class (label = 1). 2) The distance between $C(\mathbf{x}_p, \mathbf{x}_g)$ and $C(\mathbf{x}_{\hat{p}}, \mathbf{x}_{g'})$ is enlarged as far as possible. The first aspect can be regarded as a binary classification problem, which is to classify the concatenated cross-gait representation with recognition signals. The second aspect can be treated as a verification problem, which aims to make a distinction between the cross-gait representation from an identical subject and the cross-gait representation from different subjects.

To employ both recognition and verification signals for more powerful cross-gait features with smaller intra-class variations and larger inter-class differences, the loss function of cross-gait is reformulated as

$$\begin{aligned}\mathcal{L}_c\left(\mathbf{x}_p, \mathbf{x}_g, \mathbf{x}_{\hat{p}}, \mathbf{x}_{g'}, \mathbf{x}_{p''}\right) = &- \sum_{p,g,\hat{p},g'} \left[P(\mathbf{x}_{pg}) \log Q(\mathbf{x}_{pg}) + P(\mathbf{x}_{\hat{p}g'}) \log Q(\mathbf{x}_{\hat{p}g'})\right] \\ &+ \eta_c \cdot \sum_{\substack{p,g \\ \hat{p},g',p''}} \left[\delta_2 - D(C(\mathbf{x}_p, \mathbf{x}_g), C(\mathbf{x}_{\hat{p}}, \mathbf{x}_{g'})) + D(C(\mathbf{x}_{\hat{p}}, \mathbf{x}_{g'}), C(\mathbf{x}_{p''}, \mathbf{x}_{g'}))\right]_+,\end{aligned} \tag{A.9}$$

where $D(x, y) = ||x - y||_2^2$. The pair gaits $\{\mathbf{x}_p, \mathbf{x}_g\}$ come from an identical subject, while $\{\mathbf{x}_{\hat{p}}, \mathbf{x}_{g'}\}$ and $\{\mathbf{x}_{p''}, \mathbf{x}_{g'}\}$ are from different subjects. The first term in the right hand is based on the recognition signal, which denotes the classification of gait-cross representation. The second term is based on the verification signal, denoting whether two pairs of gait-cross representations are of the same pair-wise class label (both pairs from identical subjects or both pairs from different subjects, which is the case in Fig. A.4) or not (one pair from an identical subject and the other pair from different subjects).

Similar to the extension from Eq. (A.4) to Eq. A.5, the constraint in Eq. A.7 should be satisfied no matter $\mathbf{x}_p$ and $\mathbf{x}_{\hat{p}}$ are identical or not. Therefore, the two terms in the right hand of Eq. A.9 for learning cross-gait representation are respectively extended

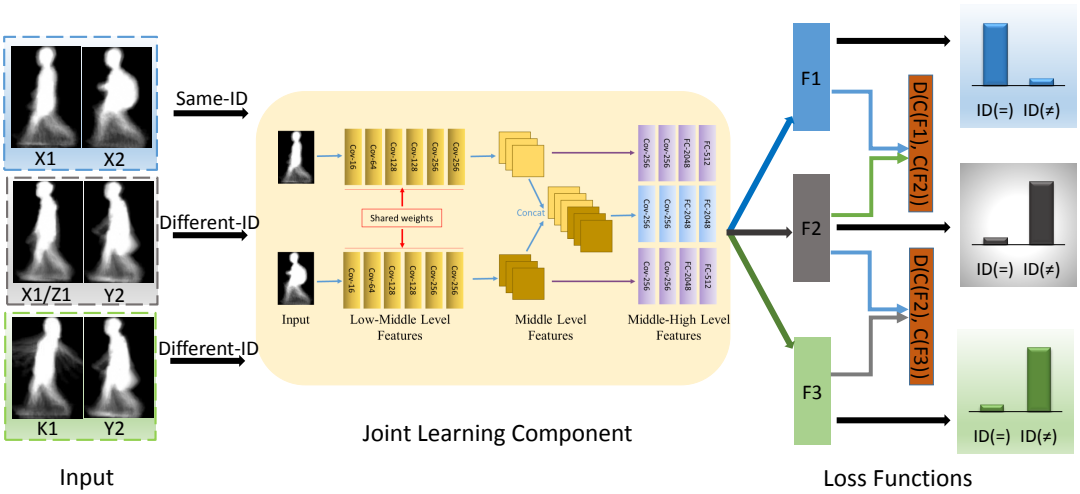


Figure A.4: **The Multi-Pair JUCNet structure based on the Quintuplet loss in the training stage.** The input to our network is several pairs of gaits. Features are extracted from each pair individually, and are processed based on the quintuplet loss. Here, the quintuplet associated with our quintuplet loss can be regarded as X1, X2, Y2, Z1, and K1.

to cover both cases ($p = \hat{p}$ and $p \neq \hat{p}$) as

$$\begin{aligned}\mathcal{L}_c(\mathbf{x}_p, \mathbf{x}_g, \mathbf{x}_{p'}, \mathbf{x}_{g'}, \mathbf{x}_{p''}) = -\sum_{\substack{p,g\\p',g'}} [(P(\mathbf{x}_{pg}) \log Q(\mathbf{x}_{pg}) \\ + P(\mathbf{x}_{pg'}) \log Q(\mathbf{x}_{pg'})) + \eta_i \cdot (P(\mathbf{x}_{pg}) \log Q(\mathbf{x}_{pg}) + P(\mathbf{x}_{p'g'}) \log Q(\mathbf{x}_{p'g'}))] \\ + \eta_c \cdot \sum_{\substack{p,g\\g',p'\\p''}} [||\delta_2 - D(C(\mathbf{x}_p, \mathbf{x}_g), C(\mathbf{x}_p, \mathbf{x}_{g'})) + D(C(\mathbf{x}_p, \mathbf{x}_{g'}), C(\mathbf{x}_{p''}, \mathbf{x}_{g'}))||_+ \\ + \eta_i \cdot ||\delta_2 - D(C(\mathbf{x}_p, \mathbf{x}_g), C(\mathbf{x}_{p'}, \mathbf{x}_{g'})) + D(C(\mathbf{x}_{p'}, \mathbf{x}_{g'}), C(\mathbf{x}_{p''}, \mathbf{x}_{g'}))||_+],\end{aligned} \tag{A.10}$$

where $\{\mathbf{x}_p, \mathbf{x}_g\}$ are from an identical subject, while $\{\mathbf{x}_p, \mathbf{x}_{g'}\}$, $\{\mathbf{x}_{p'}, \mathbf{x}_{g'}\}$, and $\{\mathbf{x}_{p''}, \mathbf{x}_{g'}\}$ come from different subjects, respectively. The hyperparameters η_c and η_i are used to balance different terms. We replace $\mathcal{L}_c$ in Eq. (A.6) with the above formulation in the training stage. As it may be noticed, there are quintuplet gait instances ($\mathbf{x}_p, \mathbf{x}_g, \mathbf{x}_{p'}, \mathbf{x}_{g'}$ and $\mathbf{x}_{p''}$) in Eq. A.10, which are the proposed quintuplet loss named after.

A.4 JUCNet

In this section, we introduce the architecture of the JUCNet, then present a Multi-Pair JUCNet model and the training procedure based on the quintuplet loss.

A.4.1 Basic JUCNet

As shown in Fig. A.3, given a pair of gray-scale gait images, our JUCNet model jointly learns the unique-gait and cross-gait representations, in both low-middle and middle-high levels. The components for learning unique-gait and cross-gait representations are presented in the **purple** and **blue** parts, respectively.

Middle-level features. The component for capturing middle-level features is shown as the yellow part in Fig. A.3, consisting of six convolutional layers. The numbers of kernels in each convolutional layer are sequentially 16, 64, 128, 128, 256, and 256, respectively. The activation function of convolutional layers is Rectified Linear Unit (ReLU). The size of all filters in this stage is 3×3 with stride 1. Each of the convolutional layers is followed by a max-pooling layer of size 2×2 and stride 2.

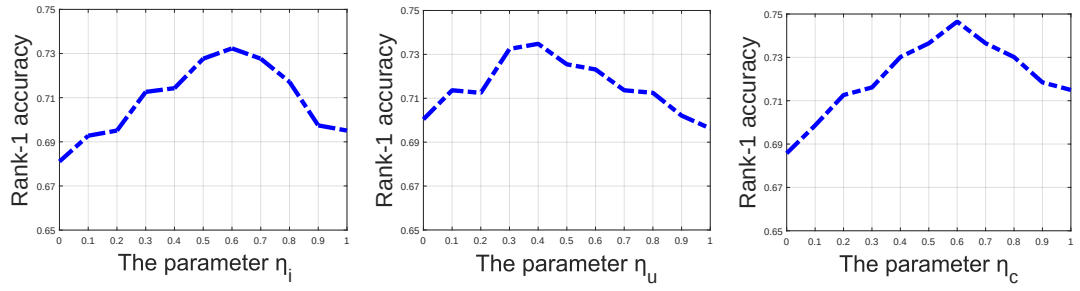
High-level features. The part learning high-level features is composed of three branches, of which two learn unique-gait representation and one learns cross-gait representation. Each branch of learning unique-gait representation includes two convolutional layers and two fully-connected layers. Middle-level feature maps with 256 channels are forwarded to the first convolutional layer with 256 kernels of size 3×3 and stride 1. The second convolutional layer also contains 256 kernels of size 3×3 and stride 1. Both of them are followed by a max-pooling layer with pooling size 2×2 and stride 2. After the convolutional layers, two fully-connected layers project feature maps extracted from previous layers into a subspace by 2048 and 512 neurons, respectively.

The component for learning cross-gait representation is also comprised of two convolutional layers and two fully-connected layers. Middle-level features are concatenated as cross-gait feature vectors, which are input into a convolutional layer with 256 kernels of size 3×3 and stride 1. The difference from the first layer of learning the unique-gait representation is that the number of kernels is doubled due to concatenation. The second convolutional layer and the first fully-connected layer are the same as those learning unique-gait representation. The second fully-connected layer contains 2048 neurons.

A.4.2 Multi-Pair JUCNet

As described above, JUCNet learns both unique-gait and cross-gait representations. The proposed quintuplet loss can enlarge inter-class differences and reduce intra-class variations simultaneously. To this end, we extend the basic JUCNet as a Multi-Pair JUCNet, which serves as the final framework during training, and train it with the quintuplet loss.

Fig. A.4 shows the overview of the Multi-Pair JUCNet. A pair of gaits can be combined as a whole, with the label of *Same-ID* or *Different-ID*. The basic JUCNet model extracts both unique-gait and cross-gait representations. For Multi-Pair JUCNet, three pairs of gaits are input to extract features. Two pairs of gaits are from different subjects, while one pair of gaits is from an identical subject. Our model learns unique-gait representation based on the loss in Eq. (A.5), and learns cross-gait representation based on the quintuplet loss in Eq. (A.10).



(a) (b) (c)

Figure A.5: **The Rank-1 accuracy by varying the weighting parameters η_i, η_u, and η_c investigated on the validation set of OU-LP-Bag β.** When varying one hyperparameter, the other two are fixed.

A.4.3 Training

We choose the popular GEIs [Man and Bhanu, 2006] as the input of Multi-Pair JUCNet because of its robustness to noise and its simplicity for computation [Iwama et al., 2012]. GEIs images are resized to the size of $256 \times 372 \times 1$. In order to augment training samples, we crop a set of $224 \times 326 \times 1$ patches from GEIs images and flip them horizontally at random. It is worth noting that a pair of GEIs are flipped at the same time to ensure the same walking direction. It is trained based on stochastic gradient descent. Weights are initialized as a Gaussian distribution with mean 0 and standard deviation 0.01. The momentum is set as 0.9. The model is updated every time after learning one mini-batch of size 32.

A.5 Experiments

To verify our model, we test it on three public datasets, which are at first introduced in this section. These datasets cover challenges like clothing variation, cross view, *etc.* in the task of the gait recognition. Based on the datasets, we then investigate effectiveness of JUCNet and the quintuplet loss. Meanwhile, comparison with the state-of-the-art methods is also reported. Finally, we study the performance of our method with the protocol of cross view.

A.5.1 Datasets

The OUTD-B dataset. The OU-ISIR Gait Database, Treadmill Dataset B (OUTD-B) [Makihara et al., 2012], is challenging due to its considerable clothing diversities, such as wearing hat, regular pants, and half shirt. It is composed of 68 subjects with up to 32 clothing conditions. There are three subsets in this dataset, a training set, a gallery set, and a probe set. The training set includes 20 subjects with 446 sequences. The gallery set and probe set are employed in the testing stage. There are 48 subjects with standard clothing types in the gallery set. The probe set contains 856 sequences

of subjects with other clothing types. Note that subjects in the gallery set and probe set are disjoint from those in the training set.

The OU-LP-Bag β dataset. The OU-LP-Bag β database [Makihara et al., 2017] is built to alleviate the problem of too small variations in existing datasets. There are one training set, one gallery set, and one probe set in this dataset. The training set includes $1,034$ subjects. For each subject, there are two sequences, one carrying objects while the other one not. The gallery and probe sets contain $1,036$ subjects which are disjoint from the subjects in the training set. Subjects in the gallery set carry objects while subjects in the probe set carry nothing. This dataset provides GEIs of all sequences, so we directly use these GEIs to carry out our experiments.

The CASIA-B gait dataset. The CASIA-B gait database [Yu et al., 2006] is composed of 124 subjects, with 110 sequences per subject. It contains eleven views and there are ten sequences per view. Among the ten sequences, six are taken under normal walking conditions (NM), two are taken when subjects are with coats (CL), and two are taken when subjects are with bags (BG).

A.5.2 Effectiveness of JUCNet and Quintuplet Loss

To demonstrate the effectiveness of the JUCNet and quintuplet loss, we develop three third-party baseline networks, *MT*, *Deeper MT*, and *CNet*. MT and Deeper MT are representative methods from [Wu et al., 2017] for learning sole cross-gait representation to predict the relationship between a pair of gaits. CNet is a simplified version of JUCNet without the component of learning unique-gait representation. We also conduct ablation analysis by comparing our full method *JUCNet (Metric & Quintuplet)* with two versions of self baseline networks, *JUCNet* and *JUCNet (Metric)*. All these networks are illustrated in the following.

- **MT** is a CNN consisting of two convolutional layers, two pooling layers, and one fully-connected layer. The input of this model is a pair of GEIs. The MT extracts features by the convolutional layers and concatenates features as the cross-gait representation by the fully-connected layer. Finally, the cross-gait representation will be input to a binary classifier to predict their relationship.
- **Deeper MT** is a deeper version of MT. It contains two additional fully-connected layers. Two convolutional layers and two fully-connected layers are utilized to learn two feature sets from the input GEIs. Then they will be concatenated as a whole to learn cross-gait representation by the third fully-connected layer. MT and Deeper MT have achieved state-of-the-art performance on some datasets [Wu et al., 2017].
- **CNet** is a network which excludes the unique-gait part from our JUCNet. It contains eight convolutional layers and two fully-connected layers. As shown in the yellow and blue parts of Fig. A.3, CNet shares a similar structure with both MT and Deeper MT. The major difference from them is that when the feature maps are concatenated as a whole, more layers are built in order to learn powerful cross-gait representation.

- **JUCNet** is our proposed network that jointly learns unique-gait and cross-gait representation. This JUCNet model is updated based on the loss functions in Eq. (A.2), (A.4), and (A.6).
- **JUCNet (Metric)** is our JUCNet model plus metric learning. It learns the metric $C(\cdot,\cdot)$ to represent the distance between a pair of gaits. The loss functions employed to train this model are Eqs. (A.4), (A.6), and (A.8) .
- **JUCNet (M & Quintuplet)** is our JUCNet model plus both metric learning and our proposed quintuplet loss. Fig. A.4 and the section of Quintuplet Loss present the details of this model and the quintuplet loss. The model is trained based on the loss functions in Eqs. (A.4), (A.6) and (A.10).

Parameter analysis. Different loss terms are weighted by hyperparameters in our loss functions. In order to set them appropriately, we utilize a part of the training set as a validation set and investigate the effect of hyperparameters by varying η_i, η_u, and η_c in Eqs. A.5, (A.6), and A.10 from 0 to 1. When hyperparameters are equal to 0, only the first term in the above equations works. With the increase of hyperparameters, the binding term plays a more and more important role in our model. When varying one hyperparameter, the other two hyperparameters are set to be fixed. According to the results shown in Fig. A.5, in general the accuracies become higher with the increase of hyperparameters until becoming lower with increased values. The best performance is achieved when $\eta_i = 0.6$, $\eta_u = 0.4$, and $\eta_c = 0.6$, which are set in our following experiments.

Results in terms of rank-n accuracy. We report the results of rank-1, rank-3, rank-5, and rank-10 accuracies of the aforementioned six models on both OU-LP-Bag β and OUTD-B, shown in Table A.1 and Table A.2, respectively.

In both tables, we observe that: 1) JUCNet achieves higher accuracies than MT, Deeper MT and CNet. This verifies the effectiveness of the proposed JUCNet by jointly learning unique-gait and cross-gait representations. As shown in Fig. A.3 , the unique-gait representation and cross-gait representation complement each other to update the shared-weight layers, leading to more powerful high-level features. 2) The improvement from JUCNet to JUCNet (Metric) reveals the advantage of metric $C(\cdot,\cdot)$, which learns to measure discrepancy between gaits driven by data automatically, in contrast to pre-defined metric like the Euclidean distance. 3) The JUCNet (Metric & Quintuplet) achieves better performance than both JUCNet and JUCNet (Metric), which additionally suggests the effectiveness of our proposed quintuplet loss. 4) The improvement from other models to JUCNet (Metric & Quintuplet) in terms of rand-1 accuracy is more evident than that in terms of rank-3, rank-5, and rank-10 accuracies. We that suspect the following reason justifies. Given a probe gait, other models may determine more than one gallery gait as from an identical subject, because they are trained with only classification loss. To the contrast, the quintuplet loss guides our model to not only obtain correct *classification* results, but also learn more powerful features ensuring enlarged inter-class differences and decreased intra-class variations, leading to correct *ranking orders*.

Table A.1: **The rank-1, rank-3, rank-5, and rank-10 accuracies [%] of different models on the OU-LP-Bag β dataset.** The best results are shown in bold, which also applies to the following tables.

Models	rank-1	rank-3	rank-5	rank-10
MT [Wu et al., 2017]	59.9	75.2	80.1	86.8
Deeper MT [Wu et al., 2017]	68.1	81.8	86.0	90.8
CNet	71.0	86.9	91.5	95.2
JUCNet	74.3	87.4	90.8	95.3
JUCNet (Metric)	74.8	88.9	92.3	95.6
JUCNet (M & Quintuplet)	**78.2**	**89.6**	**92.8**	**95.8**

Table A.2: **The rank-1, rank-3, rank-5, and rank-10 accuracies [%] of different models on the OUTD-B dataset.**

Models	rank-1	rank-3	rank-5	rank-10
MT [Wu et al., 2017]	70.7	87.7	91.9	97.9
Deeper MT [Wu et al., 2017]	72.4	90.3	95.8	98.4
CNet	71.1	88.2	94.3	97.9
JUCNet	73.2	88.9	94.2	98.0
JUCNet (Metric)	73.8	88.4	93.9	97.9
JUCNet (M & Quintuplet)	**76.4**	**91.4**	**95.2**	**98.7**

A.5.3 Comparison with State-of-the-art Methods

We have verified that the proposed JUCNet with the quintuplet loss outperforms the conventional CNN models which are solely based on cross-gait representation. In this section, we compare our method with other state-of-the-art methods, including part-based FDF [Hossain et al., 2010], part-based Entropy of the Discrete Fourier Transform (EnDFT) [Rokanujjaman et al., 2015], GEnI [Bashir et al., 2009], Masked-GEI [Bashir et al., 2010], Gabor GEI [Tao et al., 2007] and spatial metric learning methods using GEI like ranking SVM [Martín-Félez and Xiang, 2014], and a Joint Intensity and Spatial Metric Learning method (JISML) [Makihara et al., 2017].

The results in Table A.3 show that JUCNet plus metric learning outperforms the previous best method on both OU-LP-Bag β and OUTD-B databases, which reveals its effectiveness. JUCNet based on metric learning and quintuplet loss achieves better performance than JUCNet with metric, justifying the advantage of our proposed quintuplet loss again. The JISML method introduces joint learning of intensity and spatial metric in order to mitigate the large intra-class differences and leverage the subtle inter-class differences, while in our method the quintuplet loss accomplishes this task. A method proposed by Guan *et al.* [Guan et al., 2015] achieves better rank-1 accuracy on the OUTD-B dataset than ours. While their results are achieved under a different training/testing protocol. Meanwhile, their method requires a regular within-class matrix for the gallery set, so it cannot be applied on datasets including only a single probe and a single gallery per subject like the OU-LP-Bag β dataset.

In addition, Table A.3 reveals that the improvement over existing methods achieved by our method on the OU-LP-Bag β dataset is greater than that on the OUTD-B dataset with regard to the rank-1 accuracy. This is because that there are more sub-

Table A.3: **The rank-1 accuracies [%] of different methods on testing sets of OU-LP-Bag β and OUTD-B."-" indicates not provided.**

Methods	OU-LP-Bag β	OUTD-B
FDF (Part-based)	-	66.3
EnDFT (Part-based)	-	72.8
GEnl	29.5	59.0
Masked GEI	-	28.0
Gabor GEI	46.4	62.3
GEI w/o ML	24.6	55.3
GEI w/ Ranking SVM	28.3	58.4
JISML	57.4	74.5
JUCNet	74.1	73.2
JUCNet (Metric)	74.7	74.9
JUCNet (M & Quintuplet)	**79.3**	**77.6**

Table A.4: **The rank-1 accuracies [%] of different methods under the cross-view condition on the BG subset of the CASIA-B gait dataset.**

Probe	Gallery	RLTDA	MT	JUCNet
54°	36°	80.8	**92.7**	91.8
54°	72°	71.5	90.4	**93.9**
90°	72°	75.3	93.3	**95.9**
90°	108°	76.5	88.9	**95.9**
126°	108°	66.5	93.3	**93.9**
126°	144°	72.3	86.0	**87.8**
Average		73.8	90.8	**93.2**

jects (1,034) in the training set of the OU-LP-Bag β dataset, while there are only 20 subjects in the training set of OUTD-B. Larger scale of the training set benefits our model in gaining greater learning capacity. On the other hand, though there are more samples in the OU-LP-Bag β dataset, the final results regarding the rank-1 accuracy on both datasets are at the same level. We believe that, it is more difficult for models to recognize the correct subjects from the OU-LP-Bag β dataset than the OUTD-B dataset because the OUTD-B dataset includes only 48 subjects in the testing set, while there are 1,036 subjects in the testing set of the OU-LP-Bag β dataset.

It may be observed that the results of OU-LP-Bag β and OUTD-B in Table A.3 are better than those in Tables A.1 and A.2. As mentioned above, we utilize a part of the training set in these two datasets as a validation set to tune weight parameters. Thus results in Tables A.1 and A.2 are reported by models trained without the validation set. While comparing with other methods in Table A.3, we put the validation set back to the training set to re-train the model for a fair comparison, because the validation set belongs to the training set in other methods.

A.5.4 Cross-view Study

The issue of cross view is crucial for gait recognition, so we evaluate our method under the condition of cross view on the BG subset of the CASIA-B gait dataset. We evaluate our method on the more challenging BG set (the accuracy is between 86.0% and 93.3%), rather than the NM set (the accuracy is between 97.0% and 99.5%). As

shown in Table A.4, subjects in the probe and gallery sets are of different views. The comparison with Wu *et al.* [Wu et al., 2017] and Hu *et al.* [Hu, 2013] indicates that our method achieves satisfactory performance under the cross-view protocol.

A.6 Conclusion

We have proposed a JUCNet model to jointly learn unique-gait and cross-gait representations for gait recognition. The two kinds of representations complement each other to boost the performance of gait recognition. Moreover, a quintuplet loss for gait recognition was proposed to increase the inter-class differences by pushing the cross-gait representation learned from different classes apart and reduce the intra-class variations by pulling the representations learned from an identical class together. The experimental results on public datasets suggest that the JUCNet model outperforms existing CNN models based on sole cross-gait representation, demonstrating the effectiveness of the JUCNet model. JUCNet with the quintuplet loss further improves the performance, validating its superiority over the state-of-the-art methods.

Bibliography

Abuolaim, A. and Brown, M. S., 2020a. Defocus deblurring using dual-pixel data. In *European Conference on Computer Vision*. (cited on page 1)

Abuolaim, A. and Brown, M. S., 2020b. Defocus deblurring using dual-pixel data. In *European Conference on Computer Vision*, 111–126. Springer. (cited on pages 61, 62, and 64)

Aljadaany, R.; Pal, D. K.; and Savvides, M., 2019. Douglas-rachford networks: Learning both the image prior and data fidelity terms for blind image deconvolution. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 70)

Alletto, S.; Carlin, C.; Rigazio, L.; Ishii, Y.; and Tsukizawa, S., 2019. Adherent raindrop removal with self-supervised attention maps and spatio-temporal generative adversarial networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0. (cited on page 14)

Alotaibi, M. and Mahmood, A., 2017. Improved gait recognition based on specialized deep convolutional neural network. *CVIU*, (2017). (cited on page 122)

Ariyanto, G. and Nixon, M. S., 2011. Model-based 3d gait biometrics. In *IJCB*. (cited on page 122)

Bae, S. and Durand, F., 2007. Defocus magnification. *Computer Graphics Forum*, 26, 3 (2007), 571–579. (cited on page 8)

Bahat, Y.; Efrat, N.; and Irani, M., 2017. Non-uniform blind deblurring by reblurring. In *CVPR*. (cited on pages 1 and 45)

Barnum, P. C.; Narasimhan, S.; and Kanade, T., 2010. Analysis of rain and snow in frequency space. *IJCV*, (2010). (cited on pages 14 and 15)

Bashir, K.; Xiang, T.; and Gong, S., 2009. Gait recognition using gait entropy image. *IET*, (2009). (cited on page 131)

Bashir, K.; Xiang, T.; and Gong, S., 2010. Gait recognition without subject cooperation. *PRL*, (2010). (cited on page 131)

Beard, K. V. and Chuang, C., 1987. A new model for the equilibrium shape of raindrops. *Journal of Atmospheric Sciences*, 44, 11 (1987), 1509–1524. (cited on page 8)

Bodor, R.; Drenner, A.; Fehr, D.; Masoud, O.; and Papanikolopoulos, N., 2009. View-independent human motion classification using image-based reconstruction. *Image and Vision Computing*, (2009). (cited on page 122)

Bouchrika, I.; Goffredo, M.; Carter, J.; and Nixon, M., 2011. On using gait in forensic biometrics. *JFS*, (2011). (cited on page 119)

Brewer, N. and Liu, N., 2008. Using the shape characteristics of rain to identify and remove rain from video. In *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*. (cited on page 15)

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; and de la Blanca, N. P., 2017. Automatic learning of gait signatures for people identification. In *IWANN*. (cited on page 122)

Chakrabarti, A., 2016. A neural approach to blind motion deblurring. In *ECCV*. (cited on pages 11, 23, and 25)

Chang, J.-R. and Chen, Y.-S., 2018. Pyramid stereo matching network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 101)

Chang, Y.; Yan, L.; and Zhong, S., 2017. Transformed low-rank model for line pattern noise removal. In *ICCV*. (cited on page 14)

Chen, C.; Seff, A.; Kornhauser, A.; and Xiao, J., 2015. Deepdriving: Learning affordance for direct perception in autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on page 101)

Chen, D.; Yuan, L.; Liao, J.; Yu, N.; and Hua, G., 2018a. Stereoscopic neural style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 16)

Chen, D.-Y.; Chen, C.-C.; and Kang, L.-W., 2014. Visual depth guided color image rain streaks removal using sparse coding. *IEEE transactions on circuits and systems for video technology*, (2014). (cited on page 5)

Chen, F. and Ma, J., 2009. An empirical identification method of gaussian blur parameter for image deblurring. *IEEE Transactions on Signal Processing*, 57, 7 (2009), 2467–2478. (cited on page 8)

Chen, J. and Chau, L.-P., 2013. A rain pixel recovery algorithm for videos with highly dynamic scenes. *IEEE Transactions on Image Processing (TIP)*, (2013). (cited on page 15)

Chen, J.; Tan, C.-H.; Hou, J.; Chau, L.-P.; and Li, H., 2018b. Robust video content alignment and compensation for rain removal in a cnn framework. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 15, 89, 96, 97, and 98)

Chen, S.-J. and Shen, H.-L., 2015. Multispectral image out-of-focus deblurring using interchannel correlation. *IEEE Transactions on Image Processing*, 24, 11 (2015), 4433–4445. (cited on page 1)

Chen, W.; Chen, X.; Zhang, J.; and Huang, K., 2017. Beyond triplet loss: a deep quadruplet network for person re-identification. In *CVPR*. (cited on page 124)

Chen, Y.-L. and Hsu, C.-T., 2013. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *ICCV*. (cited on pages 5 and 73)

Cho, S. and Lee, S., 2009. Fast motion deblurring. *TOG*, (2009). (cited on pages 1 and 19)

Cho, T. S.; Paris, S.; Horn, B. K.; and Freeman, W. T., 2011. Blur kernel estimation using the radon transform. In *CVPR*. (cited on page 46)

Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; and Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 108)

Damera-Venkata, N.; Kite, T. D.; Geisler, W. S.; Evans, B. L.; and Bovik, A. C., 2000. Image quality assessment based on a degradation model. *IEEE Transactions on Image Processing*, 9, 4 (2000), 636–650. (cited on page 16)

Delbracio, M. and Sapiro, G., 2015. Burst deblurring: Removing camera shake through fourier burst accumulation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 39 and 41)

Eigen, D.; Krishnan, D.; and Fergus, R., 2013. Restoring an image taken through a window covered with dirt or rain. In *ICCV*. (cited on pages 14, 15, and 74)

Eslami, S. A.; Heess, N.; Weber, T.; Tassa, Y.; Szepesvari, D.; Hinton, G. E.; et al., 2016. Attend, infer, repeat: Fast scene understanding with generative models. In *Advances in Neural Information Processing Systems (NeurIPS)*. (cited on pages 16 and 101)

Feng, Y.; Li, Y.; and Luo, J., 2016. Learning effective gait features using lstm. In *ICPR*. (cited on page 122)

Feng, Y.; Ma, L.; Liu, W.; Zhang, T.; and Luo, J., 2018. Video re-localization. In *ECCV*. (cited on page 122)

Fergus, R.; Singh, B.; Hertzmann, A.; Roweis, S. T.; and Freeman, W. T., 2006. Removing camera shake from a single photograph. In *ACM SIGGRAPH*. (cited on pages 1 and 7)

Fu, X.; Huang, J.; Ding, X.; Liao, Y.; and Paisley, J., 2017a. Clearing the skies: A deep network architecture for single-image rain removal. *TIP*, (2017). (cited on pages 14, 73, 82, and 83)

Fu, X.; Huang, J.; Zeng, D.; Huang, Y.; Ding, X.; and Paisley, J., 2017b. Removing rain from single images via a deep detail network. In *CVPR*. (cited on pages 5, 14, 73, 82, 83, 85, and 98)

Gao, H.; Tao, X.; Shen, X.; and Jia, J., 2019. Dynamic scene deblurring with parameter selective sharing and nested skip connections. In *CVPR*. (cited on pages 7, 11, 12, 17, 20, 29, and 70)

Garg, K. and Nayar, S. K., 2004. Detection and removal of rain from videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 15)

Garg, K. and Nayar, S. K., 2005. When does a camera see rain? In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*. (cited on page 8)

Garg, K. and Nayar, S. K., 2006. Photorealistic rendering of rain streaks. In *TOG*. (cited on pages 15, 83, and 97)

Geiger, A.; Lenz, P.; Stiller, C.; and Urtasun, R., 2013. Vision meets robotics: The kitti dataset. *The International Journal of Robotics Research*, 32, 11 (2013), 1231–1237. (cited on pages 38, 58, and 108)

Girshick, R., 2015. Fast r-cnn. In *ICCV*. (cited on page 73)

Godard, C.; Mac Aodha, O.; and Brostow, G. J., 2017. Unsupervised monocular depth estimation with left-right consistency. In *CVPR*. (cited on pages 16, 101, and 103)

Goffredo, M.; Bouchrika, I.; Carter, J. N.; and Nixon, M. S., 2010. Self-calibrating view-invariant gait biometrics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, (2010). (cited on page 122)

Goldstein, A. and Fattal, R., 2012. Blur-kernel estimation from spectral irregularities. In *ECCV*. (cited on page 19)

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y., 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on pages 34 and 35)

Guan, Y.; Li, C.-T.; and Roli, F., 2015. On reducing the effect of covariate factors in gait recognition: a classifier ensemble method. *TPAMI*, (2015). (cited on page 131)

Gupta, A.; Joshi, N.; Zitnick, C. L.; Cohen, M.; and Curless, B., 2010. Single image deblurring using motion density functions. In *European Conference on Computer Vision (ECCV)*. (cited on page 45)

Han, J. and Bhanu, B., 2005. Individual recognition using gait energy image. *TPAMI*, (2005). (cited on pages 4 and 73)

Hao, Z.; You, S.; Li, Y.; Li, K.; and Lu, F., 2019. Learning from synthetic photorealistic raindrop for single image raindrop removal. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 0–0. (cited on page 14)

Harmeling, S.; Michael, H.; and Schölkopf, B., 2010. Space-variant single-image blind deconvolution for removing camera shake. In *NIPS*. (cited on page 45)

He, K.; Gkioxari, G.; Dollár, P.; and Girshick, R., 2017. Mask r-cnn. In *ICCV*. (cited on pages 2 and 73)

He, K.; Zhang, X.; Ren, S.; and Sun, J., 2016. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 2, 22, 23, 34, 82, 109, and 122)

Hirsch, M.; Schuler, C. J.; Harmeling, S.; and Schölkopf, B., 2011. Fast removal of non-uniform camera shake. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 20 and 46)

Hore, A. and Ziou, D., 2010. Image quality metrics: Psnr vs. ssim. In *IEEE International Conference on Pattern Recognition*. (cited on page 16)

Hossain, M. A.; Makihara, Y.; Wang, J.; and Yagi, Y., 2010. Clothing-invariant gait identification using part-based clothing categorization and adaptive weight control. *PR*, (2010). (cited on page 131)

Hossfeld, T.; Heegaard, P. E.; Varela, M.; and Möller, S., 2016. Qoe beyond the mos: an in-depth look at qoe via better metrics and their relation to mos. *Quality and User Experience*, 1, 1 (2016), 2. (cited on page 16)

Hu, H., 2013. Enhanced gabor feature based classification using a regularized locally tensor discriminant model for multiview gait recognition. *TCSVT*, (2013). (cited on page 133)

Hu, X.; Fu, C.-W.; Zhu, L.; and Heng, P.-A., 2019. Depth-attentional features for single-image rain removal. In *CVPR*. (cited on pages 79, 103, 104, 108, 111, 112, and 113)

Hu, X.; Zhu, L.; Wang, T.; Fu, C.-W.; and Heng, P.-A., 2021. Single-image real-time rain removal based on depth-guided non-local features. *IEEE Transactions on Image Processing*, 30 (2021), 1759–1770. (cited on page 14)

Huang, D.-A.; Kang, L.-W.; Wang, Y.-C. F.; and Lin, C.-W., 2013. Self-learning based image decomposition with applications to single image denoising. *TMM*, (2013). (cited on page 14)

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q., 2017. Densely connected convolutional networks. In *CVPR*. (cited on page 23)

Hummel, R. A.; Kimia, B.; and Zucker, S. W., 1987. Deblurring gaussian blur. *Computer Vision, Graphics, and Image Processing*, 38, 1 (1987), 66–80. (cited on page 8)

Hyun Kim, T.; Ahn, B.; and Mu Lee, K., 2013. Dynamic scene deblurring. In *ICCV*. (cited on page 29)

Hyun Kim, T. and Mu Lee, K., 2015. Generalized video deblurring for dynamic scenes. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 39, 41, 43, 56, and 57)

Hyun Kim, T.; Mu Lee, K.; Scholkopf, B.; and Hirsch, M., 2017. Online video deblurring via dynamic temporal blending network. In *IEEE International Conference on Computer Vision*. (cited on page 13)

Isola, P.; Zhu, J.-Y.; Zhou, T.; and Efros, A. A., 2017. Image-to-image translation with conditional adversarial networks. In *CVPR*. (cited on pages 2 and 13)

Iwama, H.; Okumura, M.; Makihara, Y.; and Yagi, Y., 2012. The ou-isir gait database comprising the large population dataset and performance evaluation of gait recognition. *TIFS*, (2012). (cited on page 128)

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al., 2015. Spatial transformer networks. In *NIPS*. (cited on page 50)

Jeon, D. S.; Baek, S.-H.; Choi, I.; and Kim, M. H., 2018. Enhancing the spatial resolution of stereo images using a parallax prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 16)

Ji, S.; Xu, W.; Yang, M.; and Yu, K., 2013. 3d convolutional neural networks for human action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35, 1 (2013), 221–231. (cited on page 33)

Jiang, P.; Ling, H.; Yu, J.; and Peng, J., 2013. Salient region detection by ufo: Uniqueness, focusness and objectness. In *IEEE International Conference on Computer Vision*. (cited on page 8)

Jiang, T.-X.; Huang, T.-Z.; Zhao, X.-L.; Deng, L.-J.; and Wang, Y., 2017. A novel tensor-based video rain streaks removal approach via utilizing discriminatively intrinsic priors. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 15)

Jiang, T.-X.; Huang, T.-Z.; Zhao, X.-L.; Deng, L.-J.; and Wang, Y., 2018. Fastderain: A novel video rain streak removal method using directional gradient priors. *IEEE Transactions on Image Processing*, (2018). (cited on pages 89, 96, 97, and 98)

Jiang, Z.; Zhang, Y.; Zou, D.; Ren, J.; Lv, J.; and Liu, Y., 2020. Learning event-based motion deblurring. *arXiv preprint arXiv:2004.05794*, (2020). (cited on page 62)

Jin, H.; Favaro, P.; and Cipolla, R., 2005. Visual tracking in the presence of motion blur. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 18–25. (cited on page 31)

Jin, M.; Meishvili, G.; and Favaro, P., 2018. Learning to extract a video sequence from a single motion-blurred image. In *CVPR*. (cited on pages 13, 46, 47, 49, 53, 54, and 56)

Johnson, J.; Alahi, A.; and Fei-Fei, L., 2016. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*. (cited on pages 24 and 49)

Jolicoeur-Martineau, A., 2019. The relativistic discriminator: a key element missing from standard gan. *ICLR*, (2019). (cited on page 24)

Kang, L.-W.; Lin, C.-W.; and Fu, Y.-H., 2011. Automatic single-image-based rain streaks removal via image decomposition. *TIP*, (2011). (cited on pages 4, 14, and 73)

Kang, S. B., 2007. Automatic removal of chromatic aberration from a single image. In *CVPR*. (cited on pages 1 and 31)

Kim, J.-H.; Lee, C.; Sim, J.-Y.; and Kim, C.-S., 2013. Single-image deraining using an adaptive nonlocal means filter. In *2013 IEEE International Conference on Image Processing*. (cited on page 5)

Kim, J.-H.; Sim, J.-Y.; and Kim, C.-S., 2014. Stereo video deraining and desnowing based on spatiotemporal frame warping. In *The IEEE International Conference on Image Processing (ICIP)*. (cited on page 16)

Kim, J.-H.; Sim, J.-Y.; and Kim, C.-S., 2015. Video deraining and desnowing using temporal correlation and low-rank matrix completion. *TIP*, (2015). (cited on pages 15 and 98)

Kim, T. H.; Lee, K. M.; Schölkopf, B.; and Hirsch, M., 2017. Online video deblurring via dynamic temporal blending network. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 40 and 41)

Köhler, R.; Hirsch, M.; Mohler, B.; Schölkopf, B.; and Harmeling, S., 2012. Recording and playback of camera shake: Benchmarking blind deconvolution with a real-world database. In *European Conference on Computer Vision*. (cited on pages 61 and 62)

Kupyn, O.; Budzan, V.; Mykhailych, M.; Mishkin, D.; and Matas, J., 2018. Deblurgan: Blind motion deblurring using conditional adversarial networks. In *CVPR*. (cited on pages 7, 12, 13, 17, 27, 66, and 67)

Kupyn, O.; Martyniuk, T.; Wu, J.; and Wang, Z., 2019. Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better. In *ICCV*. (cited on pages 7, 12, 13, 17, 66, 67, and 70)

Kurihata, H.; Takahashi, T.; Ide, I.; Mekada, Y.; Murase, H.; Tamatsu, Y.; and Miyahara, T., 2005. Rainy weather recognition from in-vehicle camera images for driver assistance. In *IV*. (cited on page 14)

Kusakunniran, W., 2014. Attribute-based learning for gait recognition using spatio-temporal interest points. *IVC*, (2014). (cited on page 122)

Kusakunniran, W.; Wu, Q.; Li, H.; and Zhang, J., 2009. Multiple views gait recognition using view transformation model based on optimized gait energy image. In *ICCV Workshops*. (cited on pages 120 and 122)

Kusakunniran, W.; Wu, Q.; Zhang, J.; Li, H.; and Wang, L., 2014. Recognizing gaits across views through correlated motion co-clustering. *TIP*, (2014). (cited on page 122)

Kusakunniran, W.; Wu, Q.; Zhang, J.; Ma, Y.; and Li, H., 2013. A new view-invariant feature for cross-view gait recognition. *TIFS*, (2013). (cited on pages 120 and 122)

Lai, W.-S.; Huang, J.-B.; Hu, Z.; Ahuja, N.; and Yang, M.-H., 2016. A comparative study for single image blind deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 62)

Lam, T. H.; Cheung, K. H.; and Liu, J. N., 2011. Gait flow image: A silhouette-based gait representation for human identification. *PR*, (2011). (cited on page 122)

Larsen, P. K.; Simonsen, E. B.; and Lynnerup, N., 2008. Gait analysis in forensic medicine. *JFS*, (2008). (cited on page 119)

Lee, H. S.; Kwon, J.; and Lee, K. M., 2011. Simultaneous localization, mapping and deblurring. In *The IEEE International Conference on Computer Vision (ICCV)*. (cited on page 31)

Lee, J.; Lee, S.; Cho, S.; and Lee, S., 2019. Deep defocus map estimation using domain adaptation. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 8)

Levin, A.; Weiss, Y.; Durand, F.; and Freeman, W. T., 2009. Understanding and evaluating blind deconvolution algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 62)

Li, B.; Lin, C.-W.; Shi, B.; Huang, T.; Gao, W.; and Jay Kuo, C.-C., 2018a. Depth-aware stereo video retargeting. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 16)

Li, B.; Ouyang, W.; Sheng, L.; Zeng, X.; and Wang, X., 2019a. Gs3d: An efficient 3d object detection framework for autonomous driving. In *CVPR*. (cited on pages 4 and 73)

Li, M.; Xie, Q.; Zhao, Q.; Wei, W.; Gu, S.; Tao, J.; and Meng, D., 2018b. Video rain streak removal by multiscale convolutional sparse coding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 15)

Li, P.; Prieto, L.; Mery, D.; and Flynn, P., 2018c. Face recognition in low quality images: a survey. *arXiv preprint arXiv:1805.11519*, (2018). (cited on page 17)

Li, R.; Cheong, L.-F.; and Tan, R. T., 2019b. Heavy rain image restoration: Integrating physics model and conditional adversarial learning. In *CVPR*. (cited on pages 79 and 111)

Li, S.; Araujo, I. B.; Ren, W.; Wang, Z.; Tokuda, E. K.; Junior, R. H.; Cesar-Junior, R.; Zhang, J.; Guo, X.; and Cao, X., 2019c. Single image deraining: A comprehensive benchmark analysis. In *CVPR*. (cited on pages 15, 78, 79, 85, 86, 101, and 104)

Li, S.; Ren, W.; Wang, F.; Araujo, I. B.; Tokuda, E. K.; Junior, R. H.; Cesar-Jr, R. M.; Wang, Z.; and Cao, X., 2021. A comprehensive benchmark analysis of single image deraining: Current challenges and future perspectives. *International Journal of Computer Vision*, 129, 4 (2021), 1301–1322. (cited on page 14)

Li, S. Z. and Jain, A., 2015. *Encyclopedia of biometrics*. Springer Publishing Company, Incorporated. (cited on page 120)

Li, X.; Wu, J.; Lin, Z.; Liu, H.; and Zha, H., 2018d. Recurrent squeeze-and-excitation context aggregation net for single image deraining. In *ECCV*. (cited on pages 14, 73, 82, 83, 111, 112, and 113)

Li, Y.; Tan, R. T.; Guo, X.; Lu, J.; and Brown, M. S., 2016. Rain streak removal using layer priors. In *CVPR*. (cited on pages 14, 82, 83, 111, and 112)

Liu, F.; Shen, C.; and Lin, G., 2015. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 101)

Liu, J.; Yang, W.; Yang, S.; and Guo, Z., 2018a. D3r-net: Dynamic routing residue recurrent network for video rain removal. *IEEE Transactions on Image Processing (TIP)*, (2018). (cited on pages 15 and 96)

Liu, J.; Yang, W.; Yang, S.; and Guo, Z., 2018b. Erase or fill? deep joint recurrent rain removal and reconstruction in videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 15, 89, and 98)

Liu, L.; Liu, B.; Huang, H.; and Bovik, A. C., 2014. No-reference image quality assessment based on spatial and spectral entropies. *Signal Processing: Image Communication*, 29, 8 (2014), 856–863. (cited on page 17)

Liu, P.; Xu, J.; Liu, J.; and Tang, X., 2009. Pixel based temporal analysis using chromatic property for removing rain from videos. *Computer and Information Science*, (2009). (cited on page 15)

Long, J.; Shelhamer, E.; and Darrell, T., 2015. Fully convolutional networks for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 34)

Lu, J. and Tan, Y.-P., 2010. Uncorrelated discriminant simplex analysis for view-invariant gait signal computing. *PRL*, (2010). (cited on page 119)

Luo, W.; Schwing, A. G.; and Urtasun, R., 2016. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 16 and 101)

Luo, W.; Sun, P.; Zhong, F.; Liu, W.; Zhang, T.; and Wang, Y., 2019. End-to-end active object tracking and its real-world deployment via reinforcement learning. *TPAMI*, (2019). (cited on page 122)

Luo, Y.; Xu, Y.; and Ji, H., 2015. Removing rain from a single image via discriminative sparse coding. In *ICCV*. (cited on pages 5, 9, 14, 111, and 112)

Makihara, Y.; Mannami, H.; Tsuji, A.; Hossain, M. A.; Sugiura, K.; Mori, A.; and Yagi, Y., 2012. The ou-isir gait database comprising the treadmill dataset. *CVA*, (2012). (cited on page 128)

Makihara, Y.; Sagawa, R.; Mukaigawa, Y.; Echigo, T.; and Yagi, Y., 2006. Gait recognition using a view transformation model in the frequency domain. *ECCV*, (2006). (cited on page 122)

Makihara, Y.; Suzuki, A.; Muramatsu, D.; Li, X.; and Yagi, Y., 2017. Joint intensity and spatial metric learning for robust gait recognition. In *CVPR*. (cited on pages 129 and 131)

Man, J. and Bhanu, B., 2006. Individual recognition using gait energy image. *TPAMI*, (2006). (cited on pages 122 and 128)

Mansur, A.; Makihara, Y.; Aqmar, R.; and Yagi, Y., 2014. Gait recognition under speed transition. In *CVPR*. (cited on page 119)

Mao, X.; Shen, C.; and Yang, Y.-B., 2016. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. In *Advances in Neural Information Processing Systems*. (cited on page 13)

Martín-Félez, R. and Xiang, T., 2014. Uncooperative gait recognition by learning to rank. *PR*, (2014). (cited on page 131)

Masia, B.; Corrales, A.; Presa, L.; and Gutierrez, D., 2011. Coded apertures for defocus deblurring. In *Symposium Iberoamericano de Computacion Grafica*. (cited on page 8)

Mathieu, M.; Couprie, C.; and LeCun, Y., 2016. Deep multi-scale video prediction beyond mean square error. In *ICLR*. (cited on page 13)

Michaeli, T. and Irani, M., 2014. Blind deblurring using internal patch recurrence. In *ECCV*. (cited on page 45)

Mitsa, T. and Varkur, K. L., 1993. Evaluation of contrast sensitivity functions for the formulation of quality measures incorporated in halftoning algorithms. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*. (cited on page 16)

Mittal, A.; Moorthy, A. K.; and Bovik, A. C., 2012a. No-reference image quality assessment in the spatial domain. *IEEE Transactions on Image Processing*, 21, 12 (2012), 4695–4708. (cited on page 17)

Mittal, A.; Soundararajan, R.; and Bovik, A. C., 2012b. Making a "completely blind" image quality analyzer. *IEEE Signal Processing Letters*, 20, 3 (2012), 209–212. (cited on page 17)

Mobahi, H.; Collobert, R.; and Weston, J., 2009. Deep learning from temporal coherence in video. In *ICML*. (cited on page 124)

Moorthy, A. K. and Bovik, A. C., 2010. A two-step framework for constructing blind image quality indices. *IEEE Signal Processing Letters*, 17, 5 (2010), 513–516. (cited on page 17)

Moorthy, A. K. and Bovik, A. C., 2011. Blind image quality assessment: From natural scene statistics to perceptual quality. *IEEE Transactions on Image Processing*, 20, 12 (2011), 3350–3364. (cited on page 17)

Mori, A.; Makihara, Y.; and Yagi, Y., 2010. Gait recognition using period-based phase synchronization for low frame-rate videos. In *2010 20th International Conference on Pattern Recognition*, 2194–2197. IEEE. (cited on page 119)

Murase, H. and Sakai, R., 1996. Moving object recognition in eigenspace representation: gait analysis and lip reading. *PRL*, (1996). (cited on page 122)

Nah, S.; Baik, S.; Hong, S.; Moon, G.; Son, S.; Timofte, R.; and Lee, K. M., 2019a. Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*. (cited on pages 60, 62, and 63)

Nah, S.; Hyun Kim, T.; and Mu Lee, K., 2017a. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*. (cited on pages 7, 12, 17, 19, 20, 23, 24, 25, 27, 28, 29, 30, 53, 56, 57, and 62)

Nah, S.; Kim, T. H.; and Lee, K. M., 2017b. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*. (cited on pages 62, 66, 67, and 70)

Nah, S.; Son, S.; and Lee, K. M., 2019b. Recurrent neural networks with intra-frame iterations for video deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 13)

Nimisha, T. M.; Kumar Singh, A.; and Rajagopalan, A. N., 2017. Blur-invariant deep learning for blind-deblurring. In *IEEE International Conference on Computer Vision*. (cited on page 11)

Pan, J.; Hu, Z.; Su, Z.; and Yang, M.-H., 2014. Deblurring text images via l0-regularized intensity and gradient prior. In *CVPR*. (cited on page 19)

Pan, J.; Sun, D.; Pfister, H.; and Yang, M.-H., 2016. Blind image deblurring using dark channel prior. In *CVPR*. (cited on page 45)

Pan, L.; Dai, Y.; Liu, M.; and Porikli, F., 2017. Simultaneous stereo video deblurring and scene flow estimation. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017). (cited on pages 38, 39, and 43)

Pan, L.; Scheerlinck, C.; Yu, X.; Hartley, R.; Liu, M.; and Dai, Y., 2019. Bringing a blurry frame alive at high frame-rate with an event camera. In *CVPR*. (cited on pages 13, 53, and 56)

Pang, J.; Sun, W.; Ren, J. S.; Yang, C.; and Yan, Q., 2017. Cascade residual learning: A two-stage convolutional neural network for stereo matching. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on page 101)

Park, P. D.; Kang, D. U.; Kim, J.; and Chun, S. Y., 2020. Multi-temporal recurrent neural networks for progressive non-uniform single image deblurring with incremental temporal training. In *European Conference on Computer Vision*. (cited on page 70)

Purohit, K. and Rajagopalan, A., 2019. Region-adaptive dense network for efficient motion deblurring. *arXiv preprint arXiv:1903.11394*, (2019). (cited on page 12)

Purohit, K.; Shah, A.; and Rajagopalan, A., 2019. Bringing alive blurred moments. In *CVPR*. (cited on pages 13, 53, 56, and 57)

Qian, R.; Tan, R. T.; Yang, W.; Su, J.; and Liu, J., 2018. Attentive generative adversarial network for raindrop removal from a single image. In *CVPR*. (cited on pages xviii, 5, 15, 74, 82, 83, 84, 85, 86, and 87)

Quan, Y.; Deng, S.; Chen, Y.; and Ji, H., 2019. Deep learning for seeing through window with raindrops. In *Proceedings of the IEEE International Conference on Computer Vision*. (cited on pages 14 and 15)

Radford, A.; Metz, L.; and Chintala, S., 2016. Unsupervised representation learning with deep convolutional generative adversarial networks. *International Conference on Learning Representations (ICLR)*, (2016). (cited on page 35)

Ren, D.; Zuo, W.; Hu, Q.; Zhu, P.; and Meng, D., 2019. Progressive image deraining networks: a better and simpler baseline. In *CVPR*. (cited on pages xviii, 81, 82, 83, 84, 86, and 87)

Ren, S.; He, K.; Girshick, R.; and Sun, J., 2015. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 2)

Ren, W.; Pan, J.; Cao, X.; and Yang, M.-H., 2017. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *ICCV*. (cited on page 15)

Riegler, G.; Liao, Y.; Donne, S.; Koltun, V.; and Geiger, A., 2019. Connecting the dots: Learning representations for active monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 101)

Rim, J.; Lee, H.; Won, J.; and Cho, S., 2020a. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*. Springer. (cited on page 61)

Rim, J.; Lee, H.; Won, J.; and Cho, S., 2020b. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *European Conference on Computer Vision*. (cited on page 62)

Rokanujjaman, M.; Islam, M. S.; Hossain, M. A.; Islam, M. R.; Makihara, Y.; and Yagi, Y., 2015. Effective part-based gait identification using frequency-domain gait entropy features. *Multimedia Tools and Applications*, (2015). (cited on pages 119 and 131)

Roser, M. and Geiger, A., 2009. Video-based raindrop detection for improved image registration. In *ICCV Workshops*. (cited on pages 14, 15, and 74)

Roser, M.; Kurz, J.; and Geiger, A., 2010. Realistic modeling of water droplets for monocular adherent raindrop recognition using bezier curves. In *ACCV*. (cited on pages 14 and 74)

Saad, M. A.; Bovik, A. C.; and Charrier, C., 2012. Blind image quality assessment: A natural scene statistics approach in the dct domain. *IEEE Transactions on Image Processing*, 21, 8 (2012), 3339–3352. (cited on page 17)

Santhaseelan, V. and Asari, V. K., 2012. A phase space approach for detection and removal of rain in video. In *Intelligent Robots and Computer Vision XXIX: Algorithms and Techniques*. (cited on page 15)

Santhaseelan, V. and Asari, V. K., 2015. Utilizing local phase information to remove rain from video. *International Journal of Computer Vision (IJCV)*, (2015). (cited on page 15)

Schmidt, U.; Rother, C.; Nowozin, S.; Jancsary, J.; and Roth, S., 2013. Discriminative non-blind deblurring. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 1)

Schuler, C. J.; Hirsch, M.; Harmeling, S.; and Schölkopf, B., 2016. Learning to deblur. *TPAMI*, (2016). (cited on page 25)

Sellent, A.; Rother, C.; and Roth, S., 2016. Stereo video deblurring. In *European Conference on Computer Vision (ECCV)*. (cited on pages 39 and 43)

Seok Lee, H. and Mu Lee, K., 2013. Dense 3d reconstruction from severely blurred images using a single moving camera. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 31)

Shaham, T. R.; Dekel, T.; and Michaeli, T., 2019. Singan: Learning a generative model from a single natural image. In *ICCV*. (cited on page 21)

Shakhnarovich, G. and Darrell, T., 2002. On probabilistic combination of face and gait cues for identification. In *FG*. (cited on page 122)

Shao, J.; Kang, K.; Change Loy, C.; and Wang, X., 2015. Deeply learned attributes for crowded scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 101)

Sheikh, H. R. and Bovik, A. C., 2006. Image information and visual quality. *IEEE Transactions on Image Processing*, 15, 2 (2006), 430–444. (cited on page 17)

Sheikh, H. R.; Bovik, A. C.; and De Veciana, G., 2005. An information fidelity criterion for image quality assessment using natural scene statistics. *IEEE Transactions on Image Processing*, 14, 12 (2005), 2117–2128. (cited on page 16)

Shen, Z.; Wang, W.; Lu, X.; Shen, J.; Ling, H.; Xu, T.; and Shao, L., 2019. Human-aware motion deblurring. In *IEEE International Conference on Computer Vision*. (cited on pages 12, 17, 62, and 70)

Shi, J.; Xu, L.; and Jia, J., 2014. Discriminative blur detection features. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 8)

Shi, J.; Xu, L.; and Jia, J., 2015. Just noticeable defocus blur detection and estimation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 31 and 45)

Shiraga, K.; Makihara, Y.; Muramatsu, D.; Echigo, T.; and Yagi, Y., 2016. Geinet: View-invariant gait recognition using a convolutional neural network. In *ICB*. (cited on page 120)

Simonyan, K. and Zisserman, A., 2015a. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*. (cited on pages 2, 22, 23, 35, and 122)

Simonyan, K. and Zisserman, A., 2015b. Very deep convolutional networks for large-scale image recognition. In *ICLR*. (cited on page 49)

Starik, S. and Werman, M., 2003. Simulation of rain in videos. In *Texture Workshop, ICCV*. (cited on page 98)

Su, S.; Delbracio, M.; Wang, J.; Sapiro, G.; Heidrich, W.; and Wang, O., 2017a. Deep video deblurring. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2017). (cited on pages 34, 37, 38, 39, 40, and 41)

Su, S.; Delbracio, M.; Wang, J.; Sapiro, G.; Heidrich, W.; and Wang, O., 2017b. Deep video deblurring for hand-held cameras. In *CVPR*. (cited on pages 12, 13, 20, 56, and 58)

Suin, M.; Purohit, K.; and Rajagopalan, A., 2020. Spatially-attentive patch-hierarchical network for adaptive motion deblurring. *arXiv preprint arXiv:2004.05343*, (2020). (cited on page 17)

Sun, D.; Yang, X.; Liu, M.-Y.; and Kautz, J., 2018. Pwc-net: Cnns for optical flow using pyramid, warping, and cost volume. In *CVPR*. (cited on page 50)

Sun, J.; Cao, W.; Xu, Z.; and Ponce, J., 2015. Learning a convolutional neural network for non-uniform motion blur removal. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 1, 11, 23, 25, 29, and 31)

Sun, L. and Hays, J., 2012. Super-resolution from internet-scale scene matching. In *IEEE International Conference on Computational Photography*. (cited on page 62)

Sun, S.-H.; Fan, S.-P.; and Wang, Y.-C. F., 2014a. Exploiting image structural similarity for single image rain removal. In *ICIP*. (cited on page 73)

Sun, Y.; Wang, X.; and Tang, X., 2014b. Deep learning face representation from predicting 10,000 classes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1891–1898. (cited on page 124)

Szeliski, R., 2010. *Computer vision: algorithms and applications*. Springer Science & Business Media. (cited on page 1)

Tanaka, Y.; Yamashita, A.; Kaneko, T.; and Miura, K. T., 2006. Removal of adherent waterdrops from images acquired with a stereo camera system. *IEICE TIS*, (2006). (cited on page 16)

Tang, C.; Zhu, X.; Liu, X.; Wang, L.; and Zomaya, A., 2019. Defusionnet: Defocus blur detection via recurrently fusing and refining multi-scale deep features. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 8)

Tang, P.; Wang, X.; Bai, X.; and Liu, W., 2017. Multiple instance detection network with online instance classifier refinement. In *CVPR*. (cited on page 122)

Tang, P.; Wang, X.; Wang, A.; Yan, Y.; Liu, W.; Huang, J.; and Yuille, A., 2018. Weakly supervised region proposal network and object detection. In *ECCV*. (cited on page 122)

Tao, D.; Li, X.; Wu, X.; and Maybank, S. J., 2007. General tensor discriminant analysis and gabor features for gait recognition. *TPAMI*, (2007). (cited on pages 119 and 131)

Tao, X.; Gao, H.; Shen, X.; Wang, J.; and Jia, J., 2018. Scale-recurrent network for deep image deblurring. In *CVPR*. (cited on pages 7, 11, 17, 20, 23, 24, 28, 29, 30, 66, 67, 68, 69, and 70)

Tesfaldet, M.; Brubaker, M. A.; and Derpanis, K. G., 2018. Two-stream convolutional networks for dynamic texture synthesis. In *CVPR*. (cited on page 50)

Tripathi, A. and Mukhopadhyay, S., 2012. Video post processing: low-latency spatiotemporal approach for detection and removal of rain. *IET Image Processing*, (2012). (cited on page 15)

Tripathi, A. K. and Mukhopadhyay, S., 2011. A probabilistic approach for detection and removal of rain from videos. *IETE Journal of Research*, 57, 1 (2011), 82–91. (cited on page 8)

Tripathi, A. K. and Mukhopadhyay, S., 2014. Removal of rain from videos: a review. *Signal, Image and Video Processing*, 8, 8 (2014), 1421–1430. (cited on page 8)

Tsuji, A.; Makihara, Y.; and Yagi, Y., 2010. Silhouette transformation based on walking speed for gait identification. In *CVPR*. (cited on page 119)

Vairy, M. and Venkatesh, Y. V., 1995. Deblurring gaussian blur using a wavelet array transform. *Pattern Recognition*, 28, 7 (1995), 965–976. (cited on page 8)

Villegas, R.; Yang, J.; Hong, S.; Lin, X.; and Lee, H., 2017. Decomposing motion and content for natural video sequence prediction. In *ICLR*. (cited on page 13)

Wagg, D. K. and Nixon, M. S., 2004. On automated model-based extraction and analysis of gait. In *FG*. (cited on page 122)

Wang, C.; Zhang, J.; Wang, L.; Pu, J.; and Yuan, X., 2012. Human identification using temporal information preserving gait template. *TPAMI*, (2012). (cited on page 122)

Wang, H.; Wang, Y.; Zhou, Z.; Ji, X.; Gong, D.; Zhou, J.; Li, Z.; and Liu, W., 2018a. Cosface: Large margin cosine loss for deep face recognition. In *CVPR*. (cited on page 119)

Wang, H.; Xie, Q.; Zhao, Q.; and Meng, D., 2020a. A model-driven deep neural network for single image rain removal. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 14)

Wang, L.; Tan, T.; Ning, H.; and Hu, W., 2003a. Silhouette analysis-based gait recognition for human identification. *TPAMI*, (2003). (cited on page 119)

Wang, T.; Yang, X.; Xu, K.; Chen, S.; Zhang, Q.; and Lau, R. W., 2019a. Spatial attentive single-image deraining with a high quality real rain dataset. In *CVPR*. (cited on pages 77 and 85)

Wang, T.-C.; Liu, M.-Y.; Zhu, J.-Y.; Liu, G.; Tao, A.; Kautz, J.; and Catanzaro, B., 2018b. Video-to-video synthesis. In *NIPS*. (cited on page 13)

Wang, X.; Chan, K. C.; Yu, K.; Dong, C.; and Change Loy, C., 2019b. EDVR: Video restoration with enhanced deformable convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition Workshop*. (cited on pages 12 and 13)

Wang, X.; Yu, K.; Wu, S.; Gu, J.; Liu, Y.; Dong, C.; Qiao, Y.; and Change Loy, C., 2018c. Esrgan: Enhanced super-resolution generative adversarial networks. In *ECCV*. (cited on page 64)

Wang, Y.; Song, Y.; Ma, C.; and Zeng, B., 2020b. Rethinking image deraining via rain streaks and vapors. In *European Conference on Computer Vision*, 367–382. Springer. (cited on page 14)

Wang, Z. and Bovik, A. C., 2002. A universal image quality index. *IEEE signal processing letters*, 9, 3 (2002), 81–84. (cited on page 17)

Wang, Z.; Bovik, A. C.; Sheikh, H. R.; and Simoncelli, E. P., 2004. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13, 4 (2004), 600–612. (cited on pages 16 and 32)

Wang, Z.; Simoncelli, E. P.; and Bovik, A. C., 2003b. Multiscale structural similarity for image quality assessment. In *Asilomar Conference on Signals, Systems and Computers (ACSSC)*. (cited on pages 16 and 32)

Wei, W.; Yi, L.; Xie, Q.; Zhao, Q.; Meng, D.; and Xu, Z., 2017. Should we encode rain streaks in video as deterministic or stochastic? In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*. (cited on pages 15, 96, and 97)

Whyte, O.; Sivic, J.; Zisserman, A.; and Ponce, J., 2012. Non-uniform deblurring for shaken images. *IJCV*, (2012). (cited on page 46)

Wu, Z.; Huang, Y.; Wang, L.; Wang, X.; and Tan, T., 2017. A comprehensive study on cross-view gait based human identification with deep cnns. *TPAMI*, (2017). (cited on pages 120, 122, 124, 125, 129, 131, and 133)

Xiong, W.; Luo, W.; Ma, L.; Liu, W.; and Luo, J., 2018. Learning to generate time-lapse videos using multi-stage dynamic generative adversarial networks. In *CVPR*. (cited on pages 13 and 50)

Xu, L. and Jia, J., 2010. Two-phase kernel estimation for robust motion deblurring. In *ECCV*. (cited on pages 1 and 19)

Xu, L.; Ren, J. S.; Liu, C.; and Jia, J., 2014a. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems (NIPS)*. (cited on page 7)

Xu, L.; Tao, X.; and Jia, J., 2014b. Inverse kernels for fast spatial deconvolution. In *European Conference on Computer Vision*. (cited on page 1)

Xu, L.; Zheng, S.; and Jia, J., 2013. Unnatural l0 sparse representation for natural image deblurring. In *CVPR*. (cited on page 19)

Yamashita, A.; Fukuchi, I.; and Kaneko, T., 2009. Noises removal from image sequences acquired with moving camera by estimating camera motion from spatio-temporal information. In *IROS*. (cited on page 14)

Yamashita, A.; Tanaka, Y.; and Kaneko, T., 2005. Removal of adherent waterdrops from images acquired with stereo camera. In *IROS*. (cited on pages 14 and 15)

Yang, G.; Song, X.; Huang, C.; Deng, Z.; Shi, J.; and Zhou, B., 2019a. Drivingstereo: A large-scale dataset for stereo matching in autonomous driving scenarios. In *CVPR*. (cited on pages 4 and 73)

Yang, W.; Liu, J.; and Feng, J., 2019b. Frame-consistent recurrent video deraining with dual-level flow. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 15, 89, and 98)

Yang, W.; Tan, R. T.; Feng, J.; Liu, J.; Guo, Z.; and Yan, S., 2017. Deep joint rain detection and removal from a single image. In *CVPR*. (cited on pages 5, 9, 14, 73, 77, 81, 82, 83, 85, 98, and 111)

Yang, W.; Tan, R. T.; Wang, S.; Fang, Y.; and Liu, J., 2020a. Single image deraining: From model-based to data-driven and beyond. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (2020). (cited on page 14)

Yang, W.; Tan, R. T.; Wang, S.; and Liu, J., 2020b. Self-learning video rain streak removal: When cyclic consistency meets temporal correspondence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. (cited on pages 15 and 16)

Yasarla, R.; Perazzi, F.; and Patel, V. M., 2019. Deblurring face images using uncertainty guided multi-stream semantic networks. *arXiv preprint arXiv:1907.13106*, (2019). (cited on page 17)

Ye, P.; Kumar, J.; Kang, L.; and Doermann, D., 2012. Unsupervised feature learning framework for no-reference image quality assessment. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 17)

You, S.; Tan, R. T.; Kawakami, R.; Mukaigawa, Y.; and Ikeuchi, K., 2015. Adherent raindrop modeling, detectionand removal in video. *TPAMI*, (2015). (cited on pages 14 and 15)

Yu, S.; Chen, H.; Reyes, E. B. G.; and Norman, P., 2017. Gaitgan: invariant gait feature extraction using generative adversarial networks. In *CVPR Workshops*. (cited on page 122)

Yu, S.; Tan, D.; and Tan, T., 2006. A framework for evaluating the effect of view angle, clothing and carrying condition on gait recognition. In *ICPR*. (cited on page 129)

Yue, Z.; Xie, J.; Zhao, Q.; and Meng, D., 2021. Semi-supervised video deraining with dynamical rain generator. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 642–652. (cited on pages 15 and 16)

Zhang, H.; Dai, Y.; Li, H.; and Koniusz, P., 2019a. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*. (cited on pages 12 and 29)

Zhang, H.; Dai, Y.; Li, H.; and Koniusz, P., 2019b. Deep stacked hierarchical multi-patch network for image deblurring. In *CVPR*. (cited on pages 66, 67, 69, and 70)

Zhang, H. and Patel, V. M., 2018a. Densely connected pyramid dehazing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 112 and 113)

Zhang, H. and Patel, V. M., 2018b. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*. (cited on pages xviii, 14, 74, 79, 82, 85, 86, 87, 111, and 112)

Zhang, H.; Sindagi, V.; and Patel, V. M., 2019c. Image de-raining using a conditional generative adversarial network. *TCSVT*, (2019). (cited on page 14)

Zhang, H. and Wipf, D., 2013. Non-uniform camera shake removal using a spatially-adaptive sparse penalty. In *NIPS*. (cited on page 45)

Zhang, J.; Pan, J.; Ren, J.; Song, Y.; Bao, L.; Lau, R. W.; and Yang, M.-H., 2018a. Dynamic scene deblurring using spatially variant recurrent neural networks. In *CVPR*. (cited on pages 11, 17, 20, and 70)

Zhang, K.; Huang, Y.; Du, Y.; and Wang, L., 2017. Facial expression recognition based on deep evolutional spatial-temporal networks. *IEEE Transactions on Image Processing*, (2017). (cited on page 122)

Zhang, K.; Li, D.; Luo, W.; Lin, W.-Y.; Zhao, F.; Ren, W.; Liu, W.; and Li, H., 2021a. Enhanced spatio-temporal interaction learning for video deraining: A faster and better framework. *arXiv preprint arXiv:2103.12318*, (2021). (cited on page 15)

Zhang, K.; Li, D.; Luo, W.; Ren, W.; Ma, L.; and Li, H., 2021b. Dual attention-in-attention model for joint rain streak and raindrop removal. *arXiv preprint arXiv:2103.07051*, (2021). (cited on page 17)

Zhang, K.; Luo, W.; Ren, W.; Wang, J.; Zhao, F.; Ma, L.; and Li, H., 2020a. Beyond monocular deraining: Stereo image deraining via semantic understanding. In *European Conference on Computer Vision*, 71–89. Springer. (cited on pages 16 and 17)

Zhang, K.; Luo, W.; Stenger, B.; Ren, W.; Ma, L.; and Li, H., 2020b. Every moment matters: Detail-aware networks to bring a blurry image alive. In *Proceedings of the 28th ACM International Conference on Multimedia*, 384–392. (cited on page 13)

Zhang, K.; Luo, W.; Yu, Y.; Ren, W.; Zhao, F.; Li, C.; Ma, L.; Liu, W.; and Li, H., 2021c. Beyond monocular deraining: Parallel stereo deraining network via semantic prior. *arXiv preprint arXiv:2105.03830*, (2021). (cited on page 17)

Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Liu, W.; and Li, H., 2018b. Adversarial spatio-temporal learning for video deblurring. *TIP*, (2018). (cited on page 12)

Zhang, K.; Luo, W.; Zhong, Y.; Ma, L.; Stenger, B.; Liu, W.; and Li, H., 2020c. Deblurring by realistic blurring. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on pages 11, 17, 66, 67, and 70)

Zhang, R.; Isola, P.; Efros, A. A.; Shechtman, E.; and Wang, O., 2018c. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 586–595. (cited on page 17)

Zhang, W. and Cham, W.-K., 2009. Single image focus editing. In *International Conference on Computer Vision Workshop*. (cited on page 8)

Zhang, X.; Dong, H.; Hu, Z.; Lai, W.-S.; Wang, F.; and Yang, M.-H., 2018d. Gated fusion network for joint image deblurring and super-resolution. *arXiv preprint arXiv:1807.10806*, (2018). (cited on page 12)

Zhang, X.; Li, H.; Qi, Y.; Leow, W. K.; and Ng, T. K., 2006. Rain removal in video by combining temporal and chromatic properties. In *ICME*. (cited on pages 15 and 73)

Zhao, H.; Shi, J.; Qi, X.; Wang, X.; and Jia, J., 2017. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 101)

Zhao, L.; Peng, X.; Tian, Y.; Kapadia, M.; and Metaxas, D., 2018. Learning to forecast and refine residual motion for image-to-video generation. In *ECCV*. (cited on page 13)

Zhao, W.; Zheng, B.; Lin, Q.; and Lu, H., 2019. Enhancing diversity of defocus blur detectors via cross-ensemble network. In *IEEE Conference on Computer Vision and Pattern Recognition*. (cited on page 8)

Zheng, L.; Shen, L.; Tian, L.; Wang, S.; Wang, J.; and Tian, Q., 2015. Scalable person re-identification: A benchmark. In *ICCV*. (cited on pages 4 and 73)

Zhou, M.; Xiao, J.; Chang, Y.; Fu, X.; Liu, A.; Pan, J.; and Zha, Z.-J., 2021. Image de-raining via continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4907–4916. (cited on page 14)

Zhou, S.; Zhang, J.; Pan, J.; Xie, H.; Zuo, W.; and Ren, J., 2019a. Spatio-temporal filter adaptive network for video deblurring. In *IEEE International Conference on Computer Vision*. (cited on page 13)

Zhou, S.; Zhang, J.; Zuo, W.; Xie, H.; Pan, J.; and Ren, J. S., 2019b. Davanet: Stereo deblurring with view aggregation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. (cited on page 16)

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A., 2017a. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*. (cited on pages 2, 13, and 52)

Zhu, J.-Y.; Zhang, R.; Pathak, D.; Darrell, T.; Efros, A. A.; Wang, O.; and Shechtman, E., 2017b. Toward multimodal image-to-image translation. In *NeurIPS*. (cited on page 22)

Zhu, L.; Deng, Z.; Hu, X.; Xie, H.; Xu, X.; Qin, J.; and Heng, P.-A., 2020. Learning gated non-local residual for single-image rain streak removal. *IEEE Transactions on Circuits and Systems for Video Technology*, (2020). (cited on page 14)

Zhu, L.; Fu, C.-W.; Lischinski, D.; and Heng, P.-A., 2017c. Joint bi-layer optimization for single-image rain streak removal. In *ICCV*. (cited on pages 14, 111, and 112)