LEARNING-BASED METHODS FOR SINGLE IMAGE RESTORATION AND TRANSLATION

BY HE ZHANG

A dissertation submitted to the

School of Graduate Studies

Rutgers, The State University of New Jersey

in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

Graduate Program in Electrical and Computer Engineering

Written under the direction of

Vishal M Patel

and approved by

____

____

____

____

New Brunswick, New Jersey

JANUARY, 2019

ABSTRACT OF THE DISSERTATION

Learning-based Methods for Single Image Restoration and Translation

by He Zhang

Dissertation Director: Vishal M Patel

In many applications such as drone-based video surveillance, self driving cars and recognition under night-time and low-light conditions, the captured images and videos contain undesirable degradations such as haze, rain, snow, and noise. Furthermore, the performance of many computer vision algorithms often degrades when they are presented with images containing such artifacts. Hence, it is important to develop methods that can automatically remove these artifacts. However, these are difficult problems to solve due to their inherent ill-posed nature. Existing approaches attempt to introduce prior information to convert them into well-posed problems. In this thesis, rather than purely relying on prior-based models, we propose to combine them with data-driven models for image restoration and translation. In particular, we develop new data-driven approaches for 1) single image de-raining, 2) single image dehazing, and 3) thermal-to-visible face synthesis.

In the first part of the thesis, we develop three different methods for single image de-raining. In the first approach, we develop novel convolutional coding-based methods for single image de-raining, where two different types of filters are learned via convolutional sparse and low-rank coding to characterize the background component and rain-streak component separately. These pre-trained filters are then used to separate the rain component from the image. In the second approach, to ensure that the restored de-rained results are indistinguishable from their corresponding clear images, we propose a novel single image de-raining method called Image De-raining Conditional General Adversarial Network (ID-CGAN) which consists of a new refined perceptual loss function and a novel multi-scale discriminator. Finally, to deal with nonuniform rain densities, we present a novel density-aware multi-stream densely connected convolutional neural network-based algorithm that enables the network itself to automatically determine the rain-density information and then efficiently remove the corresponding rain-streaks guided by the estimated rain-density label.

In the second part of the thesis, we develop an end-to-end deep learning-based method to address the single image dehazing problem. We propose to combine the physics-based image formation model with data-driven approach for single image dehazing. In particular, a new end-to-end single image dehazing method, called Densely Connected Pyramid Dehazing Network (DCPDN), is proposed which can jointly estimate the transmission map, atmospheric light and dehazed image all together. The end-to-end learning is achieved by directly embedding the atmospheric scattering model into the network, thereby ensuring that the proposed method strictly follows the physics-driven scattering model for dehazing.

In the final part of the thesis, we develop an image-to-image translation method for generating high-quality visible images from polarimetric thermal faces. Since polarimetric images contain different stokes images containing various polarization state information, we propose a Generative Adversarial Network-based multi-stream feature-level fusion technique to synthesize high-quality visible images from polarimetric thermal images. An application of this approach is presented in polarimetric thermal-to-visible cross-modal face recognition.

Acknowledgements

Firstly, I'd like express great thanks to my advisor Prof. Vishal M. Patel, who not only picked me at the begging of my academic career but also guide me and support me through my PhD career. Good luck on your new journey in Johns Hopkins.

Secondly, I'd like to thank all my PhD committee members: Prof. Kristin Dana, Prof. Peter Meer, Prof. Laleh Najafizadeh, Prof. Yingying Chen, Prof. Kevin S. Zhou. They give me considerate suggestions to my final dissertation thesis. In addition, I'd like to express my gratitude to my intern mentors: Dr. Jianming Zhang, Dr. Zhe Lin, Dr. Federico Perazzi and Dr. Mingqing Chen.

Thirdly, I sincerely express my thanks to all my collages: Dr. Shaogang Wang, Dr. Shunqiao Sun, Dr. Hang Zhang and Vishwanath Sindagi , who lead me to the gate of research and inspire me during my whole PhD life. In addition, many thanks to my lab mates and friends: Pramuditha Perera, Xing Di, Puyang Wang, Lidan Wang, Poojan Oza, Mahdi Abavisani, Jia Xue and Yi Han,. Also, a special thank to my undergraduate thesis advisor Prof. Le Yang, who greatly influenced my academic career and leads a power-electronic guy into the field of signal processing and computer vision.

Finally, my special appreciation is dedicated to my fiancée Li (Leona) Liu, who give me so much support both in work and life. And my deepest appreciation is dedicated to my parents who inspire me and support me throughout life.

New Chapter.

Dedication

This work is dedicated to my family who love, encourage, and urge me all along.

Table of Contents

List of Tables

List of Figures

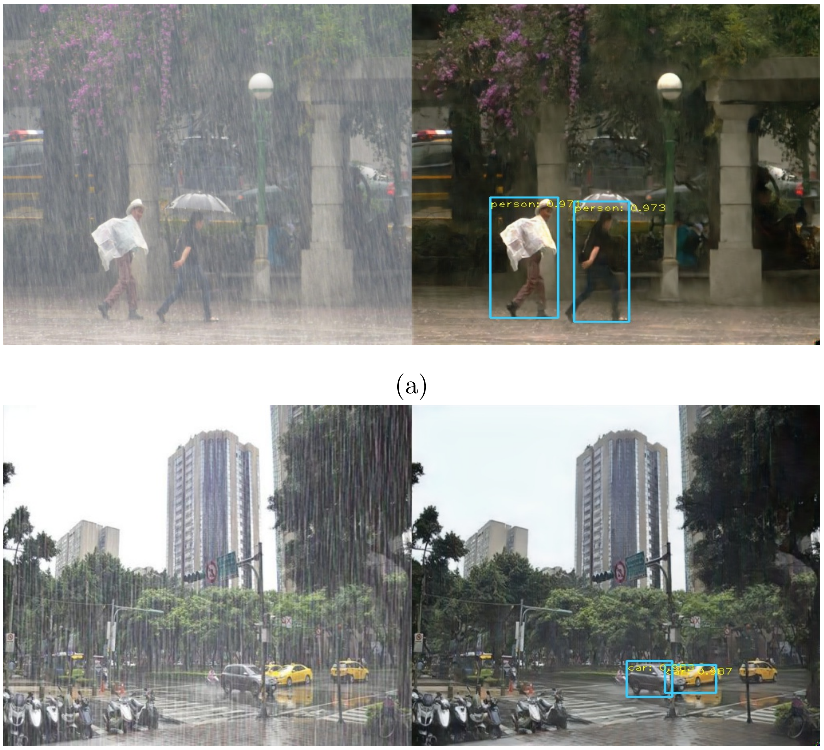
Chapter 1

Introduction

Images are ubiquitous and indispensable in modern science and everyday life. Mirroring the abilities of our own human visual system, it is natural to display observations of the world in a graphical form. Images are obtained in areas ranging from everyday photography to astronomy, remote sensing, medical imaging and microscopy. In each case, there is an underlying object or scene we wish to observe; the image is a visual representation of these observations.

Yet imaging, just as any other observation process, is never perfect, especially for the images captured in unconstrained conditions such as rain, haze or extreme illumination. It has been widely acknowledged that these unpredictable impairments adversely affect the performance of many computer vision algorithms such as detection, classification and tracking. This is primarily due to the fact that these algorithms are trained using images that are captured under well-controlled conditions. Consider the example of rainy image as shown in Figure 1.1. From this figure, it can be observed that the presence of rain-streaks greatly impairs the visual quality of the image, thus rendering object (human or car) detection and verification algorithms ineffective under such degradations. A possible solution to address this issue is to include images captured under unconstrained conditions in the training process of these algorithms. However, it may not be practical to collect such images for all classes in the training set, especially in a large scale setting. Hence, it is important to develop algorithms that can automatically remove these artifacts.

As image restoration is an ill-posed problem, some of the previous approaches have introduced different priors to solve the problem. Some of the priors include low-rank



(a)

(b)

Figure 1.1: Real-world examples of rainy images and their corresponding de-rained results. By comparing the figures on the left with the figures on the right, one can clearly see that the object detector fails to detect objects in rainly images. Once the images are restored (i.e. de-rained), the detector is able to detet the objects.

prior for image denoising [48], sparsity prior for image super-resolution [156], and dark-channel prior for the image dehazing [50]. However, purely prior-based approaches do not often work well in practice since they do not consider the other information available in the real-world data for restoration. Hence, it is important consider the data-driven knowledge into the algorithms for obtaining better results.

The main focus of this thesis is to design learning-based methods for image restoration and translation problems. Specifically, we consider three different kinds of image restoration and translation problems in this thesis. These include single image de-raining, single image de-hazing and polarimetric thermal to visible image synthesis. In what follows, we give a brief overview of these problems.

1.1 Thesis Overview

1.1.1 Single Image De-raining

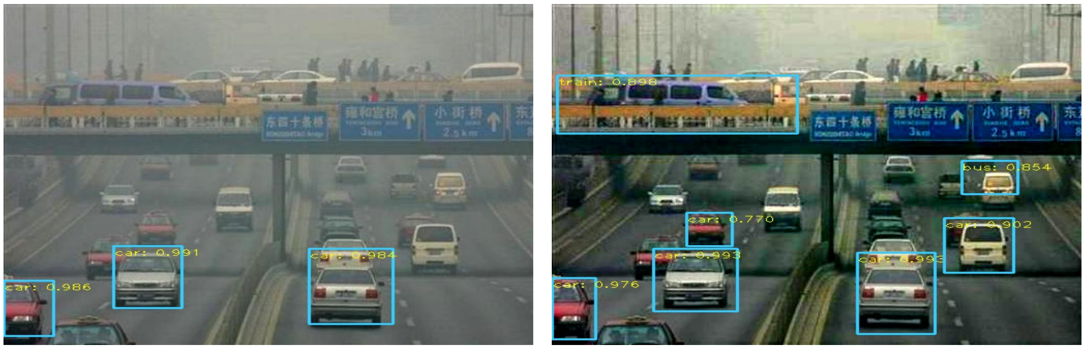
In this thesis, we propose three different learning-based approaches to address the single image de-raining problem. In Chapter 3, we propose a novel Convolutional Coding-based Rain Removal (CCRR) algorithm to automatically remove rain streaks from a single rainy image. Our method first learns a set of generic sparsity-based and low-rank representation-based convolutional filters for efficiently representing background clear image and rain streaks, respectively. To this end, we first develop a new method for learning a set of convolutional low-rank filters. Then, using these learned filters, we propose an optimization problem to decompose a rainy image into a clear background image and a rain streak image. By working directly on the whole image, the proposed rain streak removal algorithm does not need to divide the image into overlapping patches for leaning local dictionaries. Extensive experiments on synthetic and real images show that the proposed method performs favorably compared to the state-of-the-art rain streak removal algorithms.

In Chapter 4, we investigate a new point of view in addressing the single image de-raining problem. Instead of focusing only on deciding what is a good prior or a good framework to achieve good quantitative and qualitative performance, we also ensure that the de-rained image itself does not degrade the performance of a given computer vision algorithm such as detection and classification. In other words, the de-rained result should be indistinguishable from its corresponding clear image to a given discriminator. This criterion can be directly incorporated into the optimization framework by using the recently introduced conditional generative adversarial networks (GANs). To minimize artifacts introduced by GANs and ensure better visual quality, a new refined loss function is introduced. Based on this, we propose a novel single image de-raining method called Image De-raining Conditional General Adversarial Network (ID-CGAN), which considers quantitative, visual and also discriminative performance into the objective function. In addition, a new de-raining dataset is created and has been made publicly available to the research community.

Furthermore, to better characterize rain-streaks with different scales, shapes and density, we present a novel density-aware multi-stream densely connected convolutional neural network-based algorithm, called DID-MDN, for joint rain density estimation and de-raining in Chapter 4. The proposed method enables the network itself to automatically determine the rain-density information and then efficiently remove the corresponding rain-streaks guided by the estimated rain-density label. To better characterize rain-streaks with different scales and shapes, a multi-stream densely connected de-raining network is proposed which efficiently leverages features from different scales. Furthermore, a new dataset containing images with rain-density labels is created and is used to train the proposed density-aware network.

1.1.2 Single Image Dehazing

Similar to the de-raining problem, single image dehazing is also important since the presenze of haze often degrades the image quality, as shown in Figure 1.2. In Chapter 6), we propose a new end-to-end single image dehazing method, called Densely Connected Pyramid Dehazing Network (DCPDN), which can jointly learn the transmission map, atmospheric light and dehazed result all together. The end-to-end learning is achieved by directly embedding the atmospheric scattering model into the network, thereby ensuring that the proposed method strictly follows the physics-driven scattering model



Before Dehazing

After Dehazing

Figure 1.2: Real-world examples of hazy image and corresponding dehazed results. Object detection results are also shown on the images before and after restoration.

for dehazing. Inspired by the dense network that can maximize the information flow along features from different levels, we propose a new edge-preserving densely connected encoder-decoder structure with multi-level pyramid pooling module for estimating the transmission map. This network is optimized using a newly introduced edge-preserving loss function. To further incorporate the mutual structural information between the estimated transmission map and the dehazed result, we propose a joint-discriminator based on GANs framework to decide whether the corresponding dehazed image and the estimated transmission map are real or fake. An ablation study is conducted to demonstrate the effectiveness of each module evaluated at both estimated transmission map and dehazed result.

1.1.3 Thermal-to-Visible Face Synthesis

The large domain discrepancy between faces captured in polarimetric (or conventional) thermal and visible domain makes cross-domain face verification a highly challenging problem for human examiners and computer vision algorithms. Consider the images shown in Figure 1.3. On the top of the figure, thermal images are displayed while on the bottom the corresponding visible images are displayed. As can be seen from this figure, it is difficult to visually distinguish between the thermal images. However, it is much easier to distinguish between these two images in the visible domain.

Hence, it is important to develop algorithms that transfer the corresponding thermal images into the visible domain. Previous approaches utilize either a two-step procedure

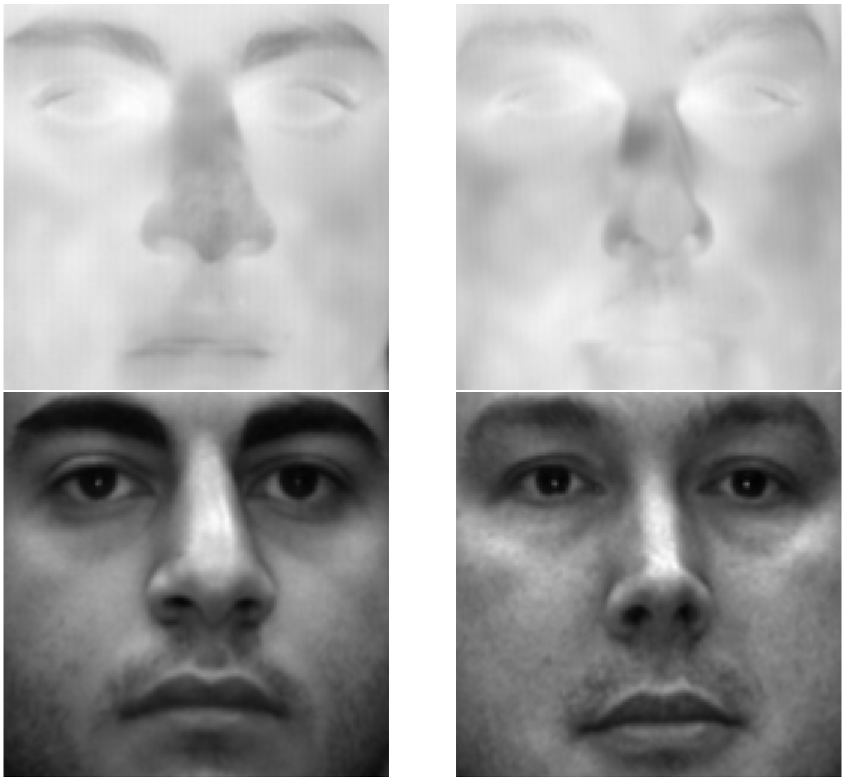


Figure 1.3: Real-world examples of thermal faces (top row) and visible faces (bottom row).

(visible feature estimation and visible image reconstruction) or an input-level fusion technique, where different polarimetric Stokes images are concatenated and used as a multichannel input to synthesize the visible image given the corresponding polarimetric signatures. Although these methods have yielded improvements, we argue that input-level fusion alone may not be sufficient to realize the full potential of the available Stokes images. We propose a GAN-based multi-stream feature-level fusion technique to synthesize high-quality visible images from prolarimetric thermal images. The proposed network consists of a generator sub-network, constructed using an encoder-decoder network based on dense residual blocks, and a multi-scale discriminator sub-network. The generator network is trained by optimizing an adversarial loss in addition to a perceptual loss and an identity preserving loss to enable photo realistic generation of visible images while preserving the discriminative characteristics. An extended dataset consisting of polarimetric thermal facial signatures of 111 subjects is also introduced.

Chapter 2

Background and Related Work

In this chapter, we review background works for single image de-raining, single image dehazing and cross-domain face synthesis. In addition, we also review some related methods, such as Convolutional Sparse Coding (CSC), GANs and perceptual loss functions.

2.1 Single Image De-raining

One can model the observed rainy image as the superposition of two images - one corresponding to rain streak and the other corresponding to clear background (see Fig. 2.1). The input rainy image can be expressed as

$$\mathbf{y} = \mathbf{y}_c + \mathbf{y}_r, \tag{2.1}$$

where $\mathbf{y} \in \mathbb{R}^{C \times M \times N}, \mathbf{y}_c \in \mathbb{R}^{C \times M \times N}$ and $\mathbf{y}_r \in \mathbb{R}^{C \times M \times N}$ are rainy image ($\mathbf{y}$), clear background image ($\mathbf{y}_c$) and rain streak image ($\mathbf{y}_r$), respectively. All three images are with C channel, height M and width N. Given, $\mathbf{y}$, the goal of rain streak removal (i.e. de-raining) is to estimate $\mathbf{y}_c$.

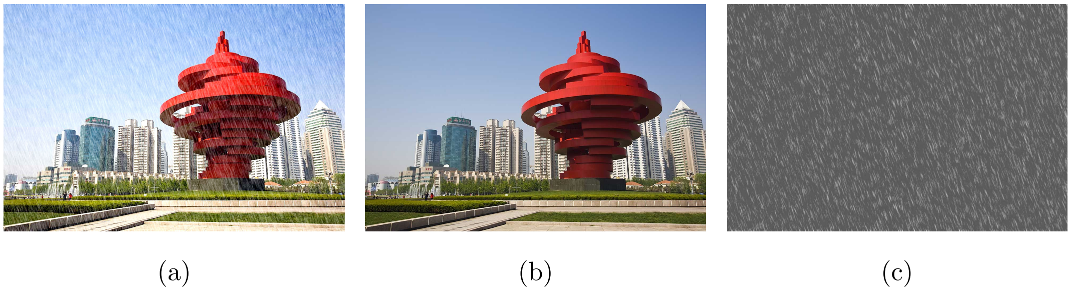

(a) (b) (c)

Figure 2.1: Rain streak removal from a single image. A rainy image (a) can be viewed as the superposition of a clean background image (b) and a rain streak image (c).

In order to solve this ill-posed problem, various methods have been developed in the literature that make use of either multiple images [112, 147, 41, 42, 176] or a prior [112, 158, 168, 186, 68] to obtain a restored image. In what follows, we review some single image de-raining methods.

2.1.1 Layer Separation Methods

The key idea here is to utilize different prior assumptions to characterize various image components separately and decompose a given rainy image into a clean background image and a rain streak component. Priors such as sparsity, low-rank and Gaussian mixture model have been investigated in the literature. The image separation model has also been explored in other applications such as image decomposition [131, 106, 97], image de-noising [152, 142] and image reflection removal.

Sparsity-based Methods

Sparse coding-based clustering method [68] is among the first ones to tackle the single image de-raining problem where the authors proposed to solve it in the image decomposition framework. They first separated the input image into low frequency and high frequency images using a bilateral filter. The high frequency image is further decomposed into rain and non-rain components based on the assumption that learned dictionary atoms can sparsely represent clear background image and rain-streak image separately. An important assumption that is made in this approach is that rain streaks usually have similar edge orientations. This may result in the removal of non-rain component as rain. Also, the method's effectiveness is dependent on the performance of the bilateral filter and clustering of basis vectors for generating sparse representation. Similar to the above approach, Luo *et al.* in [86] propose a discriminative sparse coding based method that considers the mutual exclusive property into the optimization framework. Though the authors present significant improvements as compared to previous methods, their method is ineffective in removing large rain-streaks due to the assumption that rain streaks are high frequency components. In addition, due to the same assumption, their method generates artifacts around the rain-streak components in the resulting images.

Low-rank Representation-based Methods

Chen *et al.* proposed a low-rank representation-based method [19] that uses patch-rank as a prior to characterize unpredictable rain pattern. This is inspired by the observation that rain streak components within the same rainy image share similar shapes and orientations. Hence, the rain streaks can be better characterized by the low-rank property. They use a low-rank model to capture correlated rain streaks while using the total-variation norm as the prior for the clean background image. However, the low-rank property tends to remove important texture details such as bricks on the wall in the de-rained images and hence the de-rained results tend to loose important details in the restored image.

Gaussian Mixture Model-based Methods

Li *et al.* in [83] used the image decomposition framework to propose patch-based priors for the background and the rain component. These priors are based on Gaussian Mixture Models (GMMs) which can accommodate multiple orientations and scales of rain streaks. These methods [19, 83] are based on the assumption that rain streaks have similar patterns and orientations. Due to this assumption, they tend to capture other global repetitive patterns such as brick and texture which results in removal of certain non-rain components from the background image.

2.1.2 Deep Learning-based Methods

Recently, due to the immense success of deep learning in both high-level and low-level vision tasks [52, 153, 60, 128, 171, 129, 142, 103, 179, 189, 191, 166, 105, 24], several Convolutional Neural Network (CNN) based methods have also been proposed for image de-raining [37, 38, 158]. In these methods, the idea is to learn a mapping between input rainy images and their corresponding ground truths using a CNN structure. According to the observation that both rain streaks and object details remain only in the detail layer, Fu *et al.* [38] leverage a two step procedure, where the input rainy image is decomposed into a based layer and a detail layer separately. Then, a CNN non-linear mapping is learned to remove the rain streaks in the detail layer. Built on [38], Fu *et al.* extend the network structure via leveraging Res-block [52] in [37]. Yang *et al.* proposed a CNN structure that can jointly detect the rain streaks and remove them simultaneously. Some of the other deep learning-based methods include [143, 146, 39, 33].

2.2 Convolutional Sparse Coding (CSC)

In CSC, given a set of M training samples $\{\mathbf{y}_m\}_{i=1}^{M}$, the objective is to learn a set of convolutional filters $\{\mathbf{d}_k\}_{i=1}^{K}$ by solving the following optimization problem

$$\begin{aligned} \arg\min_{\mathbf{d},\mathbf{x}} \quad & \frac{1}{2}\sum_{m=1}^{M}\left\|\mathbf{y}_m - \sum_{k=1}^{K}\mathbf{d}_k * \mathbf{x}_{m,k}\right\|_2^2 + \\ & \lambda\sum_{m=1}^{M}\sum_{k=1}^{K}\|\mathbf{x}_{m,k}\|_1 \\ \text{subject to} \quad & \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1,\cdots,K\}, \end{aligned} \tag{2.2}$$

where $\mathbf{x}_{m,k}$ are the sparse coefficients that approximate the data $\mathbf{y}_m$ when convolved with the corresponding filters $\mathbf{d}_k$ of fixed support and for an N-dimensional vector $\mathbf{x}$, $\|\cdot\|_q$ denotes the ℓ_q-norm, $0 < q < \infty$, defined as $\|\mathbf{x}\|_q = \left(\sum_{i=1}^{N}|x_i|^q\right)^{\frac{1}{q}}$. Here, $*$ represents the 2-D convolution operator and λ is a positive regularization parameter. Several methods have been proposed in the literature for solving the above optimization problem [12, 54, 150, 148]. In particular, [150], [148] developed an efficient method that jointly uses the space and Fourier domains to solve the CSC problem.

2.3 Single Image Dehazing

The image degradation (atmospheric scattering model) due to the presence of haze is mathematically formulated as

$$I(z) = J(z)t(z) + A(z)(1 - t(z)), \tag{2.3}$$

where I is the observed hazy image, J is the true scene radiance, A is the global atmospheric light, indicating the intensity of the ambient light, t is the transmission map and z is the pixel coordinates. Transmission map is the distance-dependent factor that affects the fraction of light that reaches the camera sensor. When the atmospheric light A is homogeneous, the transmission map can be expressed as $t(z) = e^{-\beta d(z)}$, where β represents attenuation coefficient of the atmosphere and d is the scene depth. In single image dehazing, given I, the goal is to estimate J.

It can be observed from (2.3) that there exists two important aspects in the dehazing process: *(1) accurate estimation of transmission map,* and *(2) accurate estimation of atmospheric light.* Apart from several works that focus on estimating the atmospheric light [8, 133], most of the other algorithms concentrate more on the accurate estimation of the transmission map and they leverage empirical rule in the estimation of the atmospheric light [50, 91, 113, 135]. This is mainly due to the common belief that good estimation of transmission map will always lead to better dehazing. These methods can be broadly divided into two main categories: (1). Handcraft prior-based methods and (2). Learning-based methods. Handcraft prior-based methods often leverage different priors in characterizing the transmission map such as dark-channel prior [50], contrast color-lines [35] and haze-line prior [7], while learning-based methods, such as those based on CNNs, attempt to learn the transmission map directly from the training data [136, 113, 13, 174, 78]. Once the transmission map and the atmospheric light are estimated, the dehazed image can be recovered using the following equation (2.4):

$$\hat{J}(z) = \frac{I(z) - \hat{A}(z)(1 - \hat{t}(z))}{\hat{t}(z)}. \tag{2.4}$$

2.3.1 Handcrafted Prior-based Dehazing Methods

Similar to the single image de-raining problem, single image dehazing is also a highly ill-posed problem. Various handcrafted prior-based methods have been explored to tackle this problem. Fattal [34] proposed a physically-grounded method by estimating the albedo of the scene. As the images captured from the hazy conditions always lack color contrast, Tan [135] *et al.* proposed a patch-based contrast-maximization method. In [74], Kratz and Nishino proposed a factorial MRF model to estimate the albedo and depths filed. Inspired by the observations that outdoor objects in clear weather have at least one color channel that is significantly dark, He. *et al.* in [50] proposed a dark-channel model to estimate the transmission map. More recently, Fattal [35] proposed a color-line method based on the observation that small image patches typically exhibit a one-dimensional distribution in the RGB color space. Similarly, Berman *et al.* [7] proposed a non-local patch prior to characterize the clean images.

2.3.2 Learning-based Dehazing Methods

Unlike some of the above mentioned methods that use different priors to estimate the transmission map, Cai *et al.* [13] introduce an end-to-end CNN network for estimating the transmission with a novel BReLU unit. In addition, they leverage the depth dataset such as NYU-depth [95] to synthesize a large scale synthetic dataset for estimation of transmission map. More recently, Ren *et al.* [113] proposed a multi-scale deep neural network to estimate the transmission map. One of the limitations of these methods is that they limit their capabilities by only considering the transmission map in their CNN frameworks. To address this issue, Li. *et al* [78] proposed an all-in-one dehazing network, where a linear transformation is leveraged to encode the transmission map and the atmospheric light into one variable. Some of the other deep learning-based single image dehazing include [159, 155, 114, 175]. Furthermore, several benchmark datasets of both synthetic and real-world hazy images for dehazing problems have also been introduced to the research community [177, 77, 1, 1].

2.4 Heterogeneous Face Recognition

In heterogeneous face recognition, one has to match two images corresponding to the same subject captured by two different modalities such as visible and thermal. Distributional change between the modalities makes the heterogeneous face recognition very difficult. Various methods have been proposed in the literature for heterogeneous face recognition such as infrared-to-visible [76, 53], thermal-to-visible [121, 118, 172, 63, 71], and sketch-to-visible [40, 101] [141]. These approaches essentialy attempt to tackle the heterogeneous face recognition problem by either synthesizing visible faces from the other domain, extracting domain-invariant features from these modalities, or projecting heterogeneous data onto a common latent space for cross-modal matching.

Klare and Jain [72] proposed an approach using kernel prototype similarities, where after geometric normalization and image filtering (e.g., Difference of Gaussian, Center-Surround Divisive Normalization [92], and Gaussian) and local features extraction (e.g., multi-scale local binary patterns, or MLBP, and scale invariant feature transform, or SIFT), the intra-domain kernel similarities between source (or target) domain images and all training images from the source (or target) domain. These intra-domain kernel similarity, which are computed using the cosine kernel, provide relational vectors for source and target domain imagery to be compared, where the main idea is that the kernel similarity between two source domain images should be similar to the kernel similarity between two corresponding target domain images.

Yi *et al.* [160] leverage the use of multi-modal Restricted Boltzmann Machines (RBMs) [94] to learn a shared representation, and for NIR-to-visible face recognition. Here, they learn a shared representation using the multi-modal RBMs locally for each patch. However, since heterogeneity is only addressed locally, they further reduce the modality gap by performing Hetra-component analysis (HCA) [82] for the holistic image representation. Hetero-component analysis is based on the theory that most of the appearance differences between imaging modalities are captured in the top eigenvectors. Therefore, a common representation is given by removing the effects from the top eigenvectors. This was shown to achieve excellent performance for NIR-to-visible face recognition. However, it is unclear how well this method would work for an emissive infrared band, such as LWIR, where the facial signatures are very different than in the visible or NIR bands.

Riggan *et al.* [116] proposed a coupled auto-associative network for learning common representation between thermal and visible face images. The authors optimize two sparse auto-encoders jointly, such that (1) the information within each modality is preserved, and (2) the inter-domain representations are similar for the corresponding images. Although this approach demonstrated some success and robustness, the constraint to preserve information for the source domain is not a necessary condition as long as the discriminability is maintain when learning the common representation.

Hu *et al.* [55] use conventional thermal images when applying a one-versus-all framework using partial least squares classifiers on the Histogram of Oriented Gradients (HOG) features. For each classifier, they introduce the concept of adding cross-domain negative samples (i.e., thermal samples from a different subject) for robustness. Later, Riggan *et al.* [118] proposed the use of a coupled neural network and a discriminative classifier for enhancing the conventional thermal-to-visible face recognition and polarimetric thermal-to-visible framework.

Riggan *et al.* [117] proposed a way to synthesize a visible image from both conventional thermal and polarimetric thermal images. This approach used a CNN to extract features from a conventional or polarimetric thermal images and then mapped those features to a corresponding visible representation using a deep perceptual mapping [88], where this representation in inverted back to the imaging domain using a forward CNN model. One potential concern is the piece-wise nature of this synthesis method. Later, built up on the success of GANs [47], Zhang *et al.* [172] improved the synthesis results by proposing an end-to-end conditional generative adversarial network (CGAN) approach, which is optimized via a newly introduced identity-preserving loss, to synthesis a visible image from a thermal image. This approach demonstrated results that were photo-realistic and discriminative.

2.5 Generative Adversarial Networks (GANs)

Generative adversarial networks were proposed by Goodfellow *et al.* in [47] to synthesize realistic images by effectively learning the distribution of training images. The authors adopted a game theoretic min-max optimization framework to simultaneously train two models: a generative model G and a discriminative model D. The goal of GANs is to train G to produce samples from training distribution such that the synthesized samples are indistinguishable from actual distribution by the discriminator D. Unlike other generative models such as Generative Stochastic Networks [137], GANs do not require a Markov chain for sampling and can be trained using standard gradient descent methods [47].

In order to learn a good generator G so as to fool the learned discriminator D and to make the discriminator D good enough to distinguish real from fake, the proposed method alternatively updates G and D. Given an actual image $\mathbf{x}$ and a random noise vector $\mathbf{z}$, the original GAN aims to learn a mapping function to generate output image $\mathbf{y}$ by solving the following optimization problem:

$$\begin{aligned}\min_G \max_D \quad & \mathbb{E}_{\mathbf{z}\sim p_{data(\mathbf{z})}},[\log(1-D(G(\mathbf{z})))]+ \\ & \mathbb{E}_{\mathbf{x}\sim p_{data(\mathbf{x})}}[\log D(\mathbf{x}))].\end{aligned} \tag{2.5}$$

Initially, the success of GANs was limited as they were known to be unstable to train, often resulting in artifacts in the synthesized images. Radford *et al.* in [107] proposed Deep Convolutional GANs (DCGANs) to address the issue of instability by including a set of constraints on their topology. Another limiting issue in GANs is that, there is no control on the modes of data being synthesized by the generator in case of these unconditioned generative models. Mirza *et al.* [93] incorporated additional conditional information in the model, which resulted in effective learning of the generator. The use of conditioning variables for augmenting side information not only increased the stability in learning but also improved the descriptive power of the generator G [69]. More recently, researchers have explored various aspects of GANs such as training improvements[120] and use of task specific cost function [23]. Also, an alternative viewpoint for the discriminator function is explored by Zhao *et al.* [182] where they deviate from the traditional probabilistic interpretation of the discriminator model. Most recently, [4] propose to minimize the Earth-Mover distance between the density of generated samples and the true data density, and they show the resultant Wasserstein GAN (WGAN) can address the vanishing gradient problem that the classic GAN suffers.

2.5.1 Applications of GANs

The success of GANs in synthesizing realistic images has led researchers exploring the GAN framework for numerous applications such as style transfer [79], image inpainting [100, 161], text to image translation [164, 154, 179], image to image translation [110, 180, 185], image super-resolution [75], texture synthesis [66] and generating outdoor scenes from attributes [69]. Isola *et al.* [64] proposed a general purpose solution for image-to-image translation using conditional adversarial networks. Apart from learning a mapping function, they argue that the network also learns a loss function, eliminating the need for specifying or hand designing a task specific loss function. Karacan *et al.* in [69] proposed a deep GAN conditioned on semantic layout and scene attributes to synthesize realistic outdoor scene images under different conditions. Recently, Jetchev *et al.* [66] proposed spatial GANs for texture synthesis. Deviating from traditional GANs, their input noise distribution constitutes a whole spatial tensor instead of a vector, thus enabling them to create architectures more suitable for texture synthesis.

2.6 Perceptual Loss Function

Loss functions form an important and integral part of learning process, especially in CNN-based reconstruction tasks. Several works [27],[85, 28, 89, 183, 26, 43, 165] have explored different loss functions and their combinations for effective learning for tasks such as super-resolution, semantic segmentation, depth estimation, feature inversion and style transfer. Initial work on CNN-based image translation or restoration optimized over pixel-wise L2-norm (Euclidean loss) or L1-norm between the predicted and ground truth images [85, 28]. Since these losses operate at pixel level, their ability to capture high level perceptual/contextual details is limited and they tend to produce blurred results [75]. Hence, many authors argue and demonstrate through their results that it would be better to optimize a perceptual loss function where the aim is to minimize perceptual difference between reconstructed image and the ground truth image [67]. In a different approach, the conditional GAN framework can also be considered as an attempt to explore a structured loss function where, a generator network is trained to minimize the discriminator's ability to correctly classify between the synthesized image and the corresponding ground truth image. Researchers have attempted to solve various reconstruction tasks such as image super-resolution and style transfer where conditional GAN framework augmented with perceptual and L2 loss function have been used to produce state-of-the-art results [75, 64].

Chapter 3

Convolutional Sparse and Low-Rank Coding-Based Rain Streak Removal

In this chapter, we introduce a novel optimization-based framework to efficiently characterize rain-streak components and background images using the learned convolutional coding prior filters. Once these priors are pre-learned, we decompose a given rainy image into separate components via newly proposed convolutional coding decomposition framework. Our method of convolutional coding de-raining is compared to several state-of-the-art methods and results on synthetic and real images show that the proposed method performs favorably compared to state-of-the-art rain streak removal algorithms.

3.1 Introduction

One of the limitations of some of the de-raining approaches such as [68, 19, 86] is that they are patch-based and they use local patches to learn local dictionaries. For example, Kang *et al* [68] divide the whole image into overlapping patches and then they learn a set of dictionary atoms in representing the overlapping patches. As a result, they often contain shifted versions of the same features [12] and the filters learned via patch-based methods are not efficient in representing the overall image. As shown in Fig. 3.1, it can be observed that patch-based sparse coding methods tend to learn shifted versions of the same features.

To deal with this issue and to make sure that the global structure information can be learned into the optimization, CSC methods have been introduced in which shift invariance is directly modeled in the objective function [163, 12, 54, 150]. Furthermore, built on the observation that rain streak components within a single image tend to share similar shapes and orientations, we propose a convolutional low-rank coding method to

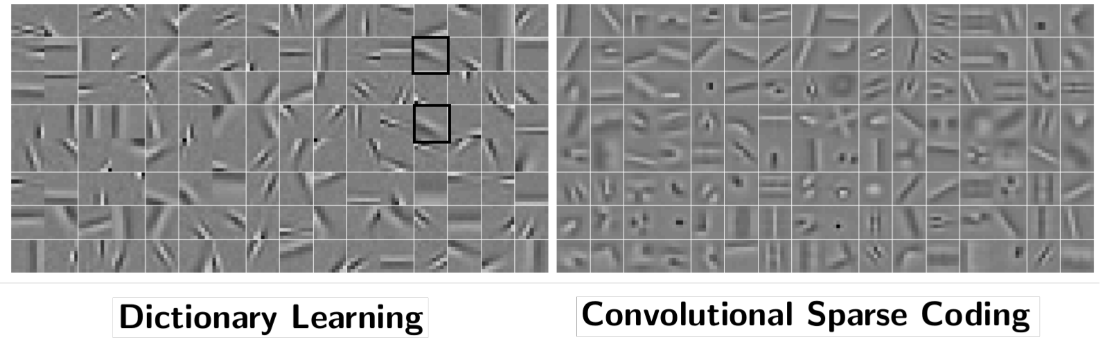


Figure 3.1: Filters learned by traditional sparse coding (dictionary learning) and convoluional sparse coding.

represent the rain streak components.

In this chapter, we present a CSC and Convolutional Low-Rank Coding (CLC) based method for rain streak removal from a single image. We first learn a set of CSC and CLC filters to efficiently represent the background image and rain streaks, respectively. Then, using the learned filters, we develop an image separation algorithm based on sparse and low-rank coding. Figure 3.2 gives an overview of the proposed Convolutional Coding based Rain Removal (CCRR) method.

In this chapter, we make the following contributions:

1. We present an optimization framework for CLC for efficiently representing low-rank rain streaks.

2. CCRR is proposed in which pre-trained CSC and CLC filters are used to efficiently represent background image and rain streaks, respectively. Using these filters, we propose an image separation method based on sparse and low-rank coding for rain streak removal.

3. We develop alternating direction method of multipliers (ADMM) based optimization frameworks [11] for solving the proposed CLC and CCRR algorithms.

3.2 Convolutional Sparse Coding

As discussed above, in CSC, given a set of M training samples $\{\mathbf{y}_m\}_{i=1}^{M}$, the objective is to learn a set of convolutional filters $\{\mathbf{d}_k\}_{i=1}^{K}$ by solving the following optimization

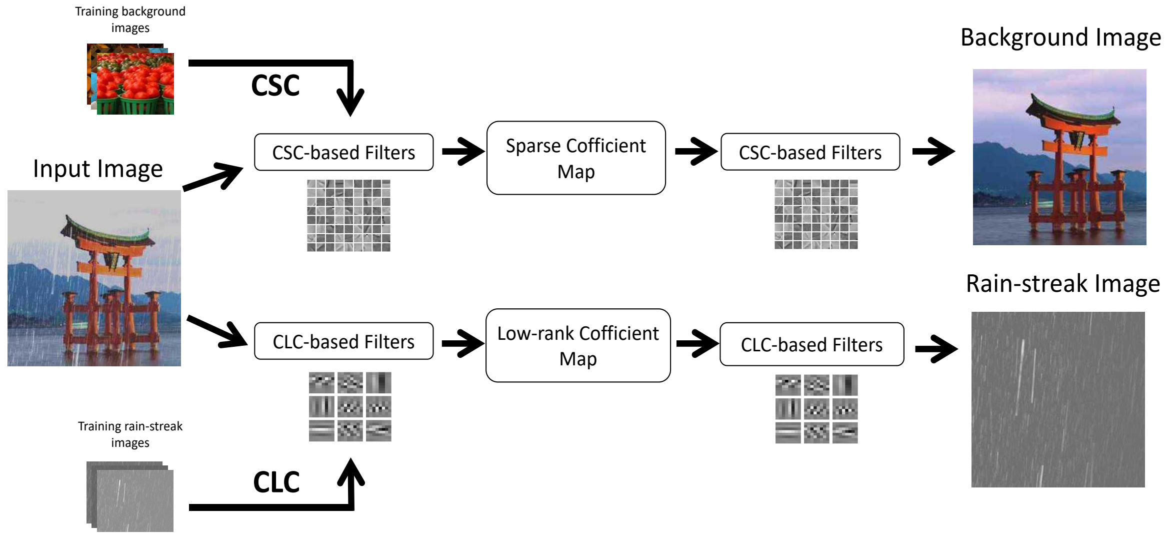


Figure 3.2: An overview of the proposed convolutional coding based rain streak removal algorithm.

problem

$$\begin{aligned} \arg\min_{\mathbf{d},\mathbf{x}} \quad & \frac{1}{2}\sum_{m=1}^{M}\left\|\mathbf{y}_m - \sum_{k=1}^{K}\mathbf{d}_k * \mathbf{x}_{m,k}\right\|_2^2 + \\ & \lambda\sum_{m=1}^{M}\sum_{k=1}^{K}\|\mathbf{x}_{m,k}\|_1 \qquad (3.1) \\ \text{subject to} \quad & \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1,\cdots,K\}, \end{aligned}$$

where $\mathbf{x}_{m,k}$ are the sparse coefficients that approximate the data $\mathbf{y}_m$ when convolved with the corresponding filters $\mathbf{d}_k$ of fixed support and for an N-dimensional vector $\mathbf{x}$, $\|\cdot\|_q$ denotes the ℓ_q-norm, $0 < q < \infty$, defined as $\|\mathbf{x}\|_q = \left(\sum_{i=1}^{N}|x_i|^q\right)^{\frac{1}{q}}$. Here, $*$ represents the 2-D convolution operator and λ is a positive regularization parameter. Several methods [149, 21] have been proposed in the literature in solving 3.1. In this chapter, we adapt the method proposed in [150] for learning the convolutional filters due to its simplicity and efficiency.

3.3 Convolutional Low-rank Coding

A rainy scene usually contains similar patterns of rain streak in different local patches. As shown in Fig 3.4, the rain streak in different patch within single rainy image share similar shapes, scales and orientations. Hence, the appearance of rain streaks can be characterized as patch-rank property. Fig 3.3 gives an example of how patch-rank works. Even though patch-rank is able to better characterize the rain streak components within a single image, yet it can be also capture repetitive texture patterns present in the rainy images, as shown in Fig 3.3 and Fig 3.4. This may result in the lost of texture details such as the brick on the wall or the texture on the clothes in the recovered de-rained image.

To overcome the texture loss issue in the recovered de-rained image, a 'supervised' low-rank model is proposed in this chapter, where we aim to enforce the learned low-rank filters are able to 'only' efficiently represent the rain streak components.

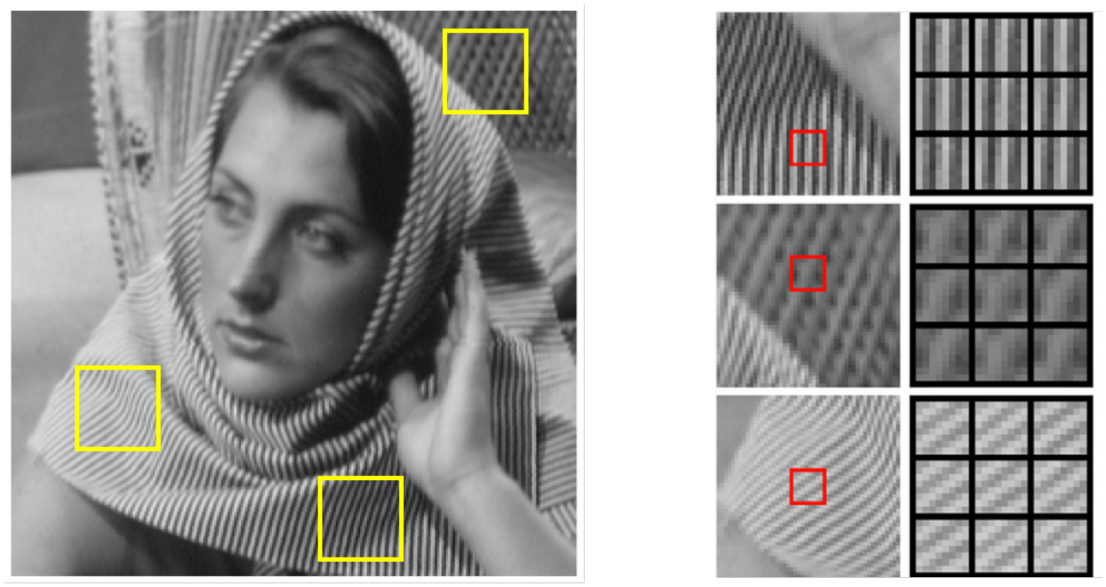


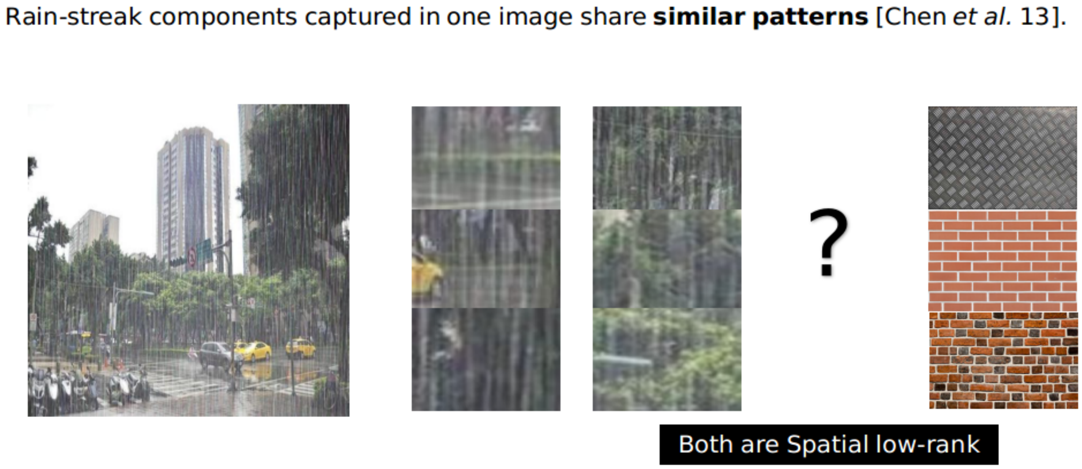
Figure 3.3: An ilustration of patch-rank.

In order to learn a set of such 'supervised' low-rank filters for efficiently representing rain streaks, we propose the following CLC problem

$$\begin{gathered} \arg\min_{\mathbf{d},\mathbf{x}} \quad \frac{1}{2}\sum_{m=1}^{M}\left\|\mathbf{y}_m - \sum_{k=1}^{K}\mathbf{d}_k * \mathbf{x}_{m,k}\right\|_2^2 + \\ \lambda_l \sum_{m=1}^{M}\sum_{k=1}^{K}\|\mathbf{x}_{m,k}\|_* \\ \text{subject to} \quad \|\mathbf{d}_k\|_2^2 \leq 1 \quad \forall k \in \{1, \cdots, K\}, \end{gathered} \tag{3.2}$$

where $\|\cdot\|_*$ is nuclear norm, representing the sum of singular values and λ_l is a positive regularization parameter.

The problem (3.2) can be solved by iteratively updating $\mathbf{d}_k$ and $\mathbf{x}_{m,k}$, as it is a bi-convex optimization problem. The updating procedure is as follows:



Rain-streak components captured in one image share **similar patterns** [Chen *et al.* 13].

Both are Spatial low-rank

Figure 3.4: It can be observed that the rain-streak components can be characterized by the patch-rank. However, it can be also observed that brick can be also characterized by the patch-rank.

3.3.1 Fix $\mathbf{x}_{m,k}$ and update $\mathbf{d}_k$

We solve the following optimization problem for updating each filter

$$\begin{aligned} &\underset{\mathbf{d}_k}{\arg\min} \quad \frac{1}{2}\sum_{m=1}^{M} \|\mathbf{y}_m - \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{x}_{m,k}\|_2^2 \\ &\text{subject to} \quad \|\mathbf{d}_k\|_2 \leq 1, \quad \forall k. \end{aligned} \tag{3.3}$$

We can regard the constrains $\|\mathbf{d}_k\|_2 \leq 1$ as post-processing after each iteration. Then, (3.3) can be rewritten as

$$\underset{\mathbf{d}_k}{\arg\min} \quad \frac{1}{2}\sum_{m=1}^{M} \|\mathbf{y}_m - \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{x}_{m,k}\|_2^2. \tag{3.4}$$

To solve (3.4) in the DFT domain, we zero pad $\mathbf{d}_k$ so that it has the same spatial support as $\mathbf{x}_{m,k}$. We form another optimization problem (3.5) that can directly include the zero-padding and normalization procedure for $\mathbf{d}_k$ in the objective [150] as

$$\begin{aligned} &\underset{\mathbf{d}_k,\mathbf{g}_k}{\arg\min} \quad \frac{1}{2}\sum_{m=1}^{M} \|\mathbf{y}_m - \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{x}_{m,k}\|_2^2 + \sum_{k=1}^{K} l_{C_{zp}}(\mathbf{g}_k) \\ &\text{subject to} \quad \mathbf{d}_k - \mathbf{g}_k = 0 \quad \forall k, \end{aligned} \tag{3.5}$$

where $l_{C_{zp}}$ is the indicator function of the constraint set C_{zp} [1]. The iterative update methods for solving (3.5) via scaled form of ADMM are as follows

$$\begin{aligned}\mathbf{d}_k{}^{(j+1)} = &\arg\min_{\mathbf{d}_k} \quad \frac{1}{2}\sum_{m=1}^{M}\|\mathbf{y}_m - \sum_{k=1}^{K}\mathbf{d}_k * \mathbf{x}_{m,k}\|_2^2 \\ &+\frac{\sigma}{2}\sum_{k=1}^{K}\|\mathbf{d}_k - \mathbf{g}_k^{(j)} + \mathbf{q}_k^{(j)}\|_2^2,\end{aligned} \tag{3.7}$$

$$\begin{aligned}\mathbf{g}_k{}^{(j+1)} = &\arg\min_{\mathbf{g}_k} \quad \sum_{k=1}^{K} l_{C_{zp}}(\mathbf{g}_k) \\ &+\frac{\sigma}{2}\sum_{k=1}^{K}\|\mathbf{d}_k^{(j+1)} - \mathbf{g}_k + \mathbf{q}_k^{(j)}\|_2^2,\end{aligned} \tag{3.8}$$

$$\mathbf{q}_k{}^{(j+1)} = \mathbf{q}_k^{(j)} + \mathbf{d}_k^{(j+1)} - \mathbf{g}_k^{(j+1)}, \tag{3.9}$$

where $\mathbf{q}$ is the scaled dual variable. The optimization problem (3.7) can be solved using the DFT based method proposed in [150], and (3.8) can be solved using a proximal algorithm [98].

3.3.2 Fix $\mathbf{d}_k$ and update $\mathbf{x}_{m,k}$

We rewrite (3.2) as

$$\begin{aligned}\arg\min_{\mathbf{x}_{m,k},\mathbf{z}_{m,k}} \quad &\frac{1}{2}\sum_{m=1}^{M}\|\mathbf{y}_m - \sum_{k=1}^{K}\mathbf{d}_k * \mathbf{x}_{m,k}\|_2^2 \\ &+ \lambda_l \sum_{m=1}^{M}\sum_{k=1}^{K}\|\mathbf{z}_{m,k}\|_* \\ \text{subject to} \quad &\mathbf{x}_{m,k} - \mathbf{z}_{m,k} = 0, \ \ \forall k.\end{aligned} \tag{3.10}$$

[1] $l_C()$ is defined as

$$l_C(p) = \begin{cases} 0, & \text{if} \quad p \in C \\ \infty, & \text{if} \quad p \notin C. \end{cases} \tag{3.6}$$

Then, the iterative update rules for solving (3.10) are as follows

$$\begin{aligned}\mathbf{x}_{m,k}{}^{(j+1)} = \arg\min_{\mathbf{x}_{m,k}} \quad & \frac{1}{2}\|\mathbf{y}_m - \sum_{k=1}^{K} \mathbf{d}_k * \mathbf{x}_{m,k}\|_2^2 \\ & + \frac{\rho}{2}\sum_{k=1}^{K} \|\mathbf{x}_{m,k} - \mathbf{z}_{m,k}^{(j)} + \mathbf{u}_{m,k}^{(j)}\|_2^2,\end{aligned} \tag{3.11}$$

$$\begin{aligned}\mathbf{z}_{m,k}{}^{(j+1)} = \arg\min_{\mathbf{z}_{m,k}} \quad & \lambda_l \sum_{k=1}^{K} \|\mathbf{z}_{m,k}\|_* + \\ & \frac{\rho}{2}\sum_{k=1}^{K} \|\mathbf{x}_{m,k}^{(j+1)} - \mathbf{z}_{m,k} + \mathbf{u}_{m,k}^{(j)}\|_2^2,\end{aligned} \tag{3.12}$$

$$\mathbf{u}_{m,k}{}^{(j+1)} = \mathbf{u}_{m,k}^{(j)} + \mathbf{x}_{m,k}^{(j+1)} - \mathbf{z}_{m,k}^{(j+1)}. \tag{3.13}$$

Problem (3.11) can be solved using the optimization method proposed in [150] and (3.12) can be solved using Singular Value Thresholding (SVT) [14].

3.4 Convolutional Coding-based Rain Removal

Assume that we have learned a set of convolutional sparsity based filters $\{\mathbf{d}_{c,k}\}$ using CSC to sparsely represent the clean background part and another set of convolutional low-rank based filters $\{\mathbf{d}_{r,k}\}$ using CLC to efficiently represent the rain streaks. That is, we have learned $\{\mathbf{d}_{c,k}\}_{k=1}^{K_c}$ and $\{\mathbf{d}_{r,k}\}_{k=1}^{K_r}$ such that $\mathbf{y}_c = \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k}$ and $\mathbf{y}_r = \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k}$, where $\mathbf{x}_{c,k}$ are the sparse coefficients and $\mathbf{x}_{r,k}$ are the low-rank coefficients that approximate $\mathbf{y}_c$ and $\mathbf{y}_r$ when convolved with the filters $\mathbf{d}_{c,k}$ and $\mathbf{d}_{r,k}$, respectively.

In order to separate the rain streaks and the background image from the mixture model, we need efficient representations for rainy component and the background image. Since rain streaks are texture like, they are inherently low-rank in nature. In fact, this assumption has been used in [19] for de-raining. Then, we propose to estimate the clean background and rain components via $\mathbf{x}_{c,k}$ and $\mathbf{x}_{r,k}$, respectively by solving the following

CCRR optimization problem

$$\begin{aligned}\hat{\mathbf{x}}_{c,k}, \hat{\mathbf{x}}_{r,k} = \arg \min_{\mathbf{x}_{c,k}, \mathbf{x}_{r,k}} & \\ & \frac{1}{2} \left\| \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} - \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k} \right\|_2^2 \\ & + \lambda_c \sum_{k=1}^{K_c} \|\mathbf{x}_{c,k}\|_1 + \lambda_r \sum_{k=1}^{K_r} \|\mathbf{x}_{r,k}\|_* \\ & + \beta TV \left(\sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} \right), \end{aligned} \tag{3.14}$$

where β, λ_r and λ_c are positive regularization parameters and TV is the total variation (i.e. sum of the absolute variations in the image). Note that in CCRR, we enforce sparsity constraint on the coefficients corresponding to the background image and low-rank constraint on the coefficients corresponding to the rain streaks. Once, $\mathbf{x}_{c,k}$ and $\mathbf{x}_{r,k}$ are estimated, the two components can be obtained by $\hat{\mathbf{y}}_c = \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \hat{\mathbf{x}}_{c,k}$ and $\hat{\mathbf{y}}_r = \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \hat{\mathbf{x}}_{r,k}$, where $\hat{\mathbf{y}}_c$ represents the de-rained image.

3.5 Optimization

In this section, we derive the framework for solving the proposed CLC and CCRR optimization problems. If we discard the TV part in (3.14), then the resulting optimization problem can be solved iteratively over $\mathbf{x}_{c,k}$ and $\mathbf{x}_{r,k}$.

3.5.1 Update step for $\mathbf{x}_{c,k}$

When $\mathbf{x}_{r,k}$ is fixed, we need to solve the following problem to obtain the sparse coefficients $\mathbf{x}_{c,k}$

$$\begin{aligned}\hat{\mathbf{x}}_{c,k} = \arg \min_{\mathbf{x}_{c,k}} \frac{1}{2} \left\| \mathbf{y} - \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} \right\|_2^2 & \\ + \lambda_c \sum_{k=1}^{K_c} \|\mathbf{x}_{c,k}\|_1 . & \end{aligned} \tag{3.15}$$

This problem can be solved using the DFT based ADMM method [150].

3.5.2 Update step for $\mathbf{x}_{r,k}$

For a fixed $\mathbf{x}_{c,k}$, we have to solve the following problem to obtain $\mathbf{x}_{r,k}$

$$\begin{aligned}\hat{\mathbf{x}}_{r,k} = \arg\min_{\mathbf{x}_{r,k}} \frac{1}{2} \left\| \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k} - \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k} \right\|_2^2 \\ + \lambda_r \sum_{k=1}^{K_r} \|\mathbf{x}_{r,k}\|_* . \end{aligned} \tag{3.16}$$

This problem is very similar to the sub-problem that we solve in CLC for finding the low-rank coefficients when $\mathbf{d}_k$ are fixed. Let $\mathbf{y}_p = \mathbf{y} - \sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \mathbf{x}_{c,k}$. Then (3.16) can be rewritten as

$$\hat{\mathbf{x}}_{r,k} = \arg\min_{\mathbf{x}_{r,k}} \frac{1}{2} \left\| \mathbf{y}_p - \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \mathbf{x}_{r,k} \right\|_2^2 + \lambda_r \sum_{k=1}^{K_r} \|\mathbf{x}_{r,k}\|_* ,$$

which can be solved using the optimization procedure described in the previous subsection for CLC.

Finally, the TV correction is applied only on the background rain-free part to control the edges in the clear image. The overall CCRR algorithm for rain streak removal is summarized in Algorithm 1, where L is the total iteration number and i is the iteration index.

```
**Algorithm 1** The CCRR Algorithm for Rain Removal.
**Input**: ${\{\mathbf{d}_{c,k}\}}_{k=1}^{K_c}$, ${\{\mathbf{d}_{r,k}\}}_{k=1}^{K_r}$, $\mathbf{y}$, $\lambda_c$, $\lambda_r$, $L$
Initialization
**for** $i = 1 : L$
    Obtain $\mathbf{x}_{c,k}$ by solving (3.14) when fixing $\mathbf{x}_{r,k}$.
    Obtain $\mathbf{y}_c$ by applying the TV correction [132].
    Use $\mathbf{y}_c$ to replace $\sum_{k=1}^{K_c} \mathbf{d}_{c,k} * \hat{\mathbf{x}}_{c,k}$ in (3.14).
    Obtain $\hat{\mathbf{x}}_{r,k}$ by solving (3.14) when fixing $\mathbf{x}_{c,k}$.
**end for**
$\hat{\mathbf{y}}_r = \sum_{k=1}^{K_r} \mathbf{d}_{r,k} * \hat{\mathbf{x}}_{r,k}$,
$\hat{\mathbf{y}}_c$=$\mathbf{y}$ - $\hat{\mathbf{y}}_r$;
**Output**: $\hat{\mathbf{y}}_c$, $\hat{\mathbf{y}}_r$
```

3.6 Experimental Results

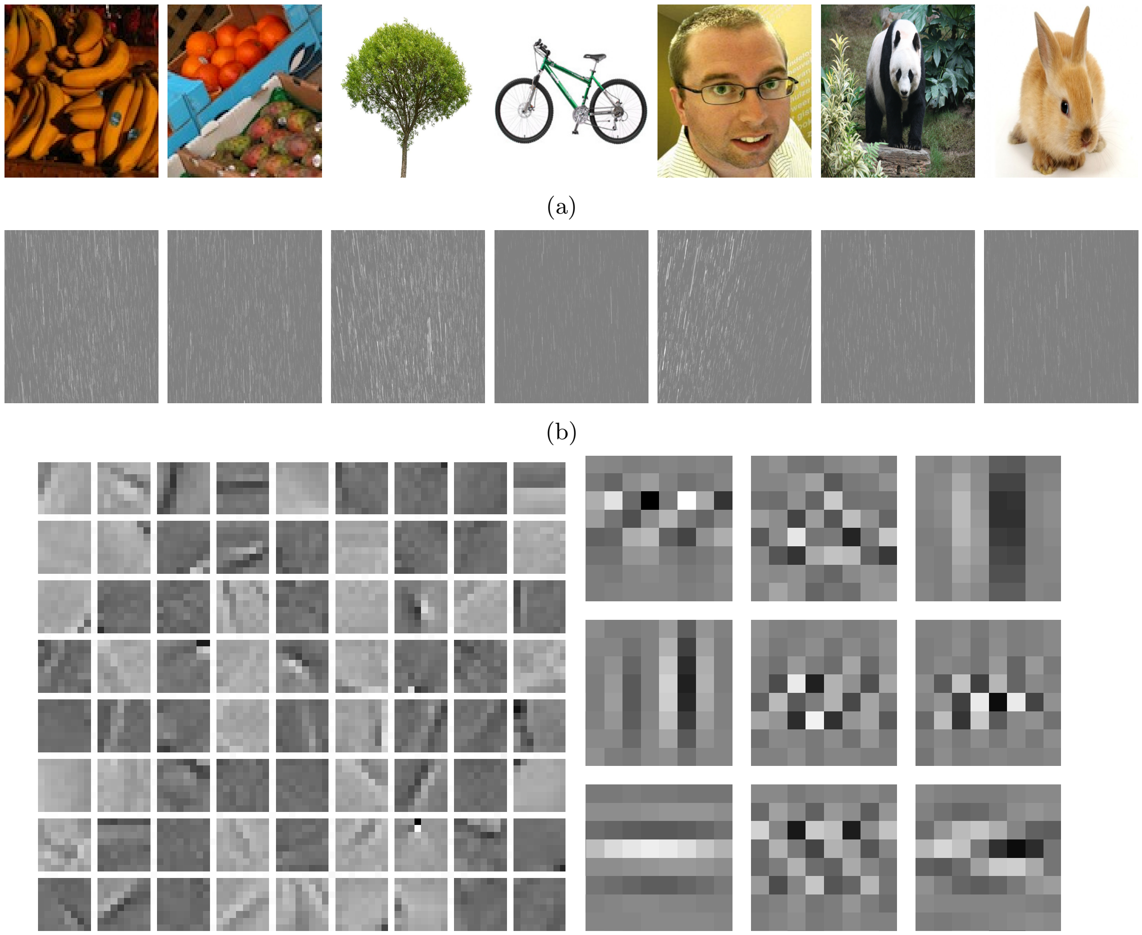
In this section, we present the results of our proposed CCRR algorithm for single image de-raining on both gray-scale and color images. We compare the performance of our method with that of four state-of-the-art single image de-raining methods - sparse dictionary based method (Auto-SP) [68], discriminative sparse coding based method (Dis-SP) [86], low-rank representation based method (Low-rank) [19] and a CNN-based method (CNN) [36]. In these experiments, we use Peak Signal to Noise Ratio (PSNR) and Structural Similarity Index (SSIM) [145] to measure the performance of the routines tested.

Sample training images shown in Figure 3.5 (a) are used to learn the convolutional sparse filters $\{\mathbf{d}_{c,k}\}$ using CSC. Similarly, some training images in the Figure 3.5 (b) are utilized to learn the convolutional low-rank filters $\{\mathbf{d}_{r,k}\}$ using CLC. The corresponding CSC and CLC learned filters are shown in Figure 3.5 (c) and (d), respectively. The size of each CSC filter is set equal to 8×8 and the size of each CLC filter is set equal to 6×6 for all experiments. From Figure 3.5 (d), one can see that these filters are oscillatory in nature and they do a good job in capturing the rain texture patters. These filters can capture the low-rank structure found in the rain streaks. Similarly, from Figure 3.5 (c), we observe that the learned filters look similar to those found in a Gabor or curvelet dictionary. They capture domain specific information found in natural images such as edges and contours.

All testing images are excluded from the training procedure. For the gray-scale images, the parameters λ_c and λ_r are set equal to $\max(0.55 - 0.090 * i, 0.001)$ and $\max(5.20 - 0.90 * i, 0.05)$, respectively. For the color images, the parameters are set as $\lambda_c = \max(1.35 - 0.435 * i, 0.001)$ and $\lambda_r = \max(5.30 - 0.72 * i, 0.82)$. The total iteration number L is set equal to to 6 for all experiments.

3.6.1 Rain Removal from Gray-scale Images

In the first set of experiments, we evaluate the quantitative performance of different methods on the two synthetic gray-scale images released by Kang *et al.* in [68]. These synthetic rainy images are shown in Figure 3.7. The performance of different de-raining methods in terms of PSNR and SSIM is tabulated in Table 3.1. As can be seen from this table, on average our method performs favorably over some of the compared methods.



(a)

(b)

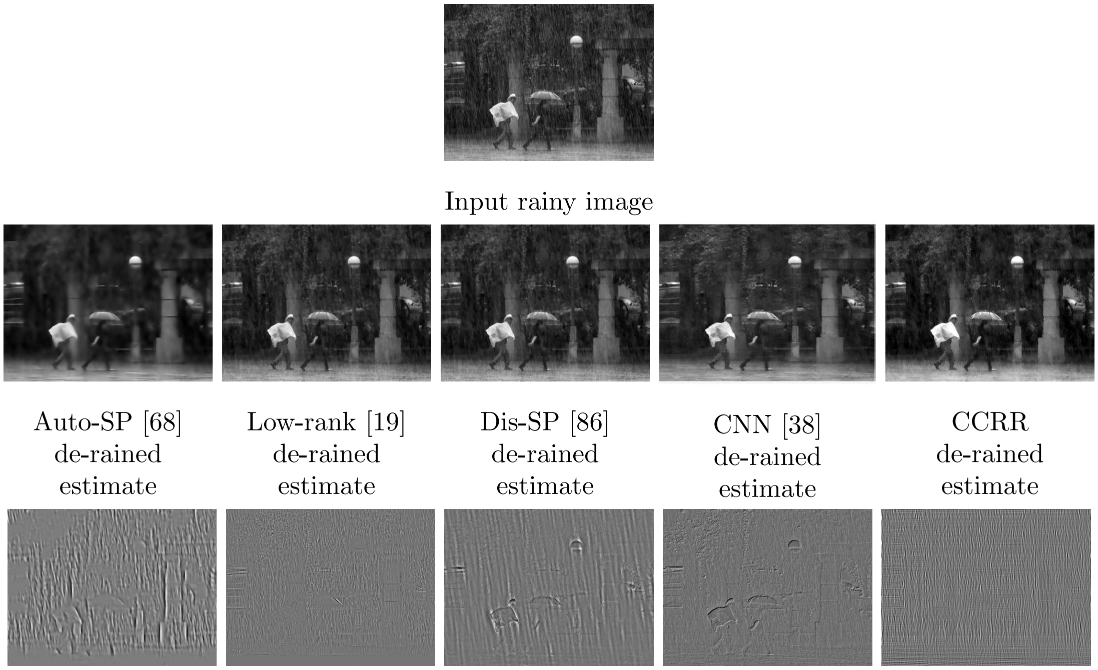
(c)

(d)

Figure 3.5: (a) Training non-rain images used for learning a set of sparsity based filters $\{\mathbf{d}_{c,k}\}$. (b) Rain-streak images used for learning a set of low-rank filters $\{\mathbf{d}_{r,k}\}$. (c) Learned non-rain filters $\{\mathbf{d}_{c,k}\}$. (d) Learned rain-streak filters $\{\mathbf{d}_{r,k}\}$.

		Rainy	CCRR	Auto-SP [68]	Low-rank [19]	Dis-sp [86]	CNN [38]
SSIM	Fig 3.7(a)	0.6602	0.7699	0.7410	0.7641	0.6738	**0.7751**
	Fig 3.7(b)	0.8579	**0.8939**	0.8654	0.8905	0.8774	0.8510
PSNR (dB)	Fig 3.7(a)	24.75	**25.78**	24.78	25.42	25.61	24.82
	Fig 3.7(b)	24.44	**25.17**	24.49	24.58	25.01	23.62

Table 3.1: Results on two synthetic gray-scale images.



Input rainy image

Auto-SP [68] de-rained estimate

Low-rank [19] de-rained estimate

Dis-SP [86] de-rained estimate

CNN [38] de-rained estimate

CCRR de-rained estimate

Auto-SP [68] rain estimate

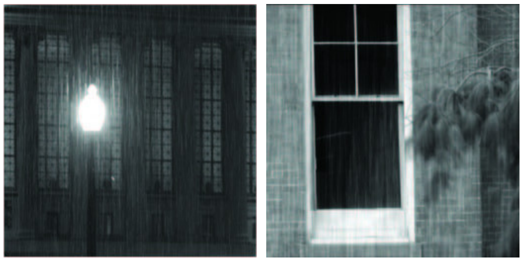
Low-rank [19] rain estimate

Dis-SP [86] rain estimate

CNN [38] rain estimate

CCRR rain estimate

Figure 3.6: Rain-streak removal results on a real gray-scale image.



(a)

(b)

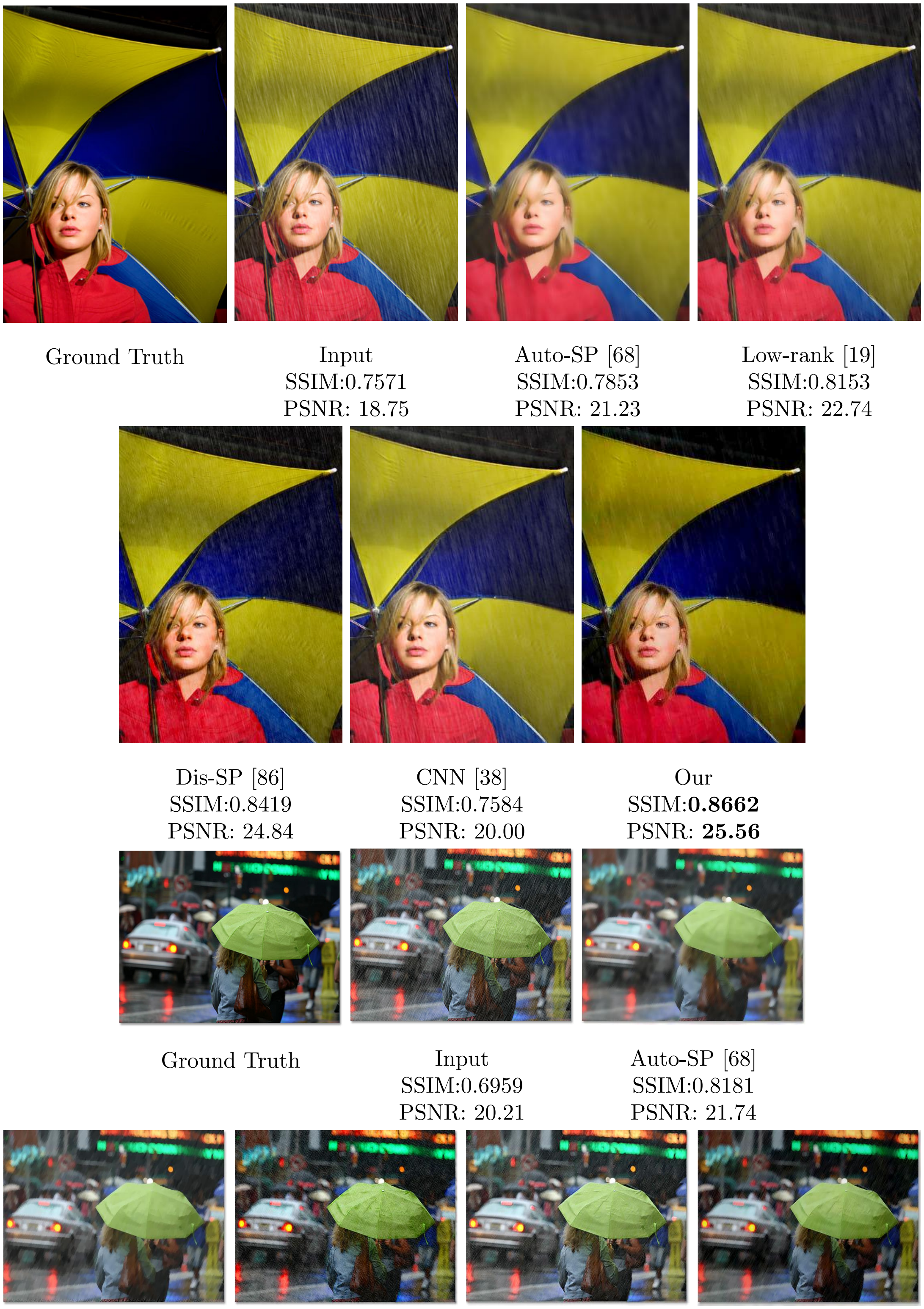
Figure 3.7: Synthetic gray-scale rainy images.

In the second set of experiments with the gray-scale images, we use a real rainy gray-scale image (shown in the first row of Figure 3.6) and visually inspect the performance of different de-raining methods by displaying the separated clear background images (shown in the second row of Figure 3.6) and rain streak images (shown in the third row of Figure 3.6) corresponding to different methods. From the third row, we can observe that our method can capture more rain streaks and less of other structures, demonstrating the advantage of using convolutional low-rank filters over just using sparsity as prior for the rain component. We also observe that the recovered rain-streak components from Dis-SP, Auto-SP and CNN methods capture more information corresponding to the background non-rain image.

3.6.2 Rain Removal from Color Images

Synthetic Images. We used two color rainy images released by [68] with its ground truth to measure the de-raining performance of different methods. The results are shown in Figure 3.8. It can be observed that our method outperforms all the other three methods quantitatively as well as qualitatively. For example, from this figure, we see that the Auto-SP method [68] tends to smooth the de-rained part when removing the rain component, while the low-rank method [19] fails to capture some rain components in the background image. Even though the Dis-SP and CNN methods have a very competitive visual quality, some rainy components still remain in the de-rained image for both of these methods. Furthermore, Dis-SP tend to enhance the contrast of some details such as the face shown in the first row in Figure 3.8 and the CNN based rain removal method has very poor quantitative performance. Similar visual and quantitative results are also observed in the second and the third row of Figure 3.8.

Real Images. We also evaluated the performance of our proposed method on many real images downloaded from the Internet. The de-rained results for all the methods and their corresponding input rainy images are shown in Figure 3.9. The first row shows the input rainy images. Results of Auto-SP [68], Low-rank based method [19], Dis-SP [86], CNN based method [38] and our CCRR method are shown in the second to fourth rows, respectively. From these de-rained images, we observe that the Auto-SP and Low-rank methods tend to smooth the details in the de-rained images even though they can remove a lot of rain streaks. This can be seen by observing the head part of the athletes in the third column of this figure. The de-rained images from the Dis-SP method still contains a lot of rain streaks. In general, the CNN based method achieves very good visual quality, however, it fails to tackle heavy rain conditions, as can be seen by comparing the results in the second and last columns of Figure 3.9. Our proposed CCRR method can preserve the details such as edges and contours while removing the low-rank rain streaks. This experiment clearly demonstrates the significance of the proposed method in removing rain streaks from real-world rainy images under a variety of different background and conditions.



Ground Truth

Input
SSIM:0.7571
PSNR: 18.75

Auto-SP [68]
SSIM:0.7853
PSNR: 21.23

Low-rank [19]
SSIM:0.8153
PSNR: 22.74

Dis-SP [86]
SSIM:0.8419
PSNR: 24.84

CNN [38]
SSIM:0.7584
PSNR: 20.00

Our
SSIM:**0.8662**
PSNR: **25.56**

Ground Truth

Input
SSIM:0.6959
PSNR: 20.21

Auto-SP [68]
SSIM:0.8181
PSNR: 21.74

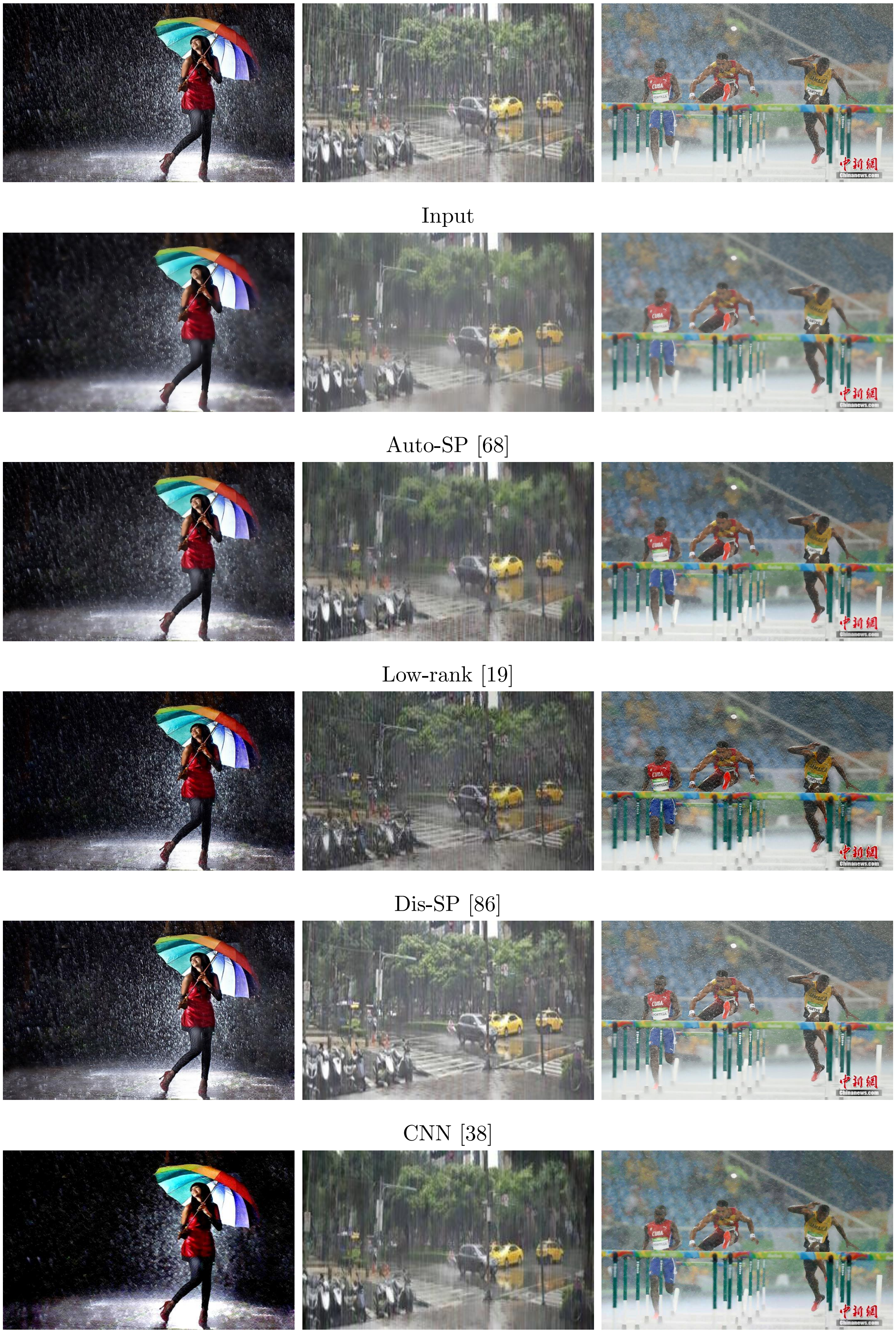
Low-rank [19]
SSIM:0.8215
PSNR: 21.87

Dis-SP [86]
SSIM:0.7915
PSNR: 24.15

CNN [38]
SSIM:0.7326
PSNR: 21.29

Our
SSIM:**0.8526**
PSNR: **26.05**

Figure 3.8: Rain-streak removal results on two synthetic color images. We compare the performance of our proposed CCRR method with other four methods: Auto-SP [68], Low-rank [19], Dis-SP [86] and CNN [38].



Input

Auto-SP [68]

Low-rank [19]

Dis-SP [86]

CNN [38]

Our

Figure 3.9: Rain-streak removal results on three real images. We compare the performance of our proposed CCRR method with the other three methods: Auto-SP [68], Low-rank [20], Dis-SP [86] and CNN [38].

Chapter 4

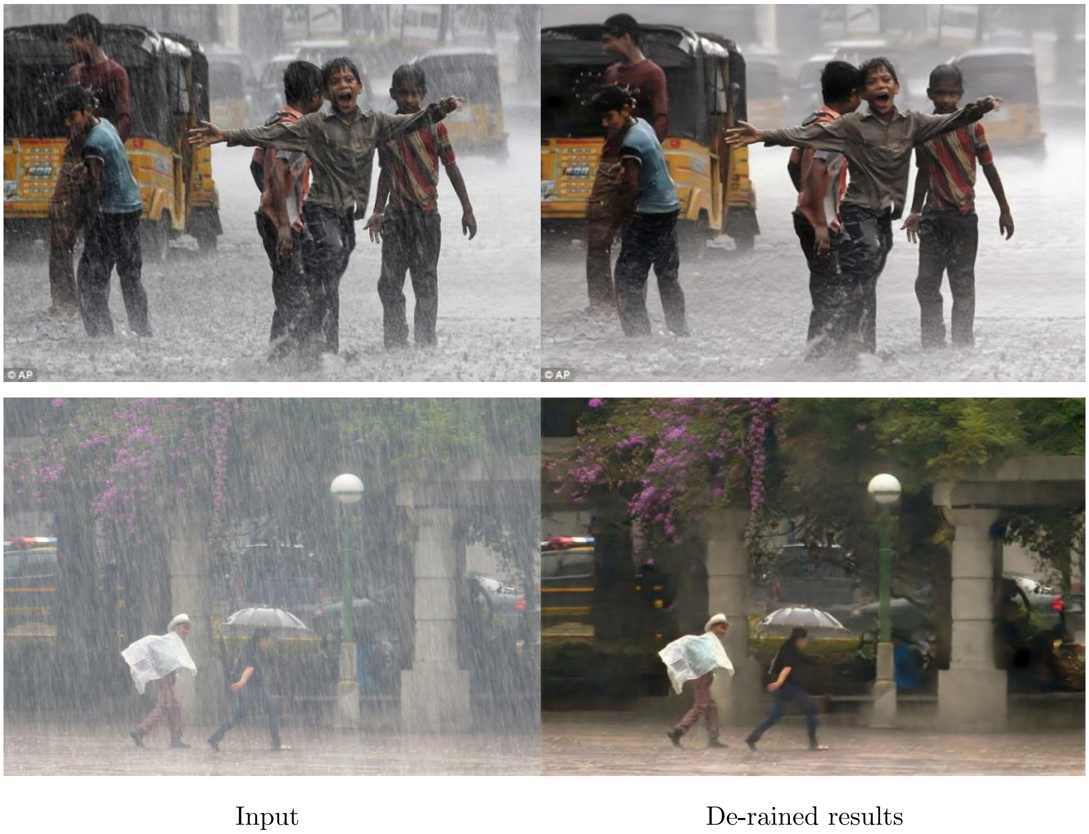
Image De-raining Using a Conditional Generative Adversarial Network

Even though prior-based methods such as sparsity prior have demonstrated their effectiveness in characterizing the clean background and rain-streak components separately, estimated clean background images tend to lose many details and results are far from optimal. To more efficiently learn the mapping between input rainy images and the corresponding clean background images, we tend to explore the strong capabilities of using CNNs in addressing the single image de-raining problem. We have proposed a conditional GAN with refined perceptual loss to include the discriminative information with perceptual quality into the learning process. In addition, a multi-scale discriminator is proposed in this method to leverage information from multiple scales to decide whether the de-rained image is real or fake. Furthermore, we synthesize a larges-scale datasets for single image de-raining problem, which has been made publicly available for future research. This method outperforms traditional prior-based methods and deep learning methods both quantitatively and qualitatively.

4.1 Introduction

Even though tremendous improvements have been achieved fpr single image de-raining problem, as discussed in Chapter 2, we note that these methods do not consider additional information into the optimization. Hence, to design a visually appealing de-raining algorithm, we must consider the following information into the optimization framework:

(a) The criterion that performance of vision algorithms such as detection and classification should not be affected by the presence of rain streaks should be considered



Input De-rained results

Figure 4.1: Sample results of the proposed ID-CGAN method for single image de-raining.

in the objective function. The inclusion of this discriminative information ensures that the reconstructed image is indistinguishable from its original counterpart.

(b) Rather than concentrating only on the characterization of rain-streaks, visual quality may also be considered into the optimization function. This can ensure that the de-rained image looks visually appealing without losing important details.

(c) Some of the existing methods adopt off-line additional image processing techniques to enhance the results [38, 68]. Instead, it would be better to use a more unified structure to deal with the problem without any additional processing.

In this work, these criteria are incorporated in a novel conditional GAN-based framework called Image De-raining Conditional Generative Adversarial Network (ID-CGAN) to address the single image de-raining problem. We aim to leverage the generative modeling capabilities of the recently introduced CGANs. While existing CNN-based approaches minimize only L2 error, these methods need additional regularization due to the ill-posed nature of the problem. In this work, adversarial loss from CGANs is used as additional regularizer leading to superior results in terms of visual quality and quantitative performance. The use of discriminator for classifying between real/fake samples provides additional feedback, enabling the generator to produce results that are visually similar to the ground-truth clean samples (real samples). Inspired by the recent success of GANs for pixel-level vision tasks such as image generation [107], image inpainting [100] and image super-resolution [75], our network consists of two sub-networks: densely-connected generator (G) and multi-scale discriminator (D). The generator acts as a mapping function to translate an input rainy image to de-rained image such that it fools the discriminator, which is trained to distinguish rainy images from images without rain. The discriminator is designed to capture hierarchical context information through multi-scale pooling. However, traditional GANs [107] are not stable to train and may introduce artifacts in the output image making it visually unpleasant and artificial. To address this issue, we introduce a new refined perceptual loss to serve as an additional loss function to aid the proposed network in generating visually pleasing outputs. Furthermore, to leverage different scale information in determining whether the corresponding de-rained image is real or fake, a multi-scale discriminator is proposed. Sample results of the proposed ID-CGAN algorithm are shown in Figure 4.1. In summary, this chapter makes the following contributions:

1. A conditional GAN-based framework to address the challenging single image de-raining problem without the use of any additional post-processing.

2. A densely-connected generator sub-network that is specifically designed for the single image de-raining task.

3. A multi-scale discriminator is proposed to leverage both local and global information to determine whether the corresponding de-rained image is real or fake.

4. Extensive experiments are conducted on publicly available and synthesized datasets to demonstrate the effectiveness of the proposed method in terms of visual quality and quantitative performance. Detailed qualitative and quantitative comparisons with existing state-of-the-art methods are presented.

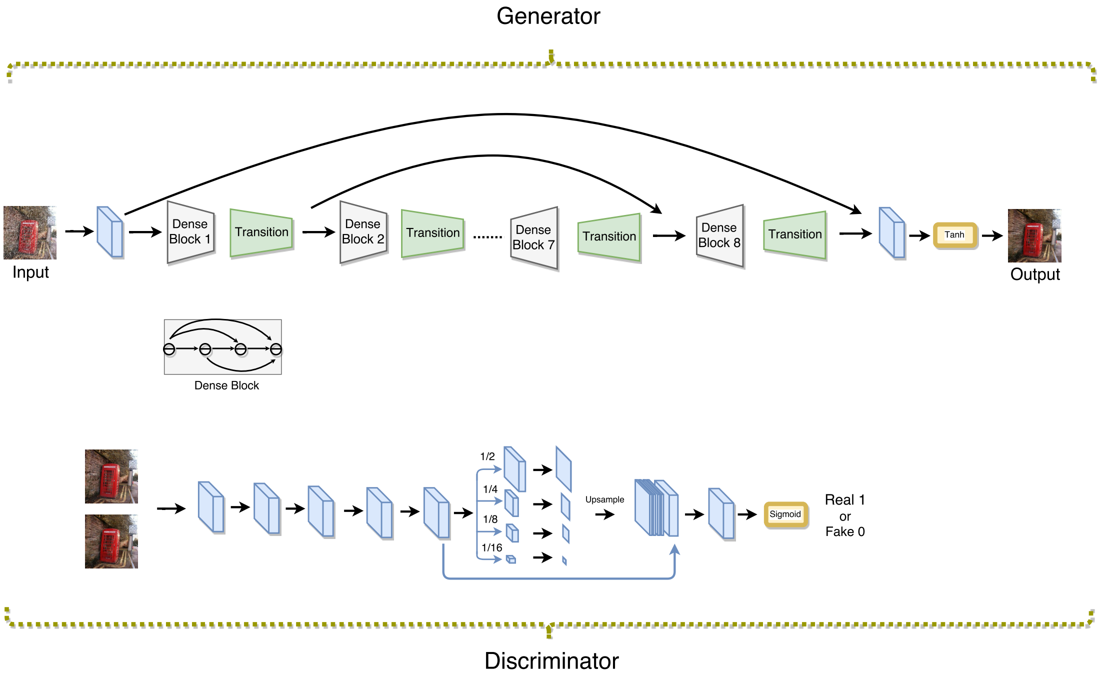


Figure 4.2: An overview of the proposed ID-CGAN method for single image de-raining. The network consists of two sub-networks: densely-connected generator G and multi-scale discriminator D.

5. Lastly, effectiveness of the proposed method in improving high-level object detection task is demonstrated on VOC dataset [32]. The detections are performed using Faster-RCNN [111].

	No addition pre- (or post) processing	End-to-end mapping	Consider discriminative performance in the optimization	Consider visual performance in the optimization	Not Patch-based	Time efficiency
SPM [68]						
PRM [19]	√					
DSC [86]	√					
CNN [38]		√			√	√
GMM [83]	√				√	√
CCR [169]	√				√	
DDN [37]		√			√	√
JORDER [158]	√	√			√	√
ID-CGAN	√	√	√	√	√	√

Table 4.1: Compared to the existing methods, our ID-CGAN has several desirable properties: 1. No additional image processing. 2. Include discriminative factor into optimization. 3. Consider visual performance into optimization.

4.2 Proposed Method

Instead of solving (2.1) in a decomposition framework, we aim to directly learn a mapping from an input rainy image to a de-rained (background) image by constructing a conditional GAN-based deep network called ID-CGAN. The proposed network is composed of three important parts (generator, discriminator and perceptual loss function) that serve distinct purposes. Similar to traditional GANs [64, 47], the proposed method contains two sub-networks: a generator sub-network G and a discriminator sub-network D. The generator sub-network G is a densely-connected symmetric deep CNN network with appropriate skip connections as shown in the top part in Figure 4.2. Its primary goal is to synthesize a de-rained image from an image that is degraded by rain (input rainy image). The multi-scale discriminator sub-network D, as shown in the bottom part in Figure 4.2, serves to distinguish 'fake' de-rained image (synthesized by the generator) from corresponding ground truth 'real' image. It can also be viewed as a guidance for the generator G. Since GANs are known to be unstable to train which results in artifacts in the output image synthesized by G, we define a refined perceptual loss functions to address this issue. Additionally, this new refined loss function ensures that the generated (de-rained) images are visually appealing.

4.2.1 Generator with Symmetric Structure

As the goal of single image de-raining is to generate pixel-level de-rained image, the generator should be able to remove rain streaks as much as possible without loosing any detail information of the background image. So the key part lies in designing a good structure to generate de-rained image.

Existing methods for solving (2.1), such as sparse coding-based methods [68, 10, 131, 106], neural network-based methods [152] and CNN-based methods [90] have all adopted a symmetric (encoding-decoding) structure. For example, sparse coding-based methods use a learned or pre-defined synthesis dictionaries to decode the input noisy image into sparse coefficient map. Then another set of analysis dictionaries are used to transfer the coefficients to desired clear output. Usually, the input rainy image is transferred to a specific domain for effective separation of background image and undesired component (rain-streak). After separation, the background image (in the new domain) has to be transferred back to the original domain which requires the use of a symmetric process.

Following these methods, a symmetric structure is adopted to form the generator sub-network. The generator G directly learns an end-to-end mapping from input rainy image to its corresponding ground truth. In contrast to the existing adversarial networks for image-to-image translation that use U-Net [119, 64] or ResNet blocks [52, 75] in their generators, we use the recently introduced densely connected blocks [59]. These dense blocks enable strong gradient flow and result in improved parameter efficiency. Furthermore, we introduce skip connections across the dense blocks to efficiently leverage features from different levels and guarantee better convergence. The jth dense block $\mathbf{D}_j$ is represented as:

$$\mathbf{D}_j = cat[D_{j,1}, D_{j,2}, ..., D_{j,6}], \tag{4.1}$$

where $D_{j,i}$ represents the features from the ith layer in dense block $\mathbf{D}_j$ and each layer in a dense block consists of three consecutive operations, batch normalization (BN), leaky rectified linear units (LReLU) and a 3×3 convolution.

Each dense block is followed by a transition block (T), functioning as up-sampling (Tu), down-sampling (Td) or no-sampling operation (Tn). To make the network efficient in training and have better convergence performance, symmetric skip connections are included into the proposed generator sub-network, similar to [90]. The generator network is as follows:

CBLP(64)-D(256)-Td(128)-D(512)-Td(256)-D(1024)-Tn(512)-D(768)-Tn(128)-D(640)-Tu(120)-D(384)-Tu(64)-D(192)-Tu(64)-D(32)-Tn(16)-C(3)-Tanh

where, $CBLP$ is a set of convolutional layers followed by batch normalization, leaky ReLU activation and pooling module, and the number inside braces indicates the number of channels for the output feature maps of each block.

4.2.2 Multi-scale Discriminator

From the point of view of a GANs framework, the goal of de-raining an input rainy image is not only to make the de-rained result visually appealing and quantitatively comparable to the ground truth, but also to ensure that the de-rained result is indistinguishable from the ground truth image. Therefore, a learned discriminator sub-network is designed to classify if each input image is real or fake. Previous methods [64, 185] have demonstrated the effectiveness of leveraging an efficient patch-discriminator in generating high quality results. For example, Isola *et al* [64] adopt a 70× 70 patch discriminator, where 70×70 indicates the receptive field of the discriminator. Though such a single scale (eg. 70× 70) patch-discriminator is able to achieve visually pleasing results, however, it is still not capable enough to capture the global context information, resulting in insufficient estimation. As shown in the zoomed-in part of the Figure 4.5 (e), it can be observed that certain tiny details are still missing in the de-rained results using a single scale discriminator. For example, it can be observed from the second row of Figure 4.5 that the front mirror of truck is largely being removed in the de-rained results. This is probably due to the fact that the receptive field size in the discriminator is 70× 70 and no additional surrounding context is provided. Hence, we argue that it is important to leverage a more powerful discriminator that captures both local and global information to decide whether it is real or fake.

To effectively address this issue, a novel multi-scale discriminator is proposed in this work. This is inspired by the usage of multi-scale features in objection detection [51] and semantic segmentation [181]. Similar to the structure that was proposed in [64], a convolutional layer with batch normalization and PReLU activation are used as a basis throughout the discriminator network. Then, a multi-scale pooling module, which pools features at different scales, is stacked at the end of the discriminator. The pooled features are then up-sampled and concatenated, followed by a 1×1 convolution and a sigmoid function to produce a probability score normalized between [0,1]. By using features at different scales, we explicitly incorporate global hierarchical context into the discriminator. The proposed discriminator sub-network D is shown in the bottom part of Figure 4.2.

4.2.3 Refined Perceptual Loss

As discussed earlier, GANs are known to be unstable to train and they may produce noisy or incomprehensible results via the guided generator. A probable reason is that the new input may not come from the same distribution of the training samples. As illustrated in Figure 4.4(c), it can be clearly observed that there are some artifacts

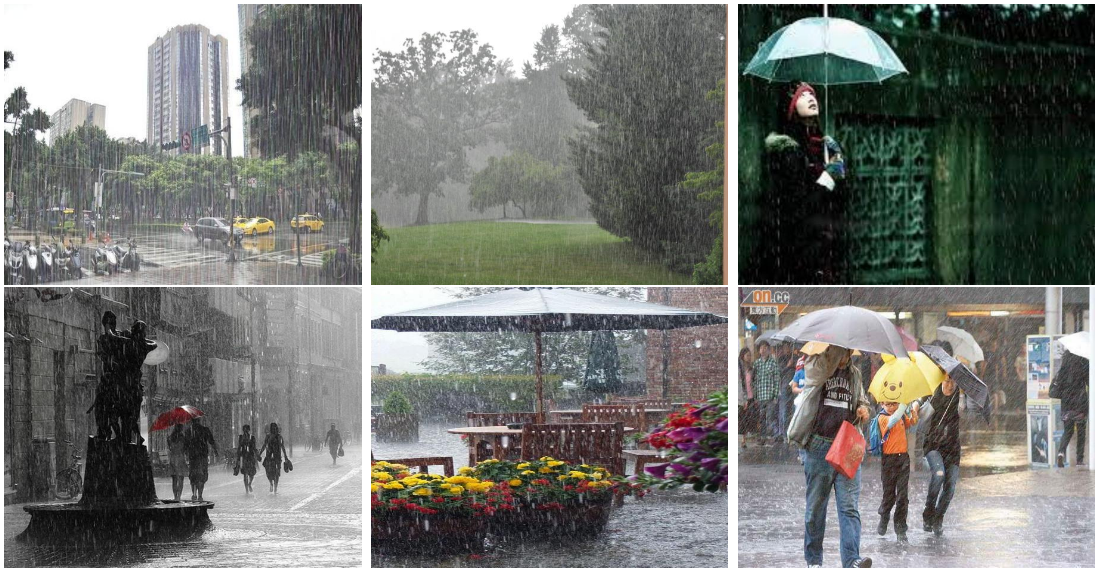


Figure 4.3: Sample images from real-world rainy dataset.

introduced by the normal GAN structure. This greatly influences the visual performance of the output image. A possible solution to address this issue is to introduce perceptual loss into the network. Recently, loss function measured on the difference of high-level feature representation, such as loss measured on certain layers in CNN [67], has demonstrated much better visual performance than the per-pixel loss used in traditional CNNs. However, in many cases it fails to preserve color and texture information [67]. Also, it does not achieve good quantitative performance simultaneously. To ensure that the results have good visual and quantitative scores along with good discriminatory performance, we propose a new refined loss function. Specifically, we combine pixel-to-pixel Euclidean loss, perceptual loss [67] and adversarial loss together with appropriate weights to form our new refined loss function. The new loss function is then defined as follows:

$$L_{RP} = L_E + \lambda_a L_A + \lambda_p L_P, \tag{4.2}$$

where L_A represents adversarial loss (loss from the discriminator D), L_P is perceptual loss and L_E is normal per-pixel loss function such as Euclidean loss. Here, λ_p and λ_a are pre-defined weights for perceptual loss and adversarial loss, respectively. If we set both λ_p and λ_a to be 0, then the network reduces to a normal CNN configuration, which aims to minimize only the Euclidean loss between output image and ground truth. If λ_p is set to 0, then the network reduces to a normal GAN. If λ_a set to 0, then the network reduces to the structure proposed in [67].

The three loss functions L_P, L_E and L_A are defined as follows. Given an image pair $\{\mathbf{x}, \mathbf{y}_b\}$ with C channels, width W and height H (i.e. $C \times W \times H$), where $\mathbf{x}$ is the input image and $\mathbf{y}_b$ is the corresponding ground truth, the per-pixel Euclidean loss is defined as:

$$L_E = \frac{1}{CWH} \sum_{c=1}^{C} \sum_{x=1}^{W} \sum_{y=1}^{H} \|\phi_E(\mathbf{x})^{c,w,h} - (\mathbf{y}_b)^{c,w,h}\|_2^2, \tag{4.3}$$

where ϕ_E is the learned network G for generating the de-rained output. Suppose the outputs of certain high-level layer are with size $C_i \times W_i \times H_i$. Similarly, the perceptual loss is defined as

$$L_P = \frac{1}{C_i W_i H_i} \sum_{c=1}^{C_i} \sum_{w=1}^{W_i} \sum_{h=1}^{H_i} \|V(\phi_E(\mathbf{x}))^{c,w,h} - V(\mathbf{y}_b)^{c,w,h}\|_2^2, \tag{4.4}$$

where V represents a non-linear CNN transformation. Similar to the idea proposed in [67], we aim to minimize the distance between high-level features. In our method, we compute the feature loss at layer relu2_2 in VGG-16 model [128].[1]

Given a set of N de-rained images generated from the generator $\{\phi_E(\mathbf{x})\}_{i=1}^{N}$, the entropy loss from the discriminator to guide the generator is defined as:

$$L_A = -\frac{1}{N} \sum_{i=1}^{N} \log(D(\phi_E(\mathbf{x}))). \tag{4.5}$$

4.3 Experiments and Results

In this section, we present details of the experiments and quality measures used to evaluate the proposed ID-CGAN method. We also discuss the dataset and training details followed by comparison of proposed methods against a set of baseline methods and recent state-of-the-art approaches.

[1] https://github.com/ruimashita/caffe-train/blob/master/vgg.train_val.prototxt

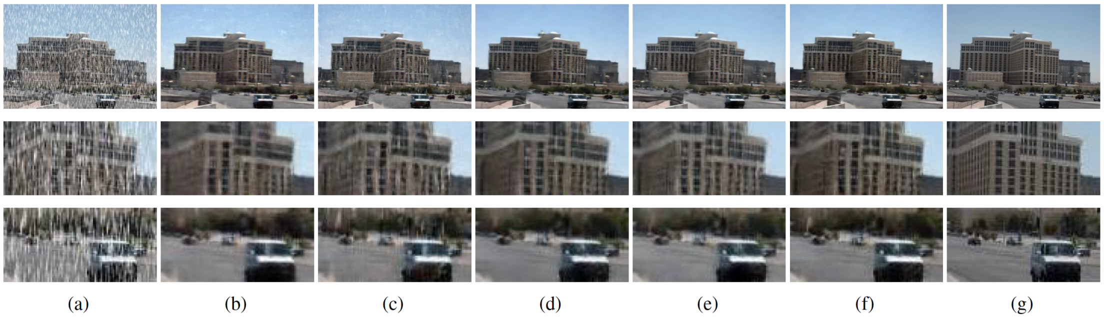


Figure 4.4: Qualitative comparisons for different baseline configurations of the proposed method. (a) Input image, (b) GEN, (c) GEN-CGAN-S, (d) GEN-P, (e) GEN-CGAN-PS, (f) ID-CGAN and (g) Target image.

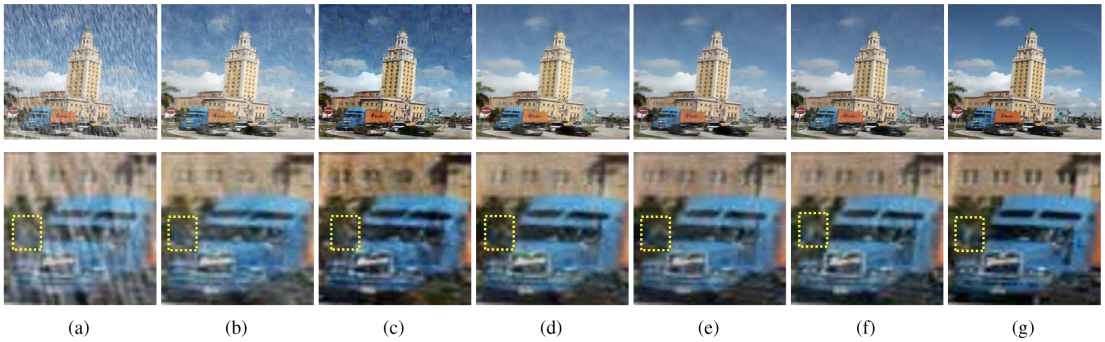


Figure 4.5: Qualitative comparisons for different baseline configurations of the proposed method. (a) Input image, (b) GEN, (c) GEN-CGAN-S, (d) GEN-P, (e) GEN-CGAN-PS, (f) ID-CGAN and (g) Target image.

4.3.1 Experimental Details

Synthetic dataset

Due to the lack of availability of large size datasets for training and evaluation of single image de-raining, we synthesized a new set of training and testing samples in our experiments. The training set consists of a total of 700 images, where 500 images are randomly chosen from the first 800 images in the UCID dataset [123] and 200 images are randomly chosen from the BSD-500's training set [3]. The test set consists of a total of 100 images, where 50 images are randomly chosen from the last 500 images in the UCID dataset and 50 images are randomly chosen from the test-set of the BSD-500 dataset [3]. After the train and test sets are created, we add rain-streaks to these images by following the guidelines mentioned in [38] using Photoshop[2]. It is ensured that rain pixels of different intensities and orientations are added to generate a diverse training and test set. Note that the images with rain form the set of observed images and the corresponding clean images form the set of ground truth images. All the training and test samples are resized to 256×256. All the images are available in https://github.com/hezhangsprinter/ID-CGAN.

Real-world rainy images dataset

In order to demonstrate the effectiveness of the proposed method on real-world data, we created a dataset of 50 rainy images downloaded from the Internet. While creating this dataset, we took all possible care to ensure that the images collected were diverse in terms of content as well as intensity and orientation of the rain pixels. A few sample images from this dataset are shown in Figure 4.3. This dataset is used for evaluation (test) purpose only.

[2] http://www.photoshopessentials.com/photo-effects/rain/

	GEN	GEN-CGAN-S	GEN-P	GEN-CGAN-PS	ID-CGAN
PSNR (dB)	**24.40**	23.55	23.77	24.08	24.34
SSIM	0.8275	0.8290	0.8305	0.8376	**0.8430**
UQI	0.6506	0.6541	0.6557	0.6705	**0.6741**
VIF	0.3999	0.4003	0.4056	0.4133	**0.4188**

Table 4.2: Quantitative comparison baseline configurations.

Quality measures

The following measures are used to evaluate the performance of different methods: Peak Signal to Noise Ratio (PSNR), Structural Similarity Index (SSIM) [145], Universal Quality Index (UQI) [144] and Visual Information Fidelity (VIF) [126]. Similar to previous methods [83], all of these quantitative measures are calculated using the luminance channel. Since we do not have ground truth reference images for the real dataset, the performance of the proposed and other methods on the real dataset is evaluated visually.

4.3.2 Model Details and Parameters

The entire network is trained on a Nvidia Titan-X GPU using the torch framework [22]. We used a batch size of 1 and number of training iterations of 100k. Adam algorithm [70] with a learning rate of 2×10^{-3} is used. During training, we set $\lambda_a = 6.6 \times 10^{-3}$ and $\lambda_p = 1$. All the parameters are set via cross-validation. A low value for λ_a is used so as to ensure that the adversarial loss does not dominate the other losses.

	SPM [68]	PRM [19]	DSC [86]	CNN [38]	GMM [83]	CCR [169]	DDN [37]	JORDER [158]	ID-CGAN
PSNR (dB)	18.88	20.46	18.56	19.12	22.27	20.56	22.28	21.09	**24.34**
SSIM	0.7632	0.7297	0.5996	0.6013	0.7413	0.7332	0.7508	0.7525	**0.8430**
UQI	0.4149	0.5668	0.4804	0.4706	0.5751	0.5582	0.5995	0.5768	**0.6741**
VIF	0.2197	0.3441	0.3325	0.3307	0.4042	0.3607	0.3916	0.3785	**0.4188**

Table 4.3: Quantitative comparisons with state-of-the-art methods evaluated on using four different criterions.

4.3.3 Comparison with Baseline Configurations

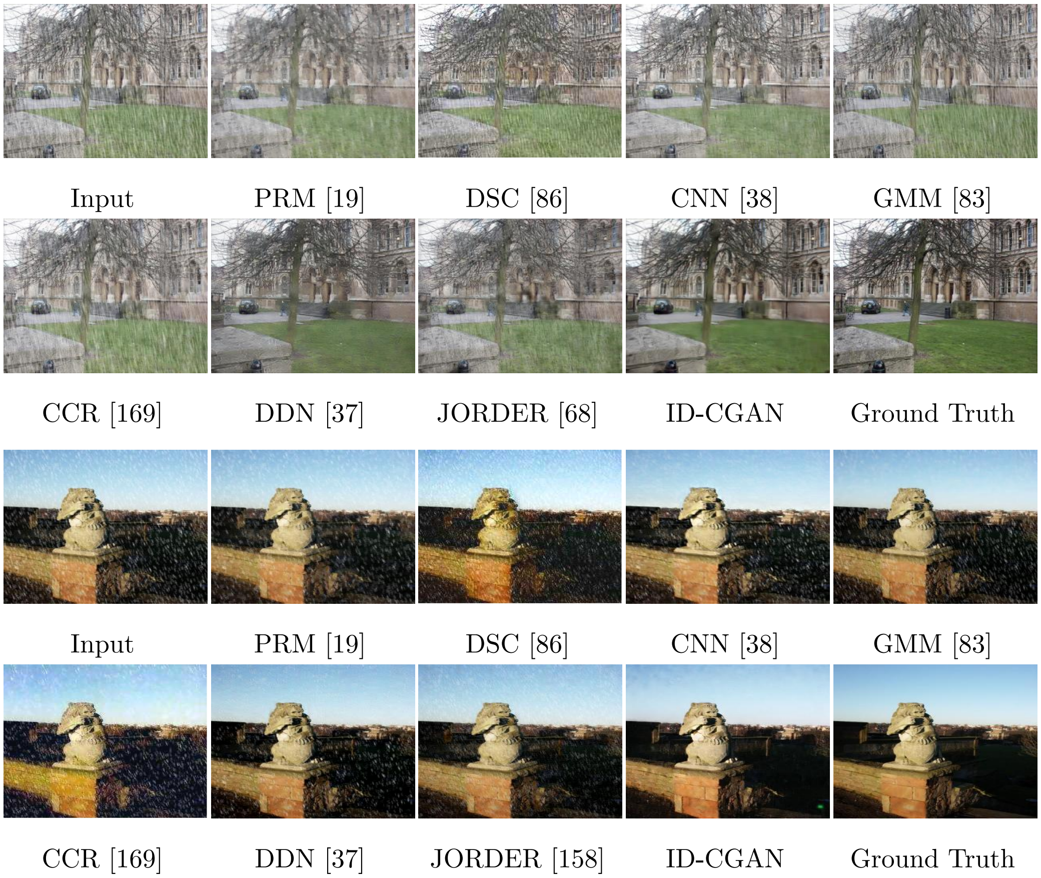
In order to demonstrate the significance of different modules in the proposed method, we compare the performance of following baseline configurations:

- GEN: Generator G is trained using per-pixel Euclidean loss by setting λ_a and λ_p to zero in (4.2). This amounts to a traditional CNN architecture with Euclidean loss.
- GEN-CGAN-S: Generator G trained using per-pixel Euclidean loss and Adversarial loss from a single-scale discriminator D (no multi-scale pooling). λ_p is set to zero in (4.2).
- GEN-P: Generator G is trained using per-pixel Euclidean loss and perceptual loss. λ_a is set to zero in (4.2).
- GEN-CGAN-PS: Generator G is trained using per-pixel Euclidean loss, perceptual loss and adversarial loss from a single scale discriminator.
- ID-CGAN: Generator G is trained using per-pixel Euclidean loss, perceptual loss and adversarial loss from multi-scale discriminator D.

All four configurations along with ID-CGAN are learned using training images from the synthetic training dataset. Results of quantitative performance, using the measures discussed earlier on test images from the synthetic dataset, are shown in Table 4.2. Sample results for the above baseline configurations on test images from real dataset are shown in Figure 4.4 and Figure 4.5. It can be observed from Figure 4.4(c), that the introduction of adversarial loss improves the visual quality over the traditional CNN architectures, however, it also introduces certain artifacts. The use of perceptual loss along with adversarial loss from a single scale discriminator reduces these artifacts while producing sharper results. However, part of the texture details are still missing in the de-rained results (Figure 4.4(e)) such as the edge of the left back part of the car (shown in third row in Figure 4.4) and the structure of truck's front mirror (shown in second row in Figure 4.5). Finally, the use of adversarial loss from multi-scale discriminator along with other loss functions (ID-CGAN) results in recovery of these texture details and achieve the best results. Quantitative results shown in Table 4.3 also demonstrate the effectiveness of the each module.

4.3.4 Comparison with State-of-the-art Methods

We compare the performance of the proposed ID-CGAN method with the following recent state-of-the-art methods for single image de-raining:



Input PRM [19] DSC [86] CNN [38] GMM [83]

CCR [169] DDN [37] JORDER [68] ID-CGAN Ground Truth

Input PRM [19] DSC [86] CNN [38] GMM [83]

CCR [169] DDN [37] JORDER [158] ID-CGAN Ground Truth

Figure 4.6: Qualitative comparison of rain-streak removal on two sample images from synthetic dataset.

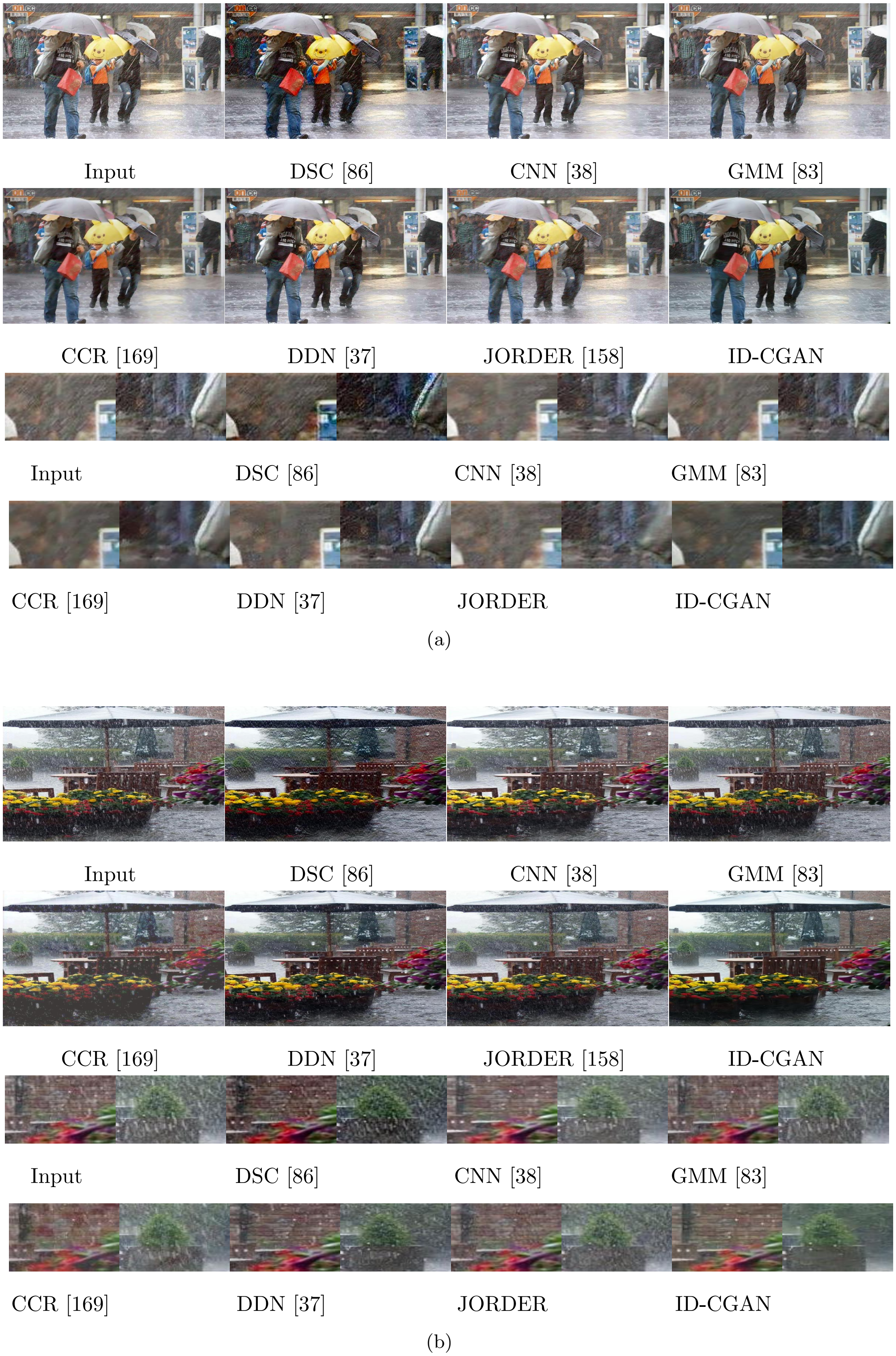


Figure 4.7: Qualitative comparison of rain-streak removal on two sample real images.

- SPM: Sparse dictionary-based method [68] (*TIP '12*)
- PRM: PRM prior-based method[19] (*ICCV '13*)
- DSC: Discriminative sparse coding-based method [86] (*ICCV '15*)
- GMM: GMM-based method [83] (*CVPR '16*)
- CNN: CNN-based method [38] (*TIP '17*)
- CCR: Convolutional-coding based method [169] (*WACV '17*)
- DDN: Deep Detail Network method [37] (*CVPR '17*)
- JORDER: CNN-based method [158] (*CVPR '17*)

Results on synthetic dataset

In the first set of experiments, we compare quantitative and qualitative performance of different methods on the test images from the synthetic dataset. As the ground truth is available for the these test images, we calculate the quantitative measures such as PSNR, SSIM, UQI and VIF. Results are shown in Table 4.3. It can be clearly observed that the proposed ID-CGAN method is able to achieve superior quantitative performance using all the measures.

To visually demonstrate the improvements obtained by the proposed method on the synthetic dataset, results on two difficult sample images are presented in Figure 4.6. Note that we selectively sample difficult images to show that our method performs well in difficult conditions. While PRM [19] is able to remove the rain-streaks, it produces blurred results which are not visually appealing. The other compared methods are able to either reduce the intensity of rain or remove the streaks in parts, however, they fail to completely remove the rain-streaks. In contrast to the other methods, the proposed method is able to successfully remove majority of the rain streaks while maintaining the details of the de-rained images.

Evaluation on Real Rainy Images

We also evaluated the performance of the proposed method and recent state-of-the-art methods on real-world rainy test images. The de-rained results for all the methods on two sample input rainy images are shown in Figure 4.7. For better visual comparison, we show zoomed versions of the two specific regions-of-interest below the de-rained results. By looking at these regions-of-interest, we can clearly observe that DSC [86] tends to add artifacts on the de-rained images. Even though the other methods GMM [83], CNN [38], CCR [169], DDN [37] and JORDER [158] are able to achieve good visual performance, rain drops are still visible in the zoomed regions-of-interest. In comparison, the proposed method is able to remove most of the rain drops while maintaining the details of the background image. One may observe that the proposed method leaves out a few rain-streaks in the output images. This is because the two image samples represent relatively difficult cases for de-raining. However, the proposed method is able to achieve better results compared to state-of-the-art methods. Additional comparisons are provided in Figure 4.8. It can be seen that the proposed method achieves better results among all the methods. In addition, more de-rained results on different rainy images, shown in Figure 4.10, demonstrate that the proposed method successfully removes rain streaks.

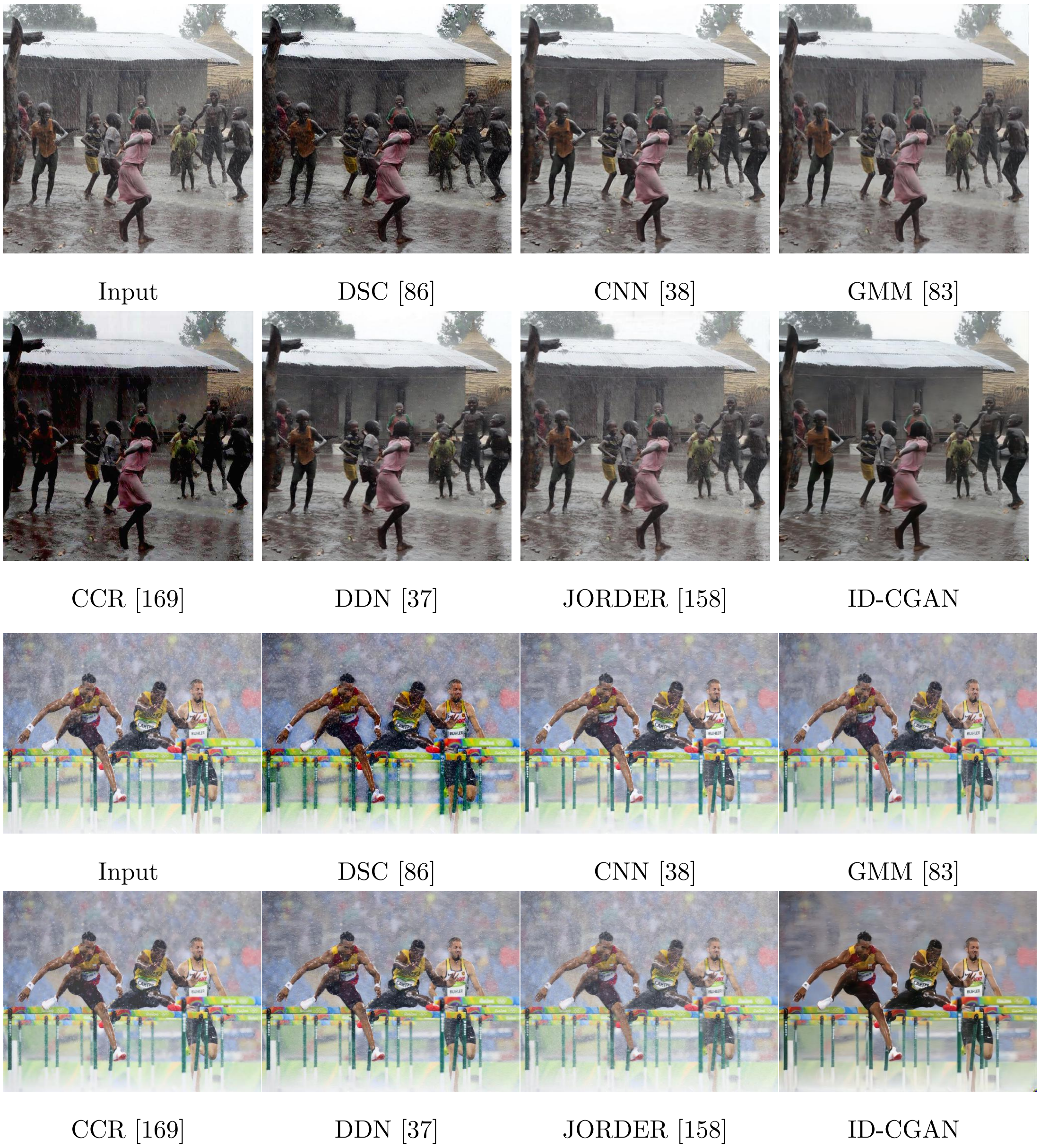
Evaluation on Object Detection Results

Single image de-raining algorithms can be used as a pre-processing step to improve the performance of other high level vision tasks such as face recognition and object detection [68]. In order to demonstrate the performance improvement obtained after de-raining using the proposed ID-CGAN method, we evaluated Faster-RCNN [111] on VOC 2007 dataset [32]. First, the VOC 2007 dataset is artificially degraded with rain streaks similar to Section IV A. Due to the degradations, object detection performance using Faster-RCNN results in poor performance. Next, the degraded images are processed by ID-CGAN method to remove the rain streaks and the de-rained images are fed to the Faster-RCNN method. We present the mean average precision (mAP) for the entire VOC dataset in Table IV. It may be noted that Faster-RCNN on degraded images results in a low average precision, however, the performance is boosted by 78% when

Table 4.4: Object detection performance using Faster-RCNN on VOC 2007 dataset.

Condition	mAP
With Rain	0.39
De-rained	0.69

the images undergo de-raining using the proposed ID-CGAN method.



Input DSC [86] CNN [38] GMM [83]

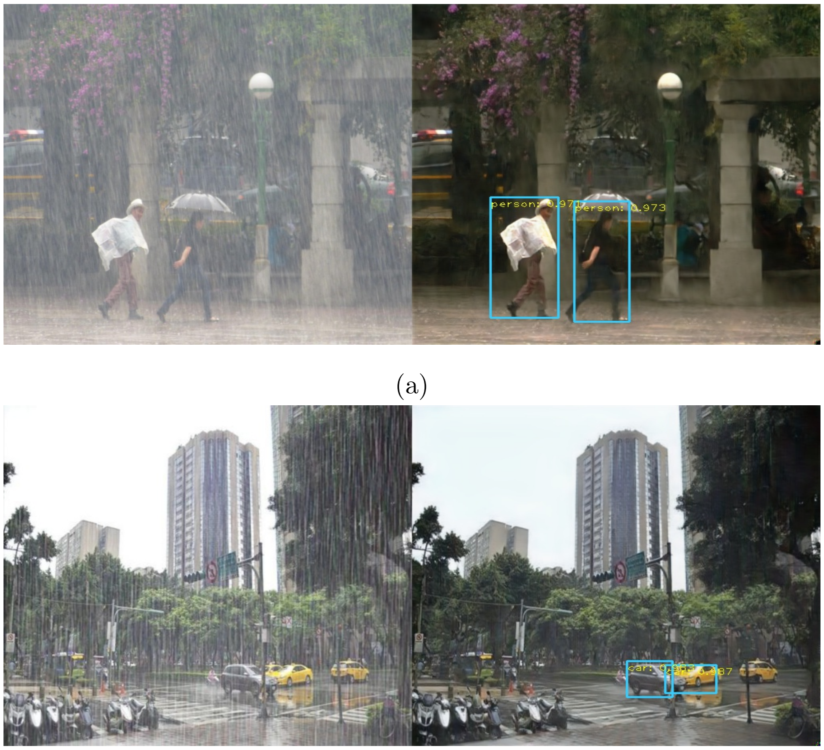
CCR [169] DDN [37] JORDER [158] ID-CGAN

Input DSC [86] CNN [38] GMM [83]

CCR [169] DDN [37] JORDER [158] ID-CGAN

Figure 4.8: Qualitative comparison of rain-streak removal on two sample real images.

Sample detection results for Faster-RCNN on real-world rainy and de-rained images are shown in Figure 4.9. The degradations result in total failure of Faster-RCNN on these images, however, after being processed by ID-CGAN, the same detection method is able successfully detect different objects in the scene.



(a)

(b)

Figure 4.9: Real-world examples of object detection (Faster-RCNN [111]) improvements obtained by the proposed ID-CGAN. *Left*: Detection results on rainy images; *Right*: Detection results on de-rained images. The detection performance is boosted when ID-CGAN is used as a pre-processing step.

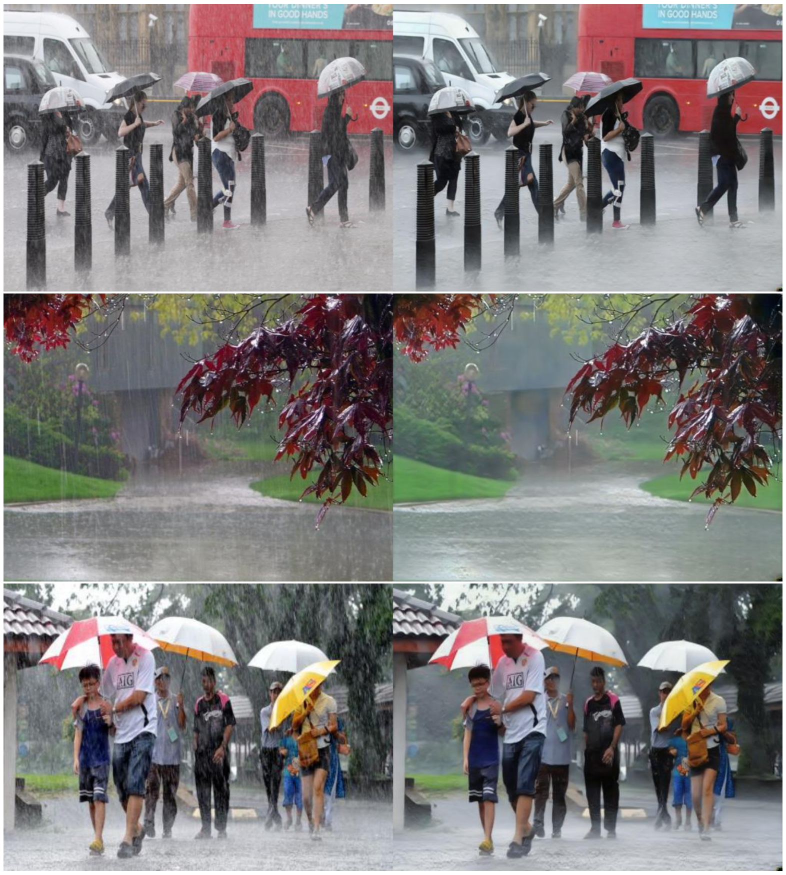


Figure 4.10: Additional de-rained results using the proposed ID-CGAN method on real-world dataset. *Left*: Input; *Right*: Derained results.

Table 4.5: Time complexity (in seconds) for different methods.

	SPM [68]	PRM [19]	DSC [86]	CNN (GPU) [38]	GMM [83]	CCR [169]	DDN (GPU) [37]	JORDER (GPU) [158]	ID-CGAN (GPU)
250X250	400.5s	40.5s	1.3s	54.9s	169.6s	150.2s	**0.2s**	0.4s	**0.2s**
500X500	1455.2s	140.3s	2.8s	189.3	674.8	600.6s	**0.3s**	1.4s	**0.3s**

Computation time

Table 4.5 compares the running time of several state-of-the-art methods. All baseline methods are implemented using MATLAB or MATLAB wrapper. Our method is implemented in Torch. It can be observed that all GPU-based CNN methods [38, 37, 158] are computationally more efficient. The proposed ID-CGAN is able achieve the fastest time[3] as compared to these methods. On an average, ID-CGAN in GPU can process and image of size 500 × 500 in about 0.3s.

[3] ID-CGAN is running as fast as Fu *et al* [37].

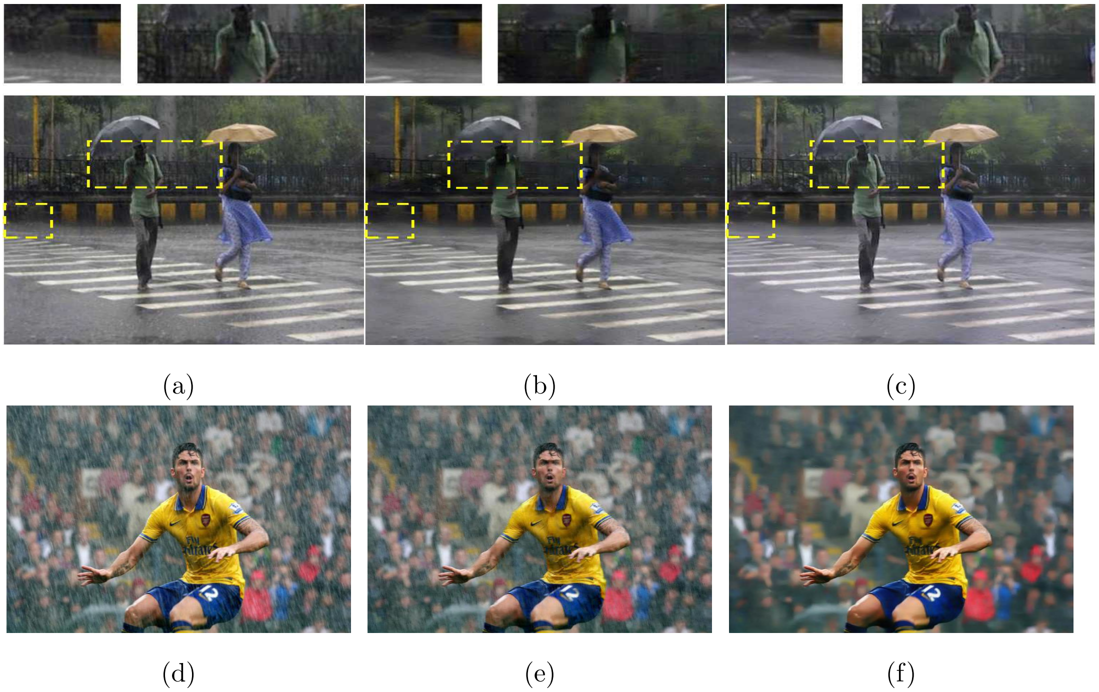
Chapter 5

Density-aware Single Image De-raining using a Multi-stream Dense Network

As discussed earlier, single image rain streak removal is an extremely challenging problem due to the presence of non-uniform rain densities in images. In this chapter, we develop a novel density-aware image de-raining method with multi-stream densely connected network (DID-MDN) for jointly rain-density estimation and de-raining. In comparison to existing approaches which attempt to solve the de-raining problem using a single network to learn to remove rain streaks with different densities (heavy, medium and light), we investigate the use of estimated rain-density label for guiding the synthesis of the de-rained image. To efficiently predict the rain-density label, a residual-aware rain-density classier is proposed in this chapter. In addition, an ablation study is performed to demonstrate the improvements obtained by different modules in the proposed method. Code and data can be found at: https://github.com/hezhangsprinter/DID-MDN

5.1 Introduction

One of the main limitations of the existing single image de-raining methods is that they are designed to deal with certain types of rainy images and they do not effectively consider various shapes, scales and density of rain drops into their algorithms. State-of-the-art de-raining algorithms such as [158, 37] often tend to over de-rain or under de-rain the image if the rain condition present in the test image is not properly considered during training. For example, when a rainy image shown in Figure 5.1(a) is de-rained using the method of Fu *et al.* [37], it tends to remove some important parts in the de-rained image such as the right arm of the person, as shown in Figure 5.1(b). Similarly,



(a) (b) (c)

(d) (e) (f)

Figure 5.1: Image de-raining results. (a) Input rainy image. (b) Result from Fu *et al.* [37]. (c) DID-MDN. (d) Input rainy image. (e) Result from Li *et al.* [158]. (f) DID-MDN. Note that [37] tends to over de-rain the image while [158] tends to under de-rain the image.

when [158] is used to de-rain the image shown in Figure 5.1(d), it tends to under de-rain the image and leaves some rain streaks in the output de-rained image. Hence, more adaptive and efficient methods, that can deal with different rain density levels present in the image, are needed.

One possible solution to this problem is to build a very large training dataset with sufficient rain conditions containing various rain-density levels with different orientations and scales. This has been achieved by Fu *et al.* [37] and Yang *et al.*[158], where they synthesize a novel large-scale dataset consisting of rainy images with various conditions and they train a single network based on this dataset for image de-raining. However, one drawback of this approach is that a single network may not be capable enough to learn all types of variations present in the training samples. It can be observed from Figure 5.1 that both methods tend to either over de-rain or under de-rain results. Alternative solution to this problem is to learn a density-specific model for de-raining. However, this solution lacks flexibility in practical de-raining as the density label information is needed for a given rainy image to determine which network to choose for de-raining.

In order to address these issues, we propose a novel Density-aware Image De-raining method using a Multi-stream Dense Network (DID-MDN) that can automatically determine the rain-density information (i.e. heavy, medium or light) present in the input image (see Figure 5.2). The proposed method consists of two main stages: rain-density classification and rain streak removal. To accurately estimate the rain-density level, a new residual-aware classifier that makes use of the residual component in the rainy image for density classification is proposed in this work. The rain streak removal algorithm is based on a multi-stream densely-connected network that takes into account the distinct scale and shape information of rain streaks. Once the rain-density level is estimated, we fuse the estimated density information into our final multi-stream densely-connected network to get the final de-rained output. Furthermore, to efficiently train the proposed network, a large-scale dataset consisting of 12,000 images with different rain-density levels/labels (i.e. heavy, medium and light) is synthesized. Figure 5.1(c) & (d) present sample results from our network, where one can clearly see that DID-MDN does not over de-rain or under de-rain the image and is able to provide better results as compared to [37] and [158].

This chapter makes the following contributions:

1. A novel DID-MDN method which automatically determines the rain-density information and then efficiently removes the corresponding rain-streaks guided by the estimated rain-density label is proposed.
2. Based on the observation that residual can be used as a better feature representation in characterizing the rain-density information, a novel residual-aware classifier to efficiently determine the density-level of a given rainy image is proposed in this chapter.
3. A new synthetic dataset consisting of 12,000 training images with rain-density labels and 1,200 test images is synthesized. To the best of our knowledge, this is the first dataset that contains the rain-density label information. Although the network is trained on our synthetic dataset, it generalizes well to real-world rainy images.

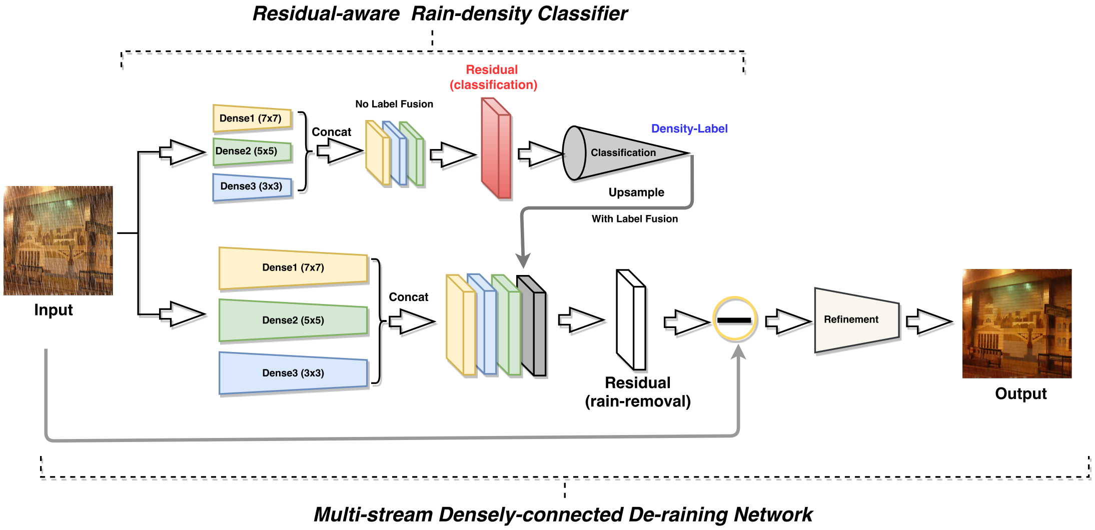


Figure 5.2: An overview of the proposed DID-MDN method. The proposed network contains two modules: (a) residual-aware rain-density classifier, and (b) multi-stream densely-connected de-raining network. The goal of the residual-aware rain-density classifier is to determine the rain-density level given a rainy image. On the other hand, the multi-stream densely-connected de-raining network is designed to efficiently remove the rain streaks from the rainy images guided by the estimated rain-density information.

4. Extensive experiments are conducted on three highly challenging datasets (two synthetic and one real-world) and comparisons are performed against several recent state-of-the-art approaches. Furthermore, an ablation study is conducted to demonstrate the effects of different modules in the proposed network.

5.2 Background and Related Work

In this section, we briefly review several recent related works on multi-scale feature aggregation.

5.2.1 Multi-scale Feature Aggregation

It has been observed that combining convolutional features at different levels (scales) can lead to a better representation of an object in the image and its surrounding context [51, 181, 60, 166]. For instance, to efficiently leverage features obtained from different scales, the FCN (fully convolutional network) method [85] uses skip-connections and adds high-level prediction layers to intermediate layers to generate pixel-wise prediction results at multiple resolutions. Similarly, the U-Net architecture [119] consists of a contracting path to capture the context and a symmetric expanding path that enables the precise localization. The HED model [153] employs deeply supervised structures, and automatically learns rich hierarchical representations that are fused to resolve the challenging ambiguity in edge and object boundary detection. Multi-scale features have also been leveraged in various applications such as semantic segmentation [181], face-alignment [102], visual tracking [81] crowd-counting [129], single image super-resolution[178], face anti-Spoofing [5], action recognition [188], depth estimation [29], single image dehazing [113, 174, 170] and also in single image de-raining [158]. Similar to [158], we also leverage a multi-stream network to capture the rain-streak components with different scales and shapes. However, rather than using two convolutional layers with different dilation factors to combine features from different scales, we leverage the densely-connected block [60] as the building module and then we connect features from each block together for the final rain-streak removal. The ablation study demonstrates the effectiveness of our proposed network compared with the structure proposed in [158].

5.3 Proposed Method

The proposed DID-MDN architecture mainly consists of two modules: (a) residual-aware rain-density classifier, and (b) multi-stream densely connected de-raining network. The residual-aware rain-density classifier aims to determine the rain-density level given a rainy image. On the other hand, the multi-stream densely connected de-raining network is designed to efficiently remove the rain streaks from the rainy images guided by the estimated rain-density information. The entire network architecture of the proposed DID-MDN method is shown in Figure 5.2.

5.3.1 Residual-aware Rain-density Classifier

As discussed above, even though some of the previous methods achieve significant improvements on the de-raining performance, they often tend to over de-rain or under de-rain the image. This is mainly due to the fact that a single network may not be sufficient enough to learn different rain-densities occurring in practice. We believe that incorporating density level information into the network can benefit the overall learning procedure and hence can guarantee better generalization to different rain conditions [112]. Similar observations have also been made in [112], where they use two different priors to characterize light rain and heavy rain, respectively. Unlike using two priors to characterize different rain-density conditions [112], the rain-density label estimated from a CNN classifier is used for guiding the de-raining process. To accurately estimate the density information given a rainy input image, a residual-aware rain-density classifier is proposed, where the residual information is leveraged to better represent the rain features. In addition, to train the classier, a large-scale synthetic dataset consisting of 12,000 rainy images with density labels is synthesized. Note that there are only three types of classes (i.e. labels) present in the dataset and they correspond to low, medium and high density.

One common strategy in training a new classifier is to fine-tune a pre-defined model such as VGG-16 [128], Res-net [52] or Dense-net [60] on the newly introduced dataset. One of the fundamental reasons to leverage a fine-tune strategy for the new dataset is that discriminative features encoded in these pre-defined models can be beneficial in accelerating the training and it can also guarantee better generalization. However, we observed that directly fine-tuning such a 'deep' model on our task is not an efficient solution. This is mainly due to the fact that high-level features (deeper part) of a CNN tend to pay more attention to localize the discriminative objects in the input image [184]. Hence, relatively small rain-streaks may not be localized well in these high-level features. In other words, the rain-streak information may be lost in the high-level features and hence may degrade the overall classification performance. As a result, it is important to come up with a better feature representation to effectively characterize rain-streaks (i.e. rain-density).

From (2.1), one can regard $\mathbf{y_r} = \mathbf{y} - \mathbf{y_c}$ as the residual component which can be used to characterize the rain-density. To estimate the residual component ($\hat{\mathbf{y_r}}$) from the observation $\mathbf{y}$, a multi-stream dense-net (without the label fusion part) using the new dataset with heavy-density is trained. Then, the estimated residual is regarded as the input to train the final classifier. In this way, the residual estimation part can be regarded as the feature extraction procedure [1], which is discussed in Section 3.2. The classification part is mainly composed of three convolutional layers (Conv) with kernel size 3×3, one average pooling (AP) layer with kernel size 9×9 and two fully-connected layers (FC). Details of the classifier are as follows:

Conv(3,24)-Conv(24,64)-Conv(64,24)-AP- FC(127896,512)-FC(512,3),

where (3,24) means that the input consists of 3 channels and the output consists of 24 channels. Note that the final layer consists of a set of 3 neurons indicating the rain-density class of the input image (i.e. low, medium, high). An ablation study, discussed in Section 4.3, is conducted to demonstrate the effectiveness of proposed residual-aware classifier as compared with the VGG-16 [128] model.

Loss for the Residual-aware Classifier:. To efficiently train the classifier, a two-stage training protocol is leveraged. A residual feature extraction network is firstly trained to estimate the residual part of the given rainy image, then a classification sub-network is trained using the estimated residual as the input and is optimized via the ground truth labels (rain-density). Finally, the two stages (feature extraction and classification) are jointly optimized. The overall loss function used to train the residual-aware classier is as follows:

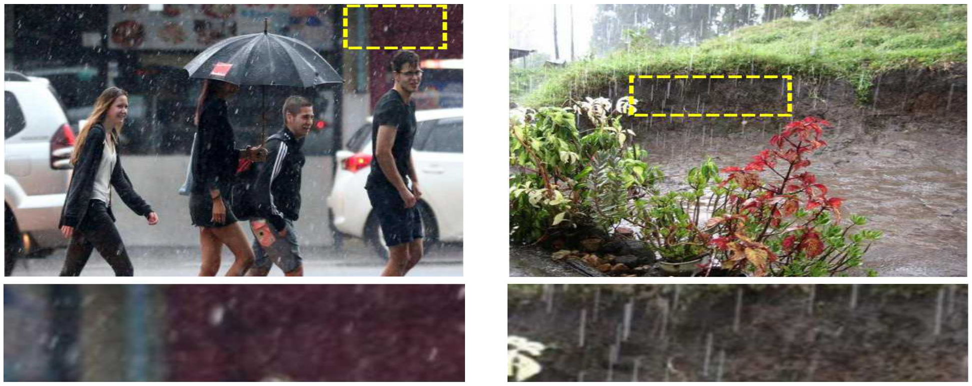
$$L = L_{E,r} + L_C, \tag{5.1}$$

where $L_{E,r}$ indicates the per-pixel Euclidean-loss to estimate the residual component and L_C indicates the cross-entropy loss for rain-density classification.

5.3.2 Multi-stream Dense Network

It is well-known that different rainy images contain rain-streaks with different scales and shapes. Considering the images shown in Figure 5.3, the rainy image in Figure 5.3 (a) contains smaller rain-streaks, which can be captured by small-scale features (with smaller receptive fields), while the image in Figure 5.3 (b) contains longer rain-streaks,

[1] Classificaiton network can be regarded as two parts: 1.Feature extractor and 2. Classifer



(a) (b)

Figure 5.3: Sample images containing rain-streaks with various scales and shapes.(a) contains smaller rain-streaks, (b) contains longer rain-streaks.

which can be captured by large-scale features (with larger receptive fields). Hence, we believe that combining features from different scales can be a more efficient way to capture various rain streak components [58, 158].

Based on this observation and motivated by the success of using multi-scale features for single image de-raining [158], a more efficient multi-stream densely-connected network to estimate the rain-streak components is proposed, where each stream is built on the dense-block introduced in [60] with different kernel sizes (different receptive fields). These multi-stream blocks are denoted by Dense1 (7×7), Dense2 (5×5), and Dense3 (3×3), in yellow, green and blue blocks, respectively in Figure 5.2. In addition, to further improve the information flow among different blocks and to leverage features from each dense-block in estimating the rain streak components, a modified connectivity is introduced, where all the features from each block are concatenated together for rain-streak estimation. Rather than leveraging only two convolutional layers in each stream [158], we create short paths among features from different scales to strengthen feature aggregation and to obtain better convergence. To demonstrate the effectiveness of our proposed multi-stream network compared with the multi-scale structure proposed in [158], an ablation study is conducted.

To leverage the rain-density information to guide the de-raining process, the up-sampled label map [2] is concatenated with the rain streak features from all three streams.

[2] For example, if the label is 1, then the corresponding up-sampled label-map is of the same dimension

Then, the concatenated features are used to estimate the residual ($\hat{\mathbf{r}}$) rain-streak information. In addition, the residual is subtracted from the input rainy image to estimate the coarse de-rained image. Finally, to further refine the estimated coarse de-rained image and make sure better details well preserved, another two convolutional layers with ReLU are adopted as the final refinement.

There are six dense-blocks in each stream. Mathematically, each stream can be represented as

$$\mathbf{s}_j = cat[DB_1, DB_2, ..., DB_6], \tag{5.2}$$

where cat indicates concatenation, $DB_i, i = 1, \cdots 6$ denotes the output from the ith dense block, and $\mathbf{s}_j, j = 1, 2, 3$ denotes the jth stream. Furthermore, we adopt different transition layer combinations[3] and kernel sizes in each stream. Details of each stream are as follows:

Dense1: three transition-down layers, three transition-up layers and kernel size 7 × 7.

Dense2: two transition-down layers, two no-sampling transition layers, two transition-up layers and kernel size 5 × 5.

Dense3: one transition-down layer, four no-sampling transition layers, one transition-up layer and kernel size 3 × 3.

Note that each dense-block is followed by a transition layer. Fig 5.4 presents an overview of the first stream, **Dense1**.

Loss for the De-raining Network:. Motivated by the observation that CNN feature-based loss can better improve the semantic edge information [67, 75] and to further enhance the visual quality of the estimated de-rained image [173], we also leverage a weighted combination of pixel-wise Euclidean loss and the feature-based loss. The loss for training the multi-stream densely connected network is as follows

$$L = L_{E,r} + L_{E,d} + \lambda_F L_F, \tag{5.3}$$

as the output features from each stream and all the pixel values of the label map are 1.

[3] The transition layer can function as up-sample transition, down-sample transition or no-sampling transition [65].

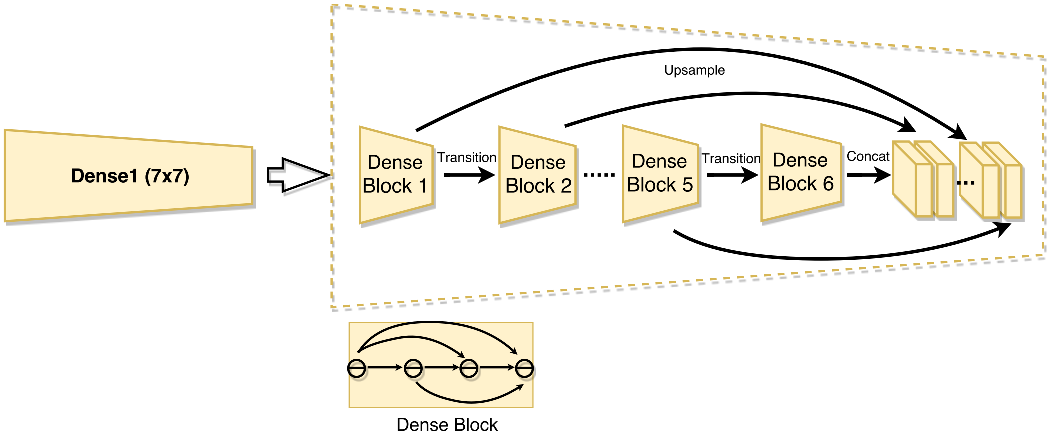


Figure 5.4: Details of the first stream *Dense1*.

where $L_{E,d}$ represents the per-pixel Euclidean loss function to reconstruct the de-rained image and L_F is the feature-based loss for the de-rained image, defined as

$$L_F = \frac{1}{CWH}\|F(\hat{\mathbf{x}})^{c,w,h} - F(\mathbf{x})^{c,w,h}\|_2^2, \tag{5.4}$$

where F represents a non-linear CNN transformation and $\hat{\mathbf{x}}$ is the recovered de-rained image. Here, we have assumed that the features are of size $w \times h$ with c channels. In our method, we compute the feature loss from the layer relu1_2 of the VGG-16 model [128].

Table 5.1: Quantitative results evaluated in terms of average SSIM and PSNR (dB) (SSIM/PSNR).

	Input	DSC [86] (ICCV'15)	GMM [83] (CVPR'16)	CNN [38] (TIP'17)	JORDER [158] (CVPR'17)	DDN [37] (CVPR'17)	JBO [186] (ICCV'17)	DID-MDN
Test1	0.7781/21.15	0.7896/21.44	0.8352/22.75	0.8422/22.07	0.8622/24.32	0.8978/ 27.33	0.8522/23.05	**0.9087**/ **27.95**
Test2	0.7695/19.31	0.7825/20.08	0.8105/20.66	0.8289/19.73	0.8405/22.26	0.8851/25.63	0.8356/22.45	**0.9092**/ **26.0745**

5.3.3 Testing

During testing, the rain-density label information using the proposed residual-aware classifier is estimated. Then, the up-sampled label-map with the corresponding input image are fed into the multi-stream network to get the final de-rained image.

5.4 Experimental Results

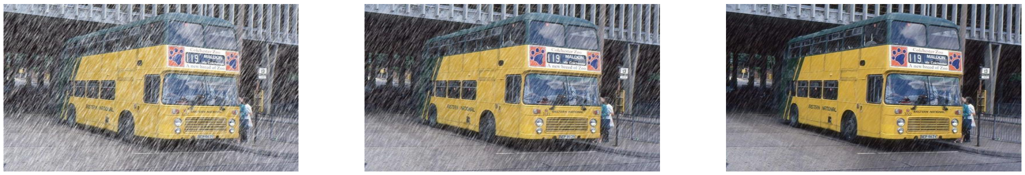
In this section, we present the experimental details and evaluation results on both synthetic and real-world datasets. De-raining performance on the synthetic data is evaluated in terms of PSNR and SSIM [145]. Performance of different methods on real-world images is evaluated visually since the ground truth images are not available. The proposed DID-MDN method is compared with the following recent state-of-the-art methods: (a) Discriminative sparse coding-based method (DSC) [86] (ICCV'15), (b) Gaussian mixture model (GMM) based method [83] (CVPR'16), (c) CNN method (CNN) [38] (TIP'17), (d) Joint Rain Detection and Removal (JORDER) method [158] (CVPR'17), (e) Deep detailed Network method (DDN) [37] (CVPR'17), and (f) Joint Bi-layer Optimization (JBO) method [186] (ICCV'17).

5.4.1 Synthetic Dataset

Even though there exist several large-scale synthetic datasets [37, 173, 158], they lack the availability of the corresponding rain-density label information for each synthetic rainy image. Hence, we develop a new dataset, denoted as *Train1*, consisting of 12,000 images, where each image is assigned a label based on its corresponding rain-density level. There are three rain-density labels present in the dataset (e.g. light, medium and heavy). There are roughly 4,000 images per rain-density level in the dataset. Similarly, we also synthesize a new test set, denoted as *Test1*, which consists of a total of 1,200 images. It is ensured that each dataset contains rain streaks with different orientations and scales. Images are synthesized using Photoshop. We modify the noise level introduced in step 3 of [4] to generate different rain-density images, where light, medium and heavy rain conditions correspond to the noise levels 5% $\sim$ 35%, 35% $\sim$ 65%, and 65% $\sim$ 95%, respectively [5]. Sample synthesized images under these three conditions are shown in Fig 5.5. To better test the generalization capability of the proposed method, we also randomly sample 1,000 images from the synthetic dataset provided by Fu [37] as another testing set, denoted as *Test2*.

[4] http://www.photoshopessentials.com/photo-effects/photoshopweather-effects-rain/

[5] The reason why we use three labels is that during our experiments, we found that having more than three rain-density levels does not significantly improve the performance. Hence, we only use three labels (heavy, medium and light) in the experiments.



Heavy Medium Light

Figure 5.5: Samples synthetic images in three different conditions.

Table 5.2: Quantitative results compared with three baseline configurations on *Test1*.

	Single	Yang-Multi [158]	Multi-no-label	DID-MDN
PSNR (dB)	26.05	26.75	27.56	**27.95**
SSIM	0.8893	0.8901	0.9028	**0.9087**

5.4.2 Training Details

During training, a 512 × 512 image is randomly cropped from the input image (or its horizontal flip) of size 586×586. Adam is used as optimization algorithm with a mini-batch size of 1. The learning rate starts from 0.001 and is divided by 10 after 20 epoch. The models are trained for up to 80×12000 iterations. We use a weight decay of 0.0001 and a momentum of 0.9. The entire network is trained using the Pytorch framework. During training, we set $\lambda_F = 1$. All the parameters are defined via cross-validation using the validation set.

5.4.3 Ablation Study

The first ablation study is conducted to demonstrate the effectiveness of the proposed residual-aware classifier compared to the VGG-16 [128] model. The two classifiers are trained using our synthesized training samples *Train1* and tested on the *Test1* set. The classification accuracy corresponding to both classifiers on *Test1* is tabulated in Table 5.3. It can be observed that the proposed residual-aware classifier is more accurate than the VGG-16 model for predicting the rain-density levels.

Table 5.3: Accuracy of rain-density estimation evaluated on *Test1*.

	VGG-16 [128]	Residual-aware
Accuracy	73.32 %	85.15 %

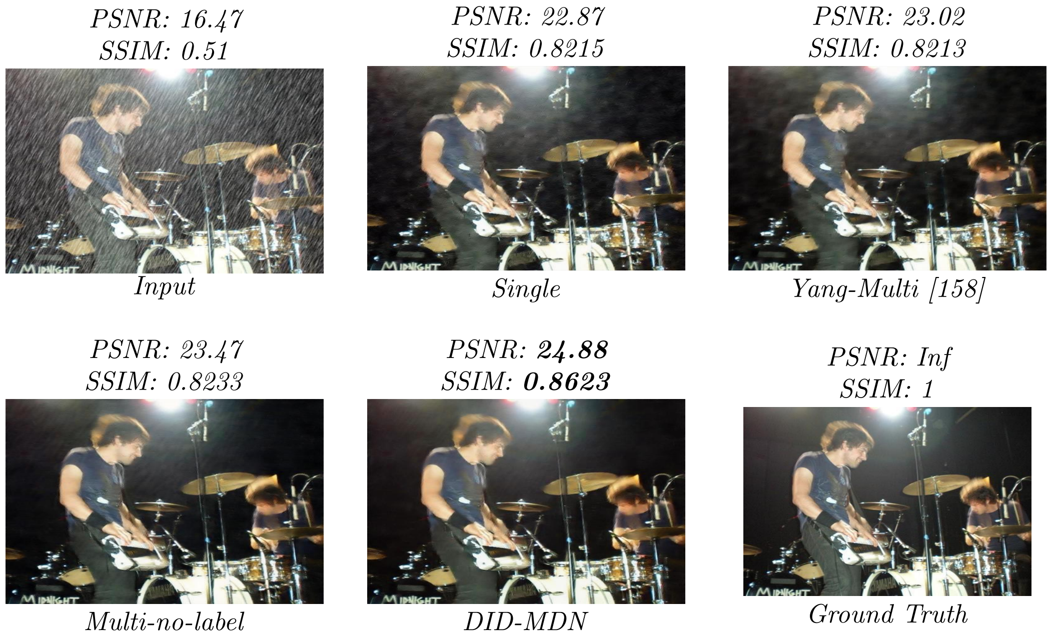


Figure 5.6: Results of ablation study on a synthetic image.

In the second ablation study, we demonstrate the effectiveness of different modules in our method by conducting the following experiments:

- **Single**: A single-stream densely connected network (**Dense2**) without the procedure of label fusion.
- **Yang-Multi [158]**[6]: Multi-stream network trained without the procedure of label fusion.
- **Multi-no-label**: Multi-stream densely connected network trained without the procedure of label fusion.
- **DID-MDN (our)**: Multi-stream Densely-connected network trained with the procedure of estimated label fusion.

The average PSNR and SSIM results evaluated on *Test1* are tabulated in Table 5.2. As shown in Figure 5.6, even though the single stream network and Yang's multi-stream network [158] are able to successfully remove the rain streak components, they both tend to over de-rain the image with the blurry output. The multi-stream network

[6] To better demonstrate the effectiveness of our proposed muli-stream network compared with the state-of-the-art multi-scale structure proposed in [158], we replace our multi-stream dense-net part with the multi-scale structured in [158] and keep all the other parts the same.

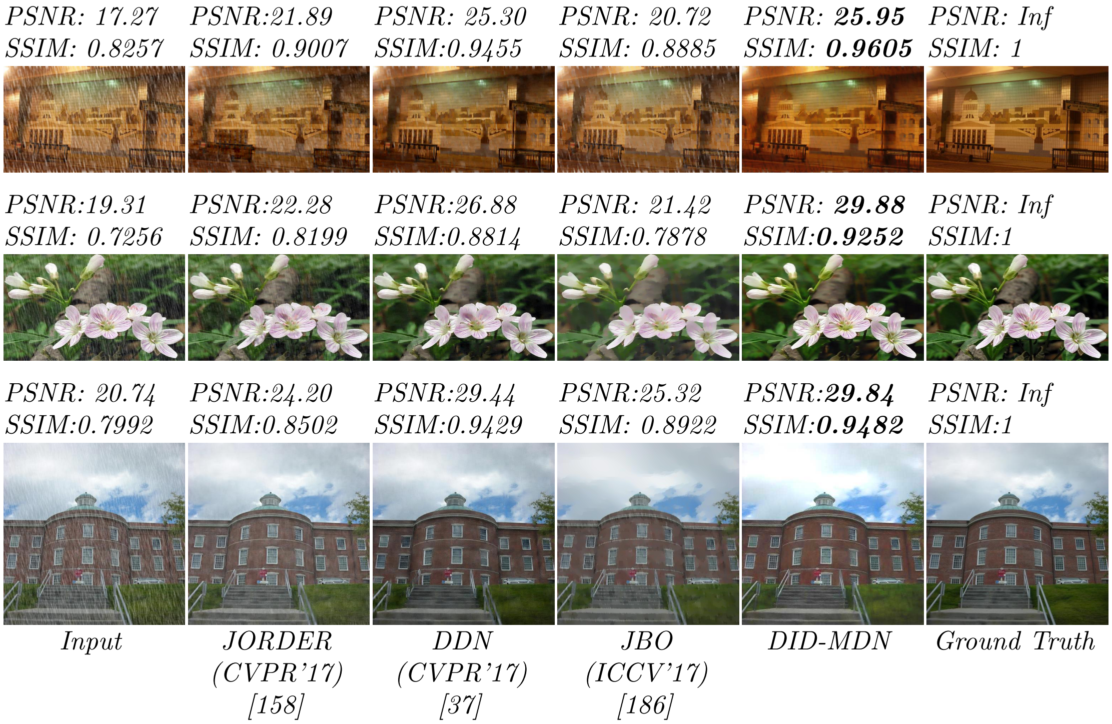


Figure 5.7: Rain-streak removal results on sample images from the synthetic datasets *Test1* and *Test2*.

without label fusion is unable to accurately estimate the rain-density level and hence it tends to leave some rain streaks in the de-rained image (especially observed from the derained-part around the light). In contrast, the proposed multi-stream network with label fusion approach is capable of removing rain streaks while preserving the background details. Similar observations can be made using the quantitative results as shown in Table 5.2.

Results on Two Synthetic Datasets

We compare quantitative and qualitative performance of different methods on the test images from the two synthetic datasets - *Test1* and *Test2*. Quantitative results corresponding to different methods are tabulated in Table 5.1. It can be clearly observed that the proposed DID-MDN is able to achieve superior quantitative performance.

To visually demonstrate the improvements obtained by the proposed method on the synthetic dataset, results on two sample images selected from *Test2* and one sample chosen from our newly synthesized *Test1* are presented in Figure 5.7. Note that we

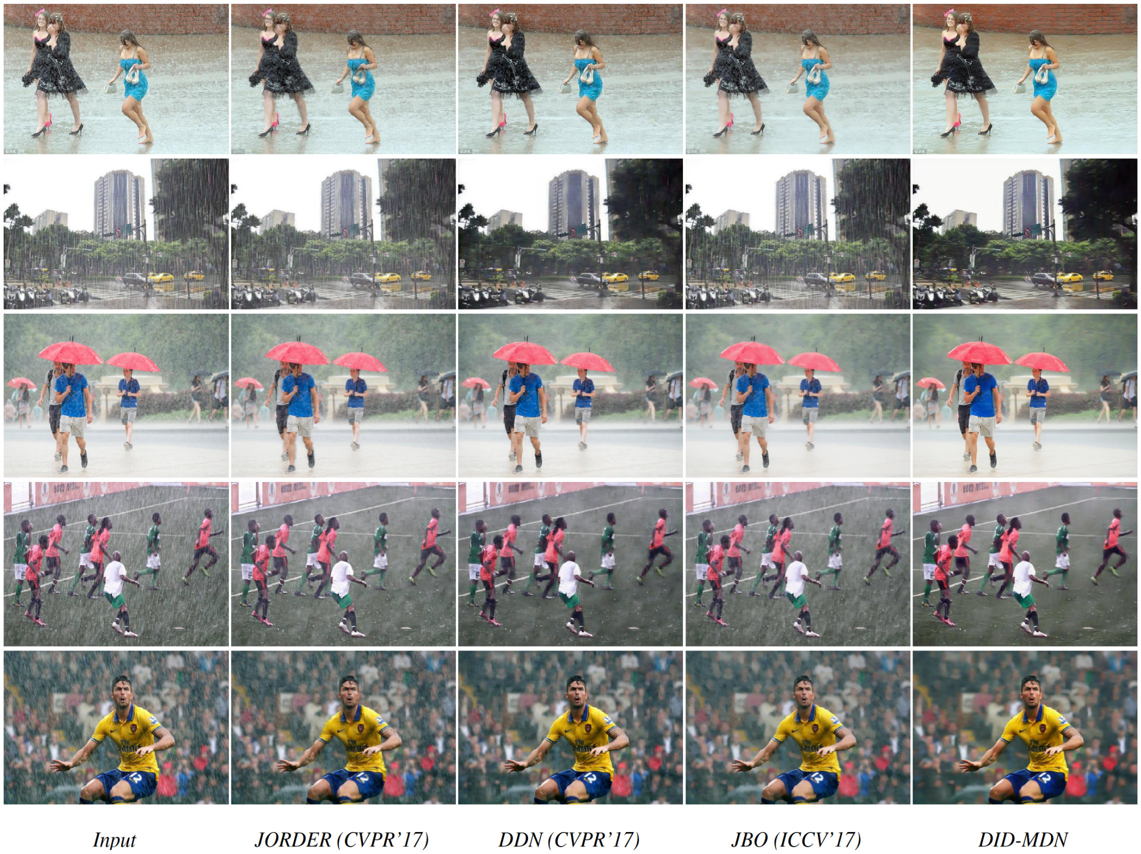


Figure 5.8: Rain-streak removal results on sample real-world images.

selectively sample images from all three conditions to show that our method performs well under different variations [7]. While the JORDER method [158] is able to remove some parts of the rain-streaks, it still tends to leave some rain-streaks in the de-rained images. Similar results are also observed from [186]. Even though the method of Fu *et al.* [37] is able to remove the rain-streak, especially in the medium and light rain conditions, it tends to remove some important details as well, such as flower details, as shown in the second row and window structures as shown in the third row (Details can be better observed via zooming-in the figure). Overall, the proposed method is able to preserve better details while effectively removing the rain-streak components.

Results on Real-World Images

The performance of the proposed method is also evaluated on many real-world images downloaded from the Internet and also real-world images published by the authors of [173, 37]. The de-raining results are shown in Fig 5.8.

As before, previous methods either tend to under de-rain or over de-rain the images. In contrast, the proposed method achieves better results in terms of effectively removing rain streaks while preserving the image details. In addition, it can be observed that the proposed method is able to deal with different types of rain conditions, such as heavy rain shown in the second row of Fig 5.8 and medium rain shown in the fifth row of Fig 5.8. Furthermore, the proposed method can effectively deal with rain-streaks containing different shapes and scales such as small round rain streaks shown in the third row in Fig 5.8 and long-thin rain-streak in the second row in Fig 5.8. Overall, the results evaluated on real-world images captured from different rain conditions demonstrate the effectiveness and the robustness of the proposed *DID-MDN* method.

Running Time Comparisons

Running time comparisons are shown in the table below. It can be observed that the testing time of the proposed DID-MDN is comparable to the DDN [37] method. On

[7] Due to space limitations and for better comparisons, we only show the results corresponding to the most recent state-of-the-art methods [158, 37, 186].

average, it takes about 0.2s to de-rain an image of size 512 × 512.

Table 5.4: Running time (in seconds) for different methods averaged on 1000 images with size 512×512.

	DSC	GMM	CNN (GPU)	JORDER (GPU)	DDN (GPU)	JBO (CPU)	DID-MDN (GPU)
512X512	189.3s	674.8s	2.8s	600.6s	0.3s	1.4s	**0.2s**

Chapter 6

Densely Connected Pyramid Dehazing Network

A new end-to-end single image dehazing method, called Densely Connected Pyramid Dehazing Network (DCPDN), which can jointly learn the transmission map, atmospheric light and dehazing all together is proposed in this chapter. This network is optimized using a newly introduced edge-preserving loss function. To further incorporate the mutual structural information between the estimated transmission map and the dehazed result, we propose a joint-discriminator based on generative adversarial network framework to decide whether the corresponding dehazed image and the estimated transmission map are real or fake. An ablation study is conducted to demonstrate the effectiveness of each module evaluated at both estimated transmission map and dehazed result. Extensive experiments demonstrate that the proposed method achieves significant improvements over the state-of-the-art methods. Code and data is made available at: https://github.com/hezhangsprinter/DCPDN

6.1 Introduction

Under severe hazy conditions, floating particles in the atmosphere such as dusk and smoke greatly absorb and scatter the light, resulting in degradations in the image quality. These degradations in turn may affect the performance of many computer vision systems such as classification and detection. To overcome the degradations caused by haze, image and video-based haze removal algorithms have been proposed in the literature [113, 13, 136, 7, 50, 73, 84, 174, 78, 187, 34, 35, 34, 114].

It can be observed from Eq. 2.3 that there exists two important aspects in the dehazing process: *(1) accurate estimation of transmission map,* and *(2) accurate estimation*

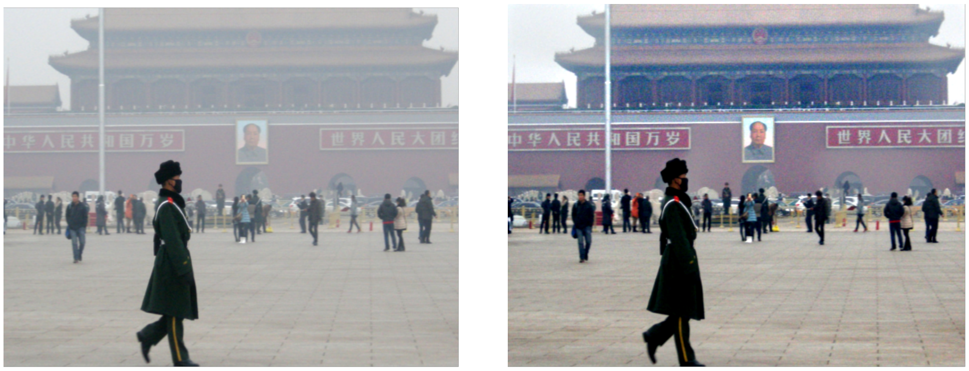


Figure 6.1: Sample image dehazing result using the proposed DCPDN method. Left: Input hazy image. Right: Dehazed result.

of atmospheric light. Apart from several works that focus on estimating the atmospheric light [8, 133], most of the other algorithms concentrate more on the accurate estimation of the transmission map and they leverage empirical rule in estimating the atmospheric light [50, 91, 113, 135]. This is mainly due to the common belief that good estimation of transmission map will lead to better dehazing. As discussed in Chapter 2, these methods can be broadly divided into two main groups: prior-based methods and learning-based methods.

Though tremendous improvements have been made by the learning-based methods, several factors hinder the performance of these methods and the results are far from optimal. This is mainly because: *1. Inaccuracies in the estimation of transmission map translates to low quality dehazed result. 2. Existing methods do not leverage end-to-end learning and are unable to capture the inherent relation among transmission map, atmospheric light and dehazed image. The disjoint optimization may hinder the overall dehazing performance.* Most recently, a method was proposed in [78] to jointly optimize the whole dehazing network. This was achieved by leveraging a linear transformation to embed both the transmission map and the atmospheric light into one variable and then learning a light-weight CNN to recover the clean image.

In this chapter, we take a different approach in addressing the end-to-end learning for image dehazing. In particular, we propose a new image dehazing architecture, called Densely Connected Pyramid Dehazing Network (DCPDN), that can be jointly optimized to estimate transmission map, atmospheric light and also image dehazing simultaneously by following the image degradation model Eq. 2.3 (see Figure 6.2). In other words, the end-to-end learning is achieved by embedding Eq. 2.3 directly into the network via the math operation modules provided by the deep learning framework. However, training such a complex network (with three different tasks) is very challenging. To ease the training process and accelerate the network convergence, we leverage a stage-wise learning technique in which we first progressively optimize each part of the network and then jointly optimize the entire network. To make sure that the estimated transmission map preserves sharp edges and avoids halo artifacts when dehazing, a new edge-preserving loss function is proposed in this chapter based on the observation that gradient operators and first several layers of a CNN structure can function as edge extractors. Furthermore, a densely connected encoder-decoder network with multi-level pooling modules is proposed to leverage features from different levels for estimating the transmission map. To exploit the structural relationship between the transmission map and the dehazed image, a joint discriminator-based GAN is proposed. The joint discriminator distinguishes whether a pair of estimated transmission map and dehazed image is a real or fake pair. To guarantee that the atmospheric light can also be optimized within the whole structure, a U-net [119] is adopted to estimate the homogeneous atmospheric light map. Shown in Figure 6.1 is a sample dehazed image using the proposed method.

This chapter makes the following contributions:

- A novel end-to-end jointly optimizable dehazing network is proposed. This is enabled by embedding Eq. 2.3 directly into the optimization framework via math operation modules. Thus, it allows the network to estimate the transmission map, atmospheric light and dehazed image jointly. The entire network is trained by a stage-wise learning method.
- An edge-preserving pyramid densely connected encoder-decoder network is proposed for accurately estimating the transmission map. Further, it is optimized via a newly proposed edge-preserving loss function.
- As the structure of the estimated transmission map and the dehazed image are highly correlated, we leverage a joint discriminator within the GAN framework to

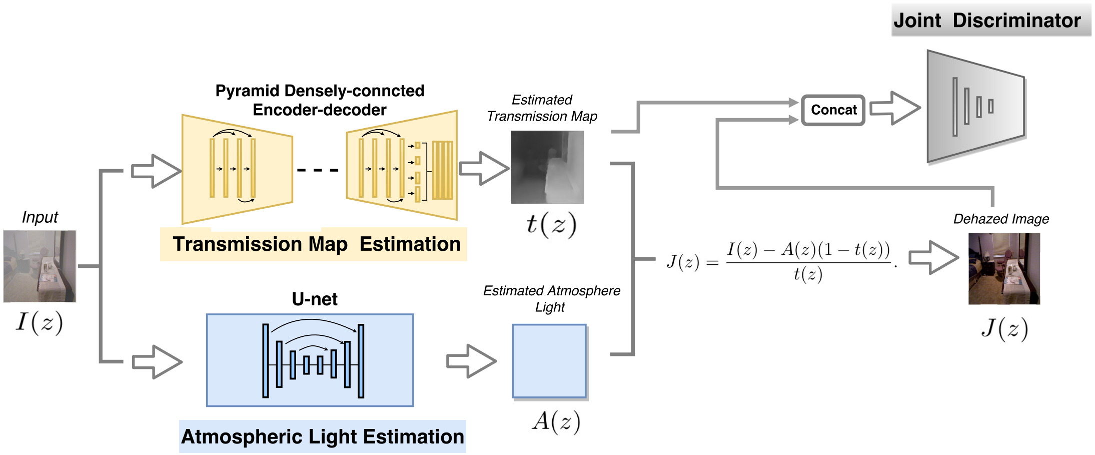


Figure 6.2: An overview of the proposed DCPDN image dehazing method. DCPDN consists of four modules: 1. Pyramid densely connected transmission map estimation net. 2. Atmospheric light estimation net. 3. Dehazing via Eq2.4. 4. Joint discriminator. We first estimate the transmission map using the proposed pyramid densely-connected transmission estimation net, followed by prediction of atmospheric light using the U-net structure. Finally, using the estimated transmission map and the atmospheric light we estimate the dehazed image via Eq. 2.4.

determine whether the paired samples (i.e. transmission map and dehazed image) are from the data distribution or not.

- Extensive experiments are conducted on two synthetic datasets and one real-world image dataset. In addition, comparisons are performed against several recent state-of-the-art approaches. Furthermore, an ablation study is conducted to demonstrate the improvements obtained by different modules in the proposed network.

6.2 Proposed Method

The proposed DCPDN network architecture is illustrated in Figure 6.2 which consists of the following four modules: 1) Pyramid densely connected transmission map estimation net, 2) Atmosphere light estimation net, 3) Dehazing via Eq. 2.4, and 4) Joint discriminator. In what follows, we explain these modules in detail.

Pyramid Densely Connected Transmission Map Estimation Network. Inspired by the previous methods that use multi-level features for estimating the transmission map [113, 13, 136, 2, 78], we propose a densely connected encoder-decoder structure that makes use of the features from multiple layers of a CNN, where the dense block is used as the basic structure. The reason to use dense block lies in that it can maximize the information flow along those features and guarantee better convergence via connecting all layers. In addition, a multi-level pyramid pooling module is adopted to refine the learned features by considering the 'global' structural information into the optimization [181]. To leverage the pre-defined weights of the dense-net [60], we adopt the first *Conv* layer and the first three *Dense-Blocks* with their corresponding down-sampling operations *Transition-Blocks* from a pre-trained dense-net121 as our encoder structure. The feature size at end of the encoding part is 1/32 of the input size. To reconstruct the transmission map into the original resolution, we stack five dense blocks with the refined up-sampling *Transition-Blocks* [65, 190] as the decoding module. In addition, concatenations are employed with the features corresponding to the same dimension.

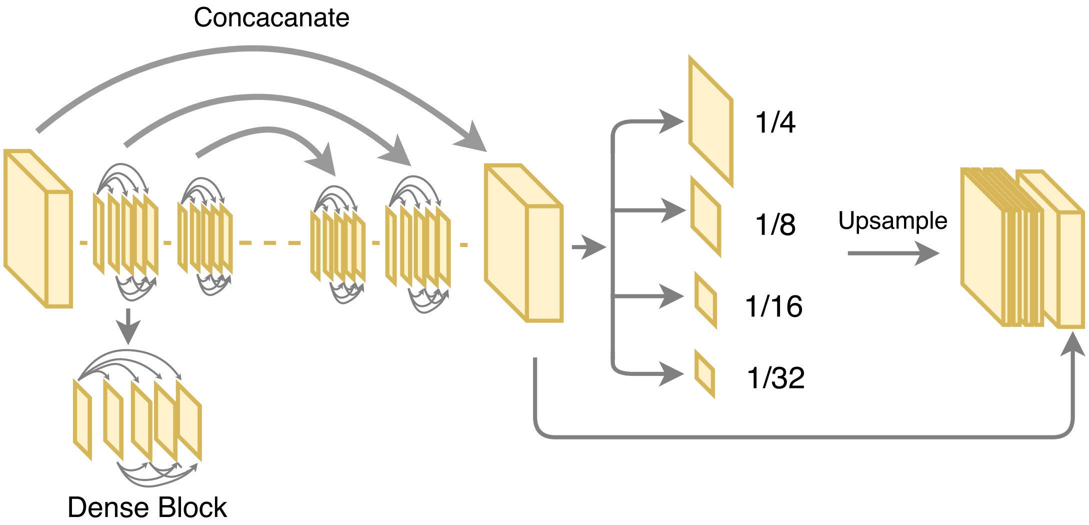


Figure 6.3: An overview of the proposed pyramid densely connected transmission map estimation network.

Even though the proposed densely connected encoder-decoder structure combines different features within the network, the result from just densely connected structure still lack of the 'global' structural information of objects with different scales. One possible reason is that the features from different scales are not used to directly estimate the final transmission map. To efficiently address this issue, a multi-level pyramid pooling block is adopted to make sure that features from different scales are embedded in the final result. This is inspired by the use of global context information in classification and segmentation tasks [181, 166, 51]. Rather than taking very large pooling size to capture more global context information between different objects [181], more 'local' information to characterize the 'global' structure of each object is needed. Hence, a four-level pooling operation with pooling sizes 1/32, 1/16, 1/8 and 1/4 is adopted. Then, all four level features are up-sampling to original feature size and are concatenated back with the original feature before the final estimation. Fig 6.3 gives an overview of the proposed pyramid densely connected transmission map estimation network.

Atmospheric Light Estimation Network. Following the image degradation model (2.3), we assume that the atmospheric light map A is homogeneous [50, 13]. Similar to previous works, the predicted atmospheric light A is uniform for a given image. In other words, the predicted A is a 2D-map, where each pixel $A(z)$ has the same value (eg. $A(z) = c$, c is a constant). As a result, the ground truth A is of the same feature size as the input image and the pixels in A are filled with the same value. To estimate the atmospheric light, we adopt a 8-block U-net [119] structure, where the encoder is composed of four *Conv-BN-Relu* blocks and the decoder is composed of symmetric *Dconv-BN-Relu* block [1].

Dehazing via (2.3). To bridge the relation among the transmission map, the atmospheric light and the dehazed image and to make sure that the whole network structure is jointly optimized for all three tasks, we directly embed (2.3) into the overall optimization framework. An overview of the entire DCPDN structure is shown in Fig 6.1.

[1] Con: Convolution, BN: Batch-normalization [62] and Dconv: Deconvolution (transpose convolution).

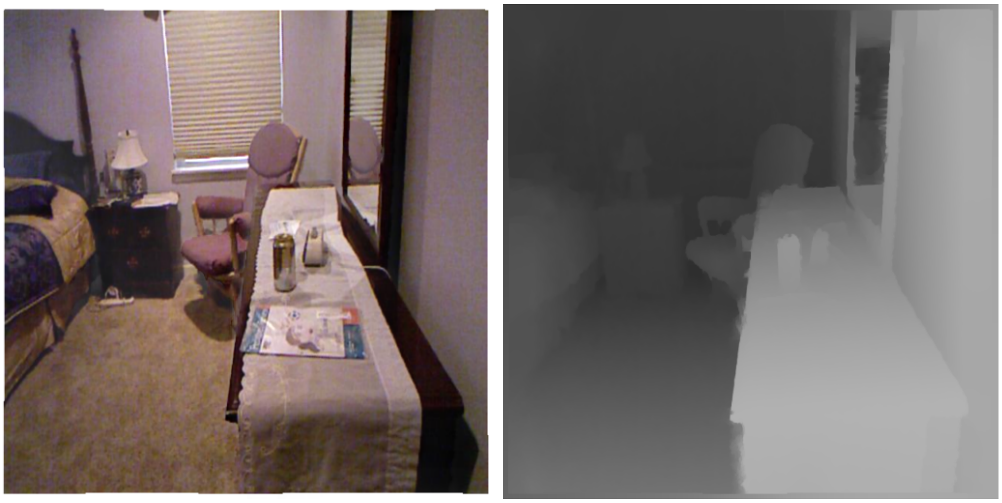


Figure 6.4: Left: a dehazed image. Right: The transmission map used to produce a hazy image from which the dehazed image on the left was obtained.

6.2.1 Joint Discriminator Learning

Let G_t and G_d denote the networks that generate the transmission map and the dehazed result, respectively. To refine the output and to make sure that the estimated transmission map $G_t(I)$ and the dehazed image $G_d(I)$ are indistinguishable from their corresponding ground truths t and J, respectively, we make use of a GAN [47] with novel joint discriminator.

It can be observed from (2.3) and also Figure 6.4 that the structural information between the estimated transmission map $\hat{t} = G_t(I)$ and the dehazed image $\hat{J}$ are highly correlated. Hence, in order to leverage the dependency in structural information between these two modalities, we introduce a joint discriminator to learn a joint distribution to decide whether the corresponding pairs (*transmission map*, *dehazed image*) are real or fake. By leveraging the joint distribution optimization, the structural correlation between them can be better exploited. Similar to previous works, the predicted air-light A is uniform for a given image. In other words, the predicted air-light A is a 2D-map, where each pixel $A(z)$ has the same value (eg. $A(z) = c,\ c$ is a constant).

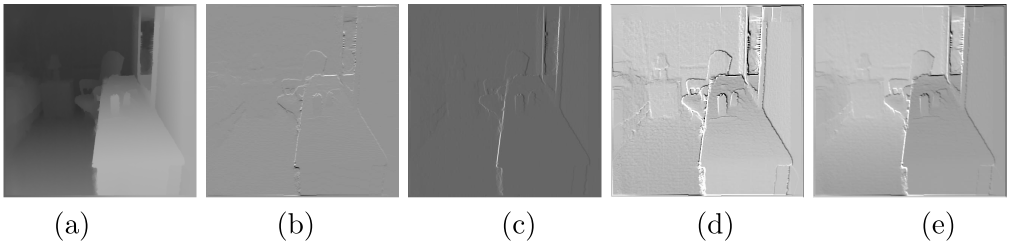


Figure 6.5: Feature visualization for gradient operator and low-level features. (a) Input transmission map. (b) Horizontal gradient output. (c) Vertical gradient output. (d) and (e) are visualization of two feature maps from relu1_2 of VGG-16 [128].

We propose the following joint-discriminator based optimization

$$\begin{aligned}\min_{G_t, G_d}\max_{D_joint} \quad & \mathbb{E}_{I\sim p_{data(I)}}[\log(1 - D_joint(G_t(I)))] + \\ & \mathbb{E}_{I\sim p_{data(I)}}[\log(1 - D_joint(G_d(I)))] + \\ & \mathbb{E}_{t,J\sim p_{data(t,J)}}[\log D_{joint}(t, J))].\end{aligned} \tag{6.1}$$

In practice, we concatenate the dehazed image with the estimated transmission map as a pair sample and then feed it into the discriminator.

6.2.2 Edge-preserving Loss

It is commonly acknowledged that the Euclidean loss ($L2$ loss) tends to blur the final result. Hence, inaccurate estimation of the transmission map with just the $L2$ loss may result in the loss of details, leading to the halo artifacts in the dehazed image [61]. To efficiently address this issue, a new edge-preserving loss is proposed, which is motivated by the following two observations. 1) Edges corresponds to the discontinuities in the image intensities, hence it can be characterized by the image gradients. 2) It is known that low-level features such as edges and contours can be captured in the shallow (first several) layers of a CNN structure [162]. In other words, the first few layers function as an edge detector in a deep network. For example, if the transmission map is fed into a pre-defined VGG-16 [128] model and then certain features from the output of layer relu1_2 are visualized, it can be clearly observed that the edge information being preserved in the corresponding feature maps (see Figure 6.5).

Based on these observations and inspired by the gradient loss used in depth estimation [140, 80] as well as the use of perceptual loss in low-level vision tasks [67, 171], we

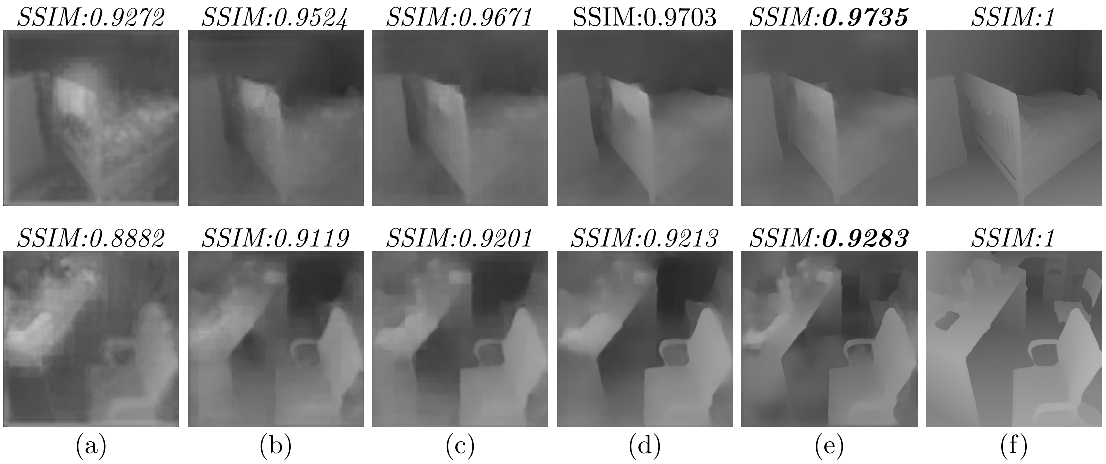


Figure 6.6: Transmission map estimation results using different modules. (a) DED; (b). DED-MLP; (c).DED-MLP-GRA; (d). DED-MLP-EP; (e). DCPDN; (f) Target. It can be observed that the multi-level pooling module is able to refine better global structure of objects in the image (observed from (a) and (b)), the edge-preserving loss can preserve much sharper edges (comparing (b), (c) and (d)) and the final joint-discriminator can better refine the detail for small objects (comparing (d) and (e)).

propose a new edge-preserving loss function that is composed of three different parts: $L2$ loss, two-directional gradient loss, and feature edge loss, defined as follows

$$L^E = \lambda_{E,l_2} L_{E,l_2} + \lambda_{E,g} L_{E,g} + \lambda_{E,f} L_{E,f}, \tag{6.2}$$

where L^E indicates the overall edge-preserving loss, L_{E,l_2} indicates the L2 loss, $L_{E,g}$ indicates the two-directional (horizontal and vertical) gradient loss and $L_{E,f}$ is the feature loss. $L_{E,g}$ is defined as follows

$$\begin{aligned} L_{E,g} = & \sum_{w,h} \|(H_x(G_t(I)))_{w,h} - (H_x(t))_{w,h}\|_2 \\ & + \|(H_y(G_t(I)))_{w,h} - (H_y(t))_{w,h}\|_2, \end{aligned} \tag{6.3}$$

where H_x and H_y are operators that compute image gradients along rows (horizontal) and columns (vertical), respectively and $w \times h$ indicates the width and height of the output feature map. The feature loss is defined as

$$\begin{aligned} L_{E,f} = & \sum_{c_1,w_1,h_1} \|(V_1(G_t(I)))_{c_1,w_1,h_1} - (V_1(t))_{c_1,w_1,h_1}\|_2 \\ & + \sum_{c_2,w_2,h_2} \|(V_2(G_t(I)))_{c_2,w_2,h_2} - (V_2(t))_{c_2,w_2,h_2}\|_2, \end{aligned} \tag{6.4}$$

where V_i represents a CNN structure and c_i, w_i, h_i are the dimensions of the corresponding low-level feature in V_i. As the edge information is preserved in the low-level features, we adopt the layers before relu1-1 and relu2-1 of VGG-16 [128] as the edge extractors V_1 and V_2, respectively. Here, $\lambda_{E,l_2}, \lambda_{E,g}$, and $\lambda_{E,f}$ are weights to balance the loss function.

6.2.3 Overall Loss Function

The proposed DCPDN architecture is trained using the following four loss functions

$$L = L^t + L^a + L^d + \lambda_j L^j, \tag{6.5}$$

where L^t is composed of the edge-preserving loss L^E, L^a is composed of the traditional $L2$ loss in predicting the atmospheric light and L^d represents the dehazing loss, which is also composed of the $L2$ loss only. L^j, which is denoted as the joint discriminator loss [2], is defined as follows

$$L^j = -\log(D_{joint}(G_t(I)) - \log(D_{joint}(G_d(I)). \tag{6.6}$$

Here λ_j is a constant.

6.2.4 Stage-wise Learning

During experiments, we found that directly training the whole network from scratch with the complex loss (6.5) is difficult and the network converges very slowly. A possible reason may be due to the gradient diffusion caused by different tasks. For example, gradients from the de-hazed image loss may 'distract' the gradients from the loss of the transmission map initially, resulting in the slower convergence. To address this issue and to speed up the training, a stage-wise learning strategy is introduced, which has been used in different applications such as multi-model recognition [30] and feature learning [6]. Hence, the information in the training data is presented to the network gradually. In other words, different tasks are learned progressively. Firstly, we optimize each task separately by not updating the other task simultaneously. After the 'initialization' for each task, we fine-tune the whole network all together by optimizing all three tasks jointly.

[2] To address the vanishing gradients problem for the generator, we also minimize (6.6) rather than the first two rows in (6.1) [47, 46].

Table 6.1: Quantitative SSIM results for ablation study evaluated on synthetic **TestA** and **TestB** datasets.

TestA

	DED	DED-MLP	DED-MLP-GRA	DED-MLP-EP	DCPDN
Transmission	0.9555	0.9652	0.9687	0.9732	0.9776
Image	0.9252	0.9402	0.9489	0.9530	0.9560

TestB

	DED	DED-MLP	DED-MLP-GRA	DED-MLP-EP	DCPDN
Transmission	0.9033	0.9109	0.9239	0.9276	0.9352
Image	0.8474	0.8503	0.8582	0.8652	0.8746

6.3 Experimental Results

In this section, we demonstrate the effectiveness of the proposed approach by conducting various experiments on two synthetic datasets and a real-world dataset. All the results are compared with five state-of-the-art methods: He *et al.* (CVPR'09) [50], Zhu *et al* (TIP'15) [187], Ren *et al.* [113] (ECCV'16), Berman *et al.* [7, 8] (CVPR'16 and ICCP'17) and Li *et al.* [78] (ICCV'17). In addition, we conduct an ablation study to demonstrate the effectiveness of each module of our network.

6.3.1 Datasets

Similar to the existing deep learning-based dehazing methods [113, 13, 78, 174], we synthesize the training samples {*Hazy /Clean /Transmission Map /Atmosphere Light*} based on (2.3). During synthesis, four atmospheric light conditions $A \in [0.5, 1]$ and the scattering coefficient $\beta \in [0.4, 1.6]$ are randomly sampled to generate their corresponding hazy images, transmission maps and atmospheric light maps. A random set of 1000 images are selected from the NYU-depth2 dataset [95] to generate the training set. Hence, there are in total 4000 training images, denoted as **TrainA**. Similarly, a test dataset **TestA** consisting of 400 (100×4) images also from the NYU-depth2 are obtained. We ensure that none of the testing images are in the training set. To demonstrate the generalization ability of our network to other datasets, we synthesize 200 {*Hazy /Clean /Transmission Map /Atmosphere Light*} images from both the Middlebury stereo database (40) [124] and also the Sun3D dataset (160) [130] as the **TestB** set.

Table 6.2: Quantitative SSIM results on the synthetic **TestA** dataset.

	Input	He. *et al.* [50] (CVPR'09)	Zhu. *et al.* [187] (TIP'15)	Ren. *et al.* [113] (ECCV'16)	Berman. *et al.* [7, 8] (CVPR'16)	Li. *et al.* [78] (ICCV'17)	DCPDN
Transmission	N/A	0.8739	0.8326	N/A	0.8675	N/A	**0.9776**
Image	0.7041	0.8642	0.8567	0.8203	0.7959	0.8842	**0.9560**

Table 6.3: Quantitative SSIM results on the synthetic **TestB** dataset.

	Input	He. *et al.* [50] (CVPR'09)	Zhu. *et al.* [187] (TIP'15)	Ren. *et al.* [113] (ECCV'16)	Berman. *et al.* [7, 8] (CVPR'16)	Li. *et al.* [78] (ICCV'17)	DCPDN
Transmission	N/A	0.8593	0.8454	N/A	0.8769	N/A	**0.9352**
Image	0.6593	0.7890	0.8253	0.7724	0.7597	0.8325	**0.8746**

6.3.2 Training Details

We choose $\lambda_{E,l_2} = 1$, $\lambda_{E,g} = 0.5$, $\lambda_{E,f} = 0.8$ for the loss in estimating the transmission map and $\lambda_j = 0.25$ for optimizing the joint discriminator. During training, we use ADAM as the optimization algorithm with learning rate of 2×10^{-3} for both generator and discriminator and batch size of 1. All the training samples are resized to 512×512. We trained the network for 400000 iterations. All the parameters are chosen via cross-validation.

6.3.3 Ablation Study

In order to demonstrate the improvements obtained by each module introduced in the proposed network, we perform an ablation study involving the following five experiments: 1) Densely connected encoder decoder structure (**DED**), 2) Densely connected encoder decoder structure with multi-level pyramid pooling (**DED-MLP**), 3) Densely connected encoder decoder structure with multi-level pyramid pooling using L2 loss and gradient loss (**DED-MLP-GRA**), 4) Densely connected encoder decoder structure with multi-level pyramid pooling using edge-preserving loss (**DED-MLP-EP**), 5) The proposed DCPDN that is composed of densely connected encoder decoder structure with multi-level pyramid pooling using edge-preserving loss and joint discriminator (**DCPDN**). [3]

The evaluation is performed on the synthesized **TestA** and **TestB** datasets. The SSIM results averaged on both estimated transmission maps and dehazed images for the various configurations are tabulated in Table 6.1. Visual comparisons are shown in

[3] The configuration 1) **DED** and 2) **DED-MLP** are optimized only with $L2$ loss.

the Fig 6.6. From Fig 6.6, we make the following observations: 1) The proposed multi-level pooling module is able to better preserve the 'global' structural for objects with relatively larger scale, compared with (a) and (b). 2) The use of edge-preserving loss is able to better refine the edges in the estimated transmission map, compared with (b), (c) and (d). 3) The final joint-discriminator can further enhance the estimated transmission map by ensuring that the fine structural details are captured in the results, such as details of the small objects on the table shown in the second row in (e). The quantitative performance evaluated on both **TestA** and **TestB** also demonstrate the effectiveness of each module.

6.3.4 Comparison with state-of-the-art Methods

To demonstrate the improvements achieved by the proposed method, it is compared against the recent state-of-the-art methods [50, 187, 113, 7, 8, 78]. on both synthetic and real datasets.

Evaluation on synthetic dataset: The proposed network is evaluated on two synthetic datasets **TestA** and **TestB**. Since the datasets are synthesized, the ground truth images and the transmission maps are available, enabling us to evaluate the performance qualitatively as well as quantitatively. Sample results for the proposed method and five recent state-of-the-art methods, on two sample images from the test datasets are shown in Figure 6.7. It can be observed that even though previous methods are able to remove haze from the input image, they tend to either over dehaze or under dehaze the image making the result darker or leaving some haze in the result. In contrast, it can be observed from our results that they preserve sharper contours with less color distortion and are more visually closer to the ground-truth. The quantitative results, tabulated in Table 6.2 and Table 6.3 [4], evaluated on both **TestA** and **TestB** also demonstrate the effectiveness of the proposed method.

Evaluation on a real dataset: To demonstrate the generalization ability of the proposed method, we evaluate the proposed method on several real-world hazy images

[4] N/A: Code released is unable to estimate the transmission map.

provided by previous methods and other challenging hazy images downloaded from the Internet.

Results for four sample images obtained from the previous methods [113, 13, 35] are shown in Figure 6.8. As revealed in Figure 6.8, methods of He *et al.* [50] and Ren *et al.* [113] (observed on the fourth row) tend to leave haze in the results and methods of Zhu *et al.* [187] and Li *et al.* [78](shown on the second row) tend to darken some regions (notice the background wall). Methods from Berman *et al.* [7, 8] and our method have the most competitive visual results. However, by looking closer, we observe that Berman *et al.* [7, 8] produce unrealistic color shifts such as the building color in the fourth row. In contrast, our method is able to generate realistic colors while better removing haze. This can be seen by comparing the first and the second row.

We also evaluate on several hazy images downloaded from the Internet. The dehazed results are shown in Figure 6.9. It can be seen from these results that outputs from He *et al.* [50] and Berman *et al.* [7, 8] suffer from color distortions, as shown in the second and third rows. In contrast, our method is able to achieve better dehazing with visually appealing results.

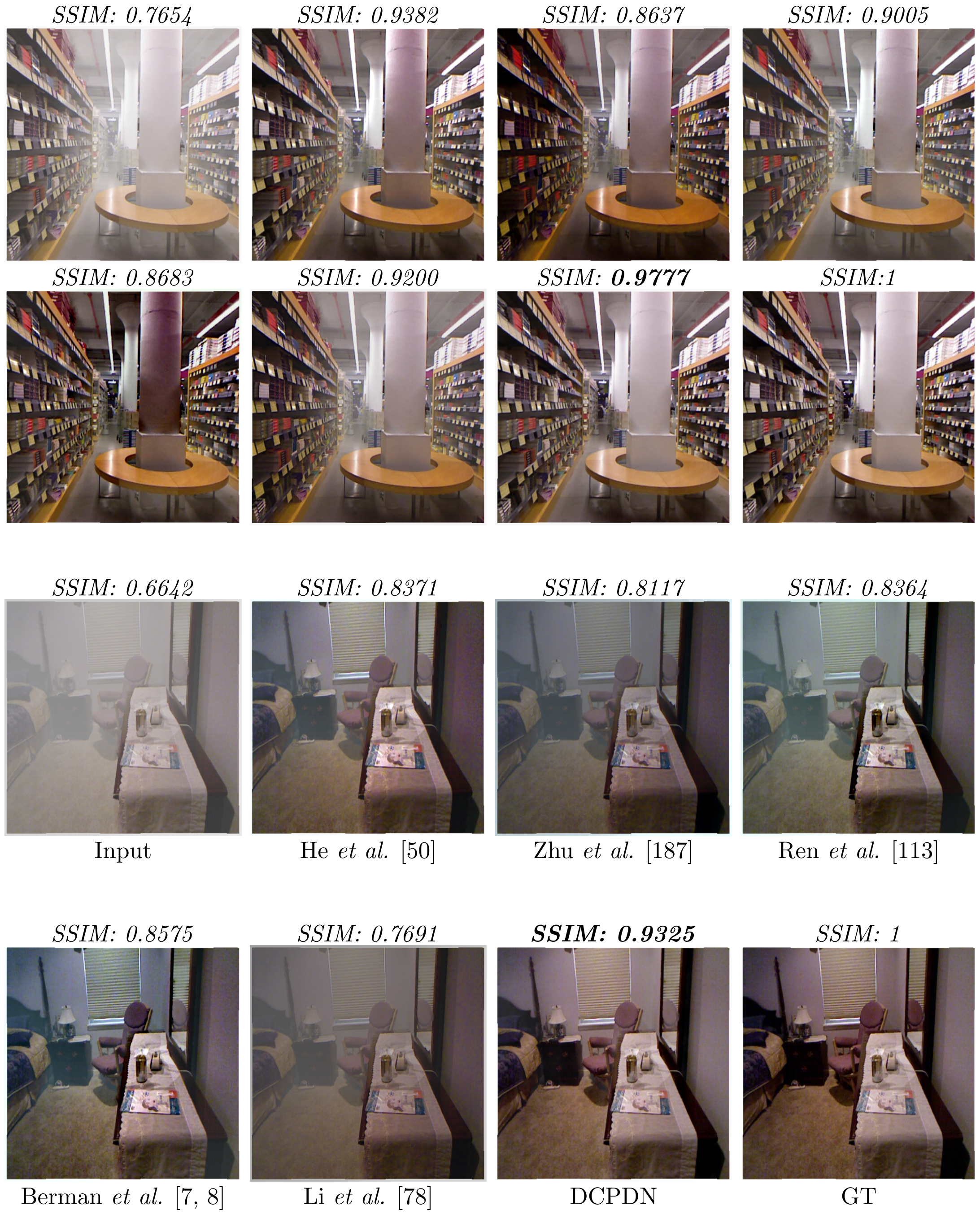


Figure 6.7: Dehazing results from the synthetic test datasets **TestA** (first row) and **TestB** (second row).

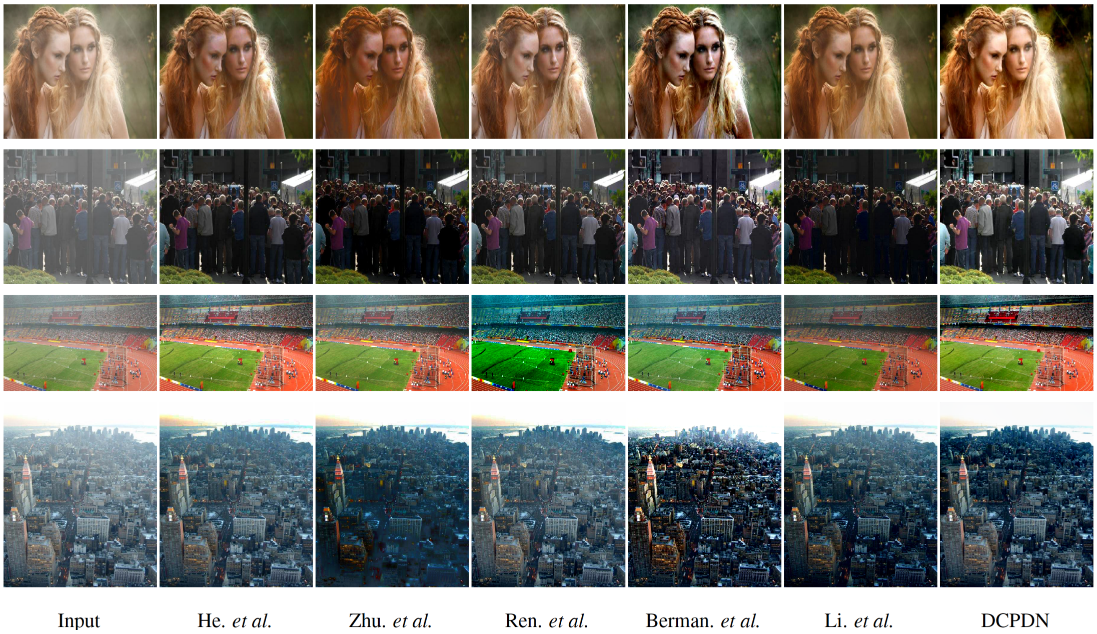


Figure 6.8: Dehazing results evaluated on real-world images released by the authors of previous methods.

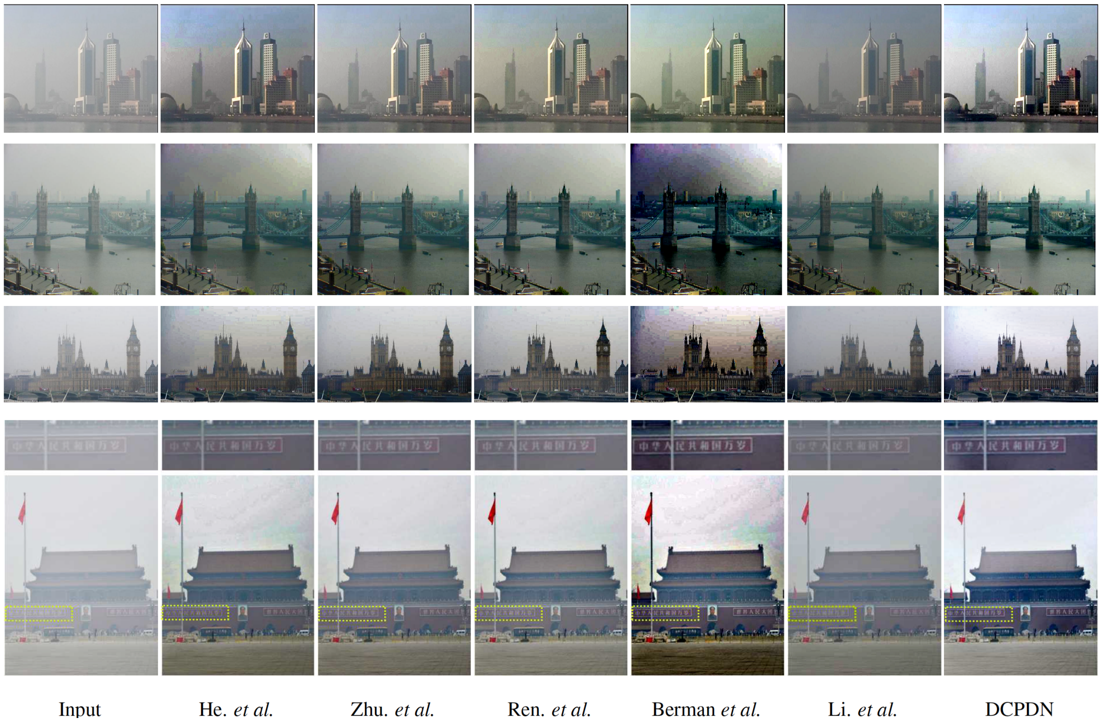


Figure 6.9: Dehazing results evaluated on real-world images downloaded from the Internet.

Chapter 7

Synthesis of High-Quality Visible Faces from Polarimetric Thermal Faces using Generative Adversarial Networks

It has been shown that faces captured in polarimetric (or conventional) thermal and visible domain are quite different and makes cross-domain face verification highly challenging. To effectively bridge the gap between these two different modalities, we propose a GAN-based multi-stream feature-level fusion technique to synthesize high-quality visible images from prolarimetric thermal images. The proposed network consists of a generator sub-network, constructed using an encoder-decoder network based on dense residual blocks, and a multi-scale discriminator sub-network. The generator network is trained by optimizing an adversarial loss in addition to a perceptual loss and an identity preserving loss to enable photo realistic generation of visible images while preserving discriminative characteristics. An extended dataset consisting of polarimetric thermal facial signatures of 111 subjects is also introduced. Multiple experiments evaluated on different experimental protocols demonstrate that the proposed method achieves state-of-the-art performance. Code will be made avaialbe at https://github.com/hezhangsprinter.

7.1 Introduction

Face is one of the most widely used biometrics for person recognition. Various face recognition systems have been developed over the last two decades. Recent advances in machine learning and computer vision methods have provided robust frameworks that achieve significant gains in performance of face recognition systems [134], [125], [16]. Deep learning methods, enabled by the vast improvements in processing hardware coupled with the ubiquity of face data and algorithmic development, have led to significant improvements in face recognition accuracy, particularly in unconstrained face imagery [108], [17], [109].

Even though these methods are able to address many challenges and have even achieved human-expert level performance on challenging databases such as the low-resolution, pose variation and illumination variation to some extent [138], [104], [16], [25], [108], they are specifically designed for recognizing face images that are collected near-visible spectrum. Hence, they often do not perform well on the face images captured from other domains such as thermal [117], [172], [55], [56], infrared [71], [96] or millimeter wave [44], [45] due to significant phenomenological differences as well as a lack of sufficient training data.

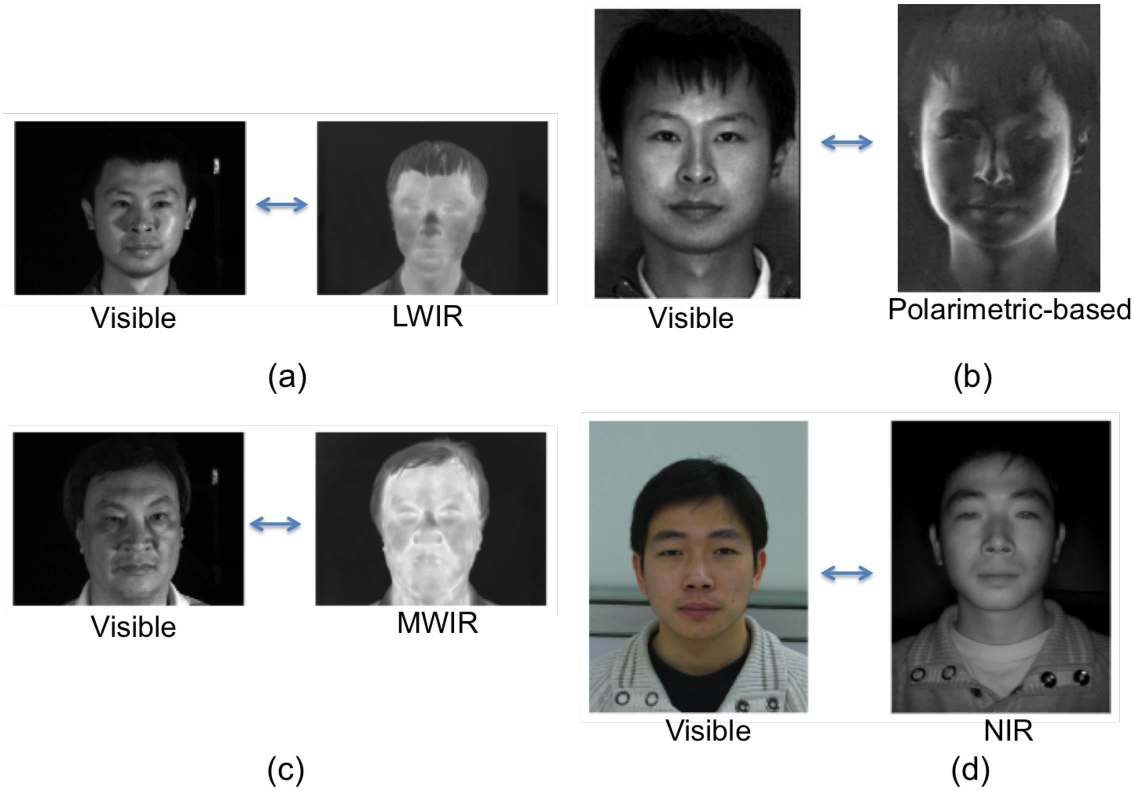


Figure 7.1: Examples of (a) visible-LWIR pair [116], (b) visible-polarimetric pair [127], (c) visible-MWIR pair [116], and (d) visible-NIR pair [116].

Thermal imaging has been proposed for night-time and low-light face recognition when external illumination is not practical due to various collection considerations. The infrared spectrum can be divided into a reflection dominated region consisting of the near infrared (NIR) and shortwave infrared (SWIR) bands, and an emission dominated thermal region consisting of the midwave infrared (MWIR) and longwave infrared (LWIR) bands [118]. In particular, recent works have been proposed to use the polarization-state information of thermal emissions to enhance the performance of thermal face recognition [56], [117], [127], [172]. It has been shown that polarimetric-thermal images capture geometric and textural details of faces that are not present in the conventional thermal facial imagery [127]. As a result, the use of polarization-state information can perform better than using only conventional thermal imaging for face recognition.

Thermal face imagery, which can be acquired passively at night, but are not carefully maintained in biometric-enabled watch lists, must be compared with visible-light face images to enable face recognition in low lighting conditions. Distributional change between thermal and visible images makes thermal-to-visible face recognition very challenging (see Figure 7.1). Various methods have been developed in the literature to bridge this gap, seeking to develop a cross-domain face recognition algorithm [121, 55, 122, 118, 63]. In particular, methods that synthesize visible faces from thermal facial signatures have gained traction in recent years [117], [172]. One of the advantages of face synthesis is that once the face images are synthesized in the visible domain, any off-the-shelf face matching algorithm can be used to match the synthesized image to the gallery of visible images.

Previous approaches utilize either a two-step procedure (visible feature estimation and visible image reconstruction) [117] or a fusion technique where different Stokes images are concatenated and used as a multi-channel input [172] to synthesize the visible image given the corresponding polarimetric signatures. Though these methods are able to effectively synthesize photo-realistic visible face images, the results are still far from optimal. One possible reason lies in that these methods treat polarimetric thermal images as multi-channel inputs without any additional attempts to capture multi-modal information inherently present in the different Stokes (modalities) of these thermal domain images. Hence, in order to efficiently leverage the multi-modal information provided by the polarimetric thermal images, we propose a novel multi-stream feature-level fusion method for synthesizing visible images from thermal domain using recently proposed Generative Adversarial Networks [47].

The proposed GAN-based network consists of a generator, a discriminator sub-network and a deep guided sub-network (see Figure 7.3). The generator is composed of a multi-stream encoder-decoder network based on dense-residual blocks, the discriminator is designed to capture features at multiple-scales for discrimination and the deep guided sub-net aims to guarantee that the encoded features contain geometric and texture information to recover the visible face. To further enhance the network's performance, it is guided by perceptual loss and an identity preserving loss in addition to adversarial loss. Once the face images are synthesized, any off-the-shelf face recognition and verification networks trained on the visible-only face data can be used for matching. Figure 7.2 illustrates the differences between visible and polarimetric thermal images. In addition, this figure also presents the photo-realistic and identity-preserving results obtained from our proposed method.

In addition to developing a novel face synthesis network, we also collected an extended dataset consisting of visible and polarimetric facial signatures from 111 subjects. A subset of this dataset consisting data from 60 subjects was described in [56]. The collected polarimetric thermal facial dataset is available to computer vision and biometrics researchers to facilitate the development of cross-spectrum and multi-spectrum face recognition algorithms.

To summarize, this chapter makes the following contributions.

1. A novel face synthesis framework based on GANs is proposed which consists of a multi-stream generator and multi-scale discriminator.

2. To embed the identity information into the objective function and make sure that the synthesized face images are photo-realistic, a refined loss function is proposed for training the network.

3. An extended dataset consisting of visible and polarimetric data from 111 subjects is collected.

4. Detailed experiments are conducted to demonstrate improvements in the synthesis

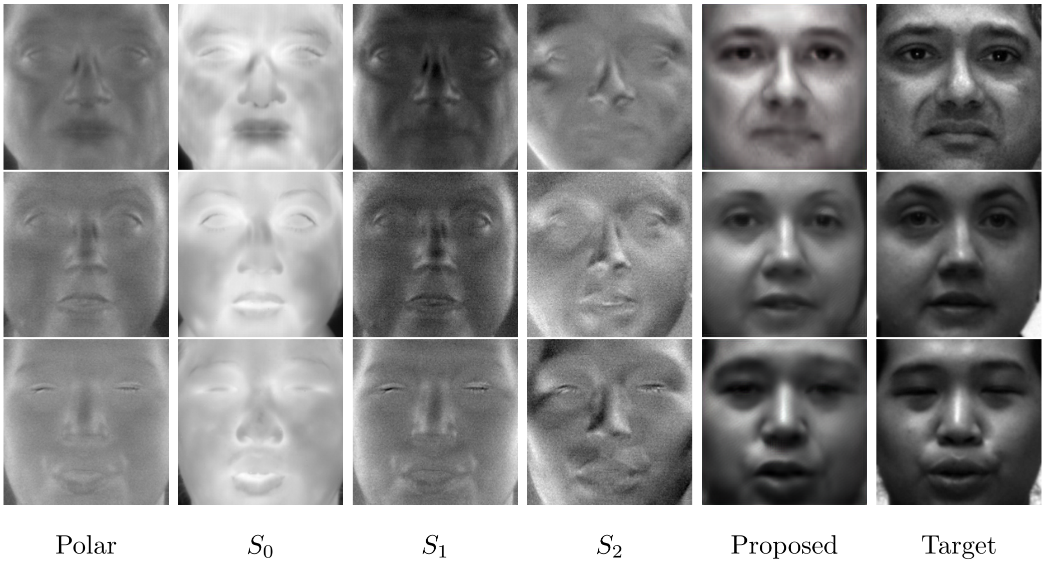


Figure 7.2: Sample results of the proposed method. (a) Input Polarimetric image. (b) Input S_0 image. (c) Input S_1 image. (d) Input S_2 image (e) Results from the proposed method, and (f) Target image.

results. Further, three ablation studies are conducted to verify the effectiveness of iterative synthesis and various loss function.

7.2 Background and Related Work

In this section, we give a brief overview of polarimetric thermal imaging.

7.2.1 Polarimetric Thermal Imaging

Polarimetric thermal imaging uses advanced optics and sensor technology to measure the polarization state of light. While traditional imaging exploits the intensity of light and infrared imaging exploits the frequency of light, polarimetric imaging exploits the orientation of light. Natural visible light exhibits no preferred polarization state. If natural light is either transmitted across a boundary from one medium to another, or is reflected by the boundary (i.e., the material is opaque) a preferential polarization state (usually linear) may occur.

This induced polarization change is a directional quantity and is a function of the angle between the surface normal and the transmitted/reflected ray. For example, unpolarized sunlight reflecting off an air-water interface results in an induced linear polarization state that is orthogonal to the plane of reflection, as defined by the surface normal and the reflected ray. A similar phenomena exists when considering light energy in the "thermal" infrared (IR) part of the spectrum, e.g., MidIR (3-5μm) and/or LWIR (8-12μm). For induced polarization in the thermal IR, the radiation is treated as either emitted and/or reflected from a surface boundary. It is this interaction at the boundary that results in an induced net linear polarization state, similar to situation seen for visible light. By capturing this thermal radiance using an IR polarimetric camera, one can exploit the additional polarization based information and reconstruct a 3D surface from a 2D polarimetric image.

Polarimetric imaging sensors capture polarization-state information by filtering light at different orientations. This is traditionally done using a rotating element [139] (i.e., division of time), but other approaches exist, such as micro-grid polarizers [139] (i.e., division of focal plane array). In essence, polarization-state information is captured at four orientations, I_0, I_{90}, I_{45}, and I_{135}. The I_0 and I_{90} measurements represent horizontal and vertical polarized light and I_{45} and I_{135} capture diagonally polarized light with respect to the camera axis. A stack of 2-D images captured using a polarimeter is summarized by Stokes images, as defined in [49], which highlight various edges of the face. A Degree of Linear Polarization (DoLP) image can be produced from the Stokes images which highlights geometric and textural features of the face. These Stokes images are illustrated in Figure 7.2 for three subjects with corresponding visible-spectrum facial signatures. The S_0 image is a total intensity polarimetric image and is representative of what a conventional thermal imager (i.e., without linear polarizer) would capture. S_1, and S_2 illustrate the additional details provided by polarimetric imaging. In this chapter, we refer to *Polar* as the three channel polarimetric image with S_0, S_1 and S_2 as the three channels.

7.3 Proposed Method

As discussed earlier, a polarimetric sample consists of three different Stokes images (S_0, S_1 and S_2), where S_0 represents the conventional thermal image and S_1 and S_2 represent the horizontal/vertical and diagonal polarization-state information, respectively. Unlike

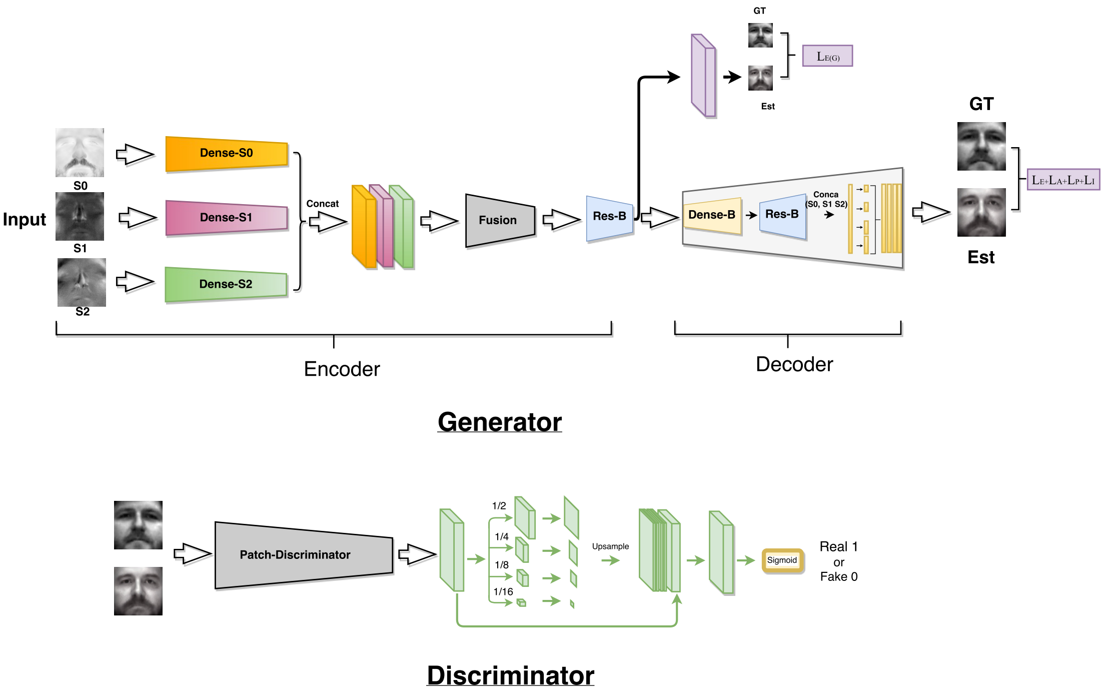


Figure 7.3: An overview of the proposed GAN-based multi-stream encoder-decoder network. The generator contains a multi-stream feature-level fusion encoder-decoder network. In addition, a deep-guided subnet is stacked at the end of the encoding part. The discriminator is composed of a multi-scale patch-discriminator structure.

traditional three-channel RGB images where each channel contains exactly the same structural content information, the S_0, S_1, S_2 images contain different geometric and texture information. For example, as shown in the first row of Figure 7.2, S_0 is able to capture the mustache information, which is not captured in S_1 and S_2. On the other hand, S_0 does not capture some of the other texture and geometric details such as wrinkles, which are well-preserved in S_1 and S_2. In other words, the Stokes images individually capture different facial features and when combined together they provide complementary information. Hence, it is important to fully utilize the information from all three Stokes images to effectively synthesize a visible face image.

Previous methods have attempted to utilize this information by exploiting input level fusion, where three Stokes images are concatenated together as a three-channel input [117, 172]. Even though the three-channel concatenation in the input level is able to generate better visible face results by bringing in the geometric and texture differences preserved in these three modalities as compared with using just a single

Stokes image as input (eg. S_0), the results are still far from optimal [172]. A potential reason is that input level fusion or mere concatenation of different Stokes images may not be sufficient enough to exploit the different geometric and texture information present in these modalities [1]. To efficiently address this problems and generate better photo-realistic visible face images, a multi-stream feature-level fusion structure is proposed in this chapter. Specifically, different encoder structures are leveraged to encode each Stokes image separately and then the embedded features from each encoder are fused together via a fusion block for further visible face reconstruction (i.e. decoding).

Synthesizing photo-realistic visible images from polarimetric images (or even any single Stokes image) is an extremely challenging problem due to information differences caused by phenomenology between polarimetric thermal images and visible images. As shown in Figure 7.2, polarimetric thermal images fail to capture fine details such as edges and gradients as compared to visible images. Due to the absence of these sharp details in the polarimetric images, reconstructing visible images from them requires joint modeling of the images from these two modalities. To efficiently leverage the training samples and guarantee better convergence with less gradient vanishing for such joint modeling, a novel dense residual structure is proposed in this chapter. Furthermore, a multi-scale patch-discriminator is utilized to classify between real and synthesized images at multiple scales. By performing the discrimination at multiple scales, we are able to effectively leverage contextual information in the input image, resulting in better high-frequency details in the reconstructed image.

To summarize, we propose a multi-stream feature-level fusion GAN structure (see Figure 7.3) which consists of the following components:

(1) Multi-stream densely-connected encoder.

(2) Deep guidance sub-network.

(3) Single-stream dense residual decoder.

(4) Multi-scale discriminator.

[1] Input level fusion can be regarded as an extreme case for low-level feature fusion, where low-level (features from shallow layers) features often preserve edge information rather than semantic mid-level or high-level class-specific information [162].

In what follows, we describe these components in detail.

7.3.1 Multi-stream Feature-level Fusion Generator

The proposed feature-level fusion method is inspired by the face dis-entangled representation work proposed by Peng *et al.* and Tran *et al.* in [104, 138, 102], where the encoded feature representations are explicitly disentangled into separate parts representing different facial priors such as identity, pose and gender. Rather than leveraging the supervised label information to enforce the disentangling factor in the embedded features, each encoder structure in the proposed method inherently learns to characterize different geometric and texture information that is captured in the Stokes images. This information is then combined with a residual block-based fusion network, followed by a decoder network, consisting of a dense network and a residual network, to reconstruct visible domain faces from the fused feature maps. Furthermore, a deep-guided sub-network is leveraged at the end of the encoding part to ensure that the encoded features preserve geometric and texture information.

Multi-stream Densely-connected Encoding. The encoder consists of three streams of sub-networks, with each sub-network having the same structure [2]. Each stream processes a particular input Stokes image. Basically, each stream is composed of a convolutional layer, rectified linear unit (ReLU) and a max-pooling operator at the front followed by three dense-blocks [60] which are stacked at the end. Each dense-block is followed by a transition-down block that performs down-sampling. Each layer $\mathbf{D}_j$ in a dense block can be represented as

$$\mathbf{D}_j = T(cat[D_1, D_2, ..., D_{j-1}]), \tag{7.1}$$

where $T(\cdot)$ indicates the combination of Batch Normalization (BN) [62], rectified linear unit (ReLU) and Convolution operator. Figure 7.4 gives an overview of a single stream in the multi-stream densely-connected encoding.

[2] Weights are not shared among each stream.

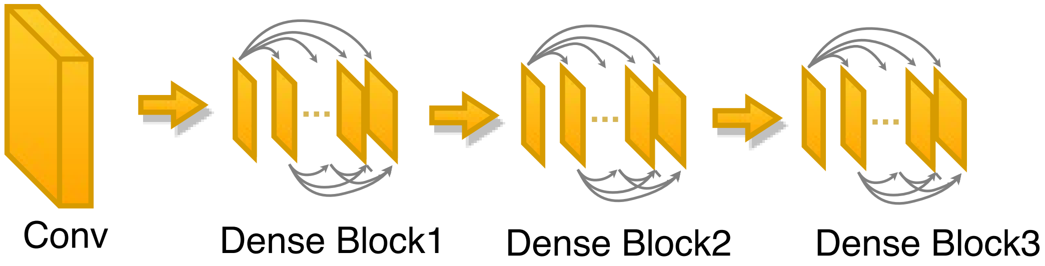


Figure 7.4: Overview of a single stream in the multi-stream densely-connected encoding part.

The dense-net blocks have a similar structure to that of Dense-net 121 structure [60], where the first dense-block contains 12 densely-connected layers, second block contains 16 densely-connected layers and the third block contains 24 densely-connected layers. The weights for each stream are initialized from the pre-trained Dense-net 121 network[60]. Feature maps from each of the three streams are of size $C \times H \times W$. These feature maps are concatenated and are forwarded to the residual-fusion block, which consists of a res-block with 1×1 convolution layer. To guarantee that the learned features contain geometric and texture facial information, a deep guidance sub-network [153] is introduced at the end of the encoding part. The deep guided sub-network is part of the network that is branching out from the end of the encoder. This sub-network is composed a 1×1 convolution layer followed by the non-linear function Tanh. Hence, the output of the guided sub network will be a three-channel RGB image with size 16×16 if the input size is 256×256.

Dense-Residual Decoder. The fused feature representations are then fed into a decoder network that is based on dense-residual decoding blocks. Specifically, each dense-residual block involves a dense-block, followed by a transition-up operator that operates as an up-sampler [189] and two res-block structures to refine the learned dense features. Once the feature maps are up-sampled to the original resolution (input resolution, e.g. 256×256), these learned features are concatenated with the three input Stokes images. Finally, a multi-level pyramid pooling block is adopted at the end of the decoding part to make sure that features from different scales are embedded in the final result. This is inspired by the use of global context information in classification and segmentation tasks [181, 167]. Rather than taking very large pooling size to capture more global context information between different objects [181], more 'local' information is leveraged here. Hence, a four-level pooling operation with sizes 1/32, 1/16, 1/8 and 1/4 are used. Then, features from all four levels are up-sampled to the original feature size and are concatenated back with the original feature maps before the final estimation. Figure 7.5 gives an overview of the proposed dense-residual decoder.

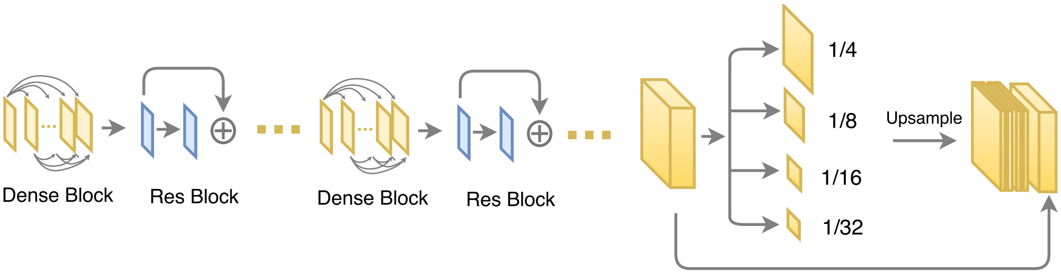


Figure 7.5: Overview of the dense-residual decoding part.

7.3.2 Multi-scale Discriminator

To ensure the synthesized visible faces are indistinguishable from real images while preserving high-frequency details, a learned multi-scale patch-discriminator sub-network is designed to decide if each input image (to the discriminator) is real or fake. Similar to the structure that was proposed in [64], a convolution layer with batch normalization and Leaky ReLU [87] activation are used as the basis throughout the patch-discriminator part. Basically, the patch-discriminator consists of the following structure:

$$\textit{CB}(K_2)\text{-}\textit{CBL}(2K_2)\text{-}\textit{CBL}(4K_2)\text{-}\textit{CBL}(8K_2)$$

where, $CBL(K_2)$ is a set of K_2-channel convolution layers followed by batch normalization and Leaky ReLU [87]. Then, a multi-scale pooling module, which pools features at different scales, is stacked at the end of the discriminator. The pooled features are then upsampled and concatenated, followed by a 1×1 convolution and a sigmoid function to produce a probability score normalized between 0 and 1. The proposed discriminator sub-network, D, is shown at the bottom of Figure 7.3.

7.3.3 Loss Functions

It is well-known that the use of Euclidean loss, L_E, alone often results in blurry results. Hence, to overcome this and to discriminate the generated visible face images from their corresponding ground truth, an adversarial loss function is employed. Even though the use of adversarial loss can generate more reasonable results compared to the L_E loss, as shown in [172], these results contain undesirable facial artifacts. To address this issue and generate visually pleasing results, perceptual loss is incorporated in our work. The perceptual loss is computed using a pre-trained VGG-16 models as discussed in [67, 165, 173, 75].

Since the ultimate goal of the our proposed synthesis method is to guarantee that human examiners or face verification systems can identify the person given his/her synthesized face images, it is also important to involve the discriminative information into consideration. Similar to the perceptual loss, we propose an identity-preserving loss that is evaluated on a certain layer of the fine-tuned VGG-Polar model. The VGG-Polar model is fine-tuned using the visible images with their corresponding labels from the newly introduced Polarimetric Visible database.

The proposed method contains the following loss functions: the Euclidean L_2 loss enforced on the reconstructed visible image, the $L_{E(G)}$ loss enforced on the guidance part, the adversarial loss to guarantee more sharp and indistinguishable results, the perceptual loss to preserve more photo realistic details and the identity loss to preserve more discriminative information for the outputs. The overall loss function is defined as follows

$$L_{\text{all}} = L_2 + L_{2(G)} + \lambda_A L_A + \lambda_P L_P + \lambda_I L_I, \tag{7.2}$$

where L_2 denotes the Euclidean loss, $L_{2(G)}$ denotes the Euclidean loss on the guidance sub-network, L_A represents the adversarial loss, L_P indicates the perceptual loss and L_I is the identity loss. Here, λ_A, λ_P and λ_I are the corresponding weights.

The L_2 and the adversarial losses are defined as follows:

$$L_2, L_{2(G)} = \sum_{w,h} \|\phi_G(S_0, S_1, S_2)^{w,h} - Y_t^{w,h}\|_2, \tag{7.3}$$

$$L_A = -\log(\phi_D(\phi_G(S_0, S_1, S_2)), \tag{7.4}$$

where S_0, S_1 and S_2 are the three different input Stokes images, Y_t is the ground truth visible image, $W \times H$ is the dimension of the input image, ϕ_G is the multi-stream feature-fusion generator sub-network G and ϕ_D is the multi-scale discriminator sub-network D.

As the perceptual loss and the identity losses are evaluated on a certain layer of the given CNN model, both can be defined as follows:

$$L_{P,I} = \sum_{c_i,w_i,h_i} \|V(\phi_G(S_0, S_1, S_2))^{c_i,w_i,h_i} - V(Y_t)^{c_i,w_i,h_i}\|_2, \tag{7.5}$$

where Y_t is the ground truth visible image, ϕ_E is the proposed generator, V represents a non-linear CNN transformation and C_i, W_i, H_i are the dimensions of a certain high level layer V, which differs for perceptual and identity losses.

7.4 Polarimetric Thermal Face Dataset

A polarimetric thermal face database of 111 subjects is used for this study, which expanded on the previously released database of 60 subjects (described in detail in Hu *et al.*, 2016 [56]). The database used in this study therefore consisted of the 60-subject database collected at the U.S. Army Research Laboratory (ARL) in 2014-2015 (referred to as Volume 1 hereinafter), and a 51-subject database collected at a Department of Homeland Security test facility (referred to as Volume 2 hereinafter). While the participants of the Volume 1 collect consisted exclusively of ARL employees, the participants of the Volume 2 collect were recruited from the local community in Maryland, resulting in more demographic diversity. Note that this extended databased is available upon request.

7.4.1 Sensors

The sensors employed to collect Volume 1 and Volume 2 were the same, consisting of a polarimetric LWIR imager and visible cameras. The LWIR polarimetric was developed by Polaris Sensor Technologies, and is based on a division-of-time spinning achromatic retarder (SAR) design which incorporated a spinning phase-retarder in conjunction with a linear wire-grid polarizer. This system has a spectral response range of 7.5-11 μm, and employed a Stirling cooler with a mercury telluride focal plane array (640×480 pixel array format). Data was recorded at 60 frames per second, using a lens with a field of view (FOV) of $10.6 \circ \times 7.9 \circ$. Four Basler Scout GigE cameras with different lens (ranging from $5\circ$ to $53\circ$) were used for Volume 1, consisting of two grayscale cameras (model # scA640-70gm; 659×494 pixel FPA) and two color cameras (model # scA640-70gc; 658×492 pixel FPA) to generate visible facial imagery at different resolutions. For Volume 2, a single Basler Scout color camera with a zoom lens was used, adjusted to produce the same facial resolution as the polarimeter.

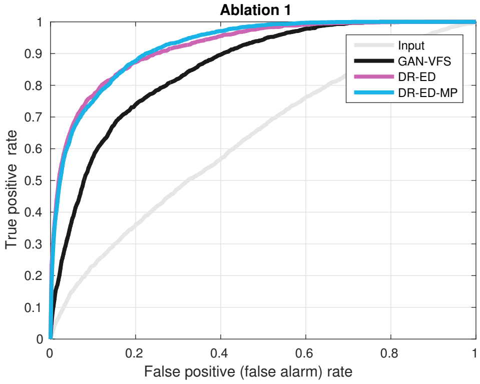


Figure 7.6: The ROC curves corresponding to **Ablation 1**.

7.4.2 Dataset

The dataset protocols for Volume 1 and Volume 2 were approved by the respective Institutional Review Boards (IRBs) where each collection occurred. The Volume 1 collection involved two experimental conditions: range and expressions. Acquisitions were made at distances of 2.5 m, 5 m, and 7.5 m. At each range, a 10 second video sequence was first collected of the subject with a neural expression, and then a 10 second

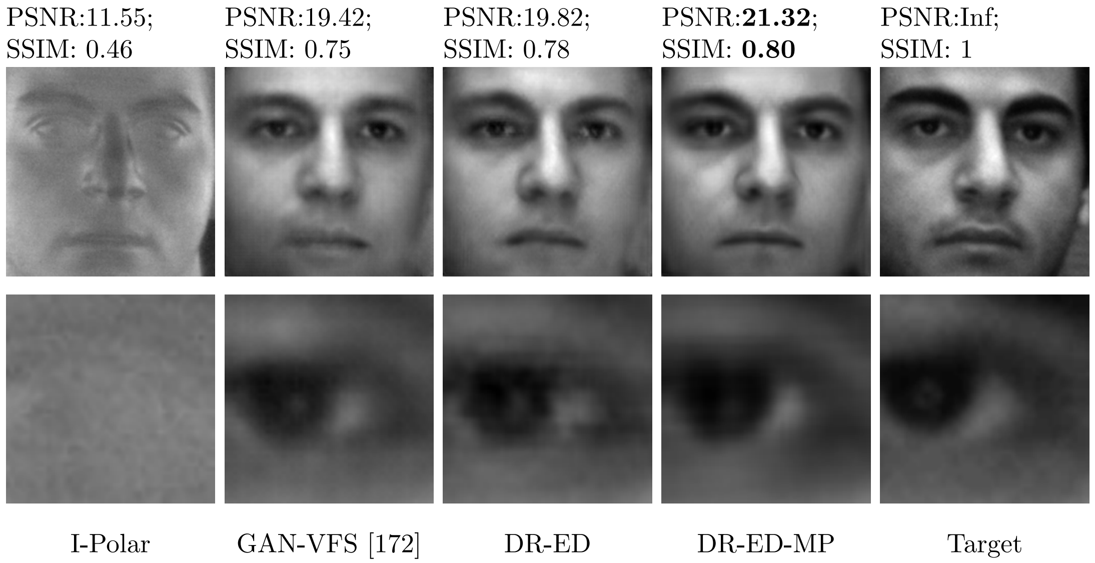


Figure 7.7: Sample results of for the **Ablation 1**. It can be observed that the dense-resisual encoder-decoder structure is able to generate better visible results and the introduced multi-level pooling module is able to preserve better structure information. Detail discussions can be found in Sec 7.5.2.

"expressions" sequence was collected as the subject counted out loud numerically from one upwards, which induced a continuous range of motions of the mouth and, to a lesser extent the eyes. In the experimental setup for Volume 1, a floor lamp was placed 1 m in front of the subject at each range to provide additional illumination.

Table 7.1: The average PSNR (dB), SSIM, EER and AUC results corresponding to different methods for **Ablation 1**.

	I-Polar	GAN-VFS [172]	DR-ED	DR-ED-MP
PSNR (dB)	11.74	18.07	18.28	**18.80**
SSIM	0.4625	0.7047	0.7128	**0.7194**
EER	41.51%	22.45%	16.51%	**15.67%**
AUC	62.93%	86.10%	91.67%	**92.55%**

The data collection setup used for Volume 2 matched that of Volume 1. However, no floor lamp was employed in the Volume 2 collect, as the DHS test facility had sufficient illumination. Furthermore, Volume 2 data was collected at a single range of 2.5 m, due to time limitations since the polarimetric face acquisition was part of a broader collection.

7.4.3 Preprocessing

The raw polarimetric thermal imagery underwent several preprocessing steps. First, a two-point non-uniformity correction (NUC) was applied on the raw data using software provided by Polaris Sensor Technologies and calibration data collected with a Mikron blackbody prior to each session. Images were sampled/extracted from the polarimetric thermal sequences. Bad pixels in the extracted images were identified, and those pixel intensities corrected via a median filter. To crop and align the facial imagery, three fiducial points (centers of the eyes, base of the nose) were first manually annotated, and an affine transform was used to normalize each face to canonical coordinates. Facial imagery was finally cropped to $m \times n$ pixels, and saved as 16-bit PNG files. The visible imagery required neither non-uniformity correction nor bad pixel correction. The same steps were used to crop and align the visible images, which were then saved as 16-bit grayscale PNG files.

7.4.4 Experimental Protocols

Even though there exist several conventional thermal-visible pair databases [31, 18], they lack the availability of the corresponding polarization state information such as S_1 and S_2. Hence, an extended database, which contains polarimetric (S_0, S_1, S_2) and visible image pairs from 111 subjects is used for evaluation in this chapter. Following the protocol defined in [117, 172], sample pairs corresponding to range 1 (baseline and expression) are used for comparisons. In particular, two different protocols are defined in this chapter for further research. To be consistent with previous methods [117, 172], the first protocol is defined as follows:

Protocol 1: The protocol 1 is evaluated on Volume 1, which contains 60 subjects, 30 subjects from Volume 1 with eight samples for each subject (in total 240 sample pairs) are used as training samples, denoted as *Train1*. Similarly, the remaining 30 subjects with eight samples for each subject (in total 240 sample pairs) are used as testing samples, denoted as *Protocol1*. All the training and testing samples are randomly chosen from the overall 60 subjects. Results are evaluated on five random splits. In Protocol 1, each split contains around 28800 pairs of templates on average (1080 positive and 27720 negative).

Protocol 2: Different from Protocol 1, the newly introduced and extended dataset with 111 subjects is used for training and testing, where 85 subjects with eight samples for each subject are randomly chosen as training samples (in total 680 sample pairs), denoted as *Train2* and the other 26 subjects are used as testing (in total 208 sample pairs), denoted as *Protocol2*. As before, results are evaluated on five random splits. In Protocol 2, each split on average contains around 21632 pairs of templates (936 positive and 20696 negative).

These protocols and splits will be made publicly available to the research community.

7.5 Experimental Results

In this section, we demonstrate the effectiveness of the proposed approach by conducting various experiments on the two defined protocols for the new polarimetric thermal dataset as described above. Once the visible images are synthesized using the proposed method, deep features can be extracted from these images using any one of many pre-trained CNNs such as VGG-face [99], Light-CNN [151], or GoogleNet [157]. In this chapter, we extract the features from the second last fully connected layer of the VGG-face network [99]. Finally, the cosine distance is used to calculate the scores. Results are compared with four state-of-the-art methods: Ben *et al.* [117], GAN-VFS [172], Pix2pix [64] and Pix2pix with BEGAN [64, 9]. In addition, three ablation studies are conducted to demonstrate the effectiveness of different modules of the proposed method. Quality of the synthesized images is evaluated using Peak Signal-to-Noise Ratio (PSNR) and Structural SIMilarity (SSIM) index [145]. The face verification performance is evaluated using the receiver operating characteristic (ROC) curve, Area Under the Curve (AUC) and Equal Error Rate (EER) measures.

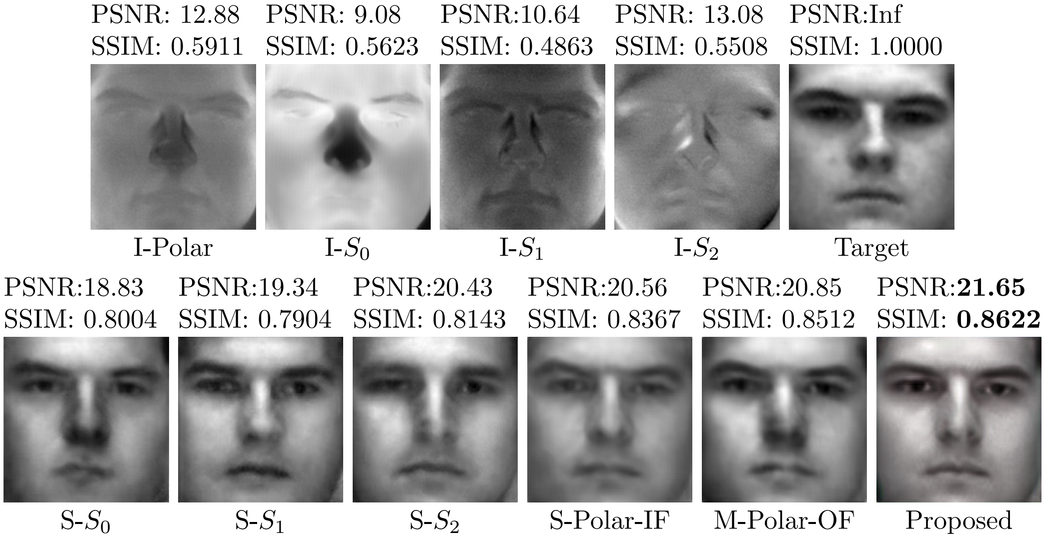


Figure 7.8: Sample results for **Ablation 2**. It can be observed that the proposed multi-stream feature-level fusion GAN is able to generate better results compared to input-level (S-Polar-IF), output-level fusion (M-Polar-OF) and also simply levering single Stokes modality. Detailed discussions can be found in Sec 7.5.2.

Table 7.2: The average PSNR (dB), SSIM, EER and AUC results corresponding to different methods for **Ablation 2**.

	I-Polar	S-S_0	S-S_1	S-S_2	S-Polar-IF	M-Polar-OF	Proposed
PSNR (dB)	11.74	17.34	17.03	17.17	18.80	18.87	**19.55**
SSIM	0.4625	0.6905	0.6852	0.6794	0.7194	0.7225	**0.7433**
EER	41.51%	23.18%	21.61%	21.56%	15.67%	15.90%	**11.78**%
AUC	62.93%	85.74%	86.64%	87.30%	92.55%	92.69%	**96.03**%

7.5.1 Implementation

The entire network is trained on a Nvidia Titan-X GPU. We choose $\lambda_A = 0.005$ for the adversarial loss, $\lambda_P = 0.8$ for the perceptual loss and $\lambda_I = 0.1$ for the identity loss. During training, we use ADAM [70] as the optimization algorithm with learning rate of 8×10^{-4} and batch size of 1 image. All the pre-processed training samples are resized to 256×256. The perceptual loss is evaluated on relu 1-1 and relu 2-1 layers in the pre-trained VGG [99] model. The identity loss is evaluated on the relu2-2 layer of the fine-tuned VGG-Polar model.

7.5.2 Ablation Study

In order to better demonstrate the effectiveness of the proposed feature-level fusion, the improvements obtained by different modules and the importance of different loss

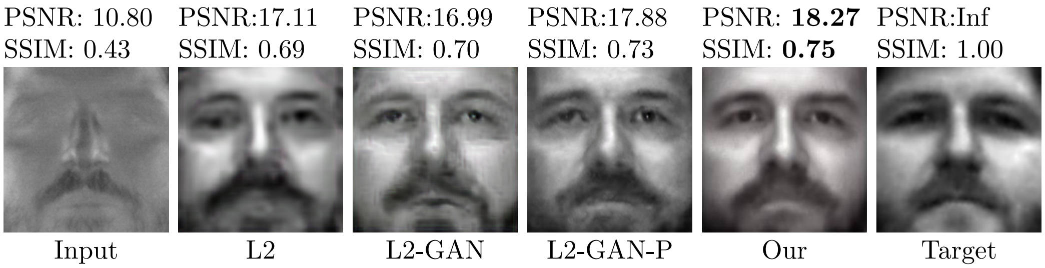


Figure 7.9: Sample results on different loss functions for **Ablation 3**.

functions in the proposed network, three ablation studies are presented in this section. All the experiments in the first two ablation studies are optimized with the same loss function discussed in Eq (2).

Ablation 1

In the first ablation study, we demonstrate the effectiveness of different modules (eg. densely connected encoder-decoder structure) in our method by conducting the following experiments. All the experimental results are evaluated using **Protocol 1** based on the polrimetric images as input:

(a) **GAN-VFS**: The GAN network proposed in [172] with polarimetric images as inputs.

(b) **DR-ED**: A single stream dense-resisual encoder-decoder structure. [3]

(c) **DR-ED-MP**: A single stream dense-resisual encoder-decoder structure with multi-level pooling.

Table 7.3: The average PSNR (dB), SSIM, EER and AUC results corresponding to different methods for **Ablation 3**.

	I-Polar	L2	L2-GAN	L2-GAN-P	Our
PSNR (dB)	11.74	17.57	17.33	18.99	**19.55**
SSIM	0.4625	0.7088	0.7115	0.7352	**0.7433**
EER	41.51%	18.07%	13.23%	11.79%	**11.78%**
AUC	62.93%	90.89%	93.64%	95.64%	**96.03%**

[3] Basically, this network is composed of one stream of the encoder part followed by the same decoder without multi-level pooling.

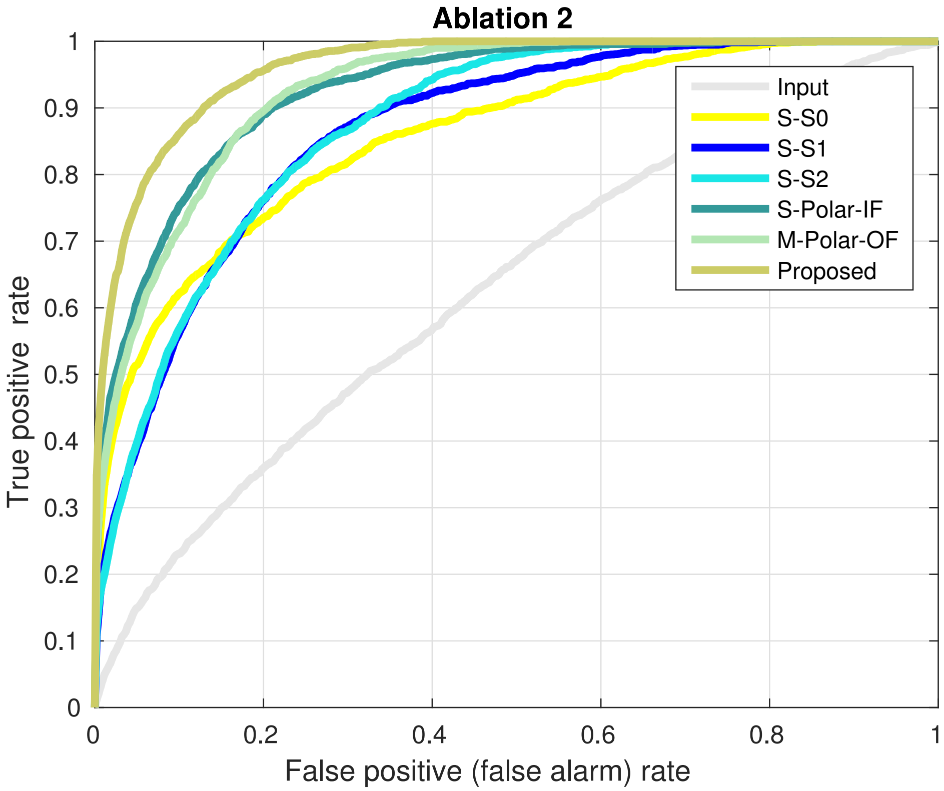


Figure 7.10: The ROC curves corresponding to **Ablation 2**.

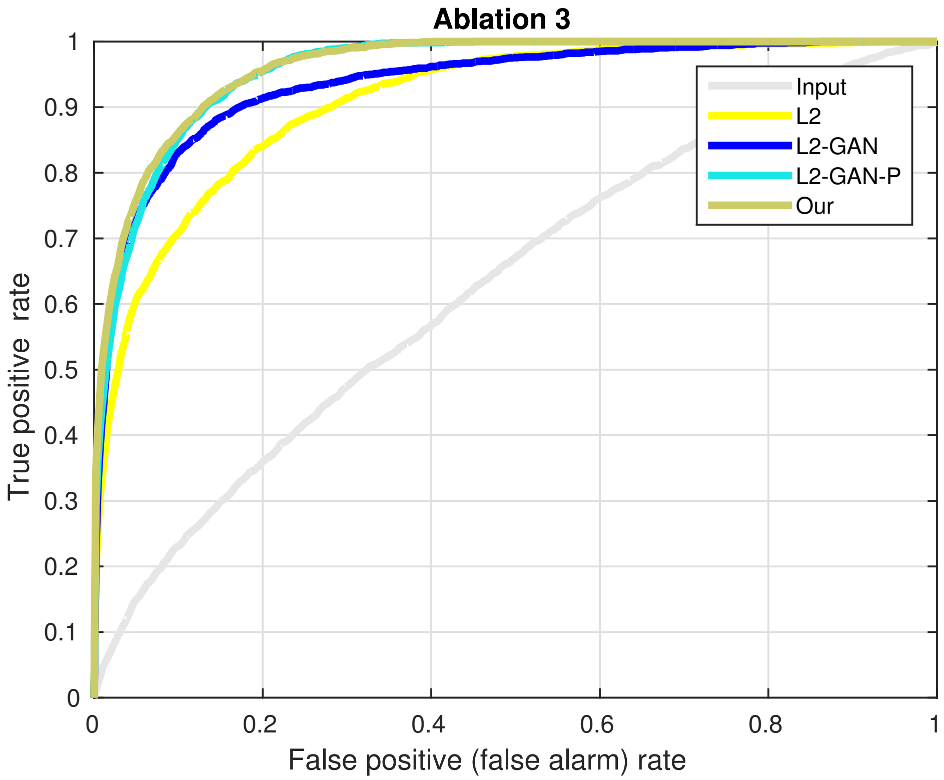


Figure 7.11: The ROC curves corresponding to **Ablation 3**.

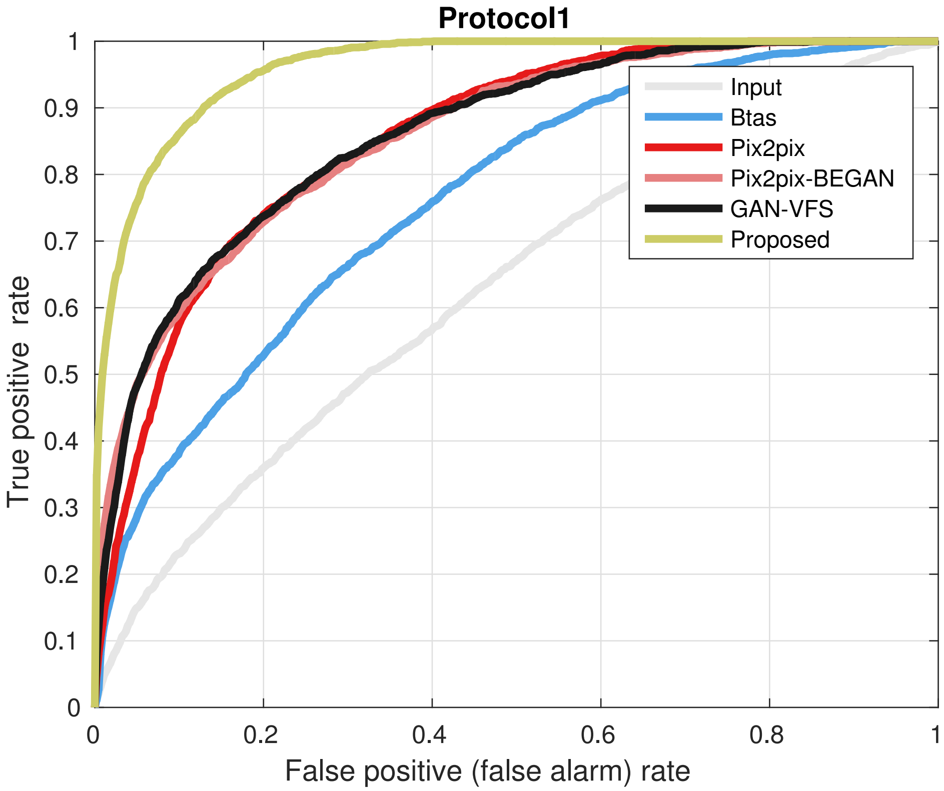


Figure 7.12: The ROC curves corresponding to *Protocol1*.

One synthesis example corresponding to **Ablation 1** is shown in Figure 7.7. It can be observed from this figure (comparing second column with third column) that the overall performance improves after leveraging the newly introduced dense-residual encoder-decoder (DR-ED) structure. This can be clearly observed from the left part of the reconstructed mouth. This essentially demonstrates the effectiveness of the proposed dense-residual encoder-decoder structure. Though the DR-ED is able to reconstruct better visible face, from the close-up of the left eye shown in the second row in Figure7.7 we observe that some structure information is missing. The multi-level pooling module at the end of the encoder-decoder structure overcomes this issue and preserves the the overall eye structure. Quantitative results evaluated based on PSNR and SSIM [145], as shown in Table 7.1, also show similar results.

In addition to comparing the performance of the synthesized images in terms of SSIM and PSNR, we also compare the contribution of each module in face verification by plotting the ROC curves. The verification results are evaluated based on the cosine

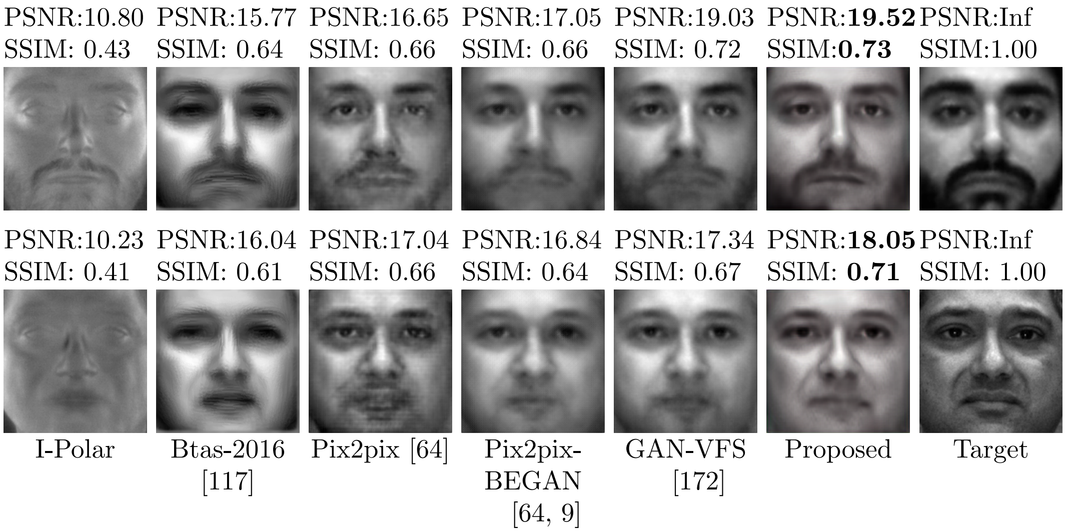


Figure 7.13: Sample results compared with state-of-the-art methods evaluated on *Protocol1*.

similarity using the deep features extracted from the pre-defined VGG-face model [99]. The results are shown in Figure 7.6. From the ROC curves, it can be clearly observed that the proposed dense-residual network with multi-level pooling can also provide some discriminative information. Similar results can also be observed from the EER and AUC comparisons, tabulated in Table 7.1.

Table 7.4: The PSNR, SSIM and EER and AUC results corresponding to *Protocol1*.

	I-Polar	Btas-2016 [117]	Pix2pix [64]	Pix2pix-BEGAN [64, 9]	GAN-VFS [172]	Proposed
PSNR (dB)	11.74	16.12	16.79	17.55	18.07	**19.55**
SSIM	0.4625	0.6785	0.6490	0.7033	0.7041	**0.7433**
EER	41.51%	26.72%	22.61%	22.56%	23.19%	**11.78**%
AUC	62.93%	81.90%	85.14%	85.30%	85.89%	**96.03**%

Ablation 2

The second ablation study is conducted to demonstrate the effectiveness of the proposed feature level multi-model fusion by conducting experiments with the following baselines:

(a) **S-S_0:** Single stream dense-resisual encoder-decoder with the proposed structure with S_0 as the input.

(b) **S-S_1:** Single stream dense-resisual encoder-decoder with the proposed structure with S_1 as the input

(c) **S-S_2:** Single stream dense-resisual encoder-decoder with the proposed structure

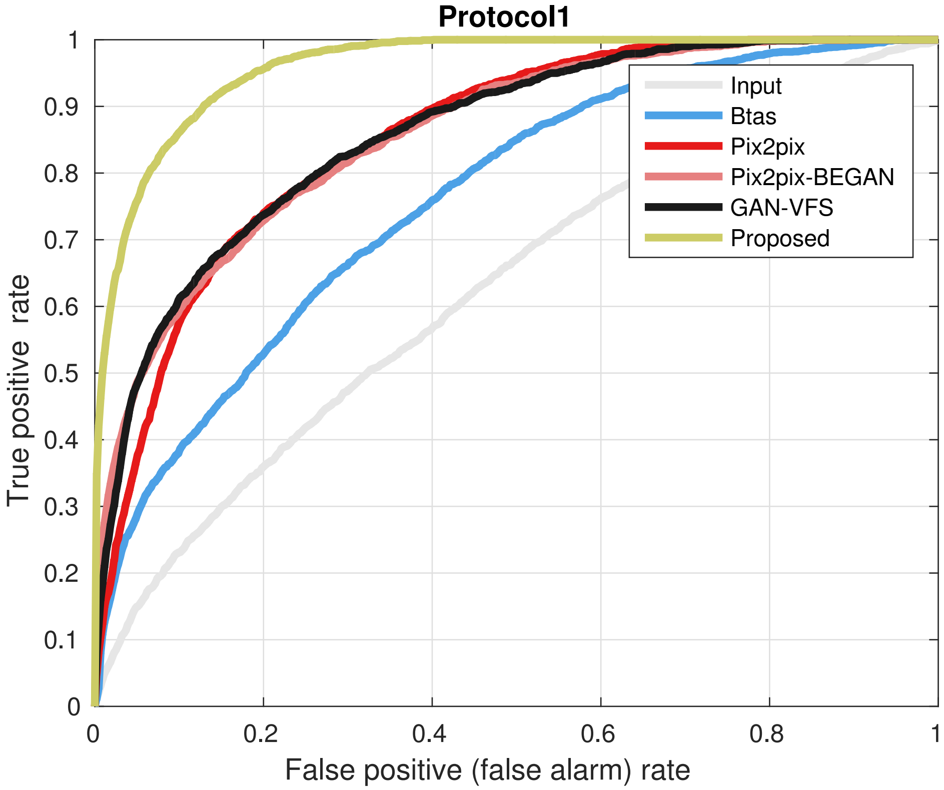


Figure 7.14: The ROC curves corresponding to *Protocol1*.

with S_2 as the input.

(d) **S-Polar-IF:** Single stream dense-resisual encoder-decoder with the proposed structure with Polar as the input (i.e. input level fusion). The S-Polar-IF model shares the exact same structure as DR-ED-ML as discussed in **Ablation 1**.

(e) **M-Polar-OF:** Multi stream dense-resisual encoder-decoder structure with output level fusion. The M-Polar-OF is basically composed of three stream dense-resisual encoder-decoder structure, where each stream shares the same structure with S-Polar-IF but with different input (S_0, S_1 and S_2) for each stream. Then, the output features from each stream are fused (concatenated) at the end of the decoding part to generate visible face images.

(f) **M-Polar-F-L2:** Multi-stream dense-resisual encoder-decoder with the proposed structure based on feature-level fusion optimized with L2 loss only.

(g) **M-Polar-F-L2-GAN:** Multi-stream dense-resisual encoder-decoder with the proposed structure based on feature-level fusion optimized with L2 and GAN loss.

(h) **M-Polar-F-L2-GAN-Perp:** Multi-stream dense-resisual encoder-decoder with the proposed structure based on feature-level fusion optimized with L2, GAN loss and perceptual loss.

(i) **Our (M-Polar-FF):** Multi-stream dense-resisual encoder-decoder with the proposed structure based on feature-level fusion with all the losses.

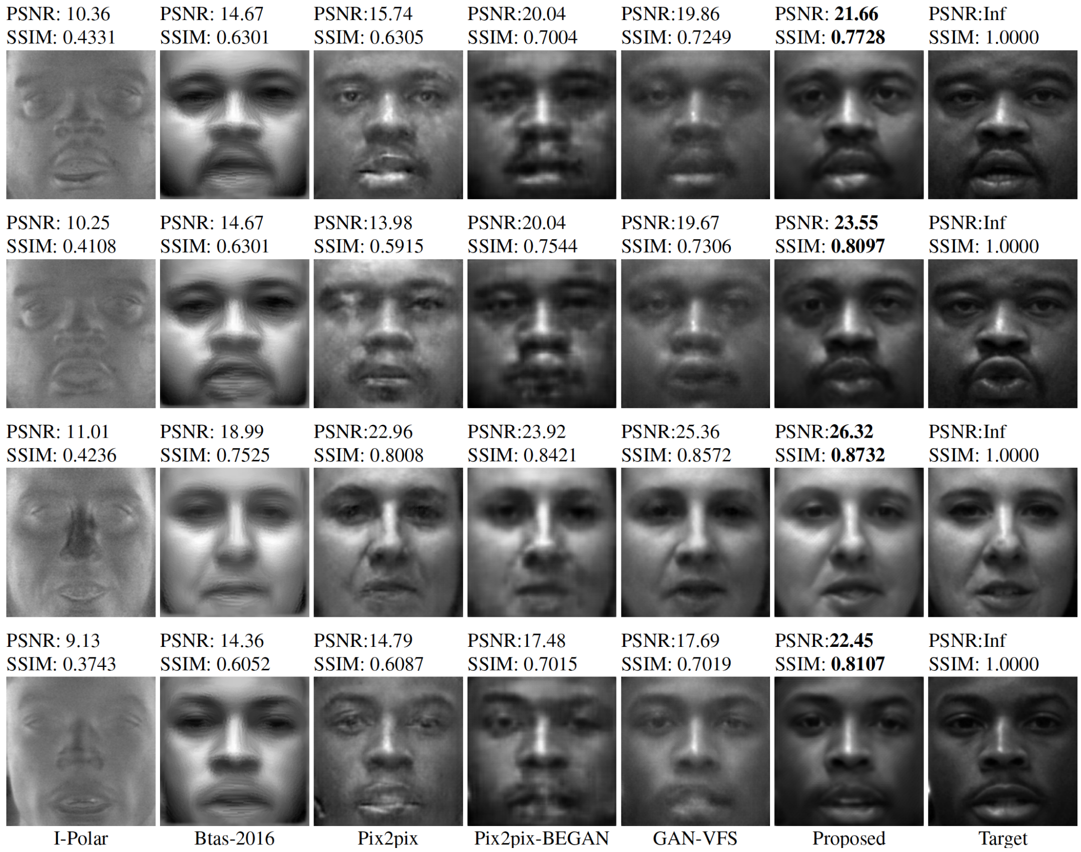


Figure 7.15: Sample results compared with state-of-the-art methods evaluated on *Protocol2*.

Sample results corresponding to **Ablation 2** is shown in Figure 7.8. It can be observed that just leveraging any one of the Stokes images as input is unable to fully capture the geometric and texture details of the whole face. For example, as shown in the first column second row in Figure 7.8, the nose is over-synthesized if just S_0 (conventional thermal) is used. Leveraging input level fusion (just concatenating three modalities as three-channel input) S-Polar-IF enables better visible face with less undesired artifacts as compared to S-S_0, S-S_1 and S-S_2. Furthermore, the proposed multi-stream feature-level fusion structure is able to preserve more geometric facial details and is

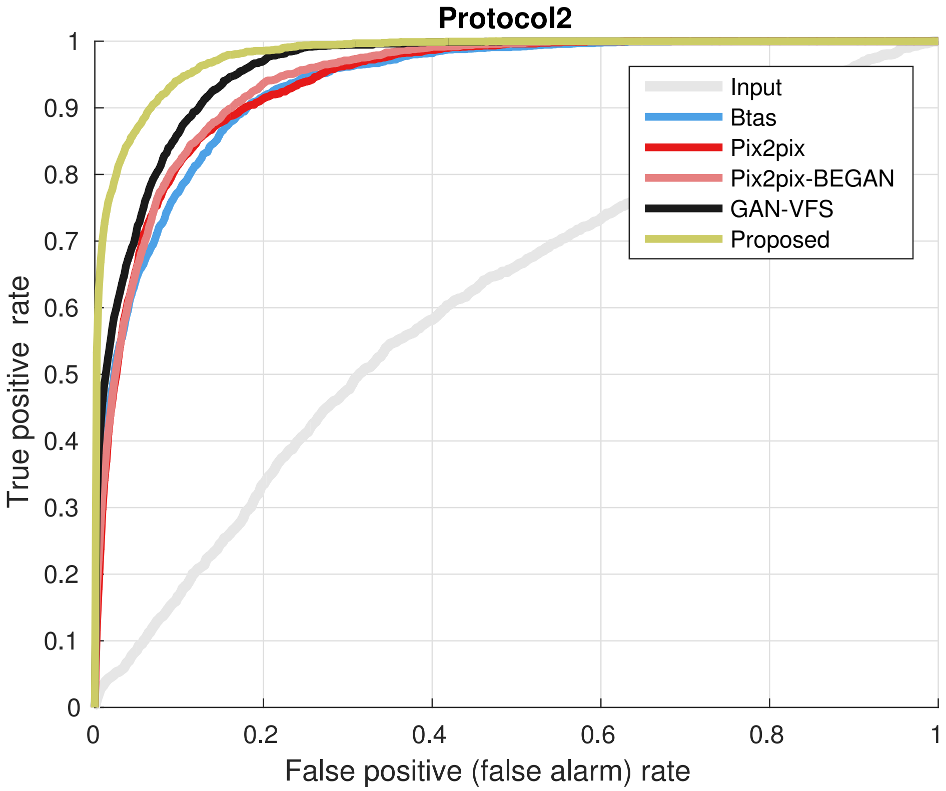


Figure 7.16: The ROC curves corresponding to the *Protocol2*.

able to generate photo-realistic visible face images. Visual results also demonstrate the effectiveness of leveraging feature level fusion over input level or output level fusion. Quantitative results evaluated in terms of PSNR and SSIM are shown in Table 7.2. Results are also consistent with our visual comparison.

Similar to Ablation study 1, the face verification results are also used as a metric to evaluate the performnace of different fusion techniques. We plot the ROC curves corresponding to the different settings discussed above. The ROC curves are shown in Figure 7.10. Again, the verification results are evaluated based the cosine similarity using the deep features extracted from the VGG-face model [99] without fine-tuning. From the ROC curves, it can be clearly observed that the proposed multi-stream feature-level fusion can bring in more discriminative information as compared to input level or output level fusion.

Ablation 3

Table 7.5: The PSNR, SSIM, EER and AUC results corresponding to *Protocol2*.

	I-Polar	Btas-2016 [117]	Pix2pix [64]	Pix2pix-BEGAN [64, 9]	GAN-VFS [172]	Proposed
PSNR (dB)	10.88	15.82	17.82	18.28	18.58	**19.18**
SSIM	0.4467	0.6854	0.6828	0.7214	0.7283	**0.7340**
EER	40.87%	14.60%	13.49%	15.81%	11.42%	**7.99**%
AUC	61.27%	93.99%	93.46%	92.50%	95.96%	**98.00**%

In the third ablation study, we demonstrate the effectiveness of different loss functions used in the proposed method (e.g. adversarial loss, perceptual loss and identity preserving loss) by conducting the following experiments. All the experimental results are evaluated using **Protocol 1** based on the polarimetric images as the input:

(a) **L2**: The proposed architecture (M-Polar-FF) optimized with the L2 loss.

(b) **L2-GAN**: The proposed architecture optimized with the L2 loss and the adversarial loss.

(c) **L2-GAN-P**: The proposed architecture optimized with the L2 loss, the adversarial loss and the perceptual loss.

(d) **Our**: The proposed architecture optimized with the L2 loss, the adversarial loss, the perceptual loss and the identity-preserving loss.

Visual results corresponding to this ablation study are shown in Figure 7.12. It can be observed from the results that the L2 loss itself generates blurry faces and many details around the eyes and the mouth regions are missing. By involving the GAN structure in the proposed method, more details are being added to the results. But it can be observed that GAN itself produces images with artifacts. Introduction of the perceptual loss in the proposed framework is able to remove some of the artifacts and makes the results visually pleasing. Finally, the combination of all the losses is able to generate more reasonable results with better facial details.

To better demonstrate the effectiveness of different losses in the proposed method, we plot the ROC curves corresponding to the above four different network settings. The results are shown in Figure 7.11. All the verification results are evaluated on the deep features extracted from the VGG-face model [99] without fine-tuning. From the ROC curves, it can be clearly observed that even though the identity loss does not produce visually different results, it can bring in more discriminative information. The corresponding PSNR, SSIM values as well as the AUC and EER values are summarized in Table 7.3.

7.5.3 Comparison with state-of-the-art Methods

To demonstrate the improvements achieved by the proposed method, it is compared against recent state-of-the-art methods [117, 64, 9, 172] on the new dataset. We compare quantitative and qualitative performance of different methods on the test images from the two distinct protocols *Protocol1* and *Protocol2* discussed earlier.

Sample results corresponding to Protocol 1 and Protocol 2 are shown in Figure 7.13 and Figure 7.15, respectively. It can be observed from these figures, Pix2pix and Pix2pix-BEGAN introduce undesirable artifacts in the final reconstructed images.

The introduction of the perceptual loss in [172] is able to remove some of these artifacts and produce visually pleasing results. However, the synthesized images still lack some geometric and texture details as compared to the target image. In contrast, the proposed method is able to generate photo-realistic visible face images while better retaining the discriminative information such as the structure of mouth and eye. Quantitative results corresponding to different methods evaluated on both protocols are tabulated in Table 7.4 and Table7.5, showing that the proposed multi-stream feature-level fusion GAN structure is able to achieve superior performance.

Similar to the ablation study, we also propose to use the performance of face verification as a metric to evaluate the performance of different methods. Figure7.14 and Figure7.16 show the ROC curves corresponding to the two experimental protocols. The AUC and EER results are reported in Table 7.4 and Table 7.5. From these results, it can be clearly observed that the proposed method is able to achieve superior quantitative performance compared the previous approaches. These results highlight the significance of using a GAN-based approach to image synthesis.

Chapter 8

Summary

In this dissertation, we developed learning-based methods for single image restoration (such as single image de-raining and single image dehazing) and translation (thermal-to-visible image synthesis). This thesis has explored the effectiveness of combining physical/empirical priors with data-driven methods in the discussed applications. We have demonstrated the effectiveness and the efficiency of the techniques theoretically and practically. Specially, we have developed the following methods:

8.1 Single Image de-raining

Firstly, we presented the CCRR algorithm for removing rain streaks from a given rainy image. Our method entails learning sparsity based and low-rank representation based filters directly from training examples. Using these learned filters, we proposed an optimization framework for de-raining. Various experiments showed the significance of our CCRR de-raining method over several recent state-of-the-art de-raining methods.

Secondly, we presented a conditional GAN-based algorithm for the removal of rain streaks form a single image. In comparison to the existing approaches which attempt to solve the de-raining problem in an image decomposition framework by using prior information, we investigated the use of generative modeling for synthesizing de-rained image from a given input rainy image. For improved stability in training and reducing artifacts introduced by GANs in the output images, we proposed the use of a new refined loss function in the GAN optimization framework. In addition, a multi-scale discriminator was proposed to leverage features from different scales to determine whether the de-rained image is real or fake. Detailed experiments and comparisons were performed on synthetic and real-world images to demonstrate that the proposed ID-CGAN method significantly outperforms many recent state-of-the-art methods. Additionally, the proposed ID-CGAN method was compared against baseline configurations to illustrate the performance gains obtained by different modules. Furthermore, experimental results evaluated on objection detection using Faster-RCNN demonstrate significant improvements in detection performance when ID-CGAN method was used as a pre-processing step.

Thirdly, we developed a novel density-aware image de-raining method with multi-stream densely connected network (DID-MDN) for jointly rain-density estimation and de-raining. In comparison to the existing approaches which attempt to solve the de-raining problem using a single network to learn to remove rain streaks with different densities (heavy, medium and light), we investigated the use of estimated rain-density label for guiding the synthesis of the de-rained image. To efficiently predict the rain-density label, a residual-aware rain-density classier was proposed in this chapter. Detailed experiments and comparisons were performed on two synthetic and one real-world datasets to demonstrate that the proposed DID-MDN method significantly outperforms many recent state-of-the-art methods. Additionally, the proposed DID-MDN method was compared against baseline configurations to illustrate the performance gains obtained by each module.

8.2 Single Image dehazing

We presented a new end-to-end deep learning-based dehazing method that can jointly optimize transmission map, atmospheric light and dehazed image. This was achieved via directly embedding the atmospheric image degradation model into the overall optimization framework. To efficiently estimate the transmission map, a novel densely connected encoder-decoder structure with multi-level pooling module was proposed and this network was optimized by a new edge-preserving loss. In addition, to refine the details and to leverage the mutual structural correlation between the dehazed image and the estimated transmission map, a joint-discriminator based GAN framework was introduced in the proposed method. Various experiments were conducted to show the significance of the proposed method.

8.3 Thermal-to-visible face synthesis

We presented a new multi-level dense-residual fusion GAN structure for synthesizing photo-realistic visible face images from the corresponding polarimetric data. In contrast to the previous methods that leverage input level fusion techniques to combine geometric and texture information from different Stokes image, we take a different approach where visual features extracted from different Stokes images were combined to synthesize the photo-realistic face images. Quantitative and qualitative experiments evaluated on a real polarimetric visible database demonstrate that the proposed method is able to achieve significantly better results as compared to the recent state-of-the-art methods. In addition, three ablation studies were performed to demonstrate the improvements obtained by the feature-level fusion methods, different modules and different loss functions in the proposed method. Furthermore, an extended polarimetric-visible database consisting of data from 111 subjects was also presented.

Chapter 9

Future Directions

There are several important topics in image restoration and translation that need further investigation. We discuss interesting and promising topics that we will pursue in the future as follows:

How to make synthetic samples realistic *(Single Image De-raining/Single Image Dehazing)*: Even the success of using synthetic samples avoiding the need of expensive annotations has demonstrated the effectiveness in single image de-raining and single image dehazing, the learning from synthetic data still does not achieve the desired performance due to the gap between synthetic and real image characteristics. Hence, it is important to explore the possibility of how to leverage the algorithms

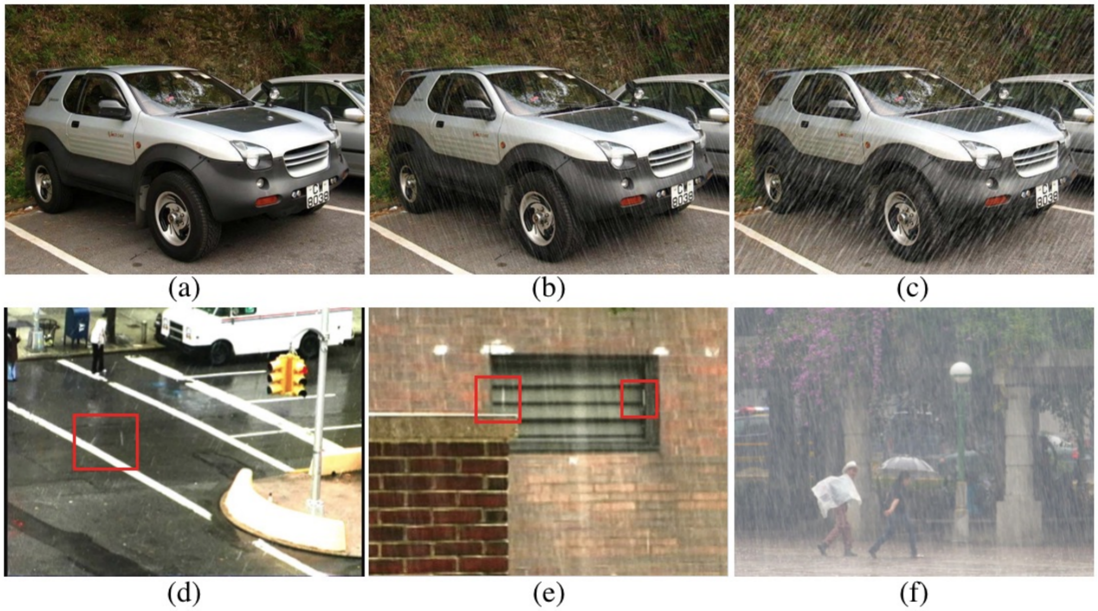


Figure 9.1: Synthetic [(a), (b), (c)] and real-world [(d), (e), (f)] examples of rainy image. It can be observed that synthetic rainy images are not realistic enough. Image credit to: *Wei, Wei, et al. "Semi-supervised CNN for Single Image Rain Removal." arXiv preprint arXiv:1807.11078 (2018).*

to automatically improve the quality of the synthetic samples. In the future, we will explore how to leverage the GAN framework to improve the quality of synthetic images.

How to Design an All-can-do Model *(Single Image De-raining/Single Image Dehazing)*: Even though many deep learning methods have achieved incredible improvements in different image restoration problems such as single image de-raining, single image dehazing, single image super-resolution and etc., yet there is still no single model that can deal with all the image restoration problems together. It will be interesting to design a model/framework which is able to address the image restoration problems all together.

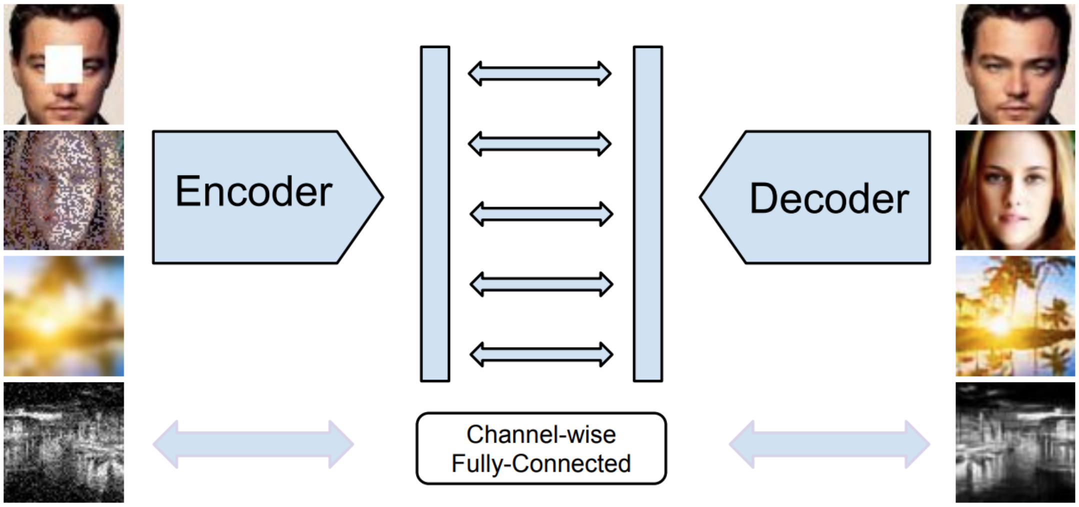


Figure 9.2: A possible overview of All-can-do model. Image credit to: *Gao, Ruohan, and Kristen Grauman. "On-demand learning for deep image restoration." Proc. IEEE Conf. Comput. Vision and Pattern Recognition. 2017.*

How to augment training samples *(Thermal-to-visible Face Synthesis)*: It has been shown that an effective deep learning framework with large-scale training pair samples is able to achieve state-of-the-art performance in image synthesis problems. However, collecting a large-scale paired samples need a lot of manual annotations. Furthermore, the collection procedure is very expensive. Hence, it is important to explore how to augment or collect a large-scale dataset with less manual efforts.

References

[1] Codruta O Ancuti, Cosmin Ancuti, Radu Timofte, and Christophe De Vleeschouwer. O-haze: a dehazing benchmark with real hazy and haze-free outdoor images. *ArXiv e-prints*, 5(6), 2018.

[2] Codruta Orniana Ancuti and Cosmin Ancuti. Single image dehazing by multi-scale fusion. *IEEE Transactions on Image Processing*, 22(8):3271–3282, 2013.

[3] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. on PAMI*, 33(5):898–916, 2011.

[4] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein gan. *arXiv preprint arXiv:1701.07875*, 2017.

[5] Yousef Atoum, Yaojie Liu, Amin Jourabloo, and Xiaoming Liu. Face anti-spoofing using patch and depth-based cnns. In *In Proceeding of International Joint Conference on Biometrics*, Denver, CO, October 2017.

[6] Elnaz Barshan and Paul Fieguth. Stage-wise training: An improved feature learning strategy for deep models. In *Feature Extraction: Modern Questions and Challenges*, pages 49–59, 2015.

[7] Dana Berman, Shai Avidan, et al. Non-local image dehazing. In *CVPR*, pages 1674–1682, 2016.

[8] Dana Berman, Tali Treibitz, and Shai Avidan. Air-light estimation using haze-lines. In *Computational Photography (ICCP), 2017 IEEE International Conference on*, pages 1–9. IEEE, 2017.

[9] David Berthelot, Tom Schumm, and Luke Metz. Began: Boundary equilibrium generative adversarial networks. *arXiv preprint arXiv:1703.10717*, 2017.

[10] J. Bobin, J. L. Starck, J. M. Fadili, Y. Moudden, and D. L. Donoho. Morphological component analysis: An adaptive thresholding strategy. *IEEE Transactions on Image Processing*, 16(11):2675–2681, Nov 2007.

[11] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine Learning*, 3(1):1–122, 2011.

[12] Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *CVPR*, pages 391–398, 2013.

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

[14] Jian-Feng Cai, Emmanuel J Candès, and Zuowei Shen. A singular value thresholding algorithm for matrix completion. *SIAM Journal on Optimization*, 20(4):1956–1982, 2010.

[15] Duan-Yu Chen, Chien-Cheng Chen, and Li-Wei Kang. Visual depth guided color image rain streaks removal using sparse coding. *IEEE transactions on circuits and systems for video technology*, 24(8):1430–1455, 2014.

[16] Jun-Cheng Chen, Vishal M Patel, and Rama Chellappa. Unconstrained face verification using deep cnn features. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–9. IEEE, 2016.

[17] Jun-Cheng Chen, Rajeev Ranjan, Swami Sankaranarayanan, Amit Kumar, Ching-Hui Chen, Vishal M. Patel, Carlos D. Castillo, and Rama Chellappa. Unconstrained still/video-based face verification with deep convolutional neural networks. *International Journal of Computer Vision*, Jul 2017.

[18] Xin Chen, Patrick J Flynn, and Kevin W Bowyer. Ir and visible light face recognition. *Computer Vision and Image Understanding*, 99(3):332–358, 2005.

[19] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *IEEE ICCV*, pages 1968–1975, 2013.

[20] Yi-Lei Chen and Chiou-Ting Hsu. A generalized low-rank appearance model for spatio-temporally correlated rain streaks. In *IEEE ICCV*, pages 1968–1975, 2013.

[21] I. Y. Chun and J. A. Fessler. Convolutional Dictionary Learning: Acceleration and Convergence. *ArXiv e-prints*, July 2017.

[22] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *BigLearn, NIPS Workshop*, 2011.

[23] Antonia Creswell and Anil A Bharath. Task specific adversarial cost function. *arXiv preprint arXiv:1609.08661*, 2016.

[24] Xing Di, Vishwanath A Sindagi, and Vishal M Patel. Gp-gan: gender preserving gan for synthesizing faces from landmarks. *arXiv preprint arXiv:1710.00962*, 2017.

[25] Hui Ding, Shaohua Kevin Zhou, and Rama Chellappa. Facenet2expnet: Regularizing a deep face recognition net for expression recognition. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 118–126. IEEE, 2017.

[26] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. *arXiv preprint arXiv:1506.02753*, 2015.

[27] Alexey Dosovitskiy and Thomas Brox. Generating images with perceptual similarity metrics based on deep networks. In *Advances in Neural Information Processing Systems*, pages 658–666, 2016.

[28] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, pages 2650–2658, 2015.

[29] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, 2014.

[30] Andreas Eitel, Jost Tobias Springenberg, Luciano Spinello, Martin Riedmiller, and Wolfram Burgard. Multimodal deep learning for robust rgb-d object recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 681–687. IEEE, 2015.

[31] Virginia Espinosa-Duró, Marcos Faundez-Zanuy, and Jiří Mekyska. A new face database simultaneously acquired in visible, near-infrared and thermal spectrums. *Cognitive Computation*, 5(1):119–135, 2013.

[32] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

[33] Zhiwen Fan, Huafeng Wu, Xueyang Fu, Yue Hunag, and Xinghao Ding. Residual-guide feature fusion network for single image deraining. *arXiv preprint arXiv:1804.07493*, 2018.

[34] Raanan Fattal. Single image dehazing. *ACM transactions on graphics (TOG)*, 27(3):72, 2008.

[35] Raanan Fattal. Dehazing using color-lines. volume 34, New York, NY, USA, 2014. ACM.

[36] X. Fu, J. Huang, X. Ding, Y. Liao, and J. Paisley. Clearing the Skies: A deep network architecture for single-image rain removal. *ArXiv e-prints*, September 2016.

[37] X. Fu, J. Huang, D. Zeng, Y. Huang, X. Ding, and J. Paisley. Removing rain from single images via a deep detail network. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1715–1723, July 2017.

[38] Xueyang Fu, Jiabin Huang, Xinghao Ding, Yinghao Liao, and John Paisley. Clearing the skies: A deep network architecture for single-image rain removal. *IEEE Transactions on Image Processing*, 26(6):2944–2956, 2017.

[39] Xueyang Fu, Borong Liang, Yue Huang, Xinghao Ding, and John Paisley. Lightweight pyramid networks for image deraining. *arXiv preprint arXiv:1805.06173*, 2018.

[40] Fei Gao, Shengjie Shi, Jun Yu, and Qingming Huang. Composition-aided sketch-realistic portrait generation. *arXiv preprint arXiv:1712.00899*, 2017.

[41] Kshitiz Garg and Shree K Nayar. Detection and removal of rain from videos. In *CVPR*, volume 1, pages I–528. IEEE.

[42] Kshitiz Garg and Shree K Nayar. Vision and rain. *IJCV*, 75(1):3–27, 2007.

[43] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[44] E. Gonzalez-Sosa, R. Vera-Rodriguez, J. Fierrez, and V. M. Patel. Exploring body shape from mmw images for person recognition. *IEEE Transactions on Information Forensics and Security*, 12(9):2078–2089, Sept 2017.

[45] E. Gonzalez-Sosa, R. Vera-Rodriguez, J. Fierrez, and V. M. Patel. Millimetre wave person recognition: hand-crafted vs. learned features. In *ISBA*, pages 1–7, 2017.

[46] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[47] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *NIPS*, pages 2672–2680, 2014.

[48] Shuhang Gu, Lei Zhang, Wangmeng Zuo, and Xiangchu Feng. Weighted nuclear norm minimization with application to image denoising. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2862–2869, 2014.

[49] Kristan P Gurton, Alex J Yuffa, and Gorden W Videen. Enhanced facial recognition for thermal imagery using polarimetric imaging. *Optics letters*, 39(13):3857–3859, 2014.

[50] Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. *IEEE Trans. on PAMI*, 33(12):2341–2353, 2011.

[51] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

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

[53] Ran He, Xiang Wu, Zhenan Sun, and Tieniu Tan. Wasserstein cnn: Learning invariant features for nir-vis face recognition. *arXiv preprint arXiv:1708.02412*, 2017.

[54] Felix Heide, Wolfgang Heidrich, and Gordon Wetzstein. Fast and flexible convolutional sparse coding. In *CVPR*, pages 5135–5143. IEEE, 2015.

[55] Shuowen Hu, Jonghyun Choi, Alex L Chan, and William Robson Schwartz. Thermal-to-visible face recognition using partial least squares. *JOSA A*, 32(3):431–442, 2015.

[56] Shuowen Hu, Nathaniel J Short, Benjamin S Riggan, Christopher Gordon, Kristan P Gurton, Matthew Thielke, Prudhvi Gurram, and Alex L Chan. A polarimetric thermal database for face recognition research. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 119–126, 2016.

[57] De-An Huang, Li-Wei Kang, Yu-Chiang Frank Wang, and Chia-Wen Lin. Self-learning based image decomposition with applications to single image denoising. *IEEE Transactions on multimedia*, 16(1):83–93, 2014.

[58] De-An Huang, Li-Wei Kang, Min-Chun Yang, Chia-Wen Lin, and Yu-Chiang Frank Wang. Context-aware single image rain removal. In *Multimedia and Expo (ICME), 2012 IEEE International Conference on*, pages 164–169. IEEE, 2012.

[59] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[60] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[61] Shih-Chia Huang, Bo-Hao Chen, and Wei-Jheng Wang. Visibility restoration of single hazy images captured in real-world weather conditions. *IEEE Transactions on Circuits and Systems for Video Technology*, 24(10):1814–1824, 2014.

[62] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of The 32nd International Conference on Machine Learning*, pages 448–456, 2015.

[63] S. M. Iranmanesh, A. Dabouei, H. Kazemi, and N. M. Nasrabadi. Deep Cross Polarimetric Thermal-to-visible Face Recognition. *ArXiv e-prints*, January 2018.

[64] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. *CVPR*, 2017.

[65] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*, pages 1175–1183. IEEE, 2017.

[66] Nikolay Jetchev, Urs Bergmann, and Roland Vollgraf. Texture synthesis with spatial generative adversarial networks. *arXiv preprint arXiv:1611.08207*, 2016.

[67] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[68] Li-Wei Kang, Chia-Wen Lin, and Yu-Hsiang Fu. Automatic single-image-based rain streaks removal via image decomposition. *IEEE TIP*, 21(4):1742–1755, 2012.

[69] Levent Karacan, Zeynep Akata, Aykut Erdem, and Erkut Erdem. Learning to generate images of outdoor scenes from attributes and semantic layouts. *arXiv preprint arXiv:1612.00215*, 2016.

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

[71] B. Klare and A. K. Jain. Heterogeneous face recognition: Matching nir to visible light images. In *ICPR*, pages 1513–1516, Aug 2010.

[72] Brendan F Klare and Anil K Jain. Heterogeneous face recognition using kernel prototype similarities. *IEEE transactions on pattern analysis and machine intelligence*, 35(6):1410–1422, 2013.

[73] Johannes Kopf, Boris Neubert, Billy Chen, Michael Cohen, Daniel Cohen-Or, Oliver Deussen, Matt Uyttendaele, and Dani Lischinski. Deep photo: Model-based photograph enhancement and viewing. In *ACM TOG*, volume 27, page 116. ACM, 2008.

[74] Louis Kratz and Ko Nishino. Factorizing scene albedo and depth from a single foggy image. In *ICCV*, pages 1701–1708. IEEE, 2009.

[75] Christian Ledig, Lucas Theis, Ferenc Huszár, Jose Caballero, Andrew Cunningham, Alejandro Acosta, Andrew Aitken, Alykhan Tejani, Johannes Totz, Zehan Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. *arXiv preprint arXiv:1609.04802*, 2016.

[76] José Lezama, Qiang Qiu, and Guillermo Sapiro. Not afraid of the dark: Nir-vis face recognition via cross-spectral hallucination and low-rank embedding. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6807–6816. IEEE, 2017.

[77] B. Li, W. Ren, D. Fu, D. Tao, D. Feng, W. Zeng, and Z. Wang. RESIDE: A Benchmark for Single Image Dehazing. *ArXiv e-prints*, December 2017.

[78] Boyi Li, Xiulian Peng, Zhangyang Wang, Jizheng Xu, and Dan Feng. An all-in-one network for dehazing and beyond. *arXiv preprint arXiv:1707.06543*, 2017.

[79] Chuan Li and Michael Wand. Precomputed real-time texture synthesis with markovian generative adversarial networks. In *ECCV*, pages 702–716, 2016.

[80] Jun Li, Reinhard Klein, and Angela Yao. A two-streamed network for estimating fine-scaled depth maps from single rgb images. In *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[81] Kunpeng Li, Yu Kong, and Yun Fu. Multi-stream deep similarity learning networks for visual tracking. In *IJCAI*, 2017.

[82] Stan Li, Dong Yi, Zhen Lei, and Shengcai Liao. The casia nir-vis 2.0 face database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 348–353, 2013.

[83] Y. Li, R. T. Tan, X. Guo, J. Lu, and M. S. Brown. Rain streak removal using layer priors. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2736–2744, June 2016.

[84] Zhuwen Li, Ping Tan, Robby T Tan, Danping Zou, Steven Zhiying Zhou, and Loong-Fah Cheong. Simultaneous video defogging and stereo reconstruction. In *CVPR*, pages 4988–4997, 2015.

[85] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3431–3440, 2015.

[86] Yu Luo, Yong Xu, and Hui Ji. Removing rain from a single image via discriminative sparse coding. In *ICCV*, pages 3397–3405, 2015.

[87] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models.

[88] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[89] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *CVPR*, pages 5188–5196. IEEE, 2015.

[90] Xiao-Jiao Mao, Chunhua Shen, and Yu-Bin Yang. Image denoising using very deep fully convolutional encoder-decoder networks with symmetric skip connections. *arXiv preprint arXiv:1603.09056*, 2016.

[91] Gaofeng Meng, Ying Wang, Jiangyong Duan, Shiming Xiang, and Chunhong Pan. Efficient image dehazing with boundary constraint and contextual regularization. In *ICCV*, pages 617–624, 2013.

[92] Ethan Meyers and Lior Wolf. Using biologically inspired features for face processing. *International Journal of Computer Vision*, 76(1):93–104, 2008.

[93] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014.

[94] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*, pages 807–814, 2010.

[95] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012.

[96] F. Nicolo and N. A. Schmid. Long range cross-spectral face recognition: Matching swir against visible light images. *IEEE Transactions on Information Forensics and Security*, 7(6):1717–1726, Dec 2012.

[97] Shintaro Ono, Takamichi Miyata, and Isao Yamada. Cartoon-texture image decomposition using blockwise low-rank texture characterization. *IEEE TIP*, 23(3):1128–1142, 2014.

[98] Neal Parikh and Stephen Boyd. Proximal algorithms. *Foundations and Trends in Optimization*, 1(3):127–239, jan 2014.

[99] O. M. Parkhi, A. Vedaldi, and A. Zisserman. Deep face recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2015.

[100] Deepak Pathak, Philipp Krahenbuhl, Jeff Donahue, Trevor Darrell, and Alexei A Efros. Context encoders: Feature learning by inpainting. *arXiv preprint arXiv:1604.07379*, 2016.

[101] Chunlei Peng, Xinbo Gao, Nannan Wang, Dacheng Tao, Xuelong Li, and Jie Li. Multiple representations-based face sketch–photo synthesis. *IEEE transactions on neural networks and learning systems*, 27(11):2201–2215, 2016.

[102] Xi Peng, Rogerio S Feris, Xiaoyu Wang, and Dimitris N Metaxas. A recurrent encoder-decoder network for sequential face alignment. In *European Conference on Computer Vision*, pages 38–56. Springer International Publishing, 2016.

[103] Xi Peng, Zhiqiang Tang, Yang Fei, Rogerio S Feris, and Dimitris Metaxas. Jointly optimize data and network training: Adversarial data augmentation in human pose estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[104] Xi Peng, Xiang Yu, Kihyuk Sohn, Dimitris N Metaxas, and Manmohan Chandraker. Reconstruction-based disentanglement for pose-invariant face recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[105] Pramuditha Perera, Mahdi Abavisani, and Vishal M Patel. In2i: Unsupervised multi-image-to-image translation using generative adversarial networks. *arXiv preprint arXiv:1711.09334*, 2017.

[106] Gabriel Peyré, Jalal Fadili, and Jean-Luc Starck. Learning the morphological diversity. *SIAM Journal on Imaging Sciences*, 3(3):646–669, 2010.

[107] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[108] R. Ranjan, S. Sankaranarayanan, A. Bansal, N. Bodla, J. C. Chen, V. M. Patel, C. D. Castillo, and R. Chellappa. Deep learning for understanding faces: Machines may be just as good, or better, than humans. *IEEE Signal Processing Magazine*, 35(1):66–83, Jan 2018.

[109] Rajeev Ranjan, Swami Sankaranarayanan, Carlos D Castillo, and Rama Chellappa. An all-in-one convolutional neural network for face analysis. In *Automatic Face & Gesture Recognition (FG 2017), 2017 12th IEEE International Conference on*, pages 17–24. IEEE, 2017.

[110] Scott Reed, Zeynep Akata, Xinchen Yan, Lajanugen Logeswaran, Bernt Schiele, and Honglak Lee. Generative adversarial text to image synthesis. *arXiv preprint arXiv:1605.05396*, 2016.

[111] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[112] Weihong Ren, Jiandong Tian, Zhi Han, Antoni Chan, and Yandong Tang. Video desnowing and deraining based on matrix decomposition. In *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, pages 4210–4219, 2017.

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

[114] Wenqi Ren, Lin Ma, Jiawei Zhang, Jinshan Pan, Xiaochun Cao, Wei Liu, and Ming-Hsuan Yang. Gated fusion network for single image dehazing. *arXiv preprint arXiv:1804.00213*, 2018.

[115] Wenqi Ren, Jinshan Pan, Xiaochun Cao, and Ming-Hsuan Yang. Video deblurring via semantic segmentation and pixel-wise non-linear kernel. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1077–1085, 2017.

[116] B. S. Riggan, C. Reale, and N. M. Nasrabadi. Coupled auto-associative neural networks for heterogeneous face recognition. *IEEE Access*, 3:1620–1632, 2015.

[117] Benjamin S Riggan, Nathanial J Short, Shuowen Hu, and Heesung Kwon. Estimation of visible spectrum faces from polarimetric thermal faces. In *Biometrics Theory, Applications and Systems (BTAS), 2016 IEEE 8th International Conference on*, pages 1–7. IEEE, 2016.

[118] Benjamin S Riggan, Nathaniel J Short, and Shuowen Hu. Optimal feature learning and discriminative framework for polarimetric thermal to visible face recognition. In *Applications of Computer Vision (WACV), 2016 IEEE Winter Conference on*, pages 1–7. IEEE, 2016.

[119] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[120] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *NIPS*, pages 2226–2234, 2016.

[121] M Saquib Sarfraz and Rainer Stiefelhagen. Deep perceptual mapping for thermal to visible face recognition. *arXiv preprint arXiv:1507.02879*, 2015.

[122] M Saquib Sarfraz and Rainer Stiefelhagen. Deep perceptual mapping for cross-modal face recognition. *International Journal of Computer Vision*, 122(3):426–438, 2017.

[123] Gerald Schaefer and Michal Stich. Ucid: an uncompressed color image database. In *Electronic Imaging 2004*, pages 472–480. International Society for Optics and Photonics, 2003.

[124] Daniel Scharstein, Heiko Hirschmüller, York Kitajima, Greg Krathwohl, Nera Nešić, Xi Wang, and Porter Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *German Conference on Pattern Recognition*, pages 31–42. Springer, 2014.

[125] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 815–823, 2015.

[126] Hamid R Sheikh and Alan C Bovik. Image information and visual quality. *IEEE TIP*, 15(2):430–444, 2006.

[127] Nathaniel Short, Shuowen Hu, Prudhvi Gurram, Kristan Gurton, and Alex Chan. Improving cross-modal face recognition using polarimetric imaging. *Opt. Lett.*, 40(6):882–885, Mar 2015.

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

[129] Vishwanath A Sindagi and Vishal M Patel. Generating high-quality crowd density maps using contextual pyramid cnns. In *ICCV*, 2017.

[130] Shuran Song, Samuel P Lichtenberg, and Jianxiong Xiao. Sun rgb-d: A rgb-d scene understanding benchmark suite. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 567–576, 2015.

[131] Jean-Luc Starck, Michael Elad, and David L Donoho. Image decomposition via the combination of sparse representations and a variational approach. *IEEE TIP*, 14(10):1570–1582, 2005.

[132] Gabriele Steidl, Joachim Weickert, Thomas Brox, Pavel Mrázek, and Martin Welk. On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides. *SIAM J. Numerical Analysis*, 42(2):686–713, 2004.

[133] Matan Sulami, Itamar Glatzer, Raanan Fattal, and Mike Werman. Automatic recovery of the atmospheric light in hazy images. In *Computational Photography (ICCP), 2014 IEEE International Conference on*, pages 1–11. IEEE, 2014.

[134] Yi Sun, Yuheng Chen, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. In *Advances in neural information processing systems*, pages 1988–1996, 2014.

[135] Robby T Tan. Visibility in bad weather from a single image. In *CVPR*, pages 1–8. IEEE, 2008.

[136] Ketan Tang, Jianchao Yang, and Jue Wang. Investigating haze-relevant features in a learning framework for image dehazing. In *CVPR*, pages 2995–3000, 2014.

[137] Eric Thibodeau-Laufer, Guillaume Alain, and Jason Yosinski. Deep generative stochastic networks trainable by backprop. 2014.

[138] Luan Tran, Xi Yin, and Xiaoming Liu. Disentangled representation learning gan for pose-invariant face recognition.

[139] J Scott Tyo, Dennis L Goldstein, David B Chenault, and Joseph A Shaw. Review of passive imaging polarimetry for remote sensing applications. *Applied optics*, 45(22):5453–5469, 2006.

[140] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. *arXiv preprint arXiv:1612.02401*, 2016.

[141] L. Wang, V. A. Sindagi, and V. M. Patel. High-quality facial photo-sketch synthesis using multi-adversarial networks. In *IEEE International Conference on Automatic Face and Gesture Recognition*, 2018.

[142] Puyang Wang, He Zhang, and Vishal M Patel. Sar image despeckling using a convolutional neural network. *IEEE Signal Processing Letters*, 24(12):1763–1767, 2017.

[143] Y.-T. Wang, X.-L. Zhao, T.-X. Jiang, L.-J. Deng, Y. Chang, and T.-Z. Huang. Rain Streak Removal for Single Image via Kernel Guided CNN. *ArXiv e-prints*, August 2018.

[144] Zhou Wang and Alan C Bovik. A universal image quality index. *IEEE Signal Processing Letters*, 9(3):81–84, 2002.

[145] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004.

[146] Wei Wei, Deyu Meng, Qian Zhao, and Zongben Xu. Semi-supervised cnn for single image rain removal. *arXiv preprint arXiv:1807.11078*, 2018.

[147] Wei Wei, Lixuan Yi, Qi Xie, Qian Zhao, Deyu Meng, and Zongben Xu. Should we encode rain streaks in video as deterministic or stochastic? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2516–2525, 2017.

[148] Brendt Wohlberg. Efficient convolutional sparse coding. In *IEEE ICASSP*, pages 7173–7177. IEEE, 2014.

[149] Brendt Wohlberg. Convolutional sparse representations as an image model for impulse noise restoration. In *Image, Video, and Multidimensional Signal Processing Workshop (IVMSP), 2016 IEEE 12th*, pages 1–5. IEEE, 2016.

[150] Brendt Wohlberg. Efficient algorithms for convolutional sparse representations. *IEEE Transactions on Image Processing*, 25(1):301–315, Jan 2016.

[151] Xiang Wu, Ran He, Zhenan Sun, and Tieniu Tan. A light cnn for deep face representation with noisy labels. *IEEE Transactions on Information Forensics and Security*, 13(11):2884–2896, 2018.

[152] Junyuan Xie, Linli Xu, and Enhong Chen. Image denoising and inpainting with deep neural networks. In *NIPS*, pages 341–349, 2012.

[153] Saining Xie and Zhuowen Tu. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, pages 1395–1403, 2015.

[154] Tao Xu, Pengchuan Zhang, Qiuyuan Huang, Han Zhang, Zhe Gan, Xiaolei Huang, and Xiaodong He. Attngan: Fine-grained text to image generation with attentional generative adversarial networks. *arXiv preprint arXiv:1711.10485*, 2017.

[155] Zheng Xu, Xitong Yang, Xue Li, and Xiaoshuai Sun. The effectiveness of instance normalization: a strong baseline for single image dehazing. *arXiv preprint arXiv:1805.03305*, 2018.

[156] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.

[157] Jiaolong Yang, Peiran Ren, Dongqing Zhang, Dong Chen, Fang Wen, Hongdong Li, and Gang Hua. Neural aggregation network for video face recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4362–4371, 2017.

[158] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1357–1366, 2017.

[159] Xitong Yang, Zheng Xu, and Jiebo Luo. Towards perceptual image dehazing by physics-based disentanglement and adversarial training. 2018.

[160] Dong Yi, Zhen Lei, and Stan Z Li. Shared representation learning for heterogenous face recognition. In *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, volume 1, pages 1–7. IEEE, 2015.

[161] Jiahui Yu, Zhe Lin, Jimei Yang, Xiaohui Shen, Xin Lu, and Thomas S Huang. Generative image inpainting with contextual attention. *arXiv preprint arXiv:1801.07892*, 2018.

[162] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[163] Matthew D Zeiler, Dilip Krishnan, Graham W Taylor, and Rob Fergus. Deconvolutional networks. In *CVPR*, pages 2528–2535. IEEE, 2010.

[164] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, 2017.

[165] Hang Zhang and Kristin Dana. Multi-style generative network for real-time transfer. *arXiv preprint arXiv:1703.06953*, 2017.

[166] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[167] Hang Zhang, Kristin Dana, Jianping Shi, Zhongyue Zhang, Xiaogang Wang, Ambrish Tyagi, and Amit Agrawal. Context encoding for semantic segmentation. In

The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.

[168] He Zhang and Vishal M Patel. Convolutional sparse and low-rank coding-based rain streak removal. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on*, pages 1259–1267. IEEE, 2017.

[169] He Zhang and Vishal M Patel. Convolutional sparse and low-rank coding-based rain streak removal. In *2017 IEEE WACV*, pages 1–9. IEEE, 2017.

[170] He Zhang and Vishal M Patel. Densely connected pyramid dehazing network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[171] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[172] He Zhang, Vishal M Patel, Benjamin S Riggan, and Shuowen Hu. Generative adversarial network-based synthesis of visible faces from polarimetric thermal faces. *International Joint Conference on Biometrics 2017*, 2017.

[173] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Image de-raining using a conditional generative adversarial network. *arXiv preprint arXiv:1701.05957*, 2017.

[174] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Joint transmission map estimation and dehazing using deep networks. *arXiv preprint arXiv:1708.00581*, 2017.

[175] He Zhang, Vishwanath Sindagi, and Vishal M Patel. Multi-scale single image dehazing using perceptual pyramid deep network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 902–911, 2018.

[176] Xiaopeng Zhang, Hao Li, Yingyi Qi, Wee Kheng Leow, and Teck Khim Ng. Rain removal in video by combining temporal and chromatic properties. In *IEEE International Conference on Multimedia and Expo*, pages 461–464. IEEE, 2006.

[177] Y Zhang, L Ding, and G Sharma. Hazerd: an outdoor scene dataset and benchmark for single image dehazing. In *Proc. IEEE Intl. Conf. Image Proc.*, pages 3205–3209, 2017.

[178] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual Dense Network for Image Super-Resolution. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[179] Z. Zhang, Y. Xie, and L. Yang. Photographic Text-to-Image Synthesis with a Hierarchically-nested Adversarial Network. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[180] Zizhao Zhang, Lin Yang, and Yefeng Zheng. Translating and segmenting multimodal medical volumes with cycle-and shapeconsistency generative adversarial network.

[181] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1–8, 2017.

[182] Junbo Zhao, Michael Mathieu, and Yann LeCun. Energy-based generative adversarial network. *arXiv preprint arXiv:1609.03126*, 2016.

[183] Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip HS Torr. Conditional random fields as recurrent neural networks. In *ICCV*, pages 1529–1537, 2015.

[184] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2921–2929, 2016.

[185] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. *IEEE International Conference on Computer Vision, 2017*, 2017.

[186] Lei Zhu, Chi-Wing Fu, Dani Lischinski, and Pheng-Ann Heng. Joint bi-layer optimization for single-image rain streak removal. In *Proceedings of the IEEE international conference on computer vision*, pages 2526–2534, 2017.

[187] Qingsong Zhu, Jiaming Mai, and Ling Shao. A fast single image haze removal algorithm using color attenuation prior. *IEEE Transactions on Image Processing*, 24(11):3522–3533, 2015.

[188] Yi Zhu, Zhenzhong Lan, Shawn Newsam, and Alexander G Hauptmann. Hidden two-stream convolutional networks for action recognition. *arXiv preprint arXiv:1704.00389*, 2017.

[189] Yi Zhu and Shawn Newsam. Densenet for dense flow. *ICIP*, 2017.

[190] Yi Zhu and Shawn Newsam. Densenet for dense flow. In *ICIP*, 2017.

[191] Yizhe Zhu, Mohamed Elhoseiny, Bingchen Liu, Xi Peng, and Ahmed Elgammal. A generative adversarial approach for zero-shot learning from noisy texts. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, 2018.