# DEEP LEARNING FOR SINGLE IMAGE DEBLURRING

A THESIS SUBMITTED TO THE UNIVERSITY OF MANCHESTER
FOR THE DEGREE OF MASTER OF PHILOSOPHY

IN THE

FACULTY OF SCIENCE AND ENGINEERING

2021

by

BOYAN XU

Department of Electrical and Electronic Engineering

School of Engineering

Supervisor:

Prof. Hujun Yin

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| AWNI | Additive White Noising Images |
| BN | Batch Normalization |
| GAN | Generative Adversarial Network |
| GCN | Graph Convolutional Network |
| CNN | Convolutional Neural Network |
| DCP | Dark Channel Prior |
| DL | Deep Learning |
| FFT | Fast Fourier Transform |
| FLOP | Floating Point Operation |
| FPN | Feature Pyramid Network |
| GPU | Graphic Processing Units |
| HR | High Resolution |
| IQA | Image Quality Assessment |
| JS | Jensen-Shannon |
| LSE | Least-Square Loss |
| LR | Low Resolution |
| LSSC | Learned Simultaneous Sparse Coding |
| MAP | Maximum-a-Posterior |
| MCU | Microcontroller Unit |
| MRF | Markov Random Field |
| MSE | Mean Squared Error |
| RCAN | Residual Channel Attention Network |
| PSF | Point Spread Function |
| PSNR | Peak Signal to Noise Ratio |

| | |
|---|---|
| ReLU | Rectified Linear Unit |
| RIR | Residual In Residual |
| RNN | Recurrent Neural Network |
| SPM | Spatial Pyramid Matching |
| SSIM | Structural SIMilarity |
| TNRD | Trainable Nonlinear Reaction Diffusion |
| TV | Total Variation |

# List of Publications

- **Xu, B.** and Yin, H., A Slimmer and Deeper Approach to Network Structures for Image Denoising and Dehazing in *International Conference on Intelligent Data Engineering and Automated Learning (IDEAL).* Springer, Cham, 2020: 268-279.

- **Xu, B.** and Yin, H., Graph Convolutional Networks in Feature Space for Image Deblurring and Super-resolution in *International Joint Conference on Neural Networks (IJCNN)* 2021. IEEE, 2021. (Oral)

- **Xu, B.** and Yin, H., DC-Deblur: A Dilated Convolutional Network for Single Image Deblurring in *International Conference on Intelligent Data Engineering and Automated Learning (IDEAL).* Springer (Accepted) 2021.

- Su, J., **Xu, B.** and Yin, H., A Survey of Deep Learning Approaches to Image Restoration, *Neurocomputing* (Accepted) 2021.

- **Xu, B.** and Yin, H., A Slimmer and Deeper Approach to Deep Network Structures for Low-Level Vision Tasks, *Expert Systems*. (Under review) 2021.

# Abstract

Blind single image deblurring is a highly ill-posed problem. It often becomes even more challenging while blur is non-uniform. Since the rise of deep learning, many recent approaches are based on Convolutional Neural Networks (CNNs). These CNN-based approaches are diverse, in terms of their structures and components. However, existing methods have many disadvantages, for instance, they require intensive computation resources, and cannot restore sharp details when image blur is severe or non-uniform. In this thesis, a review the state-of-the-art methods in image restoration is firstly given, including image denoising, image dehazing, image super-resolution and image deblurring, especially of those learning based. Then various key elements and mechanisms in deblurring networks are analysed, including backbones, frameworks and conjecture that a good balance among receptive field, depth and efficiency.

To achieve better performance, three networks are proposed in this research. By combining the strength of DenseNet and Inception-v4 to realize a balanced structure, a network is proposed and named as MixNet. A new network that uses dilated convolution and named DC-Deblur is also introduced as well as a Graph Convolutional Network (GCN) based method, termed as GCResNet. Further experiments in other image restoration tasks are given, in order to show the generalisation of the proposed methods. Quantitative evaluations in term of comprehensive image quality measures have been performed. Results show that the proposed MixNet, DS-Deblur, and GCResNet are able to elevate the state-of-the-art performance on end-to-end results for dynamic scene deblurring on all the benchmark datasets.

# Declaration

No portion of the work referred to this Thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.

# Copyright

# Acknowledgements

I would like to thank everyone at the University of Manchester who supported me throughout the research career of this MPhil, especially my supervisor, professor Yin.

# Chapter 1

# 1 Introduction

## 1.1 Background and Motivation

Image blur, resulted from various possible causes such as motion and out-of-focus, can severely degrade photo quality and its purposes. Motion blur is a common and arising type often due to camera shakes, camera on moving/flying devices, object motion, and long exposure time. With the wide use of mobile devices, efforts against image blur are of great value to many vision applications, from face recognition, object detection, to medical image processing. Its highly ill-posed nature poses a great challenge as blur mechanisms or kernels are usually unknown and, in many cases, spatially variant.

Image deblurring can be divided into several categories based on the input used or available. The most common and the most challenge case is the single image deblurring, where a single image is used as input. Besides, if a series of images or consecutive frames are available or used as the input, it is either multi-image deblurring [1] or video-deblurring [2,3]. Stereo deblurring [4] is a separate category with stereo image pairs as

the input. These methods may offer valuable ideas for single image deblurring due to the use of similar backbones. Deblurring can be specifically targeted on blur type, such as out-of-focus deblur [5,6] or motion deblur [7]. Single image motion deblurring is the most important problem and has attracted a lot of attention.

In order to restore information from a blurred image, one need to find a model to deal with the blurring process, and to recover the sharp image. Earlier research [8-15] focused on removing blurs by assuming one or more sparsity priors to help restrict the solution space and transferring it to an optimization problem. However, there is a common disadvantage in these optimization-based methods that they deal with the blurring mechanism with simplified models. Therefore, it is not easy for these methods to achieve best results in real-world cases, especially when the problem is further coupled with noise resulted from low lighting and various other issues such as depth variations and occlusions in motion boundaries.

With the boost of deep learning, esp. convolution neural networks (CNNs) [16], many learning-based methods have been proposed to learn blur kernel, i.e. [17,18]. Deep learning can achieve much better results with the high-performance computer hardware, which is under a rapid development recently. In 2014, authors in [19] first proposed a deep-layered architecture to deblur images. Method [20] formulated the image prior as a binary classifier using a deep convolutional neural network and presented an effective blind image deblurring method based on a data-driven discriminative prior. Sun *et al*. [21] predicted the probabilistic distribution of motion blur at patch level and used a Markov random field model to infer a dense non-uniform motion blur field. Chakrabarti *et al*. [22] developed a network that learns to predict the complex Fourier coefficients of a deconvolution filter to be applied to the input patch for restoration. These methods are highly dependent on modelling blur kernels. Different with non-blind image deblurring, these aforementioned methods are not depended on the known of blur kernel. However,

The first fully convolutional CNN to estimate the latent sharp image directly was proposed by Nah *et al*. [23] using multi-scale residual networks and was adopted by recent

work to further advance the state-of-the-art solutions. Methods [24] and [25] developed multi-scale structures to aggregate features in a coarse-to-fine manner, while showing the benefit of selective parameters sharing and/or recurrent layers. These methods can offer the advantage of enabling generalized dynamic scene deblurring at low latency, by circumventing the iterative optimization stage involving fitting of hand-designed motion models.

In addition, a design [26] was developed composing of multiple CNNs and Recurrent Neural Networks (RNNs) to learn spatially varying weights for deblurring. Zhang *et al*. [27] proposed a multi-patch hierarchical network and stacked its copies along depth. With the birth of the generative adversarial network (GAN), GAN-based methods have been proposed, such as DeblurGAN [28] and DeblurGAN-v2 [29], which reached a trade-off among computing time and restoration accuracy by using different backbones including Inception-ResNet and Mobilenet. It becomes clear that several deep learning models such as RNN and GAN can enhance the performance in deblurring simply by stacking networks, though a deeper understanding would be required in long term research. To this end, the proposed network makes use of the superiority of DenseNet with certain simplifying modifications, while still producing the state-of-the-art performance.

## 1.2 Aims and Objectives

The research described in this thesis aims to investigate and develop advanced deep learning-based approaches (high-performance and effective) for single image deblurring and inverse problems. Specifically, the research aims to explore different network structures, sizes of convolution kernels, network components, and learning strategies, in order to deal with image deblurring effectively. The main objectives of the research in this thesis are;

- To identify the capabilities of different network backbone or component. New single image deblurring methods will be produced and will be compared to the state-of-the-art methods. Experiments and ablation study will be done in order to analyse the value and ability of different network component, including but not limited to residual learning, densely connected structure, and dilated convolution.

- To explore the possibility to use graph neural networks in image restoration. New method will be proposed to convert feature maps in convolution neural networks to nodes in graph neural networks. Suitable graph structure which can be used in image restoration should be explored. Experiments will be done to improve whether the proposed method has a good performance.

- To develop new deblurring networks and improve their performance. The results of the experiments will be used to compare with the state-of-the-art performance. Methods published recent years should be reviewed to produce a comprehensive comparison, in order to improve the performance of the proposed network as well as the value of the study.

## 1.3 Main Contributions

The key contributions of the thesis lies mainly in the network structures for single image deblurring. To this end, the following methods have been developed.

- MixNet: A Balanced Approach to CNN Structure for Blind Single Image Deblurring. A review on the state-of-the-art methods in single image deblurring is conducted. Various key elements in deblurring networks are explored, including backbones, frameworks and parameter sharing. In this part, a new

framework is proposed, termed MixNet based on DenseNet and Inception-v4 to realize such a balanced structure. Results shows that the proposed MixNet can elevate the state-of-the-art performance of end-to-end results for dynamic scene deblurring on all benchmark datasets.

- DC-Deblur: A Dilated Convolutional Network for Single Image Deblurring. This research proposed a novel network by adopting a dilated convolution structure. The research further improves the training process by combining $l_1$ loss, MS-SSIM loss and MSE loss. The proposed network is light and fast. Quantitative and qualitative experiments indicate that the proposed method outperforms state-of-the-art models, in terms of performance and speed.

- Graph Convolutional Networks in Feature Space for Image Deblurring and Super-resolution. A novel encoder-decoder network is proposed with added graph convolutions by converting feature maps to vertexes of a pre-generated graph to synthetically construct graph-structured data. By doing this, the network inexplicitly applies graph Laplacian regularization to the feature maps, making them more structured. The experiments show that it significantly boosts performance for image restoration tasks, including deblurring and super-resolution. It can open up opportunities for GCN-based approaches in more applications.

## 1.4 Thesis Organisation

This thesis has been structured according to the guidelines provided by the University of Manchester. Accordingly, chapters 3-4 are the main contribution of my work, which are

in most instances, been published, been submitted or accepted for publication. The outline of the remainder of the thesis is as follows:

Chapter 2 provides a brief literature review of related work, looking at the basic definition of image restoration, which including image denoising, image dehazing, image super-resolution, and image deblurring. It provides a background on deep learning and reviews learning based image deblurring, as well as generative adversarial network.

Chapter 3 introduces the proposed networks for image deblurring. This chapter first analyse more about densely connected blocks, residual learning, and inception network, and further introduces a balanced network for deblurring. This chapter also analysed the influence of different size of convolution kernels on deblurring and adopted dilated convolution into image deblurring. The experiments show that the proposed methods can improve the performance of deblurring (in regard of PSNR and SSIM).

Chapter 4 introduces graph neural network on feature space for image restoration. A novel encoder-decoder network is proposed with added graph convolutions by converting feature maps to vertexes of a pre-generated graph to synthetically construct graph-structured data. By doing this, graph Laplacian regularization is applied to the feature maps, making them more structured. The experiments show that it significantly boosts performance for image restoration tasks, including deblurring and super-resolution.

Chapter 5 concludes the thesis by summarising the results and findings of the proposed methods and the experiments conducted. Some suggestions for future work are also included.

# Chapter 2

# 2 Literature Review

This research aims primarily at image deblurring using deep learning methods, including model establishment, network framework, novel network and optimizations. To give basic information about image restoration and inverse problems, this chapter first reviews the mainstream image restoration methods. First, image restoration is generally reviewed, which including image denoising, image dehazing, super-resolution, etc. Then this chapter focuses on deblurring and introduce more about single image deblurring, especially learning based deblurring methods. Deep learning (DL) and generative adversarial network (GAN) are also introduced.

## 2.1 Image Restoration

When people take photos by sensors like camera and mobile phone, there are many kinds of degradation people might face to. For instance, motion blurring, out of focus, hazing and raining. Image restoration is a process which restore sharp and clear images

from their degradation inputs. Since an input image or couple of images in a video could be degraded by different case of kernels, such as down-sample, raining and motion blur, image restoration can be divided into many sub-tasks. In this section, three mainstream image restoration tasks are introduced, which including image denoising, image dehazing, and image super-resolution.

### 2.1.1 Image Denoising

Image denoising is the operation which take a noisy image as input and estimate the clean, original image. An example of image denoising can be seen in Fig. 2.1.

(a) (b) (c) (d) (e)

**Figure 2.1: An illustration of image denoising [152]: (a) Wiener filtering [153], (b) Bilateral filtering [154], (c) PCA method [155], (d) Wavelet transform domain method [156], (e) BM3D [157].**

Image denoising techniques have attracted much attention in the past decades [30,31]. Not only because of its great value for real-world application, but also due to the general research value for other image inverse problems. Denoising is always an important task in signal processing. As is analysed in [32], image denoising can be divided as real noisy images, blind image denoising, Additive white noising images (AWBI denoising), hybrid noising images. Due to the achievement of datasets, AWBI denoising has attracted most attention.

Traditionally, WNNM [33] is an important method, where the singular values are assigned different weights. Markov random field (MRF) [34] is also used in image denoising by combining the image model and the optimization algorithm in a single unit.

Learned simultaneous sparse coding (LSSC) [35] jointly decomposing groups of similar signals on subsets of the learned dictionary, trainable nonlinear reaction diffusion (TNRD) [36] learns all the parameters (filters and influence functions) from training data.

Before the proposing of deep learning (DL), neural networks have already been used for image restoration and image denoising. For instance, [37]. Tamura *et al.* [38] made a trade-off between denoising efficiency and performance by a feedforward network. Bendini *et al.* [39] enhance the expressive ability of neural networks by combining maximum entropy and primal-dual Lagrangian multipliers. Paik *et al.* [40] introduced greedy algorithms and asynchronous algorithms in denoising neural network. Performance can also be improved by multilayer perceptron and multilevel sigmoidal function [41]. Related work can also be seen in [42, 43].

After the birth of deep neural networks (DNN) [44], learning based denoising methods have experienced a rapid development. One of the representative methods is DnCNN, which is proposed by Zhang *et al.* [45]. DnCNN is the first end-to-end image denoising method, it contains convolution layer, batch normalization (BN), rectified linear unit (ReLU) layer, and also adopts residual learning [46]. DnCNN has a great influence on learning-based image restoration, mainly on the residual learning strategy.

Other single image denoising network including FFDNet [47], GCBD [48], CBDNet [49], etc. FFDNet [47] can improve denoising speed and process blind image denoising by using different noise levels and the noisy image patch as the input of a denoising network. GCBD [48] uses generated adversarial network to handle input unpaired images. CBDNet [49] uses two sub-networks, one in charge of estimating the noise of the real noisy image, and the other for obtaining the latent clean image.

### 2.1.2 Image Dehazing

The main aim of image dehazing is to sharpen, restore, and retrieval a hazy image. Haze, fog, smog, etc. vitiate the image condition of outside scene. Haze presented in an image can be an abrasive issue as it quells the contrast and changes its colour. Many methods do not work reliably because of it. The presence of haze also cutbacks the clearness of underwater pictures. As a result, removal of these bad weather situations is a critical and unavoidable area of image processing [50]. An illustration of image dehazing can be seen in Fig. 2.2.

In this section, a brief discussion is given about single image dehazing. In almost every practical scenario the light reflected from a surface is scattered in the atmosphere before it reaches the camera. This is mainly due to the presence of aerosols which deflect light from its original course of propagation. In long distance photography or foggy scenes, this process has a substantial effect on the image in which contrasts are reduced and surface colours become faint.


**Figure 2.2: An illustration of image dehazing [51].**

In [88], He *et al.* proposed dark channel prior (DCP) for dehazing based on an observation that most local patches in outdoor haze-free images contain some pixels whose intensity is very low in at least one colour channel. DCP is a very important benchmark in image dehazing. Narasimhan *et al.* [48] presented a physics-based model that describes the appearances of scenes in uniform haze images. Shwartz *et al.* [49]

remove the haze effect through two or more images taken with different degrees of polarization. Tan et al. [52] removes haze by maximizing the local contrast of the restored image based on the observation that a haze-free image must have higher contrast compared with the input hazy image.

Under the rapid development of deep learning, many learning based dehazing methods have been proposed. For instance, Li *et al.* [53] proposed an all-in-one method to directly generate clean images through a light-weight CNN, based on a reformulated atmospheric scattering model. Qu *et al.* [54] introduced a GAN for dehazing by using the discriminator to guide the generator to create a pseudo realistic image on a coarse scale, while the enhancer following the generator was required to produce a realistic dehazing image on the fine scale. In [55], Chen *et al.* adopted smoothed dilation technique and leveraged on a gated sub-network to fuse the features from different levels.

### 2.1.3 Image Super-resolution

Single image super-resolution is also a kind of image restoration problem and a typical inverse problem, for the solution is not only one due to the ill-posed nature of image degradation. Especially, image super-resolution is to recover a high resolution (HR) image from a low resolution (LR) input, which might be produced by bicubic down sample or down sampling with other kernels. An illustration of single image super-resolution can be seen in Fig. 2.3.

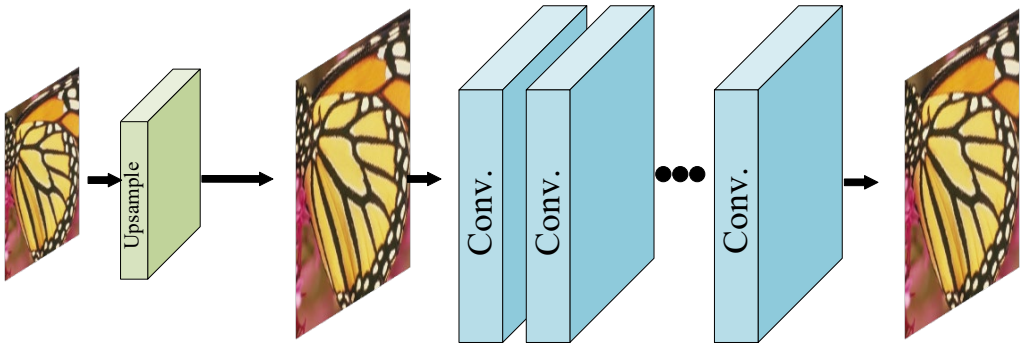



**Figure 2.3: An illustration of single image super-resolution [56].**

Traditionally, Dong *et al.* proposed SRCNN [56] to adopt deep convolutional network into solving image super-resolution. Lim *et al.* [57] proposed improved the performance by removing unnecessary modules in conventional residual networks. Based on [57], Zhang *et al.* [58] adopted residual in residual (RIR) and residual channel attention networks (RCAN) [58] to further improve the network.

## 2.2 Single Image Deblurring

Low resolution image, noisy image, and hazy image can be generalized as different kind of blur. To find a more typical scene for image inverse problem, image deblurring is selected as the main task of this thesis. Given a blurred image $y$, which is often a three dimensional image, deblurring is to recover its original sharp image $x$ by deconvoluting with kernel $k$, which is typically a two dimensional convolution kernel, as follows,

$$y = k * x + n \tag{2.1}$$

where $*$ is the convolution operator and $n$ denotes noise, which is often treated as white Gaussian. This equation generally assumes uniform blur, but in real world $k$ is not spatially invariant throughout the image.

Image deblurring can be divided into traditional method and learning based method by whether machine learning is used. Before analysing learning-based image deblurring, in this section, traditional single image deblurring methods are firstly introduced, in order to basically establish a mathematical model. Consider an imperfect optical system, a ray of light passing through the optical setup will spread over the image domain instead of converting to a single end point [7]. This process can be modelled by point spread function (PSF). Based on this fact, image deblurring can be further divided into non-blind deblurring and blind deblurring by whether the PSF is known or not. It can also be divided into uniform and non-uniform deblurring by whether the PSF is uniform throughout the

image. Since a review is made by Mahdi *et al.* in [7], several main methods are briefly introduced.

Statistical Priors: Statistical methods are to estimate the underlying image (sharp image) as a conditional probability of a given blurry observation (blurry image) by maximum-a-posterior (MAP) estimation. In the early beginning, this method was proposed by Richardson-Lucy (RL) [59,60] by recasting the solution in an iterative algorithm starting from an initial guess. The accelerated RL algorithm was proposed later in [61] by using an adaptive line searching technique. Comprehensive surveys for the early development of these methods can be seen in [62]. In recent decades, digital cameras experienced a rapid development, and raised the requirements of image deblurring. Thus, more practical deconvolution methods were proposed [9, 63-65]. To numerically solve these problems, many variants of sparse reconstruction which recasts the regularization problem as an iterative procedure where dominant feature coefficients are preserved during each iteration [7].

Tikhonov Regularization: Tikhonov Regularization (aka ridge regression / weight decay) is a regularization method of ill-posed problems. It provides improved efficiency in parameter estimation problems in exchange for a tolerable amount of bias. When the data fidelity is regularized in $l_2$-norm space to minimize a cost function, it becomes a variant of the Tikhonov regularization problem [7]. The solution can be given by quadratic minimization that can be accelerated by fast Fourier transform (FFT) which can reduce the computational complexity by an order of $O(n\ \log n)$. Early application of Tikhonov Regularization was deployed in the classical Wiener deconvolution to regulate the image spectrum in the Fourier domain using a non-linearly weighted inverse blur response [7]. Based on the requirement of faster image restoration, more methods have been developed under this framework [66].

Total Variation (TV): The priors for either the blur kernel or latent image can be regulated by the TV-norm [67], which preserves sharp edges while preventing ringing

artifacts for recovery. However, these methods suffer from visual blocking or “staircasing” artifacts.

Besides, many methods combined different approaches and used more than one regularization prior for image restoration [7]. This becomes more useful when both blur and image priors (in the blind case) could be fit into one regularization framework to address more complex formulations. The common practice is to use split variable techniques to recast the algorithms in parallel and independently update each sub-modular task [68].

## 2.3 Deblurring with Deep Learning

However, traditional method cannot deal with heavy blurred image, and optimization-based methods often cost lots of time. Recently, with the rapid development of deep learning, esp. convolution neural networks (CNNs) [16], many learning-based methods have been proposed to learn image deblurring, i.e. [17,18], which boosted the performance of single image deblurring.

Deep learning has been boosted with the development of computational hardware. In 2014, authors of [19] first proposed a deep-layered architecture to deblur images. Method [20] formulated the image prior as a binary classifier using a deep convolutional neural network and presented an effective blind image deblurring method based on a data-driven discriminative prior. Sun *et al*. [21] predicted the probabilistic distribution of motion blur at patch level and used a Markov random field model to infer a dense non-uniform motion blur field. Chakrabarti *et al*. [22] developed a network that learns to predict the complex Fourier coefficients of a deconvolution filter to be applied to the input patch for restoration. These methods are highly dependent on modelling blur kernels. The first fully convolutional CNN to estimate the latent sharp image directly was proposed by Nah *et al*. [23] using multi-scale residual networks and was adopted by recent work to further

advance the state-of-the-art solutions. Then, [24-29] are followed. Details of these methods is given in chapter 3.2.2, baseline models. In this section, relations between different approaches are focused on.

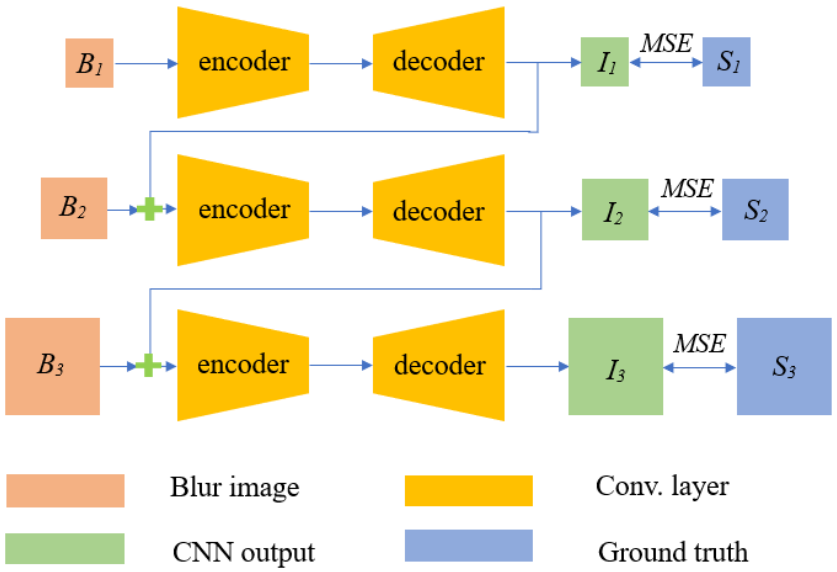



**Figure 2.4: An illustration of multi-scale structure in deblurring.**

Multiscale based deblur networks use the “coarse-to-fine" structure to restore an image with several steps. Nah *et al*. [23] first applies the multiscale structure in image deblurring, and this structure is further developed by Eigen *et al*. [111]. Tao *et al*. [24] and Gao *et al*. [25] developed. Zhang *et al*. [27] made a fundamental subsequent change to the structure and mechanism, leading to marked differences to the other three methods. A typical multiscale structure is shown in Fig. 2.4. A blur image is resized to 1/4 resolution, $B_3$, as the input. Then $I_3$ becomes $B_2$, the input of the second scale, which is transferred and resized to 1/2 resolution by up-convolution. Then $I_2$ becomes $B_1$ in the third scale, which is the same size as input image, and $I_1$ is the output of the entire network.

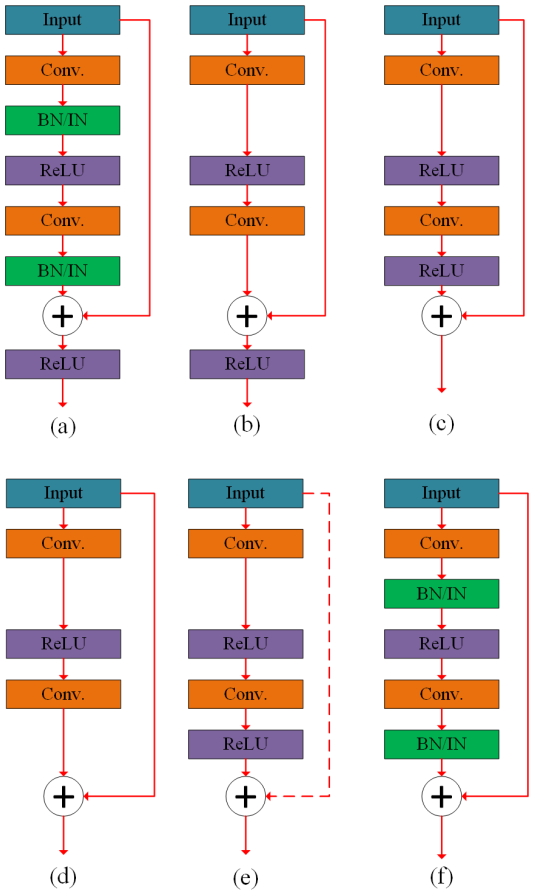



**Figure 2.5: Different modified ResBlocks: (a) original ResBlock proposed in [25], (b) modified ResBlock used in DeepDeblur [23], (c) modified ResBlock used in SRN-Deblur [24], (d) convolution blocks used in DSHMN [27], (e) convolution block adopted by PSS-NSC [25], and (f) modified ResBlocks used by DeblurGAN [28].**

The modification of convolutional blocks is also a mainstream. As shown in Fig. 2.5, many methods try to remove normalization or ReLU layers. In the original ResBlock, two BN layers are used, and there is a ReLU layer after skip connection. While in DeepDeblur [23], BN layers are removed. In SRN-Deblur, ReLU layer is placed before skip connection. A typical ResBlock in recent image deblurring network is given in Fig 2.5 (d), which proposed in DSHMN [27], there is only one ReLU layer left. According to EDSR [57], normalization is not valuable in image restoration.

It has to be noted that for the ResBlock in PSS-NSC [25], there is no residual link between the input and the output of residual blocks, but due to the adoption of densely connected structure, the connection actually exists as indicated by the dotted line. This was reported to give better performance. By comparison with some early work, it can be concluded that ResBlock has become an indispensable component for deblurring. Furthermore, PSS-NSC has the best performance among those methods using DenseNet, and RADN-Deblur [69] which also used pre-trained DenseNet as the main component.

Recently, Cai *et al.* [70] proposed a Dark and Bright Channel Priors embedded Network (DBCPeNet) with a trainable DCP layer and sparse regularization to plug the channel priors into a neural network for deblurring. Li *et al.* [71] used depth map to consider image deblurring. Zamir *et al.* [72] proposed MPRNet which first learns the contextualized features using encoder-decoder architectures and later combines them with a high-resolution branch that retains local information. Rozumnyi *et al.* [73] proposed DeFMO by a generative model which outputs the motion-blur object's appearance and position in a series of sub-frames as if captured by a high-speed camera. Chi *et al.* [74] proposed a novel self-supervised meta-auxiliary learning which enables internal training within a test image from scratch, while not fully rely on large external datasets.

Apart from single image deblurring, video deblurring / multi-image deblurring is also a hot task in image / video restoration [75-78]. Although this thesis mainly focuses on single image motion deblur, image blur is not limit to motion blur. There are many other blur conditions which have great application value. i.e., out-of-focus deblurring [79-82].

## 2.4 Generative Adversarial Network

A generative adversarial network (GAN) is a class of machine learning frameworks designed by Ian Goodfellow and his colleagues in 2014 [83]. Two neural networks contest

with each other in a game (in the form of a zero-sum game, where one agent's gain is another agent's loss).

Given a training set, this technique learns to generate new data with the same statistics as the training set. For example, a GAN trained on photographs can generate new photographs that look at least superficially authentic to human observers, having many realistic characteristics. Though originally proposed as a form of generative model for unsupervised learning, GANs have also proven useful for semi-supervised learning, fully supervised learning, and reinforcement learning.

The idea of generative adversarial networks, introduced by Goodfellow *et al.* [83], is to define a game between two competing networks: a discriminator and a generator. The generator receives information from an input and generates a sample. The discriminator learns from real and generated samples and tries to distinguish between them. The goal of the generator is to fool the discriminator by generating perceptually convincing samples that cannot be distinguished from the real samples. The game between the generator $G$ and discriminator $D$ has the minimax objective:

$$\min_G \min_D \boldsymbol{E}_{a \sim P_r} [\log D(a)] + \boldsymbol{E}_{\tilde{a} \sim P_g} [\log(1 - D(\tilde{a}))]. \quad (2.2)$$

Where $P_r$ is the data or real sample distribution and $P_g$ is the model distribution. GANs are known for its ability to generate samples of good perceptual quality in vision tasks. There is a development of GAN, which is called WS-GAN, as briefly reviewed in Appendix D.

## 2.4 Summary

Learning based methods lead to great change for image restoration. Deep learning approaches bring many benefits to image restoration such as,

- Learning based methods can boost the performance. On most benchmark datasets, deep learning based methods often outperform the traditional methods significantly.
- Deep learning makes applications more realistic. One can restore a video from degradation by taking sequential frames into consideration or fill in some missing content while it is impossible for the degradation process to be modelled mathematically (e.g., inpainting).
- By using parallel processing units such as graphic processing units (GPUs), deep learning algorithms naturally fit with computer hardware, leading to high efficiency compared to using CPUs. In addition, learning based image restoration significantly reduces running speed by using graphic processing units (GPUs), while traditional methods often need long time for iteration.

However, there are still many challenges:

- From the perspective of computational complexity, deep learning based methods still have considerable computational costs, making them difficult to deploy in real-time processing. In addition, matrix processing leads to higher requirement for computer hardware, in terms of GPUs and memory, which cannot be satisfied by the embedded systems used commonly in industry, e.g., microcontroller unit (MCU). Thus, lightweight model for image processing should be explored.
- From the perspective of performance, existing algorithms still have a lot of room for improvement. For instance, deblurring network still cannot restore clear image while blur kernel is very large.
- The design of learning-based image restoration methods still based on experience and some rules in other computer vision tasks such as image classification and detection. Deeper understand for the network structure and components for image-to-image transfer and inverse problems should be explored.

This thesis is to explore different ideas and try to find solution for the aforementioned challenges.

# Chapter 3

# 3 Deep Learning for Image Deblurring

This chapter introduces deep learning applications in image deblurring. The chapter starts with an introduction of the state-of-the-art learning based single image deblurring network, and their shortage. Then the chapter analysis the related work mainly focusses on the development of network structure and backbones. Based on the tendency of development, MixNet and DC-deblur are further introduced and proposed by different modification on network. In the subsection of experiments, the implementation of network is firstly introduced, and followed by an introduction of datasets. This chapter also presents experiment results and ablation study in order to prove that the proposed methods are effective.

## 3.1 Introduction

This chapter mainly focus on the design of image deblurring network. Motion deblur often suffer from poor performance, especially when blur kernel is very large. In Deepdeblur proposed by Nah *et al.* [23], satisfied output image cannot be obtained while the network

causes very long time to run, although the time is much shorter than traditional methods [23]. Many researchers want to deal with the problem by parameter-sharing [24, 25]. Although parameter-sharing cannot greatly reduce computational complexity, it can reduce the model size obviously.

Multi-scale model has been adopted to achieve a coarse-to-fine structure to help boost the performance. Then GAN model is proposed, to generate more details. However, there still in short of a review about the network structure, how each component work on image deblurring, and what kind of network is suitable and reasonable for image deblurring. Specifically, there are three main considerations on single image deblurring, which are network framework, learning strategy, and network backbone, respectively.

In this chapter, the new methods are introduced mainly based on these three considerations. To this end, a review of the tendency of the development of these tasks is given first, mainly focus on network framework and network backbone.

## 3.2 Related Work

### 3.2.1 Baseline Models

In this subsection, some baseline models are further introduced. They are used in experimental comparison.

DeepDeblur [23] is the first kernel-free method and hence it does not suffer from kernel-related problems. Structurally, the authors employed a modified version of residual network structure (ResBlock) as the backbone. The main change was the removal of the Rectified Linear Unit (ReLU) after the skip connection to boost convergence speed. The removal of ReLU makes all parameter values positive, which may not be useful in image restoration as compared to the mechanisms for object detection and other high-level vision tasks. To convert a coarse output to fit the input size of next finer scale, the authors adopted up-convolution layers instead of up-sampling as it showed better performance.

As shown in Fig. 2.4, each output is involved to the calculation of loss function, thus the content loss is given by

$$\mathcal{L}_{cont} = \sum_{i=1}^{n} \frac{1}{N_i} \|I_i + S_i\|^2 \tag{3.1}$$

where $I_i$ and $S_i$ are the network output and ground truth, respectively at the $i^{th}$ scale. $N_i$ is the number of elements to normalize. The network combines the content loss and adversarial loss, which is given as

$$\mathcal{L}_{adv} = \mathbb{E}_{x \sim P_{sharp}(x)} [\log D(x)] + \mathbb{E}_{y \sim P_{blurry}(y)} [\log(1 - D(G(y)))] \tag{3.2}$$

where $x$ denotes the sharp image and $y$ the blurred image. $G$ and $D$ are the generator and discriminator, which often implemented by deep neural networks, respectively. So, the total loss is given by

$$\mathcal{L} = \mathcal{L}_{cont} + \lambda \times \mathcal{L}_{adv} \tag{3.3}$$

where the weight constant $\lambda = 1 \times 10^{-4}$. The multiscale structure has profound influence in image deblurring.

Following DeepDeblur, Scale-Recurrent Network (SRN-Deblur) was proposed by Tao *et al.* [24] to further enhance the performance. The authors used Recurrent Neural Network (RNN) to record the time-domain changes in blur images. RNNs were added between the encoder and the decoder. SRN-Deblur followed the simplified ResBlock used in [23] and further implemented parameter sharing. Three different scales used the same parameters, significantly reducing the number of trainable parameters. This is similar to using data multiple times and could be implemented as data augmentation. Besides, it enables cross-scales restoration for the hidden states to capture useful information.

PSS-NSC network [25] was influenced by densely connected structures, DenseNet [38]. Its framework is similar to that of SRN-Deblur, while replacing ResBlocks with DenseNet and removing the LSTM block. The DenseNet blocks in PSS-NSC network, termed as DenseBlocks, were modified based on [38]. The parameter sharing is adapted in both intra-scale and cross-scale, which means for the whole 3-scale network, there are

six different DenseNet blocks, three in the encoder and three in the decoder. In each forward process, totally 36 DenseNet blocks are used, and each block is used six times.

Feature aggregation over multiple patches has been used in image classification [84]. For example, spatial pyramid matching (SPM) [84] was proposed to divide images into coarse-to-fine grids in which histograms of features were computed. Focusing on the problem that deconvolution operations is time consuming in the coarse-to-fine scheme, Zhang *et al.* [27] presented a deep hierarchical multi-patch network inspired by SPM. Although this network has a multiscale structure, it is different to the three aforementioned methods. In contrast to DeepDeblur that uses deconvolution links, feature map concatenations are used in this method. The authors also proposed a novel stacking paradigm for deblurring. Instead of making the network deeper vertically (adding finer levels), they proposed to increase the depth horizontally by stacking multiple network models to perform deblurring. It does not combine all the scale outputs in the calculation of loss but only uses the last output.

DeblurGAN was proposed by Kupyn *et al.* [28] as the first to exploit conditional GANs for deblurring. It uses the residual network block [25] as the main component of the generator and blur images as the input. The generator produces a latent image, and the discriminator tries to distinguish it from the ground truth. The result of the discriminator leads to the adversarial loss, which is combined with the content loss, i.e., the MSE loss. The network achieves good performance, esp. in terms of structural similarity (SSIM).

DeblurGAN-v2 [29] is an update of DeblurGAN. feature pyramid network (FPN), originally proposed for object detection [85,86], was used as the generator. A relativistic discriminator [87] with a least-square loss (LSE) wrapped [88] inside was proposed for the discriminator, to evaluate both global (image) and local (patch) scales. For the backbone, a sophisticated Inception-ResNet-v2 backbone was plugged to pursue the state-of-the-art deblurring quality, and towards being more efficient, the authors adopted the MobileNet-V2 [89] and further created its variant with depth-wise separable convolutions (MobileNet-DSC). DeblurGAN-v2 also uses blur-sharp image pairs to train.

The authors of [90] presented an end-to-end deblurring network by unsupervised CNNs. The previous supervised deep-learning networks extensively rely on large sets of paired images, which are highly challenging to obtain, while unsupervised training scheme with unpaired data can achieve the same. The model is also a GAN that learns a strong prior on clean or sharp images with adversarial loss and maps blurred images to their clean equivalent. To improve stability of the model and to preserve the image correspondence, the authors introduced an additional CNN module to reblur the generated output to match with the blur input. Along with these two modules, the authors also made use of the blurred images to self-guide the network to constrain the solution space of generated sharp images, by imposing a scale-space gradient error with an additional gradient module.

Another approach to image restoration is to design special convolution blocks. These methods give further and deeper view of image blur processes and design special blocks base on these processes. But they cannot be easily used in other image restoration problems such as super-resolution and denoising.

Zhang *et al.* [26] gave a view that the deblurring process was equivalent to an Infinite Impulse Response (IIR) model, which could be approximated by RNNs. They further presented the structure of the spatially variant RNN for dynamic deblurring, with the RNN weights learned by CNNs.

Authors of [91] proposed a novel network called Dr-Net. They used the Douglas-Rachford iterations to solve deblurring problem because it was a more applicable optimization procedure than the proximal gradient descent algorithm. Two proximal operators originate from these iterations, one for the data fidelity and one for the image prior. It is non-trivial to design a hand-crafted function to represent these proximal operators which would work with real-world image distributions. The authors therefore approximated both these proximal operators using deep networks.

In this thesis, the comparisons are mainly based on the aforementioned methods due to their similarity of the proposed method.

### 3.2.2 Network Structure

The balance between performance and speed is a main challenge of existing algorithms. A balanced approach to CNN structure relies on a comprehensive understanding of key factors of networks. Based on the literature reviews in the previous section, this section provides an analysis and further discuss guidelines on two key factors in the CNN-based approach to deblurring: backbones and frameworks, which affect the performance, efficiency and memory usage.

Because the choice of network backbone has a direct impact on deblurring quality and efficiency, researchers have explored and adopted various backbones in the literature and hence backbone can be regarded as a main difference among various methods. To pursue better network ability, better backbones should be found in this work.

Some researchers used special blocks in the middle of the network, between encoder and decoder. Special module based deblurring methods often have good ability. Main reasons can be summarized to two aspects. First, technical tricks, especially image priors, can restrict the space of parameters in training. Second, deblurring networks often have two or more downsample operations to enlarge receptive field and reduce computational complexity. Thus, features in encoded feature map are more abstract and more efficient. Therefore, adding these tricks can bring benefits. On the basis of this, finding suitable modules that are capable of these is an important and challenging issue for deblurring.

Another factor is parameter sharing, first introduced in deblurring in. It can be divided as intra-scale sharing and cross-scale sharing. As discussed in [24], it can reduce the number of parameters. The training of shared weights works in a way similar to using data multiple times, which actually amounts to data augmentation. In [25], both intra-scale sharing and cross-scale sharing were used. The experiments indicate that combining of intra-scale sharing and cross-scale sharing may suffer from containment due to complex interactions in training, thus may not lead to better performance. Such results

can be seen in section 3.5. In addition, recent work used simple encoder-decoder without parameter sharing and achieved excellent performance.

The most basic framework used in image deblurring is the encoder-decoder structure, which has been shown to perform well [24]. For some multiscale network like DeepDeblur, encoder and decoder are used in each scale, thus the input image will experience several encoders and decoders in forward inference. Multiscale deblurring structure was used in the first end-to-end deblur network, a coarse-to-fine structure, and also achieved good performance in other image restoration tasks. However, without parameter sharing, multiscale structure would lead to a large number of parameters, making training and testing slower.

FPN was used in deblurring firstly in DeblurGANv2. Extensively experimented the FPN structure, as reported in section 3.5, but the results indicate that FPN is not particularly helpful for deblurring, though it shows advantages in some other inverse problems such as image denoising. In general, FPN structure has simple and thin decoder stage in terms of image restoration, thus not helpful to improving reconstruction. Although the multiscale structure can improve the performance as indicated in [24], it does not always, especially when being combined with complex parameter sharing.

Based on these observations on multiscale structure and FPN, it can be found that complex structures do not necessarily bring significant performance improvement. Hence, this work further proposes principles on using simple network structures to achieve good performance. The principles can be summarised as follows:

- Residual structure is indispensable for image deblurring, e.g., ResBlock, and DenseNet.
- Special modules based on blur mechanism should be added in between encoder and decoder to maximize the use of semantic extraction capabilities of CNNs.
- Simpler is better. A complex architecture may not necessarily perform well in deblurring.

## 3.3 MixNet

### 3.3.1 Residual Learning

Residual learning is proposed by He *et al.* in [46]. It is proposed to avoid the problem of vanishing gradients, or to mitigate the degradation (accuracy saturation) problem in order to help training network. Then, in DnCNN [45], residual learning is used in image restoration, thus the network is actually learning the residual information. An illustration of residual learning can be seen in Fig. 3.1

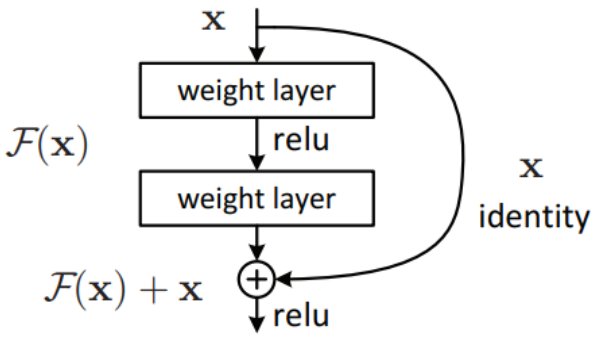



**Figure 3.1: An illustration of Residual Learning [46].**

### 3.3.2 Densely Connected Blocks

Densely connected network is proposed in [92] and termed as DenseNet, as is shown in Figure 3.2. In DenseNet, each layer obtains additional inputs from all preceding layers and passes on its own feature-maps to all subsequent layers. Each layer is receiving a "collective knowledge" from all preceding layers. An illustration of DenseNet can be seen in Fig. 3.2.

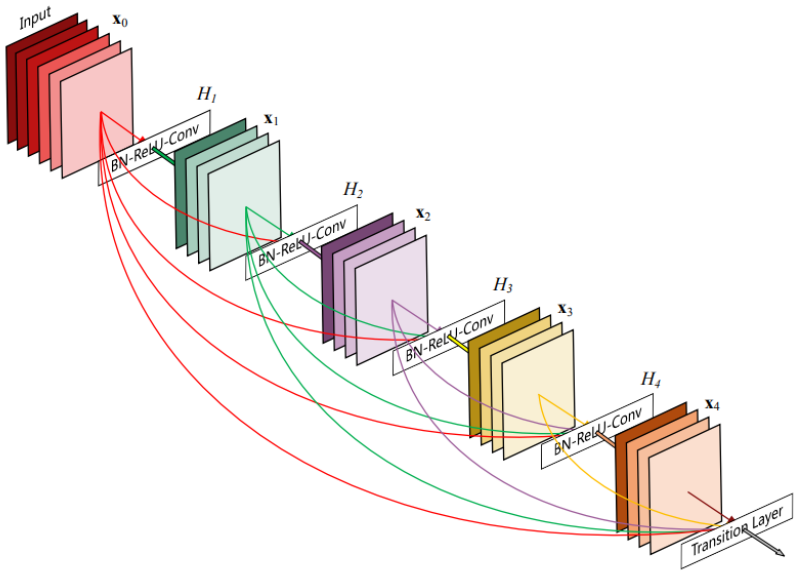



**Figure 3.2: An illustration of DenseNet [92].**

### 3.3.3 Proposed Network with Mixed Backbone

As is analysed in section 3.3.2, DenseNet should be used in network to improve the performance. In addition, since a simple network structure is the goal of this work, parameter sharing is removed. Theoretically, parameter sharing improves the network mainly due to more convolution layers. The value of shared parameters is producing a deeper network while make the network not very hard to train.

In this thesis, we also use Inception-v4 [158]. Inception architecture that has been shown to achieve very good performance at relatively low computational cost. Inception structure can extract information from different scales by its parallel strategy, which can replace the function of feature pyramid structure. Since Inception-v4 is used as a special block, it should be inserted between encoder and decoder based on the conclusion in section 3.3.2.

As shown in Fig. 3.3, the network consists of convolutional layers, DenseBlocks and Inception-A blocks, where $I_b$ and $I_s$ denote input blur image and output image, respectively. The backbone of the network uses DenseBlocks with ResBlock as its elements for enhanced receptive field. By default, each nonlinear DenseBlock module has four processing units. Structures of the ResBlocks used are indicated in Fig. 2.5 (e), composed of two convolutional layers. For the framework, based on the analysis in section 3.2, both multiscale structure and parameter sharing mechanism are removed in the proposed network, leading to a simplified network. An encoder-decoder structure is used based on 12 DenseBlocks with independent parameters. Four Inception-A blocks are added in the middle of the network, to widen the feature maps. Inception-A is one of the components of Inception-v4 and has suitable input size and feature map width for image restoration, containing 382 channels. For Inception-v4, there are five kinds of blocks: Inception-A, Reduction-A, Inception-B, Reduction-B and Inception-C. But these blocks except for Inception-A have over 1000 features and hence are not suitable for image reconstruction. By adopting inception-A, the proposed network has mixed feature extraction mechanism combined with DenseNet and Inception-A network and therefore is termed as MixNet. Different from using kernel size of $5 \times 5$ [23,24], kernel size of $3 \times 3$ is used to control the model size, since 2 layers with $3 \times 3$ kernels can cover the same receptive fields as one layer with $5 \times 5$ kernels but saving around 25 % of parameters.

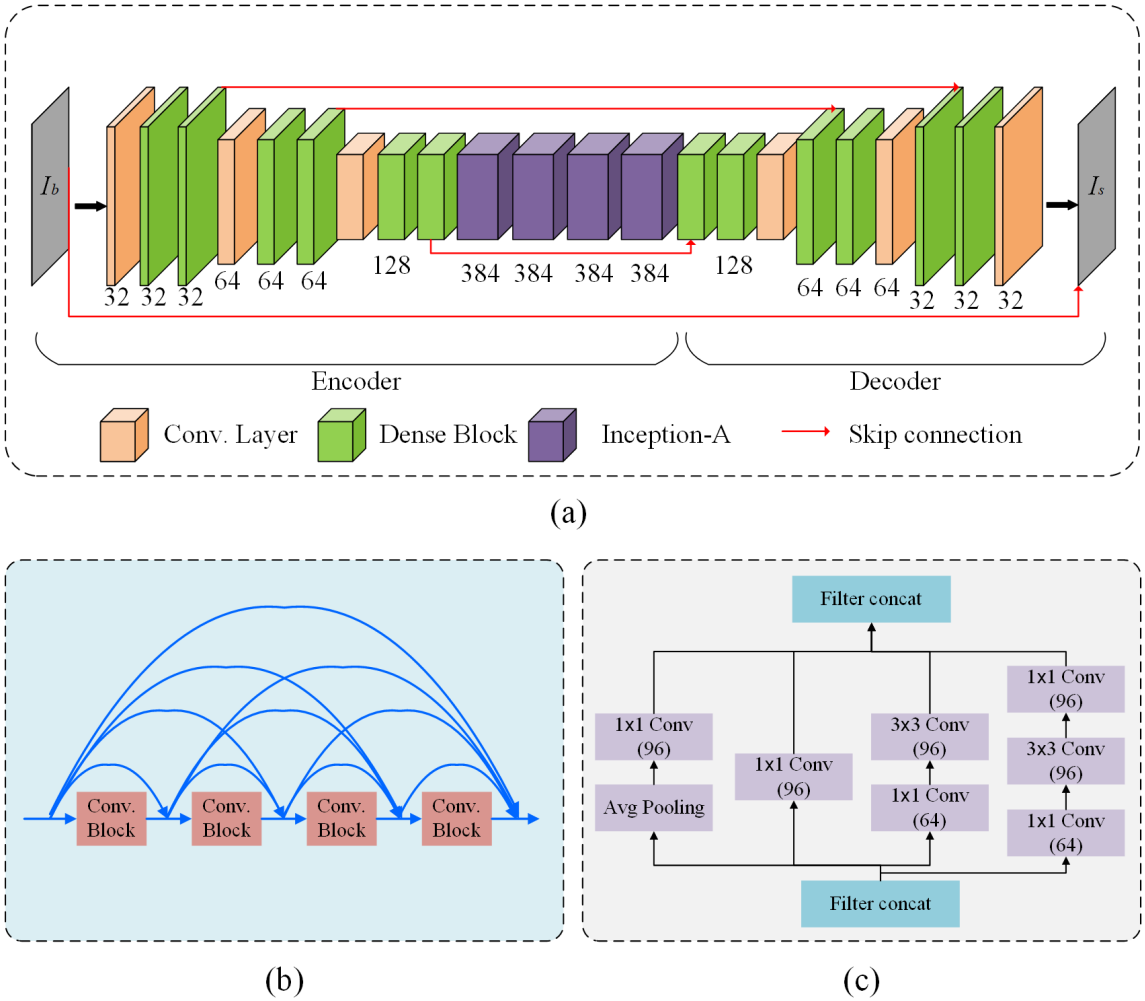



**Figure 3.3: The proposed network for image deblurring, (a) the overall architecture, (b) the DenseBlock structure, and (c) Inception-A block.**

Loss function is another significant element of image deblurring network. Based on the review in section 3.2, the MSE loss is referred to as the most important loss function in image deblurring. It has direct relationship with PSNR, which is one of the most important measures in evaluation, given as,

$$psnr = 10 \log_{10} \frac{255^2}{mse} \tag{3.4}$$

In this work, MSE loss is adopted as the loss function. In the experience, adding other auxiliary loss such as SSIM loss or adversarial loss may not always have significant effect.

## 3.4 DC-Deblur

### 3.4.1 Motivation of Using Dilated Structure

With rapid development of Deep Convolutional Neural Networks (DCNNs) [24], learning based methods have become the mainstream for image restoration, e.g., image denoising [45], super-resolution [56] and image deblurring [23]. In some cases, a network that performs well on one of the aforementioned tasks can also work on other image inverse problems. For instance, the method in [124] performs well on super-resolution and edge-preserving filtering and has been shown in [55] to work well on image dehazing and deraining. However, in many cases such transferable capability is hard to find. Image deblurring is a common restoration problem, yet networks that perform well on denoising or dehazing cannot be adopted for deblurring directly. For instance, adopting [93,55] for deblurring has led to poor performance. To investigate the reason, it has been observed that deblurring networks often require larger model sizes and more parameters than other image restoration networks. For example, a state-of-the-art dehazing and deraining network Gated Context Aggregation Network (GCANet) [108] has a size of 2.7 MB, yet typical deblurring networks like PSSNSC [25] have a size of 46.5 MB. Note that PSSNSC adopts parameter sharing, thus, consider the full computational complexity, PSSNSC would be more expensive. Earlier deblurring networks such as DeepDeblur [23] have an even larger size of 303.6 MB.

The requirement for large reception field is the core reason for this condition. Due to the mechanism of image blurring, blur kernels could be very large, and usually dozens of pixels involving in blur kernels. Thus, deblur networks need large reception fields both in feature extraction and restoration in order to model and handle the blur kernels. Generally, previous work increases the reception field by using more convolution layers [23]. It is undeniable that deeper networks could increase the performance, but it would also increase the complexity, making the forward processing slow. Many researchers have tried to develop faster and better networks, but architectures of these networks are still very complex [29].

To make a network more efficient, the network structure should be able to cover a large reception field without adding too many parameters. To this end, a dilated convolution network approach is adopted. Dilated convolution (or atrous convolution) was originally developed for wavelet decomposition [94]. The main idea is to insert "holes" (zeros) between pixels in convolutional kernels to increase the convolution range, thus enabling dense feature extraction in DCNNs. In this section, a dilated convolution based network for single image deblurring is proposed, termed as DC-Deblur. For the framework and backbones of the network, an encoder-decoder structure is adopted and use densely connected structure to extract features and help feature restoration.

### 3.4.2 Dilated Convolution in Image Deblurring

Dilated convolution, also known as atrous convolution, was first proposed in [95], and has been widely used in many computer vision tasks such as object detection [96], audio generation [97], and video modelling [98]. It can significantly increase the reception field and extract global information by inserting zeros between weights, without requiring additional parameters. Consider a one-dimension input $x$, the output $y$ for a dilated convolution at location $i$ is defined as

$$y(i) = \sum_{j=1}^{J} f(i + r * j)\, x(j) \tag{3.5}$$

Where $f$ is the filter implemented by convolutional layer with kernel size $J$, and dilation rate $r$. For image deblurring, if consider standard convolutions as dilated convolutions with a dilation rate of $r = 1$, a downsampling layer with a subsampling rate of 2 can be removed by letting the dilation rate of all subsequent layers be 2. This results in dilated convolutional layers with dilation rates of $r = 2, 4, 8$, etc.

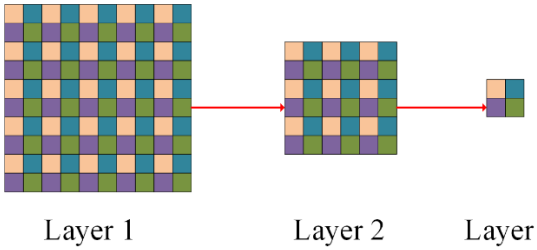



**Figure 3.4 Illustration of the gridding problem. Consider a $3 \times 3$ convolution kernel and a dilation rate $r = 2$, there are four neighbouring units with different colours in layer 3. Their actual reception fields in layer 2 and layer 1 are marked using the same colour. Obviously, the reception fields completely separate with each other.**

However, when using dilated convolutions to restore an image, the gridding artifacts affect the models significantly. In [99], smoothed dilated convolution was proposed by using the original dilated convolution parameters (with the hole 0 removed) to perform convolutions and then periodically sampling the original feature map into 4 feature maps with reduced resolution, and then the convolution results were combined by up-sampling.

In this section, the proposed DC-Deblur network is introduced. The overall structure of the network is shown in Fig. 3.7. Given a blurry image $I_{in}$, $I_{in}$ is encoded by an encoder and process the information by a dilated convolution structure and gated fusion. Then the gated feature map will be decoded to a blur residue by a decoder structure, and finally add to the restored sharp image by using residual link. Densely connected structure blocks [92] are used for the encoder and the decoder, and in the rest of the section, they are termed as DenseBlocks.

Dilated convolutions often suffer from the gridding artifacts, as shown in Fig. 3.4. To handle this problem, smoothed dilated convolution proposed by Wang *et al.* [99], was successfully adapted for image dehazing and deraining in GCANet [108]. However, some

details can be further improved for deblurring. Specifically, separable and shared (SS) convolutions are used for technical implementation. Consider a convolution layer with input and output channels $C$ (e.g., 64)and kernel $n \times n$, for instance, $3 \times 3$, separable convolution handles each channel separately. In contrast with a standard convolution that connects all $C$ channels in input to all $C$ channels in outputs, leading to $n^2 \times C^2$ parameters, a separable convolution only connects the $i^{th}$ output channel to the $i^{th}$ input channel, resulting in only $n^2 \times C^2$ parameters. Based on separable convolutions, shared convolution means the same $n^2 \times C$ parameters are shared by all pairs of input and output channels. SS convolutions only have one filter scanning all spatial locations for input and output of $C$ channels and share this filter across all channels. $3 \times 3$ in SS convolution is used, and thus can help fusing the information of neighbour pixels to help avoid gridding artifacts, as is shown in Fig. 3.5.

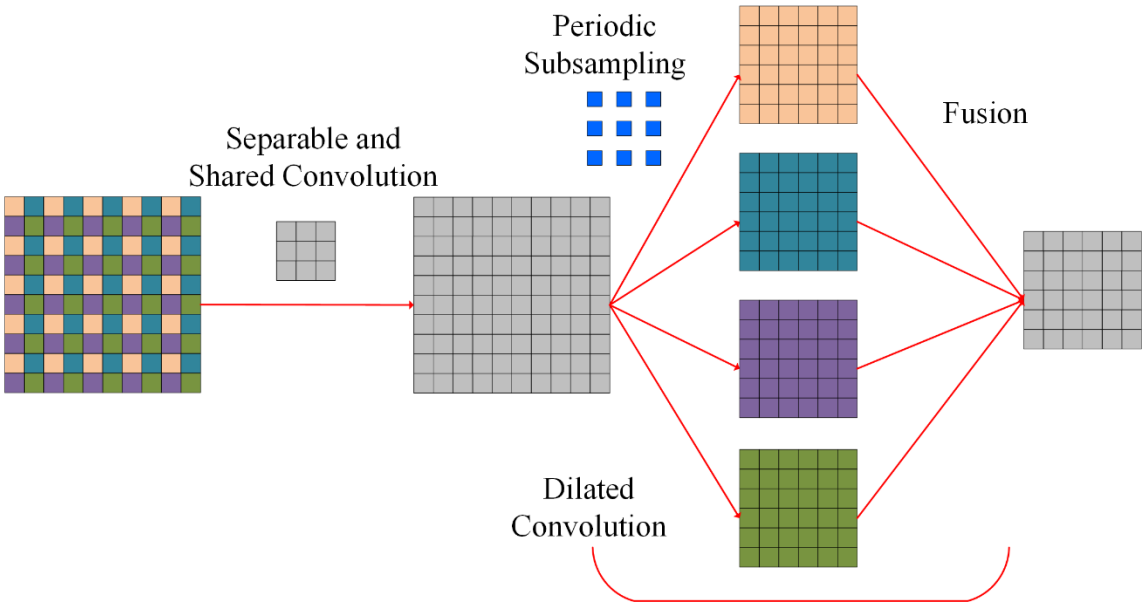



**Figure 3.5 Illustration of the smoothed dilated convolution method. Separable and Shared convolution is used to produce a smoothed feature map, and then periodic subsampling is used to implement dilated feature extraction. The four outputs are then fused into a new smoothed dilated layer. The grey map represents smoothed feature maps.**

This section further analyses the difference between image dehazing/deraining and image deblurring and modify the dilated convolution blocks by removing normalization layers, as is shown in Fig. 3.6. In DeepDeblur [23], it was found that removing the batch normalization (BN) unit in the original residual building block benefited the convergence and training time. Since normalization layers normalize the features, they reduce the range flexibility by normalizing the features [57]. Hence it is better to remove them in image restoration. Based on the previous analysis, removing these normalization layers would benefit the performance. Experiments verified this idea, and further details are given in section 3.5.

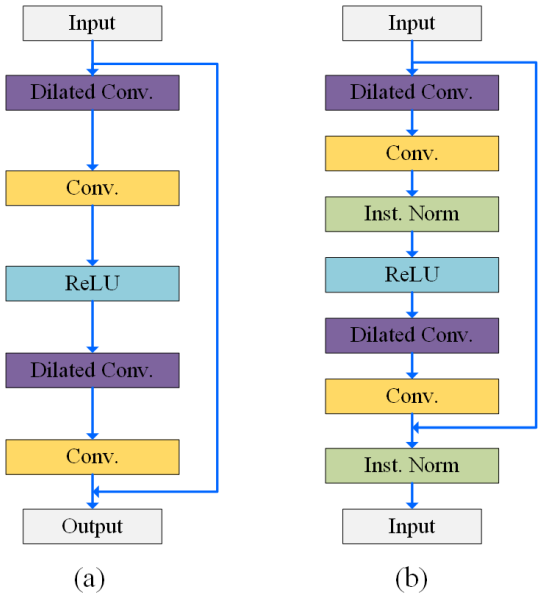



**Figure 3.6 Comparison of the used dilated convolution blocks and GCANet: (a) dilated convolution blocks of GCANet, and (b) modified dilated convolution blocks of proposed DC-Deblur network.**

Following the similar network design principles in [23-25], the overall network is designed as a simple encoder-decoder structure, as shown in Fig. 3.7, where $I_{in}$ denotes the input blurry image, and $I_{out}$ is the output sharp image. DenseBlocks [38] are adopted in the proposed network, and the network used the modified ResBlocks [23]. These ResBlocks consist of two convolution layers and a ReLU layer. Note that the residual links are implemented by the densely connected links when using ResBlocks in DenseBlock, as shown in Fig. 3.7. Each DenseBlock consists of five ResBlocks. The

proposed network uses DenseBlocks with 64 channels, two DenseBlocks in the encoder, two others in the decoder. DC-Deblur uses a limited number of IN layers out of dilated blocks to make the training stable. Dilated blocks are used after the first downsample. By the using of dilated convolution structure, the second downsample is avoided, which means that DC-Deblur uses only one downsample and one upsample in the deblurring network. In contrast, previous work such as DeepDeblur [23], SRN-Deblur [24], PSSNSC [25] all used two or more subsamples.

In this paper, a gated fusion structure is further adopted inspired by [55] via incorporation of an extra gated fusion sub-network $G$, as shown in Fig. 3.7. DC-Deblur first extracts the feature maps from different levels, denoting them as $F_1$, $F_2$, and $F_3$, which are typically 4 dimensional [107] tensor and feed them into the gated fusion sub-network. The output of the gated fusion sub-network are three different importance weights $M_1$, $M_2$, and $M_3$, which are also 4 dimensional tensor, corresponding to each feature level. Finally, these three feature maps $F_1$, $F_2$, and $F_3$ from different levels are linearly combined with the regressed importance weights,

$$(M_1, M_2, M_3) = G(F_1, F_2, F_3) \tag{3.6}$$

$$F_{out} = M_1 \times F_1 + M_2 \times F_2 + M_3 \times F_3 \tag{3.7}$$

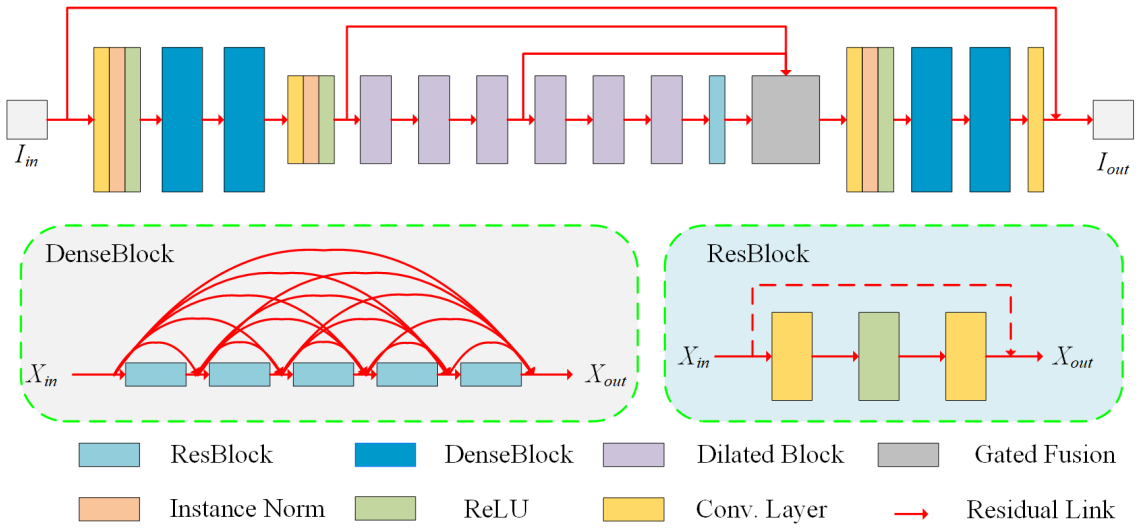



**Figure 3.7 Overall network structure of the proposed network, following an auto-encoder structure. It consists of two convolution blocks and two DenseBlocks as the**

**encoder part, and one deconvolution block and DenseBlocks as the decoder part. Several smoothed dilated resblocks are inserted between them to aggregate context information without gridding artifacts. To fuse the features from different levels, an extra gate fusion sub-network is leveraged. The proposed network will predict the residues between target sharp image and blurry input image in an end-to-end way.**

### 3.4.2 Loss Function

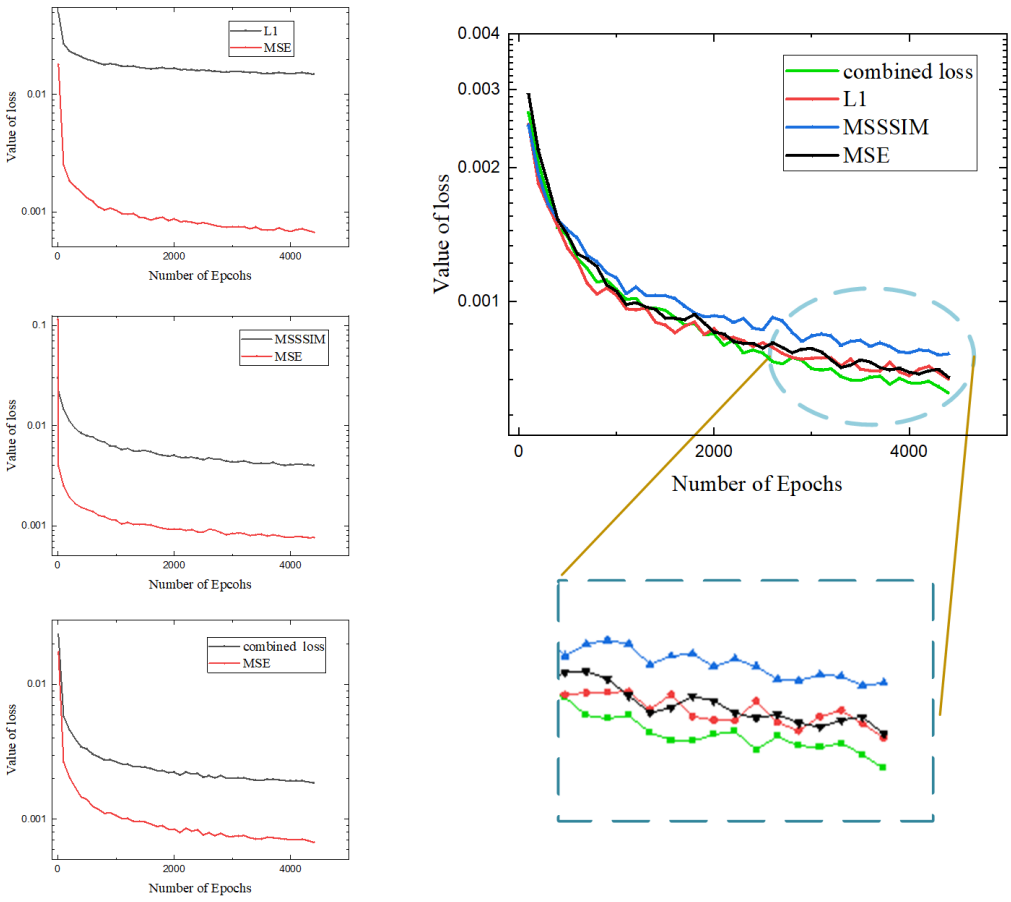
Loss function is another key part in image restoration and deblurring. In [24], $l_2$ loss was used, which also known as the MSE loss. MSE loss has close relationship with peak signal to noise ratio (PSNR), an important image quality assessment (IQA) index.

Another important loss function is the $l_1$ loss. In [100], the authors explored loss functions in image restoration, especially in denoising, super-resolution and JPEG artifacts removal by proposing a mixed loss by combining $l_1$ loss and Multi-Scale Structural SIMilarity (MS-SSIM) loss. MS-SSIM is based on Structural SIMilarity (SSIM), which is a well-known criterion and brings IQA from pixel-based to structure-based.

In the experiments, MSE loss gave similar or even better performances compared with the $l_1$ loss in image deblurring. This might be because that blur kernels are larger and complex. Thus, the Euclidean distance between a blurry image and sharp image is greater than that of a low-resolution and high-resolution image. In this work, DC-Deblur further combine $l_1$ loss, MSE loss and MS-SSIM loss. The loss function of the proposed network is denoted by

$$\mathcal{L}^{Mix} = \omega \times \mathcal{L}^{ms-ssim} + \gamma \times \mathcal{L}^{1} + \lambda \times \mathcal{L}^{mse} \tag{3.8}$$

$l_1$ loss, MSE loss and MS-SSIM loss have different numerical factors. For instance, the value of $l_1$ loss is often 10 times bigger than that of MSE loss. This is negative for the convergence of the network. To balance these parts in the loss, DC-Deblur set $\omega = 0.1$, $\gamma = 0.05$ and $\lambda = 1$.




## 3.5 Experiments and Results

Experiments on the proposed MixNet and DC-Deblur are conducted and compared with the state-of-the-art methods on dynamic scene deblurring and non-uniform deblurring.

The methods compared include DeepDeblur [23], Scale Recurrent Network (SRN-Deblur) [49], DSHMN [27], DeblurGAN [28], DeblurGANv2 [29], Unsupervised deblur [90], Domain spacific [101], SVRNN [26], Dual Residual [102], Douglas-Rachford network [91], Region Adaptive [69], Blur2Flow [103], Extreme Channels Prior [104], and Phase-only Kernel Estimation [105]. For kernel-based methods including Blur2Flow [103] and optimization-based methods, the experiments tested them on released model codes.

### 3.5.1 Implementation

The proposed MixNet is implemented by TensorFlow [106] and the proposed DC-Deblur is implemented by PyTorch [107], both on a NVIDIA Tesla P100 GPU. During training, a $256 \times 256$ region from a blurred and its ground truth image at the same location were randomly cropped out as the training input. The batch size was set to 16 during training. All weights were initialized using the Xavier method [108], and biases were initialized to zero. The network was optimized using the Adam method [109] with default setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate was initially set to 0.0001 and exponentially decayed to 0 using power of 0.3. According to the experiments, 2000 epochs were sufficient for the network to converge.

### 3.5.2 Datasets

Datasets are also especially important to learning-base deblurring approach. As reported in [25], a poor dataset would have negative influence on training, while a good dataset could improve quality of the test result and provide better evaluations. In this chapter, a brief review is given on a few main datasets for learning-based image deblurring. In addition, a better method to generate blurred images has been given in [110].

*GoPro Dataset:* GoPro dataset [23] is the most used end-to-end deblurring dataset by far. It was synthesized by averaging consecutive frames from a high-speed video and contains 2,103 pairs of images for training and 1,111 pairs for evaluation. However, as the first kernel-free dataset, many issues are not fully considered. First, many sharp images also have certain degrees of blur, due to the image acquisition process. Second, in some pictures, a large portion of the pixels are occupied by asphalt road or sky, resulting in a large solid colour area in the pictures. When they are cropped into training images, it can lead to deviation of the loss function.

*New GoPro Dataset:* In [25], the authors pointed out that there exist flaws in some of the ground truth or sharp images in the GoPro training set, as aforementioned. To improve the training performance, the authors established a new dataset following the procedure of Nah *et al.* [23] using GoPro Hero6 and iPhone7 at 240 fps and collected 5,290 blurred/sharp image pairs. Here, this thesis denotes it as the *New GoPro* Dataset. Due to more reasonable image acquisition and clearer original images, this dataset can improve training performance. In this chapter, the experiments also give the effect and a comparison on adding the New GoPro Dataset into training.

*REDS Dataset:* Authors of [111] provided the REalistic and Dynamic Scenes (REDS) dataset for video deblurring and super-resolution. The authors manually recorded 300 RGB video clips, paying attention to the quality of each frame, diversity of source contents (scenes and locations) and dynamics of various motions. They used the GoPro HERO6 Black camera to record videos of 1080x1920 resolution, at 120 fps. In contrast to the previous datasets for deblurring that captured videos in higher frame rate (240 fps). This is a larger and clearer dataset.

*HIDE Dataset:* In [112] the authors pointed out the GoPro dataset is mainly concerned with wide-range scenes, ignoring significant moving objects in background, especially in close-up shots. To fully capture the dynamic blurs caused by the passive device interference and initiative actions, HIDE dataset was proposed to cover both wide-range and close-range scenes to address human-aware motion deblurring. The HIDE

dataset is divided into two parts: HIDE I consists of 1304 long-shot pictures, while HIDE II 7118 close-ups pictures. The training set combines HIDE I and HIDE II, leading to 6397 for training, 1063 of HIDE I and 962 of HIDE II for testing. HIDE dataset has a large number of human faces and complex objects, including high-speed moving vehicles and strong light interference. Thus, it is better in evaluation and more useful in real-world applications. In this chapter, the experiments also provide a comparison on HIDE Test in section 3.5.3.

### 3.5.3 Results and Ablation Study

This section conducted experiments on the proposed MixNet and DC-Deblur and compared with the state-of-the-art methods on dynamic scene deblurring and non-uniform deblurring on the GoPro dataset. The methods compared include DeepDeblur [23], SRN-Deblur [24], DSHMN [27], DeblurGAN [28], DeblurGANv2 [29], Unsupervised deblur [90], Disentangled Representation [127], SVRNN [26], Dual Residual [133], Dr-Net [134], RADN-Deblur [101], Blur2Flow [103], Extreme Channels Prior [104], and Phase-only Kernel Estimation [105]. Quantitative results and evaluations are presented in Table 3.1, and a visual comparison of MixNet is shown in Fig. 3.9. The results were generated by the models trained only on the default GoPro training dataset and then tested on the GoPro test dataset. For unsupervised methods, Unsupervised deblur and Disentangled Representation, blur images from GoPro training dataset, and sharp images from New GoPro dataset, which have higher resolution, is used. For kernel-based methods including Blur2Flow and optimization-based methods, the experiments tested them on released model codes. Special convolution modules could result in better performance, e.g., Dr-Net and RADN-Deblur methods have good performance both in PSNR and run time. However, more specific the model is, the better performance it may produce for a specific task, its generalization ability would sacrifice.

It is clear that the proposed MixNet outperformed others in most of the benchmarks, including PSNR, SSIM, and FSIM. The proposed DC-Deblur also have good performance. In addition, the proposed method was also better in speed of forward processing and had smaller model size. By using dilated convolutions, the network size is reduced for about 60 % compared with the state-of-the-art methods. Note that SRN-Deblur and PSSNSC used parameter sharing when training. For instance, PSSNSC [25] shared the parameters of each DenseBlocks six times, thus the number parameters involved in processing was much larger. Visual results are shown in Fig. 3.13. the proposed approach led to sharper restoration.


**Figure 3.9: Testing results of the proposed MixNet for image deblurring on GoPro dataset.**

This section also evaluated and compared the proposed method on the HIDE dataset and quantitative results are shown in Table 3.2. These results were generated by the models trained on the default HIDE training dataset. The visual comparison of MixNet is shown in Fig. 3.9, and that of DC-Deblur is in Fig. 3.13. As illustrated, the proposed MixNet either outperformed or matched these state-of-the-art methods in all these evaluation criteria. Visual comparisons on the HIDE evaluation dataset are also shown in

Figure. 3.9. For a fair and comprehensive comparison, comparison is based on all the state-of-the-art learning-based methods, including top performers, SRN-Deblur [24], DSHMN [27], and PSS-NSC [25]. As shown in the figure, the propose MixNet has markedly improved performance over these methods, especially in the reconstruction of human faces and features. A comparison of MixNet on RWBI dataset and our own data is also given, which is shown in Fig. 3.11 and Fig. 3.12.

The proposed DC-Deblur also has good performance, as shown in Fig. 3.14 For the windows which have a far distance in the pictures, the proposed network can produce sharper edges and one can also recover better human faces with DC-Deblur.


**Figure 3.10: Testing results of the proposed MixNet for image deblurring on HIDE dataset.**

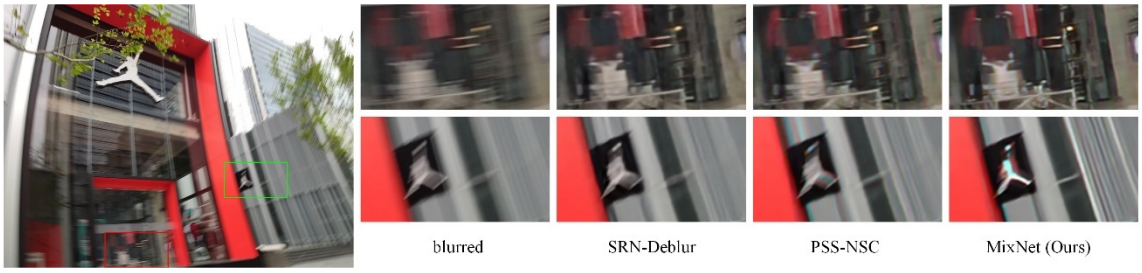



**Figure 3.11: Testing results of the proposed MixNet for image deblurring on RWBI dataset.**

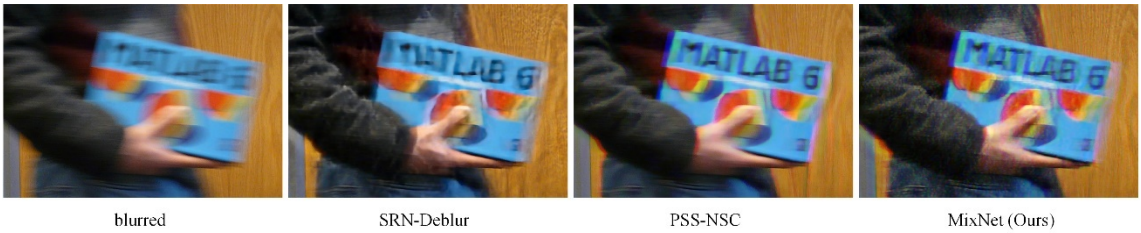



**Figure 3.12: Testing results of the proposed MixNet for image deblurring on a real blur image.**

**Table 3.1 Test results of the proposed methods on GoPro dataset.**

| Algorithm | PSNR | SSIM | FSIM | IFC | VIF | RunTime(s) |
|---|---|---|---|---|---|---|
| DeepDeblur [23] | 29.08 | 0.9135 | 0.9633 | 3.2916 | 0.9955 | 3.09 |
| SRN [24] | 30.26 | 0.9432 | 0.9653 | 3.0750 | 0.9878 | 0.86 |
| PSS-NSC [25] | 30.92 | 0.9421 | 0.9756 | 3.4136 | 0.9914 | 0.68 |
| DeblurGAN [28] | 28.70 | 0.9580 | 0.9688 | 3.1124 | 0.9732 | 0.85 |
| DeblurGAN-v2 [29] | 29.55 | 0.9340 | 0.9527 | 2.6929 | 0.9697 | <u>0.35</u> |
| SVRNN [26] | 29.19 | 0.9306 | 0.9446 | 2.8934 | 0.9822 | 1.40 |
| Unsupervised deblur [90] | 18.64 | 0.6219 | 0.8220 | 1.0233 | 0.6401 | 1.23 |
| Domain specific [101] | 20.15 | 0.6623 | 0.8317 | 1.1790 | 0.6788 | 0.96 |
| Dual Residual [100] | 29.90 | 0.9100 | 0.9436 | 2.6387 | 0.9948 | 1.14 |
| Dr-Net [91] | 30.35 | **0.9610** | <u>0.9812</u> | 3.4540 | 0.9964 | 1.20 |
| DSHMN [27] | 31.50 | 0.9483 | 0.9742 | 3.2697 | 0.9915 | 0.552 |
| RADN-Deblur [69] | <u>31.76</u> | 0.9530 | 0.9804 | 3.4298 | 0.9965 | 0.038 |
| Blur2Flow [103] | 26.99 | 0.9233 | 0.9562 | 3.1432 | 0.9842 | 1.70 |
| Extreme Channels Prior [113] | 23.70 | 0.8273 | 0.9034 | 1.7055 | 0.9388 | 271.68 |
| MixNet | **32.21** | <u>0.9534</u> | **0.9827** | **3.6804** | **0.9987** | 0.54 |
| DC-Deblur | 31.27 | 0.9448 | 0.9743 | <u>3.5414</u> | <u>0.9970</u> | **0.29** |

**Table 3.2 Test results of the proposed methods on HIDE dataset.**

| Algorithm | HIDE I (long-shot) | | | HIDE II (close-ups) | | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | FSIM | PSNR | SSIM | FSIM |
| DeepDeblur [23] | 27.43 | 0.9020 | 0.9622 | 26.18 | 0.8780 | 0.9339 |
| SRN [24] | 29.41 | 0.9137 | 0.9681 | 27.54 | 0.9070 | 0.9393 |
| PSS-NSC [25] | 29.98 | 0.9234 | 0.9728 | 28.14 | 0.9021 | 0.9470 |
| DeblurGAN [28] | 26.44 | 0.8900 | 0.9440 | 25.37 | 0.8670 | 0.9244 |
| SVRNN [26] | 28.69 | 0.9038 | 0.9614 | 26.68 | 0.8702 | 0.9231 |
| DSHMN [27] | 29.79 | 0.9247 | 0.9736 | 28.33 | 0.9099 | 0.9501 |
| MixNet | **30.72** | **0.9277** | **0.9745** | 28.72 | **0.9137** | 0.9517 |
| DC-Deblur | 30.26 | 0.9297 | 0.9801 | **28.86** | 0.9133 | **0.9545** |

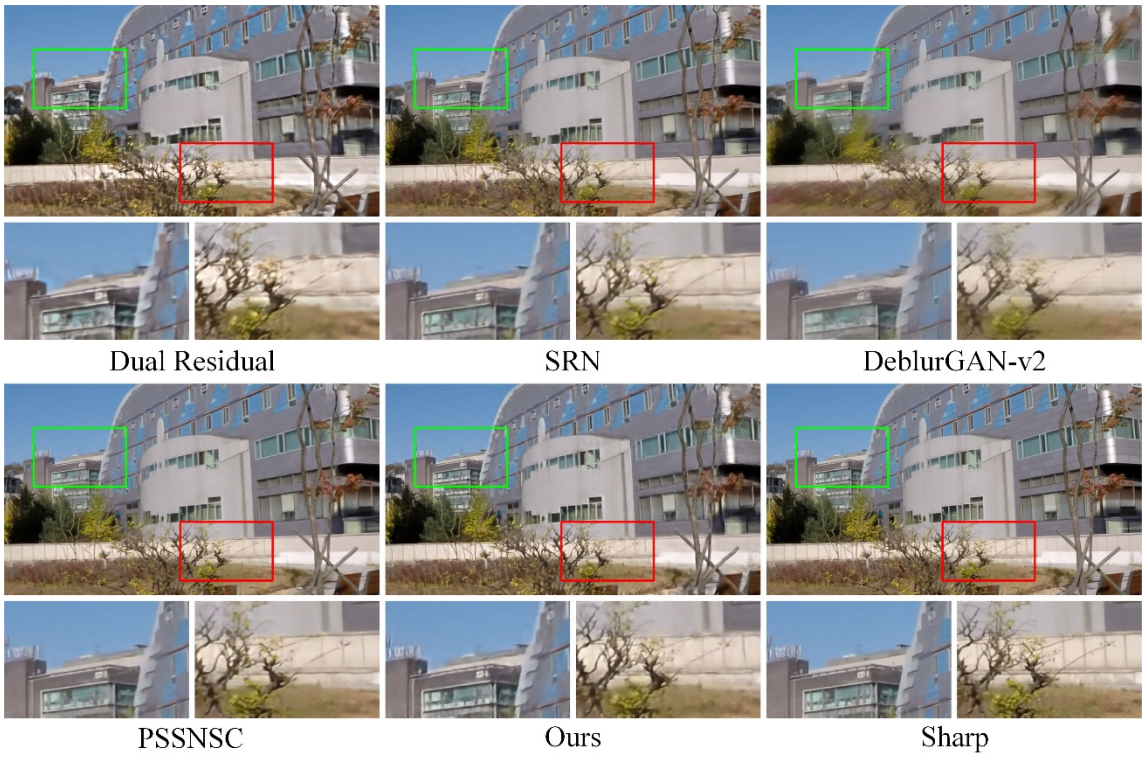



**Figure 3.13 Visual comparison of DC-Deblur on GoPro dataset, with Dual Residual Network [102], SRN [24], DeblurGAN-v2 [28], PSSNSC [25].**

**Table 3.3 Testing results of ablation studies of DC-Deblur on GoPro dataset. The adopted network with dilated convolution and combined loss performs the best.**

| Method | PSNR | SSIM | Size (MB) |
|---|---|---|---|
| DenseNet Deblur | 30.26 | 0.9345 | 27.0 |
| Ours with Norm | 31.05 | 0.9403 | 10.1 |
| Ours with MSE loss | 31.14 | 0.9433 | 10.1 |
| Ours with $l_1$ loss | 31.09 | 0.9426 | 10.1 |
| Ours with MS-SSIM loss | 30.65 | 0.9388 | 10.1 |
| Ours (adopted) | **31.21** | **0.9448** | **10.1** |


**Figure 3.14 Visual comparison of DC-Deblur on HIDE dataset, with SRN [24], DeblurGAN-v2 [29], PSSNSC [25].**

In this section, further experiments on different settings of DC-Deblur is given to study its effectiveness and efficiency. Generally, the ablation study focused on network framework and loss functions.

To measure the ability of the proposed model, two more experiments are done on the network structure. Firstly, the network performance without dilated convolution is tested. However, directly removing the dilated blocks would lead to performance deterioration obviously. DenseBlocks are used to replace the dilated convolutions. Specially, 6 DenseBlocks are used in the encoder, and 6 DenseBlocks for the decoder. This network was similar to PSSNSC deblurring network to some degrees, while parameter sharing and multi-scale structure are not used, and all of the DenseBlocks were 64-channel. The results are shown in Table 3.3. The proposed method has better performance than DenseNet Deblur, even though DenseNet Deblur had a much larger size. This shows that the proposed method can achieve better performance and reduce the complexity of the model. Secondly, network is tested with normalization layers. To this end, the dilated convolution structure in Fig. 3.6 (b) is used, which was also the dilated convolution block used in GCANet [55]. The results are shown in Table 3.3. The network performed better when normalization layers were removed. In addition, deblurring networks based on dilated convolution with normalization layers were also better than DenseNet Deblur, indicating that the adoption of dilated structure itself improved the model. In conclusion, dilated convolution is an effective and efficient approach for image deblurring and removing instance normalization is an effective modification toward dilated convolution blocks in deblurring.

Loss function was another task that should be further examined. In the previous section, it is pointed out that multi loss function can better help the training of network rather than using single one. This section also gives further comparisons of different loss functions. The experiments on loss function were based on the network structure in Fig. 3.7. First of all, MSE loss is used only, and the results are shown in Table 3.3, denoted

as *Ours with MSE loss*. DC-Deblur model based on $l_1$ loss only and MS-SSIM loss only are also tested. Their results are also shown in Table 3.3, denoted as *Ours with* $l_1$ *loss* and *Ours with MS-SSIM loss*, respectively. The training process is shown in Fig. 3.8. Based on these experiments, the MS-SSIM only training had the worst performance, which conforms with the result in [100]. The network trained by $l_1$ loss only was slightly worse than that of MSE loss only, this is different with the result of [100], mainly because of the difference between image deblurring and super-resolution. And it is obviously that the proposed method with combined loss had the best performance. In Fig. 3.8, the speed of convergence of $l_1$ loss was slightly faster, but the final result was nearly the same as that of MSE loss. And the curve of MS-SSIM loss was obviously higher than others, especially after 500 epochs. However, based on the experiments, the loss function combined with MSE, $l_1$ and MS-SSIM gave better overall ability.

## 3.6 Conclusions

This chapter have introduced a new learning-based model, termed as MixNet by integrating the DenseNet and Inception-v4 mechanisms. The MixNet uses Inception-A to increase the channels of feature maps and receptive fields while maintaining reasonable complexity, so that the network is simple and easy to implement. Experiments on various benchmark datasets show the advantage of the proposed approach in dealing with single image deblurring. Performances of the proposed MixNet markedly exceed that of the state-of-the-art methods in a range of image quality measures.

In this chapter, a novel image deblurring network termed as DC-Deblur is proposed. The network is based on a dilated convolution structure, which is able to increase the reception field without incurring additional parameters. The requirement of large reception field is unavoidable for image deblurring due to the ill-posed nature of deblurring and possible large blur kernels. DC-Deblur can reduce the number of parameters, making the network efficient. This work further improves the network

performance by combining $l_1$ loss, MS-SSIM loss and MSE loss. Experiments and quantitative and qualitative evaluations indicate that the proposed method outperforms the state-of-the-art models, in both performance and speed.

# Chapter 4

# 4 Graph Neural Networks on Image Restoration

## 4.1 Introduction

Graph convolutional networks (GCNs) have recently been shown outstanding ability in dealing with data of non-Euclidean structures, such as point clouds and graphs. Recently they have gained much increased attention in the signal/image processing and machine learning communities. Many existing applications of GCNs have focused on graph data or data exhibiting graph structures, such as social networks [115,116], physical systems [117], and knowledge graphs [118].

The main advantage of graph networks is to express dissemination among information and interaction of data; hence GCNs are powerful tools to represent the intra-relationship of input data. In [119], GCN was used in classification to describe relationships between multiple labels. In this case, the original data was not graph-structured (i.e., images in the Euclidean space), while high-level features were abstracted and further processed by knowledge graphs. Similar work can be also seen in [120]. GCN

in image classification is often applied on the result of encoder, as indicated in Fig. 4.1 (a), and the existing research suggests that GCN works well on these encoded data. Such applications are based on high-level semantics. By comparing network structures of classification and image restoration, it can be reasonably argued that semantic relationships should also exist in low-level features, for instance, intermediate feature maps in the convolutional neural networks (CNNs). This chapter, explores the use of GCNs in an encoder-decoder structure, as illustrated in Fig. 4.1 (b).

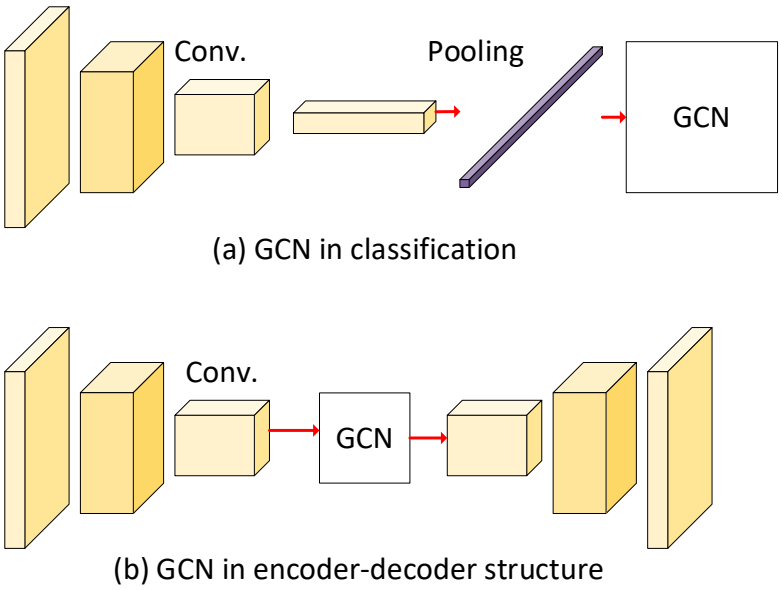



**Figure 4.1: Comparison of GCN in classification and the encoder-decoder structure.**

In the forward process of a CNN, features produced by the convolutions may be used to infer their topological relationships. Some features may be more important and subsequent layers may depend more on these key features. These topological relationships can be described by graph networks. Therefore, a GCN-based encoder-decoder network is proposed to exploit such relationships among features. For efficiency, this work first produces an artificially constructed graph structure and then fit features into the graph, followed by their corresponding weight updating. By doing this, the features extracted contain certain structural relationships useful to many image restoration tasks such as

deblurring and super-resolution. To our best knowledge, there is no similar approach before. Such use of GCNs also broadens the application of GCNs.

The proposed network adds graph convolutions by converting feature maps to vertexes of a pre-generated graph to extract topological structures of the features. By doing this, this work inexplicitly applies graph Laplacian regularization to the feature maps, making them more structured. Furthermore, residual learning is used to moderately deepen the graph network in order to increase performance. The experiments show that GCN in the feature space can significantly improve the performance in the task of image restoration (we use image deblurring and super-resolution as examples).

## 4.2 Related Work

### 4.2.1 Graph Convolutional Networks

From the perspective of aggregator, GCNs can be divided into spectral-based and spatial-based. Authors in [121] firstly developed graph convolutions based on spectral graph theory using the Fourier basis of a given graph in the spectral domain. Many extensions subsequently apply extensions, improvements and approximations on spectral-based GCNs [116,122]. Spatial-based GCNs [116,123] directly define graph convolution operations on the graph by operating on spatially close neighbours.

In the recent rapid and fruitful development of GCNs, most methods employed shallow GCNs. Some attempted different ways of training deeper GCNs [115,124]. However, these networks are limited to 10 layers in depth before performance degrades. Inspired by the benefit of training deep CNN-based networks [25], DeepGCNs [125] proposed to train a very deep GCN (56 layers) by adapting residual/dense connections (ResGCN/DenseGCN) to GCN.

### 4.2.2 Graph Networks on computer vision

GCN has been wildly used in computer vision. Yan *et al.* [126] use GCN on human body skeletons by automatically learning both the spatial and temporal patterns. Yang *et al.* [127] proposed Graph R-CNN with a relation proposal Network and an attentional Graph Convolutional Network. Johnson *et al.* [128] generating images from scene graphs, enabling explicitly reasoning about objects and their relationships. Wang *et al.* [129] developed EdgeConv which acts on graphs dynamically computed in each layer of the network. Qi *et al.* [130] used graph network in detecting and recognizing human-object interactions. Guo *et al.* [131] focused on fewshot 3D Action Recognition by leveraging the inherent structure of 3D data through a graphical representation. Narasimhan *et al.* [132] develop an entity graph and use a graph convolutional network to 'reason' about the correct answer by jointly considering all entities for factual visual question answering. However, GCN in image restoration has not been studied.

## 4.3 Proposed GCN Based Deblurring

### 4.3.1 Overall Scheme

The overall scheme is shown in Fig. 4.3. Since graph convolutions are used on features of a CNN, one of the difficulties is the conversion between feature maps and graph nodes, i.e., basic units of a graph. Usually, the implementation of graph convolution is based on adjacency matrix and degree matrix of a given graph, thus the data structure in training a deblurring or super-resolution network need to fit in those matrix operations. In addition, using too many filters to convolve can lead to much increased dimensions and computational complexity. For instance, using 3 filters to convolve a 2D grey-scale image can produce a 3-dimensional matrix. To deal with these problems, a new method is proposed, given in Algorithm 1. Consider an input $X$, by converting dimension $C$ to the graph, yielding a new dimension $F$, the data can appear in $C \times F$ structure with

each element a feature map for left multiplication, and is the same as the mathematical form given in [116].

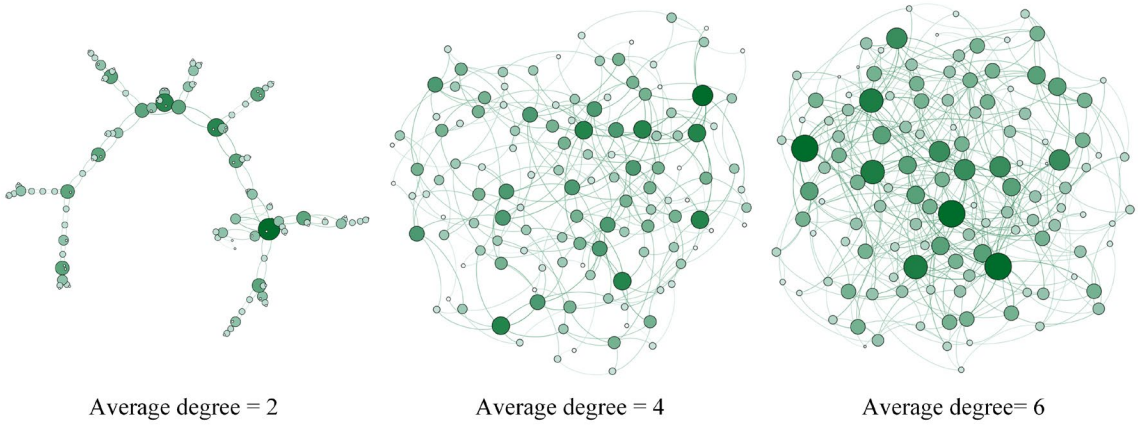



**Figure 4.2: The adopted graph with different mean node degrees, which are 2, 4, and 6 respectively.**

### 4.3.2 Graph Structure

Graph structure is important in GCNs. However, it can be found that using any suitable structure from natural laws and adding graph convolutions can help improve the CNN performance in image restoration. Generating the graph structures at each iteration during training can be extremely computationally intensive and may not produce stable results. This work proposed to use a pre-generated graph structure throughout to alleviate the burden of generating graphs dynamically. With a pre-generated graph, transforming conventional convolutions to graph convolutions incurs little extra cost. Here, the Watts–Strogatz (WS) model [133], which is a random generated graph that has small-world network properties such as clustering and short average path length, is used. For a small-world graph, the average minimum path length is usually small and produces some hub nodes, to reflect importance of the features. The proposed method generates the graph based on number of features. For instance, for a 96-channel feature map, the graph has 96 nodes. Although the proposed method randomly creates the graph for training, different

random graphs would not result in marked differences [134] in graph properties, which is mainly determined by degree centrality [135]. which is defined as the number of links upon a node.

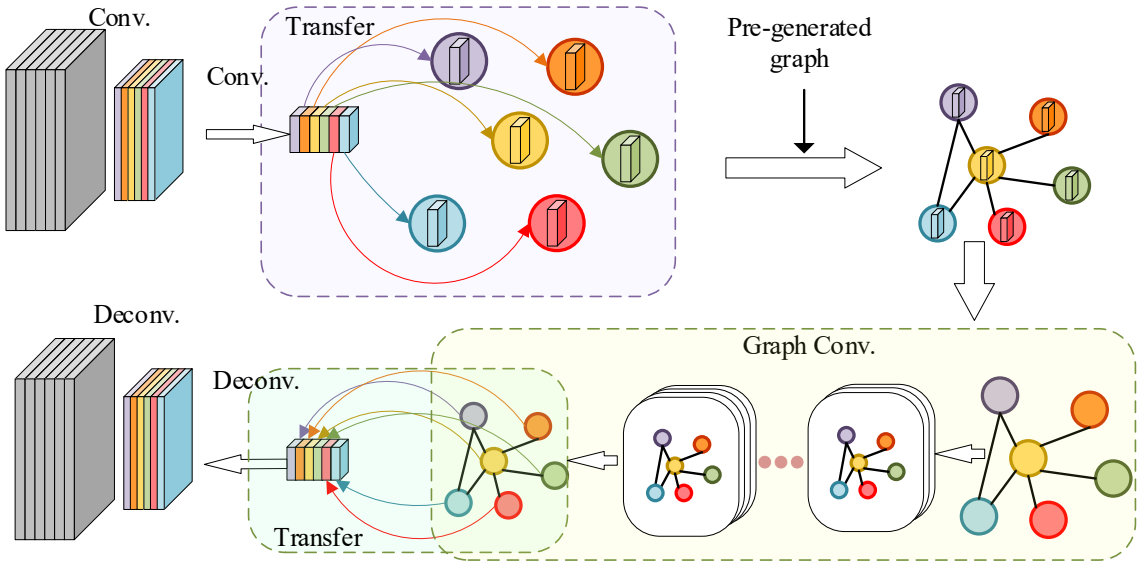



**Figure 4.3: Proposed method converts feature maps from CNN into a graph, treating each channel as an identity or node and connecting them by a pre-defined adjacency matrix. The output layer of GCN is converted back to feature maps.**

Specifically, the core properties of a graph are determined by the degree of the graph. The average degree of a set of random graphs of same number of nodes becomes stable when the number increases. There are some nodes that are more significant than others, thus the weights of CNN will adapt to the graph structure when training with a new graph. The order of nodes does not affect much the network performance. Exemplar graphs of different average degrees are illustrated in Fig. 4.2.

### 4.3.3 Aggregator and Updator

For GCN, propagation contains aggregators to obtain hidden states of nodes. Various GCNs utilise different aggregators to gather information from each node's neighbours and

specific updaters to update nodes' weights. Kipf *et al.* [116] developed an aggregator for spectral GCNs. Consider an undirected graph $\mathcal{G}$, the aggregator is given as

$$T = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}X \tag{4.1}$$

Where $\widetilde{\boldsymbol{A}} = \boldsymbol{A} + \boldsymbol{I_N}$ denotes the adjacency matrix of the graph $\mathcal{G}$ with added self-connections produced by identity matrix $I_N$. $N$ is the number of nodes. $T$ is the aggregator. Based on the renormalisation trick proposed in [116], $\boldsymbol{I_N} + \boldsymbol{D}^{-\frac{1}{2}}\boldsymbol{A}\boldsymbol{D}^{-\frac{1}{2}} \rightarrow \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$, where $D$ is the degree matrix. From [116,136]

$$X^{l+1} = T^{l}\boldsymbol{\Theta}^{l} \tag{4.2}$$

where $\boldsymbol{\Theta}^{l} \in \mathbb{R}^{C \times F}$ is a matrix of graph convolution (GC) filter parameters of $C$ input channels and $F$ filters in the $l^{th}$ GC layer, $T^{l}$ is the aggregator of the $l^{th}$ layer, $X^{l+1}$ is the convolved output after the $l^{th}$ layer. Thus,

$$GraphConv(X) = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}X\boldsymbol{\Theta} \tag{4.3}$$

as a representation of the graph convolution used in the proposed network.

To further improve the performance, also consider ResGCN [125] to make the network deeper. As being analysed in [125,137], deepening the network is useful. It is reasonable that such conclusion can be also applied in GCNs. Compared with [125], the proposed network faces to a fresh challenge with pre-generated graph, and the aggregator will be different from the original ResGCN. Thus, the proposed method removed normalization and limited the number of ResGCN blocks within 10 to avoid computational complexity. Based on Eq. 4.2, the ResGCN block used in this paper can be given as

$$X^{out} = GraphConv\big(\alpha\,(GraphConv(X^{in}))\big) \tag{4.4}$$

where $\alpha(\cdot)$ denotes the activation function, e.g., ReLU. In this chapter, the influence of different number of ResGCN blocks is compared and analysed in the experiments.

**Algorithm 1** Conversion between feature maps and nodes

**Input** $X \in \mathbb{R}^{B_s \times C \times W \times H}$: feature map, where $B_s$ is batch size, $W$ and $H$ is width and height of input feature, $C$ is the number of features;
**Output** $X^{out} \in \mathbb{R}^{B_s \times C \times W \times H}$: the feature map after graph convolution.
Convert $X \in \mathbb{R}^{B_s \times C \times W \times H}$ to $X^* \in \mathbb{R}^{B_s \times H \times W \times C}$;
**for** all data in an epoch **do**
  Convert $X$ to $X^* \in \mathbb{R}^{B_s \times H \times W \times C \times F}$, added $F$ as graph feature dimension;
  **for** numbers of GC layers **do**
    $X^* \leftarrow GraphConv(X^*)$;
  **end for**
  Convert $X^*$ to $X \in \mathbb{R}^{B_s \times H \times W \times C}$, with graph feature dimension $F$ removed;
**end for**
Convert $X \in \mathbb{R}^{B_s \times H \times W \times C}$ to $X \in \mathbb{R}^{B_s \times C \times W \times H}$;

**Algorithm 1: Conversion between feature maps and nodes.**

## 4.4 Experiments and Results

Graph convolutions are adopted in the residual blocks (ResBlocks) in the proposed network for deblurring and term it as Graph Convolution ResNet (GCResNet), which is shown in Fig.4.4. Normalization layers are removed based on the analysis of [24]. The network is based on an encoder-decoder structure with residual link from the import of the network to the last convolution layer. 18 ResBlocks are used in encoder and 18 ResBlocks in decoder. Graph convolution layers are adopted between encoder and decoder. MSE loss is used, as it is the most suitable loss function and widely used in image deblurring. We also do experiments on super-resolution.

### 4.4.1 Implementation

Implementation of the proposed method consists of production of a graph network and training the network. We used MATLAB [138] to produce the WS graph with $\rho = 0.9$ (slightly different values would not result in significant differences).

We implemented the proposed network by PyTorch [107] on a NVIDIA Tesla P100 GPU. During training, we randomly cropped a $256 \times 256$ region from a blurred image and used it and its ground truth image at the same location as the training input. The batch size was set to 12. All weights were initialized by the Xavier method [108], and biases were initialized to zero. The network was optimized by using the Adam [109] with default setting $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. The learning rate was initially set to 0.0001 and linearly decayed to 0.

### 4.4.2 Datasets

Deblurring datasets which have been introduced in chapter 3 is used in these experiments. In addition, super-resolution datasets are also used here.

*Set 5 / Set 14*: The Set5 dataset [139] is a dataset consisting of 5 images ("baby", "bird", "butterfly", "head", "woman") commonly used for testing performance of Image Super-Resolution models. The Set14 dataset [140] is a dataset consisting of 14 images commonly used for testing performance of Image Super-Resolution models.

*Urban100*: The Urban100 dataset [141] contains 100 images of urban scenes. It commonly used as a test set to evaluate the performance of super-resolution models.

*B100*: BSD [142] is a dataset used frequently for image denoising and super-resolution. Of the sub datasets, BSD100 is a classical image dataset having 100 test images proposed by Martin et al. The dataset is composed of a large variety of images ranging from natural images to object-specific such as plants, people, food etc. BSD100 is the testing set of the Berkeley segmentation dataset BSD300.

*DIV2K:* The DIV2K dataset [143] is divided into train data and test data. Train data starts from 800 high-definition high-resolution images corresponding low-resolution images and provide both high- and low-resolution images for 2, 3, and 4 downscaling factors. In validation data, 100 high-definition high resolution images are used for

generating low resolution corresponding images, the low res are provided from the beginning of the challenge and are meant for the participants to get online feedback from the validation server; the high-resolution images will be released when the final phase of the challenge starts. In test data, 100 diverse images are used to generate low resolution corresponding images; the participants will receive the low-resolution images when the final evaluation phase starts, and the results will be announced after the challenge is over and the winners are decided.

In this chapter, DIV2K dataset is used to train the super-resolution network.

### 4.4.3 Results and Ablation Study

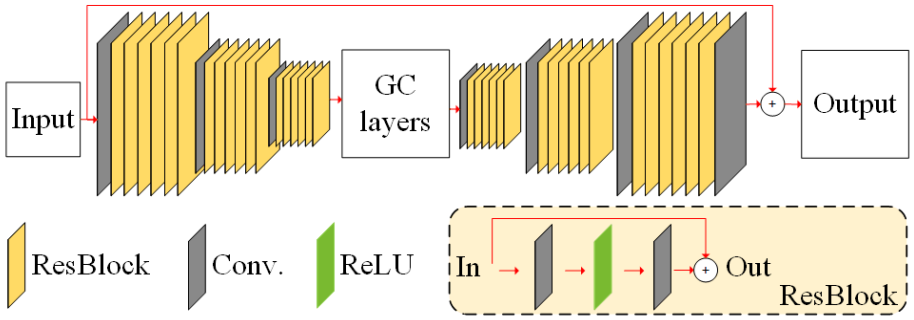



**Figure 4.4: Structure of proposed network for image deblurring. The red lines denote skip connections.**

This section evaluated the GCResNet on the GoPro dataset [23] and HIDE dataset [112]. Peak Signal-to-Noise Ratio (PSNR), Structural SIMilarity (SSIM) and Feature SIMilarity (FSIM) index are used for image quality assessment. The proposed network was compared with the mainstream methods: DeepDeblur [23], SRN-Deblur [24], PSS-NSC [25], DeblurGANv2 [29] and SVRNN [26]. Results are shown in Table. 4.1 and Table. 4.2. The proposed method has the best performance. A visual comparison is shown in Fig. 4.5, indicating that GCResNet has restored clearer and sharper details.


**Figure 4.5: Visual comparison of image deblurring on GoPro, with DeblurGAN-v2 [29], SRN [24], PSS-NSC [25], and DSHMN [27]. GCResNet produced clearer details especially on person's hair and stripes on the shutter door.**

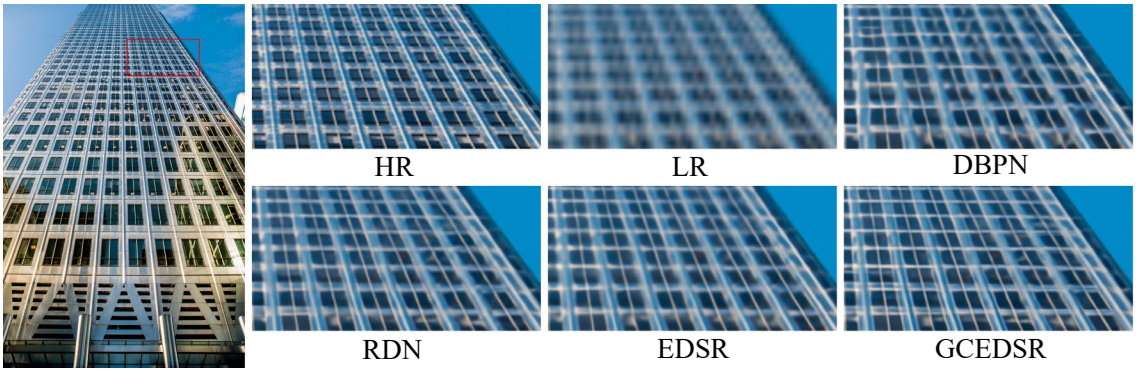



**Figure 4.6: Visual comparison of image super-resolution on Urban100, with RDN [144], DBPN [145], and EDSR [57].**

This section conducted an ablation study on the GCN structure, to show differences between different sets of GC layers. Results are shown in Fig. 4.7. When keeping the number of ResGCN blocks and enlarged the average degree, the performance would decrease for $degree > 4$. Note that the average degree cannot be odd number [133]. Such phenomenon might be due to too sparse or too dense connections, and the graph on $degree = 4$ showed the best topological properties. The performance of $degree = 2$ was slightly better than that of $degree = 4$. It can be implied that the performance would be close to the network without GCN when the average degree kept increasing, because too many connections would make GC meaningless on information. When keeping $degree = 4$ and increased the number of ResGCN blocks, the performance kept increasing while the growth rate decreased with too many blocks. The network

without ResGCN had the worst performance. In addition, performance could be improved by using more features, but that would also lead to heavier network in CNN. Consider the limit of memory, network with 128 features and degree of 4 showed the best-balanced performance. This work used 5 ResGCN blocks in the GCResNet.

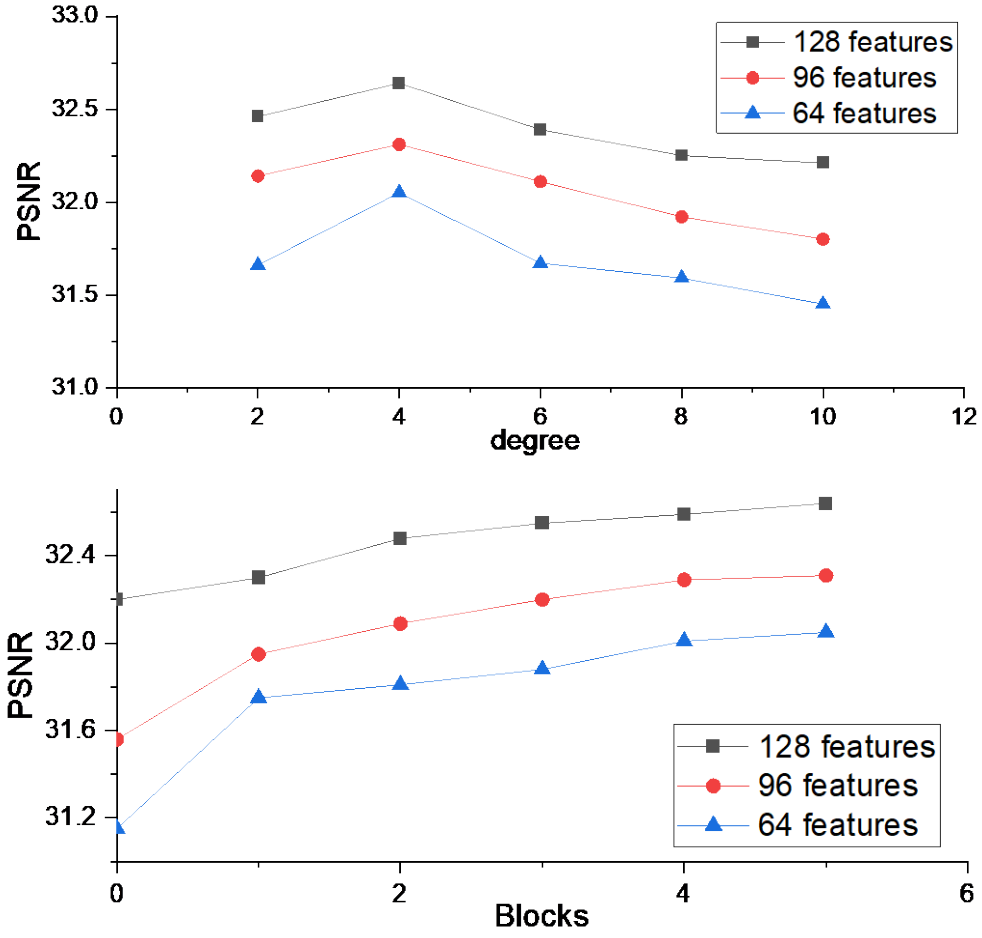



**Figure 4.7: Comparison of different network settings in image deblurring. Degree= 4 achieved the best results with other settings remained the same. Degree = 2was better than degree = 6, for degree = 6 was too dense so that the advantage of graph convolution could not be produced and hence the poor improvement. Deeper networks showed better performance, but too many residual blocks would not bring further improvement.**

**Table 4.1 Test results of GCResNet on GoPro dataset.**

| Algorithm | PSNR | SSIM | FSIM | IFC | VIF | RunTime(s) |
|---|---|---|---|---|---|---|
| DeepDeblur [23] | 29.08 | 0.9135 | 0.9633 | 3.2916 | 0.9955 | 3.09 |
| SRN [24] | 30.26 | 0.9432 | 0.9653 | 3.0750 | 0.9878 | 0.86 |
| PSS-NSC [25] | 30.92 | 0.9421 | 0.9756 | 3.4136 | 0.9914 | 0.68 |
| DeblurGAN [28] | 28.70 | 0.9580 | 0.9688 | 3.1124 | 0.9732 | 0.85 |
| SVRNN [26] | 29.19 | 0.9306 | 0.9446 | 2.8934 | 0.9822 | 1.40 |
| DSHMN [27] | 31.50 | 0.9483 | 0.9742 | 3.2697 | 0.9915 | 0.552 |
| GCResNet | **32.64** | **0.9580** | **0.9802** | **3.6804** | **0.9987** | **0.54** |

**Table 4.2 Test results of GCResNet on HIDE dataset.**

| Algorithm | HIDE I | | | HIDE II | | |
|---|---|---|---|---|---|---|
| | PSNR | SSIM | FSIM | PSNR | SSIM | FSIM |
| DeepDeblur [23] | 29.08 | 0.9135 | 0.9633 | 26.18 | 0.8780 | 0.9339 |
| SRN [24] | 30.26 | 0.9432 | 0.9653 | 27.54 | 0.9070 | 0.9393 |
| PSS-NSC [25] | 30.92 | 0.9421 | 0.9756 | 28.14 | 0.9021 | 0.9470 |
| DeblurGAN [28] | 28.70 | 0.9580 | 0.9688 | 25.37 | 0.8670 | 0.9244 |
| SVRNN [26] | 29.19 | 0.9306 | 0.9446 | 26.68 | 0.8702 | 0.9231 |
| DSHMN [27] | 31.50 | 0.9483 | 0.9742 | 28.33 | 0.9099 | 0.9501 |
| GCResNet | **30.04** | **0.9240** | **0.9827** | **28.62** | **0.9132** | **0.9827** |

**Table 4.3 Test results of GCEDSR on Super-resolution.**

| Algorithm | Scale | Set5 | Set14 | Urban100 | B100 |
|---|---|---|---|---|---|
| SRCNN [56] | × 2 | 36.66 / 0.9542 | 32.45 / 0.9067 | 29.50 / 0.8946 | 31.36 / 0.8879 |
| FSRCNN [146] | × 2 | 37.05 / 0.9560 | 32.66 / 0.9090 | 29.88 / 0.9020 | 31.53 / 0.8879 |
| VDSR [147] | × 2 | 37.53 / 0.9590 | 33.05 / 0.9130 | 30.77 / 0.9140 | 31.90 / 0.8960 |
| RDN [144] | × 2 | 38.24 / 0.9614 | 34.01 / 0.9212 | 32.89 / 0.9353 | 32.34 / 0.9017 |
| DBPN [145] | × 2 | 38.09 / 0.9600 | 33.85 / 0.9190 | 32.55 / 0.9324 | 32.27 / 0.9000 |
| EDSR [57] | × 2 | 38.11 / 0.9602 | 33.92 / 0.9195 | 32.93 / 0.9351 | 32.32 / 0.9013 |
| GCEDSR | × 2 | **38.29 / 0.9615** | **34.05 / 0.9213** | **33.12 / 0.9386** | **32.39 / 0.9023** |
| SRCNN [56] | × 4 | 30.48 / 0.8628 | 27.50 / 0.7513 | 24.52 / 0.7221 | 26.90 / 0.7101 |
| FSRCNN [146] | × 4 | 30.72 / 0.8660 | 27.61 / 0.7550 | 24.62 / 0.7280 | 26.98 / 0.7150 |
| VDSR [147] | × 4 | 31.35 / 0.8830 | 28.02 / 0.7680 | 25.18 / 0.7540 | 27.29 / 0.7260 |
| RDN [144] | × 4 | 32.47 / 0.8990 | 28.81 / 0.7871 | 26.61 / 0.8028 | 27.72 / 0.7419 |
| DBPN [145] | × 4 | 32.47 / 0.8980 | 28.82 / 0.7860 | 26.38 / 0.7946 | 27.72 / 0.7400 |
| EDSR [57] | × 4 | 32.46 / 0.8968 | 28.80 / 0.7876 | 26.64 / 0.8033 | 27.71 / 0.7420 |
| GCEDSR | × 4 | **32.61 / 0.9001** | **28.89 / 0.7885** | **26.72 / 0.8079** | **27.76 / 0.7439** |
| SRCNN [56] | × 8 | 25.33 / 0.6900 | 23.76 / 0.5910 | 21.29 / 0.5440 | 24.13 / 0.5660 |
| FSRCNN [146] | × 8 | 20.13 / 0.5520 | 19.75 / 0.4820 | 21.32 / 0.5380 | 24.21 / 0.5680 |
| VDSR [147] | × 8 | 25.93 / 0.7240 | 24.26 / 0.6140 | 21.70 / 0.5710 | 24.49 / 0.5830 |
| DBPN [145] | × 8 | 27.21 / 0.7840 | 25.13 / 0.6480 | 22.73 / 0.6312 | 24.88 / 0.6010 |
| EDSR [57] | × 8 | 26.96 / 0.7762 | 24.91 / 0.6420 | 22.51 / 0.6221 | 24.81 / 0.5985 |
| GCEDSR | × 8 | **27.39 / 0.7876** | **25.18 / 0.6503** | **23.14 / 0.6370** | **24.92 / 0.6027** |

The network for super-resolution consists of graph convolutions with the EDSR [35], thus it is named as Graph Convolution EDSR (GCEDSR). Graph convolution is adopted after 8 Resblocks, and then followed by another 8 Resblocks until the final SR image produced. 5 ResGCN blocks is used and $degree = 2$, based on the ablation study on deblurring. For a fair comparison, other settings in the EDSR are not changed, except for adding graph convolutions. All the convolution layers had 256 channels.

$l_1$ loss is used as the loss function, as it is the most suitable loss function and widely used in single image super-resolution [56,57].

Quantitative results are demonstrated in Table 4.3 and qualitative results in Fig. 4.6. One can observe from Fig. 4.6 that the proposed network outperforms the previous methods at $\times 4$ super-resolution levels. More results can be seen in Appendix A.

## 4.5 Conclusions

A new convolutional neural network for image deblurring and super-resolution is proposed by adapting graph network in CNNs. While existing graph neural networks are for image classification, the proposed network explore graph structures in the feature maps of CNNs for effective image restoration. Experiments demonstrate that such adaptation with a predefined graph structure can achieve improved performance in image restoration with little added computational costs. Exploring topological relationships among feature maps is beneficial to many images processing tasks.

# Chapter 5

# 5 Conclusions and Future Work

This research explores deep learning based image deconvolution, especially image deblurring, in order to provide efficient and sharp restoration of real-world blurred images. The research work focused on the network structure and learning strategy. Summary of each work has been provided in the relevant chapters. In this chapter, a brief outline is given, which is not only a survey of the aforementioned work, but also a view of future research. Firstly, a discussion is given on the network structure and the potential of GCN based image inverse problems.

## 5.1 Discussion

### 5.1.1 Network Backbone and Framework

This thesis gives research on different network backbones on image deblurring. Network backbone including ResNet, DenseNet, Inception, etc. According to the experiments in chapter 3 and chapter 4, Resnet is an important network backbone. However, original ResNet in [46] is not suitable for image restoration. The data structure for image restoration, which is an image-to-image transformation, is different with classification and detection in computer vision. Modification is done on Resnet to make it reasonable for image restoration. In addition, pretrain model is not used directly in training, mainly because that the pretrain model is trained on general computer vision dataset (e.g., ImageNet), which is not very suitable for low-level vision and image processing. DenseNet often leads to good results but leads to more training time. Inception network can produce better results but also cost much more computational resources.

For network framework, FPN and multi-scale structure attracts lots of attention. FPN is not reasonable for image restoration because the decoder part is so simple. Multi-scale is the most important framework in image deblurring, as it has also been used recently in [72] and achieved very good performance. However, consider the application scene, multi-scale structure often costs more time.

Different backbones and frameworks can lead to different computational complexity. Here, to know the complexity of the three proposed method, we compare the proposed methods by floating point operations (FLOPs), as shown in Table 5.1.

**Table 5.1: Comparison of computational complexity of three proposed methods. GFLOPs index is used to evaluate the complexity.**

| Method | PSNR on GoPro dataset | GFLOPs |
|---|---|---|
| MixNet | 32.21 | 7.76 |
| DC-Deblur | 31.27 | 1.96 |
| GCResNet | 32.64 | 3.57 |

In Table 5.1, although DC-Deblur has a lower PSNR compared with MixNet on GoPro dataset, the cost of floating point operations is highly reduced by dilated convolution. GCResNet has acceptable complexity while the highest PSNR, which shows the value and performance of graph neural networks..

### 5.1.2 The Potential of GCN

In this subsection, the potential of GCN is analysed. First, the relationship between GCN and Laplacian Regularization / channel attention is given first.

*With Laplacian Regularization:* In the proposed GCN, feature maps of the encoder are first placed onto a graph structure. Then in the GC layers, these features, now nodes, further undergo graph convolutions. These GC layers inexplicitly apply graph Laplacian regularization [125], to the resulting the feature maps, $X\,\Theta$, of the encoder. The proposed approach combines the efficacy of CNNs in feature extraction and effectiveness of GNNs for constraining feature relationships. With a predefined graph structure, the method is also extremely efficient.

*With Channel Attention:* The proposed method can be regarded as an expansion of channel attention mechanism. Consider a simple example: for a network that only connects all nodes to 8 special nodes respectively. Hence the graph convolution in the proposed method is similar with channel-wise attention focused on these 8 channels, which has a similar structure in [58]. The graph which is used in the thesis is small-world

graph, with small average path length, thus key nodes can extract information from other vertexes within few GC layers.

According to the review in section 4.2, GCN is mainly used in human backbone, 3D point cloud, and object detection. These tasks have obvious topological structure, thus the adoption of graph neural network is reasonable. However, for image restoration, graph structure has not yet been fully explored. As analysed above, graph structure can process the relation of different features. Such inner relationships are of great research value. In addition, the vertex of graph should not be limited to features. Many other data can be considered as a node in a graph, for instance, a group of features, and a part of input image. Graph structure can also be considered to estimated blur kernels, especially non-uniform blur kernels.

## 5.2 Conclusions

Blind image deblurring is a challenging task in image processing and computer vision and a great deal of progress has been made recently. In this thesis, three novel image deblurring networks are proposed, which are MixNet, DC-Deblur, and GCResNet. The MixNet uses Inception-A to increase the channels of feature maps and receptive fields while maintaining reasonable complexity, so that the network is simple and easy to implement. Extensive experiments on various benchmark datasets and real-world images show the advantage of the proposed approach in dealing with single image deblurring. Performances of the proposed MixNet match or exceed markedly that of the state-of-the-art methods in a range of image quality measures. Tests on real-world blurry images also show the advantages of the proposed method.

DC-Deblur is based on a dilated convolution structure, which is able to increase the reception field without incurring additional parameters. The requirement of large reception field is unavoidable for image deblurring due to its ill-posed nature and possible large blur kernels. DC-Deblur can reduce the number of parameters and make the network

efficient. This work further improves the network performance by combining $l_1$ loss, MS-SSIM loss and MSE loss. Experiments and quantitative and qualitative evaluations indicate that the proposed method outperforms state-of-the-art models, in both performance and speed.

GCResNet is introduced in chapter 4 for image deblurring and super-resolution by adapting graph network in CNNs. While existing graph neural networks are for image classification, the proposed network explore graph structures in the feature maps of CNNs for effective image restoration. Experiments demonstrate that such adaptation with a predefined graph structure can achieve improved performance in image restoration with little added computational costs. Exploring topological relationships among feature maps is beneficial to many image processing tasks.

## 5.3 Future Work

This research focuses on the network structure of learning-based image deblurring. However, learning strategy is also important. Here, three possible research direction are listed for future work.

### 5.3.1 Model Compression and Performance of Algorithm

One of the challenges of image deblurring based on deep learning facing today is how to turn a large model into a small model that can be used on a variety of mobile devices. For image deblurring, neural network should be easy to train and have simple structure.

### 5.3.2 Better Dataset and Learning from Unlabeled Data

For image deblurring, datasets are critical to the effectiveness of training. Porav *et al.* [148] used a stereo dataset recorded using a system that allows one lens to be affected by real water droplets while keeping the other clear. If a fast lens and a time-lapse lens are used, suitable dataset for deblurring can be achieved. In [28], authors proposed a novel method for synthetic motion blurred images to allow realistic dataset augmentation. Aittala *et al.* [149] train the network with richly varied synthetic data, so that the framework can be used in both camera-shaking and noising. [74] proposed a novel self-supervised meta-auxiliary learning which enables internal training within a test image from scratch, while not fully rely on large external datasets. Reinforcement learning method is also considerable in deblurring, if a function that can judge the degree of blur is available, I can use reinforcement learning to convert a blurred image to the sharpest state.

### 5.3.3 Transfer Learning in image deblurring

Transfer Learning [150] is another angle of deblurring model generalization ability. For a real blurred image, blur may come from many aspects, such as out of focus, noise, low illumination, etc. A practical deblurring algorithm needs to be able to process images with multiple sources of blur at the same time. If there is only defocus or motion-blur in a data set, what kind of neural network can present a best result when Gaussian blur or low illumination is included in the actual scene? It is undeniable that this is a relatively difficult problem to solve, but it is very important for the development and application of image deblurring - the blurring of actual images is often unpredictable.

# References


[1] Peña F A G, Fernández P D M, Ren T I, et al. “Burst ranking for blind multi-image deblurring,” *IEEE Transactions on Image Processing*, 2019, 29: 947-958.

[2] Wang X, Chan K C K, Yu K, et al. “Edvr: Video restoration with enhanced deformable convolutional networks,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019: 0-0.

[3] Purohit K, Shah A, Rajagopalan A N. “Bringing alive blurred moments,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 6830-6839.

[4] Zhou S, Zhang J, Zuo W, et al. “Davanet: Stereo deblurring with view aggregation,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 10996-11005.

[5] Kim B, Son H, Park S J, et al. “Defocus and motion blur detection with deep contextual features,” *Computer Graphics Forum*. 2018, 37(7): 277-288.

[6] Lee J, Lee S, Cho S, et al. “Deep defocus map estimation using domain adaptation,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 12222-12230.

[7] Hosseini M S, Plataniotis K N. “Convolutional deblurring for natural imaging,” *IEEE Transactions on Image Processing*, 2019, 29: 250-264.

[8] Lai W S, Huang J B, Hu Z, et al. “A comparative study for single image blind deblurring,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2016: 1701-1709.

[9] Whyte O, Sivic J, Zisserman A, et al. “Non-uniform deblurring for shaken images,” *International journal of computer vision*, 2012, 98(2): 168-186.

[10] Pan J, Hu Z, Su Z, et al. “Deblurring text images via L0-regularized intensity and gradient prior,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2014: 2901-2908.

[11] Chan T F, Wong C K. “Total variation blind deconvolution,” *IEEE transactions on Image Processing*, 1998, 7(3): 370-375.

[12] Levin A, Weiss Y, Durand F, et al. “Understanding and evaluating blind deconvolution algorithms,” *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009: 1964-1971.

[13] Pan J, Sun D, Pfister H, et al. “Blind image deblurring using dark channel prior,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2016: 1628-1636.

[14] Yan Y, Ren W, Guo Y, et al. “Image deblurring via extreme channels prior,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2017: 4003-4011.

[15] Pan L, Hartley R, Liu M, et al. “Phase-only image based kernel estimation for single image blind deblurring,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 6034-6043.

[16] Krizhevsky A, Sutskever I, Hinton G E. “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, 2012, 25: 1097-1105.

[17] Sun J, Cao W, Xu Z, et al. “Learning a convolutional neural network for non-uniform motion blur removal,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2015: 769-777.

[18] Wieschollek P, Hirsch M, Scholkopf B, et al. “Learning blind motion deblurring,” *Proceedings of the IEEE International Conference on Computer Vision*. 2017: 231-240.

[19] Schuler C J, Hirsch M, Harmeling S, et al. “Learning to deblur,” *IEEE transactions on pattern analysis and machine intelligence*, 2015, 38(7): 1439-1451.

[20] Li L, Pan J, Lai W S, et al. “Blind image deblurring via deep discriminative priors,” *International journal of computer vision*, 2019, 127(8): 1025-1043.

[21] Sun L, Cho S, Wang J, et al. “Edge-based blur kernel estimation using patch priors,” *IEEE International Conference on Computational Photography (ICCP)*. IEEE, 2013: 1-8.

[22] Chakrabarti A. “A neural approach to blind motion deblurring,” *European conference on computer vision*. Springer, Cham, 2016: 221-235.

[23] Nah S, Hyun Kim T, Mu Lee K. ‘Deep multi-scale convolutional neural network for dynamic scene deblurring,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 3883-3891.

[24] Tao X, Gao H, Shen X, et al. “Scale-recurrent network for deep image deblurring,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2018: 8174-8182.

[25] Gao H, Tao X, Shen X, et al. “Dynamic scene deblurring with parameter selective sharing and nested skip connections,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 3848-3856.

[26] Zhang J, Pan J, Ren J, et al. “Dynamic scene deblurring using spatially variant recurrent neural networks,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018: 2521-2529.

[27] Zhang H, Dai Y, Li H, et al. “Deep stacked hierarchical multi-patch network for image deblurring,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 5978-5986.

[28] Kupyn O, Budzan V, Mykhailych M, et al. “Deblurgan: Blind motion deblurring using conditional adversarial networks,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018: 8183-8192.

[29] Kupyn O, Martyniuk T, Wu J, et al. “Deblurgan-v2: Deblurring (orders-of-magnitude) faster and better,” *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019: 8878-8887.

[30] Bernstein R. “Adaptive nonlinear filters for simultaneous removal of different kinds of noise in images,” *IEEE Transactions on Circuits and Systems*, 1987, 34(11): 1275-1291.

[31] Xu J, Zhang L, Zuo W, et al. “Patch group based nonlocal self-similarity prior learning for image denoising,” *Proceedings of the IEEE International Conference on Computer Vision*. 2015: 244-252.

[32] Tian C, Fei L, Zheng W, et al. “Deep learning on image denoising: An overview,” *Neural Networks*, 2020.

[33] Gu S, Zhang L, Zuo W, et al. “Weighted nuclear norm minimization with application to image denoising,” *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2014: 2862-2869.

[34] Schmidt U, Roth S. “Shrinkage fields for effective image restoration,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2014: 2774-2781.

[35] Mairal J, Bach F, Ponce J, et al. “Non-local sparse models for image restoration,” *2009 IEEE 12th international conference on computer vision.* IEEE, 2009: 2272-2279.

[36] Chen Y, Pock T. “Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration,” *IEEE transactions on pattern analysis and machine intelligence*, 2016, 39(6): 1256-1272.

[37] Chiang Y W, Sullivan B J. “Multi-frame image restoration using a neural network,” *Proceedings of the 32nd Midwest Symposium on Circuits and Systems*. IEEE, 1989: 744-747.

[38] Tamura S. “An analysis of a noise reduction neural network,” *International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 1989: 2001-2004.

[39] Bedini L, Tonazzini A. “Neural network use in maximum entropy image restoration,” *Image and Vision Computing*, 1990, 8(2): 108-114.

[40] Paik J K, Katsaggelos A K. “Image restoration using a modified Hopfield network,” *IEEE Transactions on image processing*, 1992, 1(1): 49-63.

[41] Sivakumar K, Desai U B. “Image restoration using a multilayer perceptron with a multilevel sigmoidal function,” *IEEE transactions on signal processing*, 1993, 41(5): 2018-2022.

[42] Lee C C, de Gyvez J P. “Color image processing in a cellular neural-network environment,” *IEEE Transactions on neural networks*, 1996, 7(5): 1086-1098.

[43] Zamparelli M. “Genetically trained cellular neural networks,” *Neural networks*, 1997, 10(6): 1143-1151.

[44] Krizhevsky A, Sutskever I, Hinton G E. “Imagenet classification with deep convolutional neural networks,” *Advances in neural information processing systems*, 2012, 25: 1097-1105.

[45] Zhang K, Zuo W, Chen Y, et al. “Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising,” *IEEE transactions on image processing*, 2017, 26(7): 3142-3155.

[46] He K, Zhang X, Ren S, et al. “Deep residual learning for image recognition,” *Proceedings of the IEEE conference on computer vision and pattern recognition.* 2016: 770-778.

[47] Zhang K, Zuo W, Zhang L. “FFDNet: Toward a fast and flexible solution for CNN-based image denoising,” *IEEE Transactions on Image Processing*, 2018, 27(9): 4608-4622.

[48] Narasimhan S G, Nayar S K. “Contrast restoration of weather degraded images,” *IEEE transactions on pattern analysis and machine intelligence*, 2003, 25(6): 713-724.

[49] Shwartz S, Schechner Y Y. “Blind haze separation,” *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE, 2006, 2: 1984-1991.

[50] Shaw S, Gupta R, Roy S. “A Review on Different Image De-hazing Methods,” *Emerging Technology in Modelling and Graphics*. Springer, Singapore, 2020: 533-540.

[51] Xu B, Yin H. “A Slimmer and Deeper Approach to Network Structures for Image Denoising and Dehazing,” *International Conference on Intelligent Data Engineering and Automated Learning*. Springer, Cham, 2020: 268-279.

[52] Tan R T. “Visibility in bad weather from a single image,” *2008 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2008: 1-8.

[53] Li B, Peng X, Wang Z, et al. “An all-in-one network for dehazing and beyond,” *arXiv preprint* arXiv:1707.06543, 2017.

[54] Qu Y, Chen Y, Huang J, et al. “Enhanced pix2pix dehazing network,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 8160-8168.

[55] Chen D, He M, Fan Q, et al. “Gated context aggregation network for image dehazing and deraining,” *2019 IEEE winter conference on applications of computer vision (WACV)*. IEEE, 2019: 1375-1383.

[56] Dong C, Loy C C, He K, et al. “Image super-resolution using deep convolutional networks,” *IEEE transactions on pattern analysis and machine intelligence*, 2015, 38(2): 295-307.

[57] Lim B, Son S, Kim H, et al. “Enhanced deep residual networks for single image super-resolution,” *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017: 136-144.

[58] Zhang Y, Li K, Li K, et al. “Image super-resolution using very deep residual channel attention networks,” *Proceedings of the European conference on computer vision (ECCV)*. 2018: 286-301.

[59] Richardson W H. “Bayesian-based iterative method of image restoration,” *JoSA*, 1972, 62(1): 55-59.

[60] Lucy L B. “An iterative technique for the rectification of observed distributions,” *The astronomical journal*, 1974, 79: 745.

[61] Biggs D S C, Andrews M. “Acceleration of iterative image restoration algorithms,” *Applied optics*, 1997, 36(8): 1766-1775.

[62] Kundur D, Hatzinakos D. “Blind image deconvolution,” *IEEE signal processing magazine*, 1996, 13(3): 43-64.

[63] Amizic B, Molina R, Katsaggelos A K. “Sparse Bayesian blind image deconvolution with parameter estimation,” *EURASIP Journal on Image and Video Processing*, 2012, 2012(1): 1-15.

[64] Levin A, Weiss Y, Durand F, et al. “Efficient marginal likelihood optimization in blind deconvolution,” *CVPR 2011*. IEEE, 2011: 2657-2664.

[65] Levin A, Weiss Y, Durand F, et al. “Understanding blind deconvolution algorithms,” *IEEE transactions on pattern analysis and machine intelligence*, 2011, 33(12): 2354-2367.

[66] Li J, Luisier F, Blu T. “PURE-LET image deconvolution,” *IEEE Transactions on Image Processing*, 2017, 27(1): 92-105.

[67] Afonso M V, Bioucas-Dias J M, Figueiredo M A T. “Fast image recovery using variable splitting and constrained optimization,” *IEEE transactions on image processing*, 2010, 19(9): 2345-2356.

[68] Liu G, Chang S, Ma Y. “Blind image deblurring using spectral properties of convolution operators,” *IEEE Transactions on image processing*, 2014, 23(12): 5047-5056.

[69] Purohit K, Rajagopalan A N. “Region-adaptive dense network for efficient motion deblurring. arxiv eprints, page,” *arXiv preprint* arXiv:1903.11394, 2019, 2(5).

[70] Cai J, Zuo W, Zhang L. “Dark and bright channel prior embedded network for dynamic scene deblurring,” *IEEE Transactions on Image Processing*, 2020, 29: 6885-6897.

[71] Li L, Pan J, Lai W S, et al. ”Dynamic scene deblurring by depth guided model,” *IEEE Transactions on Image Processing*, 2020, 29: 5273-5288.

[72] Zamir S W, Arora A, Khan S, et al. “Multi-stage progressive image restoration,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021: 14821-14831.

[73] Rozumnyi D, Oswald M R, Ferrari V, et al. “DeFMO: Deblurring and Shape Recovery of Fast Moving Objects,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021: 3456-3465.

[74] Chi Z, Wang Y, Yu Y, et al. “Test-Time Fast Adaptation for Dynamic Scene Deblurring via Meta-Auxiliary Learning,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021: 9137-9146.

[75] Gu C, Lu X, He Y, et al. “Blur removal via blurred-noisy image pair,” *IEEE Transactions on Image Processing*, 2020, 30: 345-359.

[76] Zhong Z, Gao Y, Zheng Y, et al. “Efficient spatio-temporal recurrent neural network for video deblurring,” *European Conference on Computer Vision*. Springer, Cham, 2020: 191-207.

[77] Li D, Xu C, Zhang K, et al. “Arvo: Learning all-range volumetric correspondence for video deblurring,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021: 7721-7731.

[78] Son H, Lee J, Lee J, et al. “Recurrent Video Deblurring with Blur-Invariant Motion Estimation and Pixel Volumes,” *ACM Transactions on Graphics (TOG)*, 2021, 40(5): 1-18.

[79] Lee J, Son H, Rim J, et al. “Iterative Filter Adaptive Network for Single Image Defocus Deblurring,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021: 2034-2042.

[80] Zhao W, Shang C, Lu H. “Self-Generated Defocus Blur Detection via Dual Adversarial Discriminators,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021: 6933-6942.

[81] Abuolaim A, Timofte R, Brown M S. “Ntire 2021 challenge for defocus deblurring using dual-pixel images: Methods and results,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2021: 578-587.

[82] Son H, Lee J, Cho S, et al. “Single Image Defocus Deblurring Using Kernel-Sharing Parallel Atrous Convolutions,” *arXiv preprint* arXiv:2108.09108, 2021.

[83] Goodfellow I, Pouget-Abadie J, Mirza M, et al. “Generative adversarial nets,” *Advances in neural information processing systems*, 2014, 27.

[84] Lazebnik S, Schmid C, Ponce J. “Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories,” *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*. IEEE, 2006, 2: 2169-2178.

[85] Lin T Y, Dollár P, Girshick R, et al. “Feature pyramid networks for object detection,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 2117-2125.

[86] Kirillov A, Girshick R, He K, et al. “Panoptic feature pyramid networks,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 6399-6408.

[87] Jolicoeur-Martineau A. “The relativistic discriminator: a key element missing from standard GAN,” *arXiv preprint* arXiv:1807.00734, 2018.

[88] Mao X, Li Q, Xie H, et al. “Least squares generative adversarial networks,” *Proceedings of the IEEE international conference on computer vision*. 2017: 2794-2802.

[89] Mao X, Li Q, Xie H, et al. “Least squares generative adversarial networks,” *Proceedings of the IEEE international conference on computer vision*. 2017: 2794-2802.

[90] Madam N T, Kumar S, Rajagopalan A N. “Unsupervised class-specific deblurring,” *European Conference on Computer Vision*. Springer, Cham, 2018: 358-374.

[91] Aljadaany R, Pal D K, Savvides M. “Douglas-rachford networks: Learning both the image prior and data fidelity terms for blind image deconvolution,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 10235-10244.

[92] Huang G, Liu Z, Van Der Maaten L, et al. “Densely connected convolutional networks,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 4700-4708.

[93] Pan J, Liu S, Sun D, et al. “Learning dual convolutional neural networks for low-level vision,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018: 3070-3079.

[94] Holschneider M, Kronland-Martinet R, Morlet J, et al. “A real-time algorithm for signal analysis with the help of the wavelet transform,” *Wavelets*. Springer, Berlin, Heidelberg, 1990: 286-297.

[95] Yu F, Koltun V. “Multi-scale context aggregation by dilated convolutions,” *arXiv preprint* arXiv:1511.07122, 2015.

[96] Dai J, Li Y, He K, et al. “R-fcn: Object detection via region-based fully convolutional networks,” *Advances in neural information processing systems*. 2016: 379-387.

[97] Oord A, Dieleman S, Zen H, et al. “Wavenet: A generative model for raw audio,” *arXiv preprint* arXiv:1609.03499, 2016.

[98] Kalchbrenner N, Oord A, Simonyan K, et al. “Video pixel networks,” *International Conference on Machine Learning*. PMLR, 2017: 1771-1779.

[99] Wang Z, Ji S. “Smoothed dilated convolutions for improved dense prediction,” *Data Mining and Knowledge Discovery*, 2021: 1-27.

[100] Zhao H, Gallo O, Frosio I, et al. “Loss functions for image restoration with neural networks,” *IEEE Transactions on computational imaging*, 2016, 3(1): 47-57.

[101] Lu B, Chen J C, Chellappa R. “Unsupervised domain-specific deblurring via disentangled representations,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 10225-10234.

[102] Liu X, Suganuma M, Sun Z, et al. “Dual residual networks leveraging the potential of paired operations for image restoration,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 7007-7016.

[103] Gong D, Yang J, Liu L, et al. “From motion blur to motion flow: A deep learning solution for removing heterogeneous motion blur,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017: 2319-2328.

[104] Yan Y, Ren W, Guo Y, et al. “Image deblurring via extreme channels prior,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017: 4003-4011.

[105] Pan L, Hartley R, Liu M, et al. “Phase-only image based kernel estimation for single image blind deblurring,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 6034-6043.

[106] Abadi M, Agarwal A, Barham P, et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems,” *arXiv preprint* arXiv:1603.04467, 2016.

[107] Paszke A, Gross S, Massa F, et al. “Pytorch: An imperative style, high-performance deep learning library.” *Advances in neural information processing systems*, 2019, 32: 8026-8037.

[108] Glorot X, Bengio Y. “Understanding the difficulty of training deep feedforward neural networks,” *Proceedings of the thirteenth international conference on artificial intelligence and statistics*. JMLR Workshop and Conference Proceedings, 2010: 249-256.

[109] Kingma D P, Ba J. “Adam: A method for stochastic optimization,” *arXiv preprint* arXiv:1412.6980, 2014.

[110] Brooks T, Barron J T. “Learning to synthesize motion blur,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2019: 6840-6848.

[111] Nah S, Baik S, Hong S, et al. “Ntire 2019 challenge on video deblurring and super-resolution: Dataset and study,” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*. 2019: 0-0.

[112] Shen Z, Wang W, Lu X, et al. “Human-aware motion deblurring,” *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019: 5572-5581.

[113] Yan Y, Ren W, Guo Y, et al. “Image deblurring via extreme channels prior,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 2017: 4003-4011.

[114] Hamilton W L, Ying R, Leskovec J. “Inductive representation learning on large graphs,” *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017: 1025-1035.

[115] Kipf T N, Welling M. “Semi-supervised classification with graph convolutional networks,” *arXiv preprint* arXiv:1609.02907, 2016.

[116] Sanchez-Gonzalez A, Heess N, Springenberg J T, et al. “Graph networks as learnable physics engines for inference and control,” *International Conference on Machine Learning. PMLR*, 2018: 4470-4479.

[117] Hamaguchi T, Oiwa H, Shimbo M, et al. “Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach,” *arXiv preprint* arXiv:1706.05674, 2017.

[118] Lee C W, Fang W, Yeh C K, et al. “Multi-label zero-shot learning with structured knowledge graphs,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018: 1576-1585.

[119] Wang X, Ye Y, Gupta A. “Zero-shot recognition via semantic embeddings and knowledge graphs,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018: 6857-6866.

[120] Bruna J, Zaremba W, Szlam A, et al. “Spectral networks and locally connected networks on graphs,” *arXiv preprint* arXiv:1312.6203, 2013.

[121] Bruna J, Zaremba W, Szlam A, et al. “Spectral networks and locally connected networks on graphs,” *arXiv preprint* arXiv:1312.6203, 2013.

[122] Henaff M, Bruna J, LeCun Y. “Deep convolutional networks on graph-structured data,” *arXiv preprint* arXiv:1506.05163, 2015.

[123] Gao H, Wang Z, Ji S. “Large-scale learnable graph convolutional networks,” *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 2018: 1416-1424.

[124] Xu B, Shen H, Cao Q, et al. “Graph wavelet neural network,” *arXiv preprint* arXiv:1904.07785, 2019.

[125] Li G, Muller M, Thabet A, et al. “Deepgcns: Can gcns go as deep as cnns? *Proceedings of the IEEE/CVF International Conference on Computer Vision.* 2019: 9267-9276.

[126] Yan S, Xiong Y, Lin D. “Spatial temporal graph convolutional networks for skeleton-based action recognition,” *Thirty-second AAAI conference on artificial intelligence*. 2018.

[127] Yang J, Lu J, Lee S, et al. “Graph r-cnn for scene graph generation,” *Proceedings of the European conference on computer vision (ECCV)*. 2018: 670-685.

[128] Johnson J, Gupta A, Fei-Fei L. “Image generation from scene graphs,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018: 1219-1228.

[129] Wang Y, Sun Y, Liu Z, et al. “Dynamic graph cnn for learning on point clouds,” *ACM Transactions on Graphics (TOG)*, 2019, 38(5): 1-12.

[130] Qi S, Wang W, Jia B, et al. “Learning human-object interactions by graph parsing neural networks,” *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018: 401-417.

[131] Guo M, Chou E, Huang D A, et al. “Neural graph matching networks for fewshot 3d action recognition.” *Proceedings of the European conference on computer vision (ECCV).* 2018: 653-669.

[132] Narasimhan M, Lazebnik S, Schwing A G. “Out of the box: Reasoning with graph convolution nets for factual visual question answering.” *Advances in Neural Information Processing Systems*, 2018, 2018: 2654-2665.

[133] Watts D J, Strogatz S H. “Collective dynamics of ‘small-world’ networks,” *nature*, 1998, 393(6684): 440-442.

[134] Xu B, Yin H. “Graph Convolutional Networks in Feature Space for Image Deblurring and Super-resolution.” *International Joint Conference on Neural Networks 2021*. IEEE, 2021.

[135] Borgatti S P. “Centrality and network flow,” *Social networks*, 2005, 27(1): 55-71.

[136] Zhou J, Cui G, Hu S, et al. “Graph neural networks: A review of methods and applications,” *AI Open*, 2020, 1: 57-81.

[137] Li G, Xiong C, Thabet A, et al. “Deepergcn: All you need to train deeper gcns,” *arXiv preprint* arXiv:2006.07739, 2020.

[138] Higham D J, Higham N J. “MATLAB guide.” *Society for Industrial and Applied Mathematics*, 2016.

[139] Bevilacqua M, Roumy A, Guillemot C, et al. “Low-Complexity Single-Image Super-Resolution based on Nonnegative Neighbor Embedding.” *British Machine Vision Conference (BMVC)*. 2012.

[140] Zeyde R, Elad M, Protter M. “On single image scale-up using sparse-representations.” *International conference on curves and surfaces. Springer, Berlin, Heidelberg*, 2010: 711-730.

[141] Huang J B, Singh A, Ahuja N. “Single image super-resolution from transformed self-exemplars.” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2015: 5197-5206.

[142] Martin D, Fowlkes C, Tal D, et al. “A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics.” *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*. IEEE, 2001, 2: 416-423.

[143] Agustsson E, Timofte R. “Ntire 2017 challenge on single image super-resolution: Dataset and study,” *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2017: 126-135.

[144] Zhang Y, Tian Y, Kong Y, et al. “Residual dense network for image super-resolution.” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018: 2472-2481.

[145] Haris M, Shakhnarovich G, Ukita N. “Deep back-projection networks for super-resolution,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018: 1664-1673.

[146] Dong C, Loy C C, Tang X. “Accelerating the super-resolution convolutional neural network,” *European conference on computer vision*. Springer, Cham, 2016: 391-407.

[147] Kim J, Lee J K, Lee K M. “Accurate image super-resolution using very deep convolutional networks,” *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016: 1646-1654.

[148] Porav H, Bruls T, Newman P. “I can see clearly now: Image restoration via de-raining,” *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019: 7087-7093.

[149] Aittala M, Durand F. “Burst image deblurring using permutation invariant convolutional neural networks,” *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018: 731-747.

[150] Pan S J, Yang Q. “A survey on transfer learning,” *IEEE Transactions on knowledge and data engineering*, 2009, 22(10): 1345-1359.

[151] Arjovsky M, Chintala S, Bottou L. “Wasserstein generative adversarial networks,” *International conference on machine learning*. PMLR, 2017: 214-223.

[152] Fan L, Zhang F, Fan H, et al. “Brief review of image denoising techniques,” *Visual Computing for Industry*, Biomedicine, and Art, 2019, 2(1): 1-12.

[153] Baxes G A. “Digital image processing: principles and applications,” *John Wiley & Sons, Inc.*, 1994.

[154] Tomasi C, Manduchi R. “Bilateral filtering for gray and color images,” *International Conference on Computer Vision (ICCV)*. IEEE, 1998: 839-846.

[155] Muresan D D, Parks T W. “Adaptive principal components and image denoising,” *Proceedings International Conference on Image Processing (ICIP)*. IEEE, 2003, 1: I-101.

[156] Mallat S G. “A Theory for Multiresolution Signal Decomposition: The Wavelet Representation,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 1989, 11(07): 674-693.

[157] Dabov K, Foi A, Katkovnik V, et al. “Image denoising by sparse 3-D transform-domain collaborative filtering,” *IEEE Transactions on image processing*, 2007, 16(8): 2080-2095.

[158] Szegedy, Christian, Sergey Ioffe, Vincent Vanhoucke, and Alexander A. Alemi. "Inception-v4, inception-resnet and the impact of residual connections on learning." In *Thirty-first AAAI conference on artificial intelligence*. 2017

# Appendix A Super-resolution Results of GCEDSR

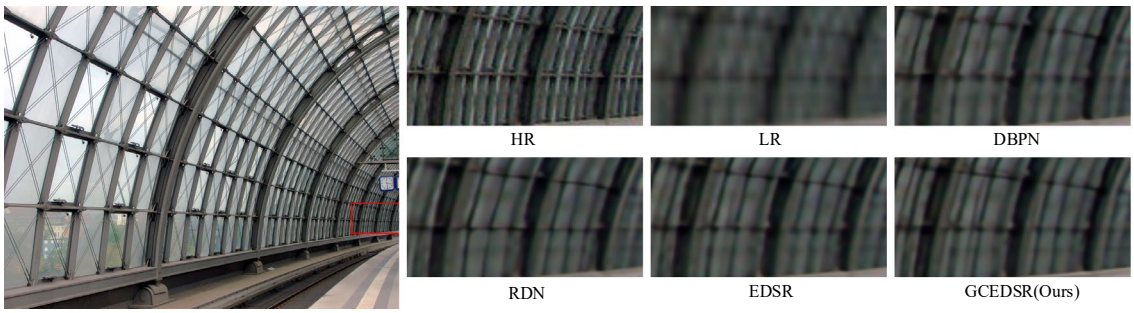



**Figure A.1 Testing results of Urban100_002 on single image super resolution.**

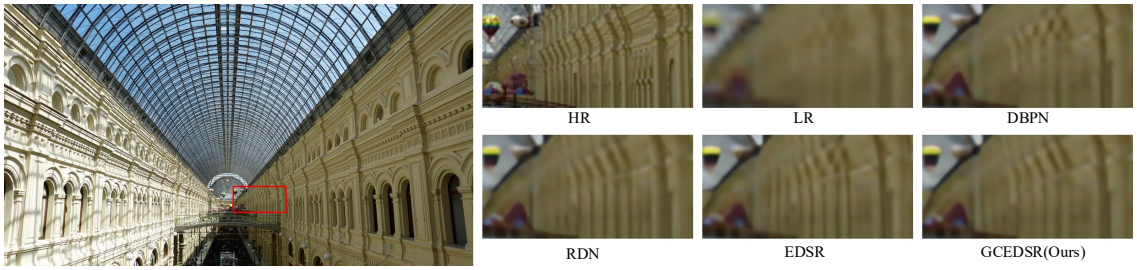



**Figure A.2 Testing results of Urban100_008 on single image super resolution.**

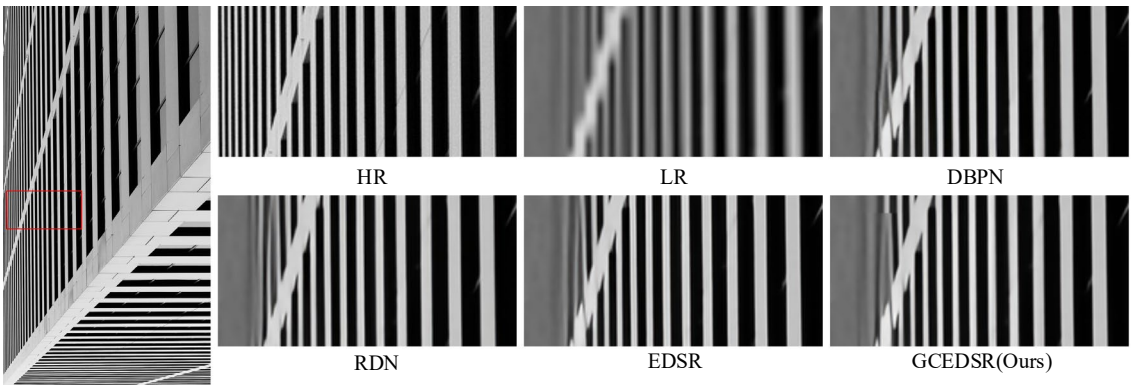



**Figure A.3 Testing results of Urban100_011 on single image super resolution.**

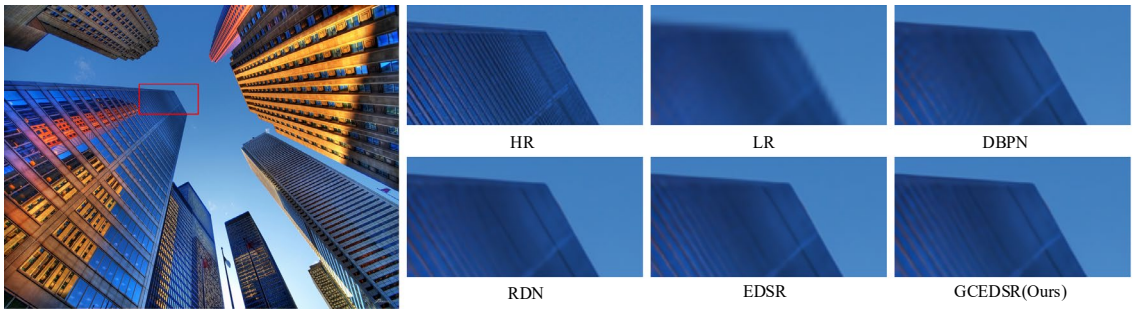



**Figure A.4 Testing results of Urban100_012 on single image super resolution.**


**Figure A.5 Testing results of Urban100_017 on single image super resolution.**

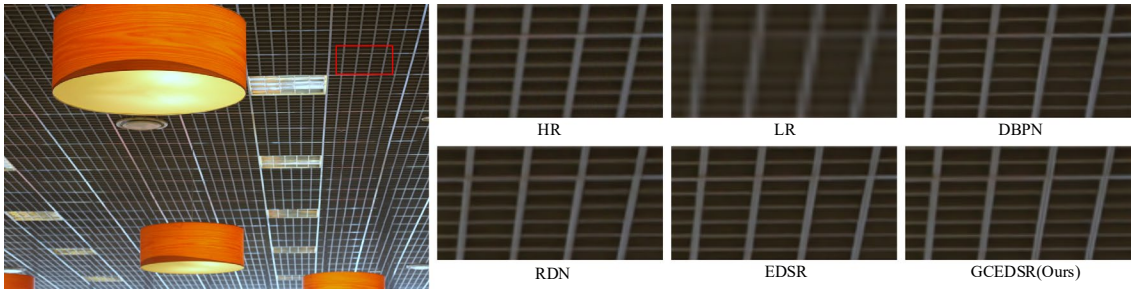



**Figure A.6 Testing results of Urban100_044 on single image super resolution.**

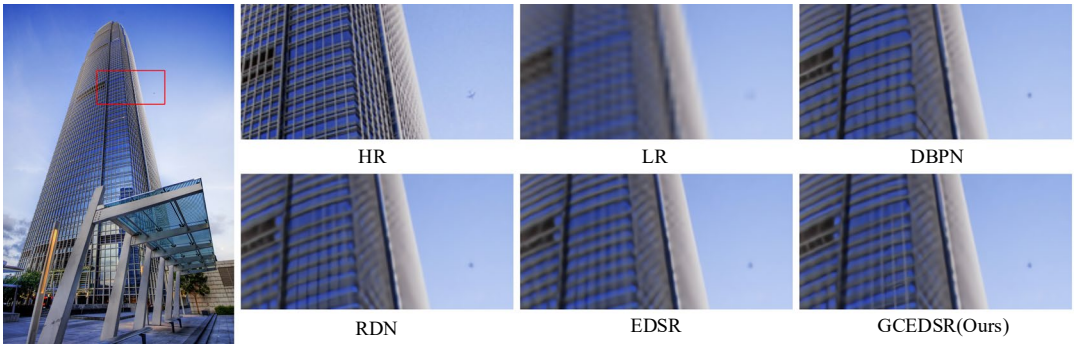


**Figure A.7 Testing results of Urban100_046 on single image super resolution.**

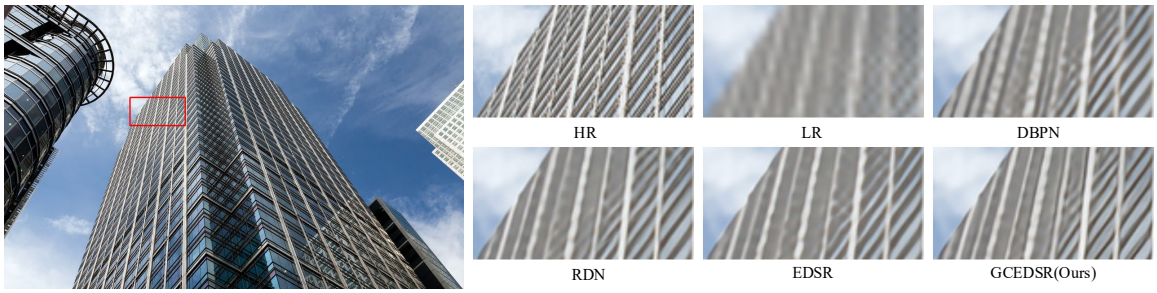



**Figure A.8 Testing results of Urban100_047 on single image super resolution.**

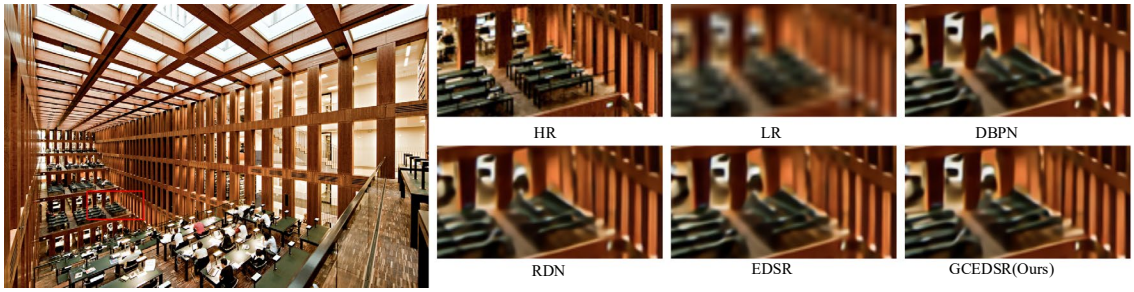



**Figure A.9 Testing results of Urban100_049 on single image super resolution.**

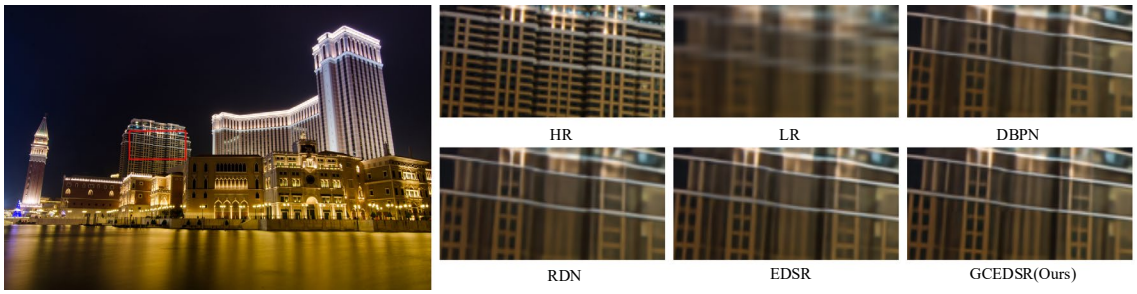



**Figure A.10 Testing results of Urban100_085 on single image super resolution.**

# Appendix B Network Structure

**Table B.1 Main network hyper-parameters of our proposed MixNet. All encoders and decoders are DenseBlocks. Inception network is added between encoder3_3 and decoder3_2. The structure of Inception network is not given in this table.**

| Layer | Structure | In_channels | Out_channels | Kernel size |
|---|---|---|---|---|
| Encoder1_1 | Conv. Layer | 3 | 32 | 3 |
| Encoder1_2 | DenseBlocks | 32 | 32 | 3 |
| Encoder1_3 | DenseBlocks | 32 | 32 | 3 |
| Encoder2_1 | Down-sample | 32 | 64 | 3 |
| Encoder2_2 | DenseBlocks | 64 | 64 | 3 |
| Encoder2_3 | DenseBlocks | 64 | 64 | 3 |
| Encoder3_1 | Down-sample | 64 | 128 | 3 |
| Encoder3_2 | DenseBlocks | 128 | 128 | 3 |
| Encoder3_3 | DenseBlocks | 128 | 128 | 3 |
| Inception | | - | - | - |
| Decoder3_2 | DenseBlocks | 128 | 128 | 3 |
| Decoder3_1 | DenseBlocks | 128 | 96 | 3 |
| Decoder2_3 | Up-sample | 96 | 64 | 4 |
| Decoder2_2 | DenseBlocks | 64 | 64 | 3 |
| Decoder2_1 | DenseBlocks | 64 | 64 | 3 |
| Decoder1_3 | Up-sample | 64 | 32 | 4 |
| Decoder1_2 | DenseBlocks | 32 | 32 | 3 |
| Decoder1_1 | DenseBlocks | 32 | 32 | 3 |
| Inp_pred | Conv. Layer | 32 | 3 | 3 |

**Table B.2 Structure of our proposed DC-Deblur network. SDBlcok denotes smooth dilated residual block.**

| Layer | Structure | In_channels | Out_channels | Kernel size |
|---|---|---|---|---|
| Head | Conv. Layer | 3 | 32 | 3 |
| Ebody1_1 | DenseBlock | 32 | 32 | 3 |
| Ebody1_2 | DenseBlock | 32 | 32 | 3 |
| Ebody1_3 | Down-sample | 32 | 64 | 3 |
| Ebody2_1 | SDBlcok | 64 | 64 | 3 |
| Ebody2_2 | SDBlcok | 64 | 64 | 3 |
| Ebody2_3 | SDBlcok | 64 | 64 | 3 |
| Ebody2_4 | SDBlcok | 64 | 64 | 3 |
| Ebody2_5 | SDBlcok | 64 | 64 | 3 |
| Ebody2_6 | SDBlcok | 64 | 64 | 3 |
| Ebody2_7 | ResNet | 64 | 64 | 3 |
| Gated Fusion | | - | - | - |
| Dbody4 | Up-ample | 64 | 32 | 3 |
| Dbody3 | DenseBlock | 32 | 32 | 3 |
| Dbody2 | DenseBlock | 32 | 32 | 3 |
| Dbody1 | Conv. Layer | 32 | 3 | 3 |

**Table B.3 Structure of the encoder and decoder of our proposed GCResNet. Graph network is added between enc_3 and dec_3.**

| Layer | Structure | In_channels | Out_channels | Kernel size |
|---|---|---|---|---|
| m_head | Conv. Layer | 3 | 128 | 3 |
| Encoder1 | ResBlock $\times n$ | 128 | 128 | 3 |
| Downsample1 | Down-sample | 128 | 128 | 3 |
| Encoder2 | ResBlock $\times n$ | 128 | 128 | 3 |
| Downsample2 | Down-sample | 128 | 128 | 3 |
| Encoder3 | ResBlock $\times n$ | 128 | 128 | 3 |
| Downsample3 | Down-sample | 128 | 128 | 3 |
| Upsample_graph | Up-sample | 128 | 128 | 3 |
| Decoder3 | ResBlock $\times n$ | 128 | 128 | 3 |
| Upsample2 | Up-sample | 128 | 128 | 3 |
| Decoder2 | ResBlock $\times n$ | 128 | 128 | 3 |
| Upsample1 | Up-sample | 128 | 128 | 3 |
| Decoder1 | ResBlock $\times n$ | 128 | 128 | 3 |
| m_tail | Conv. Layer | 128 | 3 | 3 |

# Appendix C WS-GAN

Training of the vanilla version of GAN suffers from many problems such as mode collapse and vanishing gradients etc., as described in [151]. Minimizing the value function in GAN is equal to minimizing the Jensen-Shannon (JS) divergence between the data and model distributions on Arjovsky *et al.* [151] discussed the difficulties in GAN training caused by JS-divergence approximation and proposed to use the Earth-Mover (also called Wasserstein-1) distance $W(q, p)$.The value function for Wasserstein-GAN is constructed usingthe Kantorovich-Rubinstein duality [6]:

$$\min_{G} \min_{D \in \boldsymbol{D}} \boldsymbol{E}_{a \sim P_r}[D(a)] + \boldsymbol{E}_{\tilde{a} \sim P_g}[D(\tilde{a})]. \tag{D.2}$$

Where $\boldsymbol{D}$ is the set of 1-Lipschitz functions and $P_g$ is once again the model distribution. The idea here is that critic value approximates $K \times W(P_r, P_\theta)$, where $K$ is a Lipschitz constant and $W(P_r, P_\theta)$ is a Wasserstein distance. In this setting, the discriminator network is called critic and it approximates the distance between the samples. To enforce Lipschitz constraint in WGAN, Arjovsky *et al.* [151] added weight clipping to $[-c, c]$. Gulrajanib *et al.* proposed to add a gradient penalty term to the value function as an alternative way to enforce the Lipschitz constraint instead:

$$\lambda \, \mathbb{E}_{\tilde{a} \sim P_a}[(\|\nabla_{\tilde{a}} D(\tilde{a})\|_2 - 1)^2] \tag{D.3}$$

This approach is robust to the choice of generator architecture and requires almost no hyper-parameter tuning. This is crucial for image deblurring as it allows to use novel light weight architectures in contrast to the standard Deep ResNet architectures, previously used for image deblurring.