





Universitat Autònoma de Barcelona

**ADVERTIMENT.** L'accés als continguts d'aquesta tesi queda condicionat a l'acceptació de les condicions d'ús establertes per la següent llicència Creative Commons:  [http://cat.creativecommons.org/?page\\_id=184](http://cat.creativecommons.org/?page_id=184)

**ADVERTENCIA.** El acceso a los contenidos de esta tesis queda condicionado a la aceptación de las condiciones de uso establecidas por la siguiente licencia Creative Commons:  <http://es.creativecommons.org/blog/licencias/>

**WARNING.** The access to the contents of this doctoral thesis it is limited to the acceptance of the use conditions set by the following Creative Commons license:  <https://creativecommons.org/licenses/?lang=en>



**Universitat Autònoma  
de Barcelona**

# Towards Efficient and Robust Convolutional Neural Networks for Single Image Super-Resolution

A dissertation submitted by **Parichehr Behjati** at  
Universitat Autònoma de Barcelona to fulfil the de-  
gree of **Doctor of Philosophy**.

Bellaterra, February 16, 2022

Co-Director	<b>Dr. Jordi Gonzalez Sabaté</b> Dept. Ciències de la computació & Centre de Visió per Computador Universitat Autònoma de Barcelona (UAB), Spain
Co-Director	<b>Dr. Pau Rodriguez Lopez</b> ElemantAI - ServiceNow Montreal, Canada
Co-Director	<b>Dr. F. Xavier Roca Marvà</b> Dept. Ciències de la computació & Centre de Visió per Computador Universitat Autònoma de Barcelona (UAB), Spain
Thesis committee	<b>Dr. David Masip Rodó</b> Dept. Informàtica, Multimèdia i Telecomunicació Universitat Oberta de Catalunya, Barcelona Spain
	<b>Dr. Jorge Bernal del Nozal</b> Dept. Ciències de la computació & Centre de Visió per Computador Universitat Autònoma de Barcelona (UAB), Spain
	<b>Dr. Antoni Jaume-i-Capó</b> Dept. de C. Matemàtiques i Informàtica Universitat de les Illes Balears, Palma de Mallorca, Spain




---

This document was typeset by the author using  $\text{\LaTeX} 2_{\epsilon}$ .

The research described in this book was carried out at the Centre de Visió per Computador, Universitat Autònoma de Barcelona. Copyright © 2022 by **Parichehr Behjati**. All rights reserved. No part of this publication may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopy, recording, or any information storage and retrieval system, without permission in writing from the author.

ISBN: 978-84-124793-1-7

Printed by Ediciones Gráficas Rey, S.L.

Our greatest weakness lies in giving up.  
The most certain way to succeed is always to try just one more time.  
— Thomas Edison

To my family...



# Acknowledgements

Working as a Ph.D. student was a magnificent as well as challenging experience for me. In all these years, many people were instrumental directly or indirectly in shaping up my academic career. It was hardly possible for me to thrive in my doctoral work without the precious support of these personalities. Here is a small tribute to all these people.

First and foremost, my sincere gratitude and *muchísimas gracias* go to my dissertation director Jordi Gonzalez for believing me from the beginning and letting me fulfill my dream. Not so many Ph.D. candidates are as lucky as I am to have the supportive learning environment, intellectual freedom, and invaluable advice that he has provided. Without his help, advice, expertise, and encouragement this research and dissertation would not have happened. Words cannot express how thankful I am to have you as my director. Thank you for your patient support and for all of the opportunities I was given to further my research.

This dissertation would also not have been possible without Pau Rodriguez. The success I have enjoyed in research, the motivation I have felt towards my work, the drive of wanting to achieve something new every day, and the burning desire to continue my work well into the future, would never have been possible if it wasn't for the tirelessness of him. I am extremely grateful for the time and effort he has invested into my work and will forever cherish the time we have spent together, discussing ideas, brainstorming, and coming up with new directions in my work. I will forever be indebted to him for pushing me to my limits, without which I would not have been who I am today.

I also would like to express my special gratitude to Isabelle Hupont and Carles Fernandez for their invaluable support, assistance, and expert guidance. I must also thank all the people from the CVC administration for all the hard work, dedication, help, and sympathy provided over the years. Among them, I would like to express my sincere gratitude, particularly to Montse, and Gisele for all their kind help and support.

Last but not means least is the recognition of the support given by my family and many friends. It was my parents' unconditional love, care, and tolerance that made the hardship of writing the thesis worthwhile. Without their support, I don't think that I could overcome the difficulties during these years. I am forever indebted to my parents for giving me the opportunities and experiences that have made me who I am. Thank you mom and dad for everything.



# Abstract

Single image super-resolution (SISR) is an important task in image processing, which aims to enhance the resolution of imaging systems. Recently, SISR has witnessed great strides with the rapid development of deep learning. Recent advances on SISR are mostly devoted to designing deeper and wider networks to enhance their representation learning capacity. However, as the depth of networks increases, deep learning-based methods are faced with the challenge of computational complexity in practice. Moreover, most existing methods rarely leverage the intermediate features and also do not discriminate the computation of features by their frequencial components, thereby achieving relatively low performance. Aside from the aforementioned problems, another desired ability is to upsample images to arbitrary scales using a single model. Most current SISR methods train a dedicated model for each target resolution, losing generality and increasing memory requirements. In this thesis, we address the aforementioned issues and devise solutions in each chapter: i) We present a novel frequency-based enhancement block which treats different frequencies in a heterogeneous way and also models inter-channel dependencies, which consequently enrich the output feature. Thus it helps the network generate more discriminative representations by explicitly recovering finer details. ii) We introduce OverNet which contains two main parts: a lightweight feature extractor that follows a novel recursive framework of skip and dense connections to reduce low-level feature degradation, and an overscaling module that generates accurate SR image by internally constructing an overscaled intermediate representation of the output features. Then, to solve the problem of reconstruction at arbitrary scale factors, we introduce a novel multi-scale loss, that allows the simultaneous training of all scale factors using a single model. iii) We propose a directional variance attention network which leverages a novel attention mechanism to enhance features in different channels and spatial regions. Moreover, we introduce a novel procedure for using attention mechanisms together with residual blocks to facilitate the preservation of finer details. Finally, we demonstrate that our approaches achieve considerably better performance than previous state-of-the-art methods, in terms of both quantitative and visual quality.

**Key words:** *Single image super-resolution, deep learning, image processing*





## Resumen

El Análisis Super-Resolución de Imágenes (SISR) es una de las tareas más importantes en procesamiento de imágenes, ya que su objetivo es, entre otros generar imágenes de alta fidelidad usando imágenes de muy baja resolución. Recientemente, y como en otros tantos campos, SISR ha sido testigo de grandes avances con el rápido desarrollo de nuevos métodos de aprendizaje profundo. Los avances recientes se han dedicado principalmente al diseño de redes más profundas y amplias para mejorar su capacidad de aprendizaje de representación. Sin embargo, a medida que aumenta la profundidad de las redes, los métodos basados en aprendizaje profundo se enfrentan al desafío de la complejidad computacional. Además, la mayoría de los métodos existentes rara vez aprovechan las características que se generan en las capas intermedias, y tampoco discriminan el cálculo de características por sus componentes frecuenciales, por lo que logran un rendimiento relativamente bajo. En esta Tesis, se abordan los problemas mencionados anteriormente: i) Presentamos un nuevo método de mejora basado en la frecuencia y dependencias entre canales, lo que en consecuencia enriquece la función de salida y, por tanto, ayuda a la red poder generar representaciones más discriminatorias y ser capaces de recuperar explícitamente detalles más finos. ii) Presentamos una nueva estructura de red llamada OverNet, que contiene dos partes principales: un extractor de características que sigue un nuevo esquema recursivo de conexiones densas y de salto para reducir la degradación de características de bajo nivel, y un módulo de sobreescala que genera una imagen super-resolución precisa mediante la construcción interna de una representación intermedia sobreescala a partir de las características de la salida de la red. Posteriormente, para resolver el problema de la reconstrucción a factores de escala arbitrarios, introducimos una nueva función de pérdida multiescala, que permite el entrenamiento simultáneo de múltiples factores de escala utilizando un único modelo. Y iii) proponemos una nueva arquitectura de red de variación direccional que saca provecho de un nuevo mecanismo de atención para mejorar las características en diferentes canales y regiones espaciales. Además, presentamos un nuevo procedimiento para usar dichos mecanismos de atención junto con bloques residuales para facilitar la preservación de los detalles más finos. Finalmente, demostramos que nuestros enfoques logran un rendimiento considerablemente mejor que los métodos del estado del arte más actuales, en términos de calidad cuantitativa y visual.

**Palabras clave:** *Superresolución de imágenes, aprendizaje profundo, procesamiento de imágenes*



## Resum

L'Anàlisi Super-Resolució d'Imatges (SISR) és una de les tasques més importants en processament d'imatges, ja que el seu objectiu és, entre d'altres, generar imatges d'alta fidelitat usant imatges de resolució molt baixa. Recentment, i com en altres camps, SISR ha estat testimoni de grans avenços amb el ràpid desenvolupament de nous mètodes d'aprenentatge profund. Els avenços recents s'han dedicat principalment al disseny de xarxes més profundes i àmplies per millorar-ne la capacitat d'aprenentatge de representació. Tot i això, a mesura que augmenta la profunditat de les xarxes, els mètodes basats en aprenentatge profund s'enfronten al desafiament de la complexitat computacional. A més, la majoria dels mètodes existents poques vegades aprofiten les característiques que es generen a les capes intermèdies, i tampoc discriminen el càlcul de característiques pels seus components freqüencials, per la qual cosa aconseguen un rendiment relativament baix. En aquesta Tesi, s'aborden els problemes esmentats anteriorment: i) Presentem un nou mètode de millora basat en la freqüència i dependències entre canals, cosa que en conseqüència enriqueix la funció de sortida i, per tant, ajuda a la xarxa poder generar representacions més discriminatòries i ser capaç de recuperar explícitament detalls més fins. ii) Presentem una nova estructura de xarxa anomenada OverNet, que conté dues parts principals: un extractor de característiques que segueix un nou esquema recursiu de connexions denses i de salt per reduir la degradació de característiques de baix nivell, i un mòdul de sobreescala que genera una imatge superresolució precisa mitjançant la construcció interna d'una representació intermèdia sobreescala a partir de les característiques de la sortida de la xarxa. Posteriorment, per resoldre el problema de la reconstrucció a factors d'escala arbitraris, introduïm una nova funció de pèrdua multiescala, que permet l'entrenament simultani de múltiples factors d'escala utilitzant un únic model. I iii) proposem una nova arquitectura de xarxa de variació direccional que treu profit d'un nou mecanisme d'atenció per millorar les característiques a diferents canals i regions espacials. A més, presentem un nou procediment per fer servir aquests mecanismes d'atenció juntament amb blocs residuals per facilitar la preservació dels detalls més fins. Finalment, demostrem que els nostres enfocaments aconseguen un rendiment considerablement millor que els mètodes de l'estat de l'art més actuals, en termes de qualitat quantitativa i visual.

**Paraules clau:** *Superresolució d'imatges, aprenentatge profund, processament d'imatges*



# Contents

<b>Abstract (English/Spanish/Catalan)</b>	<b>iii</b>
<b>List of figures</b>	<b>xiii</b>
<b>List of tables</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Single Image Super-Resolution via Deep Learning . . . . .	3
1.2 Objectives . . . . .	4
1.3 Thesis Outline and Contributions . . . . .	6
<b>2 Background</b>	<b>9</b>
2.1 Problem Definition . . . . .	9
2.2 Benchmark Datasets . . . . .	10
2.2.1 Training and Test Datasets . . . . .	11
2.3 Assessment Methods . . . . .	12
2.3.1 Image Reconstruction Accuracy . . . . .	12
2.3.2 Image Perceptual Quality . . . . .	13
2.3.3 Reconstruction Efficiency . . . . .	15
2.4 Super-resolution Frameworks . . . . .	15

2.4.1	Pre-upsampling Super-resolution . . . . .	16
2.4.2	Post-upsampling Super-resolution . . . . .	17
2.4.3	Progressive Upsampling Super-resolution . . . . .	17
2.4.4	Up-and-down Sampling Super-resolution . . . . .	18
2.5	Upsampling Methods . . . . .	18
2.5.1	Interpolation-based Upsampling . . . . .	18
2.5.2	Learning-based Upsampling . . . . .	19
2.6	Optimization Objective . . . . .	22
2.6.1	Learning Strategy . . . . .	22
2.6.2	Loss Functions . . . . .	23
2.6.3	Other Improvements . . . . .	25
2.7	Most Related CNN-based Frameworks for SISR . . . . .	26
2.7.1	Evolution of Architectures . . . . .	26
2.7.2	Frequency-based Networks . . . . .	28
2.7.3	Attention Mechanisms . . . . .	29
2.7.4	Reconstruction Methods . . . . .	30
2.8	Summary . . . . .	31
<b>3</b>	<b>Frequency-based Enhancement Network for Efficient Super-Resolution</b>	<b>33</b>
3.1	Motivation . . . . .	33
3.2	Frequency-based Enhancement Network . . . . .	35
3.2.1	Network overview . . . . .	36
3.2.2	Frequency-based Enhancement Block (FEB) . . . . .	37
3.2.3	Discussion . . . . .	39

3.3	Experimental Results . . . . .	40
3.3.1	Settings . . . . .	40
3.3.2	Ablation Study . . . . .	41
3.3.3	Comparison With state-of-the-art Methods . . . . .	46
3.4	Summary . . . . .	52
<b>4</b>	<b>Lightweight Multi-Scale Super-Resolution with Overscaling Network</b>	<b>53</b>
4.1	Motivation . . . . .	53
4.2	Proposed Overscaling Network . . . . .	54
4.2.1	Feature Extractor . . . . .	56
4.2.2	Overscaling Module . . . . .	57
4.2.3	Multi-Scale Loss . . . . .	58
4.2.4	Difference with Other SR Methods . . . . .	59
4.3	Experimental Results . . . . .	59
4.3.1	Settings . . . . .	59
4.3.2	Ablation Studies . . . . .	60
4.3.3	Comparison with State-of-the-art Methods . . . . .	65
4.4	Summary . . . . .	72
<b>5</b>	<b>Directional Variance Attention Networks</b>	<b>73</b>
5.1	Motivation . . . . .	73
5.2	Directional Variance Attention Network (DiVANet) . . . . .	76
5.2.1	Network Overview . . . . .	76
5.2.2	Directional Variance Attention (DiVA) . . . . .	77



## Contents

---

5.2.3	Residual Attention Feature Group (RAFG) . . . . .	80
5.3	Experimental Results . . . . .	81
5.3.1	Settings . . . . .	81
5.3.2	Ablation Study . . . . .	82
5.3.3	Comparison with State-of-the-art Lightweight Methods . . . . .	88
5.4	Summary . . . . .	97
<b>6</b>	<b>Conclusions and Future work</b>	<b>99</b>
6.1	Conclusions . . . . .	99
6.2	Future Perspective . . . . .	100
6.3	Scientific Articles . . . . .	101
6.3.1	Submitted Journals . . . . .	101
6.3.2	International Conferences and Workshops . . . . .	102
6.4	Contributed Code . . . . .	102
<b>A</b>	<b>Experiments on Other Image Enhancement Tasks</b>	<b>103</b>
A.1	Experimental Settings . . . . .	104
A.2	Results on Image Dehazing . . . . .	105
A.3	Results on JPEG Compression Artifacts Reduction . . . . .	107
A.4	Summary . . . . .	110
	<b>Bibliography</b>	<b>124</b>

# List of Figures

1.1	The taxonomy of existing super-resolution techniques [17]. . . . .	2
1.2	Example of a low-resolution image compared to a high-resolution sample of the same scene. . . . .	3
1.3	Example of single image super-resolution (SISR) pipeline. . . . .	4
2.1	Downsampling and upsampling in super-resolution. Noise is added to simulate realistic degradation within an image. . . . .	10
2.2	Representative test images from six super-resolution datasets used for comparing and evaluating algorithms. . . . .	11
2.3	Super-resolution model frameworks based on deep learning. The cube size represents the output size. The gray ones denote predefined upsampling, while the green,yellow and blue ones indicate learnable upsampling, downsampling and convolutional layers, respectively. And the blocks enclosed by dashed boxes represent stackable modules [127]. . . . .	16
2.4	Transposed convolution layer. The blue boxes denote the input, and the green boxes indicate the kernel and the convolution output. . . .	19
2.5	Sub-pixel layer. The blue boxes denote the input, and the boxes with other colors indicate different convolution operations and different output feature maps. . . . .	20
2.6	Meta upscale module. The blue boxes denote the projection patch, and the green boxes and lines indicate the convolution operation with predicted weights. . . . .	21

3.1	The visual comparison of SR results by the networks with different building modules for scale factor $\times 4$ . The residual block is used as building module for EDSR. In EDSR-FEB, we replace residual block with proposed FEB. . . . .	35
3.2	Proposed frequency-based enhancement network (FENet) for SISR, which consists of non-linear mapping and reconstruction modules. .	36
3.3	Schematic illustration of the proposed Frequency-based Enhancement Network (FEB). As it can be seen, the original filters are separated into two processing lines, each of which is in charge of a different functionality. . . . .	38
3.4	Visual activation feature maps of input $\mathbf{X}_2, \mathbf{T}_2$ , and obtained high-frequency information ( $\mathbf{T}_3$ ). . . . .	39
3.5	Visual comparisons of SR results using FENet with different SR blocks for scale factor $\times 4$ . . . . .	43
3.6	The visual comparison of SR results by the networks with different building modules for $\times 4$ scale factor. The residual blocks followed by channel attentions are used as building modules for RCAN. In RCAN-FEB, we replace its blocks with proposed FEBs. . . . .	45
3.7	Visual results of <b>BI</b> degradation model ( $\times 4$ ). . . . .	48
3.8	Visual results of <b>BD</b> degradation model ( $\times 3$ ). . . . .	49
3.9	Visual results of <b>DN</b> degradation model ( $\times 3$ ). . . . .	50
3.10	Comparing capacity vs performance for lightweight state-of-the-art SISR models on Urban100 ( $\times 4$ ). Circle sizes are set proportional to the number of multiplications and additions (Multi-Adds). . . . .	51
4.1	Demonstration of our proposed overscaling network with short and long skip connections. As the maximum scale factor in this particular example is set to $N = 4$ , the required overscaling is $\times 5$ . . . . .	56
4.2	Visual results of <b>BI</b> degradation model for scale factor $\times 4$ . . . . .	67
4.3	Visual results of <b>BD</b> degradation model for scale factor $\times 3$ . . . . .	69

4.4	Visual results of <b>DN</b> degradation models for scale factor $\times 3$ . . . . .	69
4.5	Comparative capacity and performance of state-of-the-art SISR models. The <b>red stars</b> represents our methods. . . . .	70
5.1	(a) Basic residual block without attention mechanisms. (b) Residual channel attention block proposed in previous works. (c) Our proposed directional variance attention (DiVA), which has its own dedicated computational path. . . . .	74
5.2	<b>Top:</b> Proposed directional variance attention network (DiVNet) architecture for SISR. <b>Bottom:</b> residual attention feature group (RAFG), containing residual blocks (RB) and the proposed directional variance attention (DiVA). . . . .	76
5.3	Visual comparison of SR results using DiVNet with different attention mechanisms ( $\times 4$ scale factor). . . . .	84
5.4	Average feature maps of residual blocks (RBs). <b>Top:</b> Attention is applied within the residual (classic approach). <b>Bottom:</b> Attention is applied outside the residual (our approach). . . . .	87
5.5	Visual results of <b>BI</b> degradation model for $\times 4$ scale factor. . . . .	90
5.6	Visual results of <b>BD</b> degradation model for $\times 3$ scale factor. . . . .	92
5.7	Visual results of <b>DN</b> degradation model for $\times 3$ scale factor. . . . .	93
5.8	Comparing capacity vs performance for lightweight state-of-the-art SISR models on Urban100 ( $\times 4$ ). Circle sizes are set proportional to the number of multiplications and additions (Multi-Adds). . . . .	94
5.9	Comparing capacity vs performance for non-lightweight state-of-the-art SISR models in the B100 dataset ( $\times 4$ ). The red stars represent our proposed methods. . . . .	95
A.1	(a): High-resolution image (HR). (b): hazy image. (c): the JPEG-compressed image, where we could see blocking artifacts, ringing effects and blurring on the eyes, abrupt intensity changes on the face. 104	
A.2	Qualitative comparisons on SOTS dataset (indoor). . . . .	106

## List of Figures

---

A.3	Qualitative comparisons on SOTS dataset (outdoor). . . . .	106
A.4	Qualitative comparisons on realistic hazy images. . . . .	107
A.5	Image compression artifacts reduction results with JPEG quality $\sigma = 10$ .	109

# List of Tables

3.1	Average PSNR obtained when FEB using different pooling methods on five benchmark datasets for scale factor $\times 4$ . . . . .	41
3.2	Average PSNR to show the effect of downsampling rate on the performance on Set5 dataset. We record the results in $10 \times 10^4$ iterations. . . . .	42
3.3	Average PSNR obtained with FENet when using different number of FEBs on five benchmark datasets for scale factor $\times 4$ . . . . .	42
3.4	Average PSNR obtained with FENet when using different SR blocks on five benchmark datasets for scale factor $\times 4$ . . . . .	43
3.5	Average PSNR obtained with FENet when using different attention mechanisms on five benchmark datasets scale factor $\times 4$ . . . . .	44
3.6	Average PSNR obtained with state-of-the-art SR methods when using FEB on five benchmark datasets for scale factor $\times 4$ . . . . .	45
3.7	Average PSNR/SSIM values for models with the same order of magnitude of parameters. Performance is shown for scale factors $\times 2$ , $\times 3$ and $\times 4$ with <b>BI</b> degradation model. The best and second best results are highlighted in <b>red</b> and <b>blue</b> respectively. . . . .	47
3.8	Quantitative results with <b>BD</b> and <b>DN</b> degradation models. The best and second best results are highlighted in <b>red</b> and <b>blue</b> respectively. . . . .	49
3.9	Average running time (s) and memory consumption (MB) comparison on Urban100 for $\times 4$ scale factor. . . . .	51
3.10	Perceptual index comparison of the proposed method with recent lightweight state-of-the-art methods on five datasets for $\times 4$ . The lower is better. All of the output SR images are provided officially. . . . .	52

4.1	Effects of skip connections (SCs) in local and global dense groups (LDG, GDG) measured on Urban100 with $\times 3$ . The best result is <b>highlighted</b> . . . . .	61
4.2	PSNR results of different OSM upscaling methods trained for arbitrary scales. The test dataset is B100. Best results are <b>highlighted</b> , second best <u>underlined</u> . . . . .	62
4.3	Average PSNR of state-of-the-art methods using OSM instead of their typical upsampling module. The best results are <b>highlighted</b> . . . . .	63
4.4	Average PSNR to show the performance of OverNet across scales. The test dataset is Set5. Best results are <b>highlighted</b> . . . . .	64
4.5	Effect of multi-scale loss. OverNet-S uses single-scale loss, OverNet-M multi-scale loss. Best results are <b>highlighted</b> . . . . .	64
4.6	Average PSNR/SSIM values for models with the same order of magnitude of parameters. Performance is shown for scale factors $\times 2$ , $\times 3$ and $\times 4$ with <b>BI</b> degradation model. The best and second best results are highlighted in <b>red</b> and <b>blue</b> respectively. . . . .	66
4.7	Quantitative results with <b>BD</b> and <b>DN</b> degradation models. The best and second best results are highlighted in <b>red</b> and <b>blue</b> respectively. . . . .	68
4.8	Average running time (s) and memory consumption (MB) comparison on Urban100 for $\times 4$ scale factor. . . . .	71
4.9	Perceptual index comparison of the proposed method with recent lightweight state-of-the-art methods on five datasets for $\times 4$ . The lower is better. All of the output SR images are provided officially. . . . .	71
5.1	Effect of different pooling methods for DiVA. Average PSNR on five benchmark datasets with scale factor $\times 4$ are shown. . . . .	83
5.2	Average PSNR obtained with DiVANet when using different attention mechanisms on five benchmark datasets (scale factor $\times 4$ ). . . . .	84
5.3	The results of adding DiVA in different networks. Average PSNR on five benchmark datasets with scale factor $\times 4$ are shown. . . . .	85

5.4	Average PSNR for a regular ResNet architecture (Baseline) vs one using the proposed feature banks on five benchmark dataset with $\times 4$ scale factor. . . . .	86
5.5	Average PSNR obtained on the ResNet baseline network, when placing the DiVA attention mechanism within ( <i>Baseline_in</i> ) or outside ( <i>Baseline_out</i> ) the residual blocks. Results are shown on five benchmark datasets and with a $\times 4$ scale factor. . . . .	87
5.6	Average PSNR/SSIM values for models with the same order of magnitude of parameters. Performance is shown for scale factors $\times 2$ , $\times 3$ and $\times 4$ with <b>BI</b> degradation model. The Multi-Adds is calculated corresponding to a $1280 \times 720$ HR image. The best and second best results are highlighted in <b>red</b> and <b>blue</b> respectively. . . . .	89
5.7	Quantitative results with <b>BD</b> and <b>DN</b> degradation models. Performance is shown for scale factor $\times 3$ . The best and second best results are highlighted in <b>red</b> and <b>blue</b> respectively. . . . .	91
5.8	Average running time (s) and memory consumption (MB) comparison on Urban100 for $\times 4$ . . . . .	96
5.9	Perceptual index comparison of the proposed methods with recent lightweight state-of-the-art methods on five datasets for $\times 4$ . The lower is better. All of the output SR images are provided officially. . . . .	97
A.1	Quantitative comparisons (average PSNR and SSIM) on SOTS for different methods. Best and second best performance are in <b>red</b> and <b>blue</b> colors, respectively. . . . .	105
A.2	Quantitative comparisons (average PSNR and SSIM) with state-of-the-art methods for JPEG compression artifact reduction on benchmark datasets. Best and second best performance are in <b>red</b> and <b>blue</b> colors, respectively. . . . .	108





# 1 Introduction

Over the past decades, digital images have become ubiquitous in our daily lives. Artistic photography, computer vision, remote sensing, astronomy, medical imaging, and microscopy are just a few of the numerous uses for images. In each case, the captured images produce a source from which we can observe or understand an object or a scene. Therefore, the demands for images with higher resolution have dramatically increased.

The resolution of a digital imaging system can be classified in four different ways: spatial resolution, spectral resolution, radiometric resolution, and temporal resolution, in which spatial resolution is of the greatest challenge. The spatial resolution of a digital imaging system is primarily defined by the pixel density in the image space, which is measured in pixels per unit area. Spatial resolution in the object space represents the level of spatial detail that can be discerned in an image; the higher the resolution, the more image details. For instance, a medical doctor in the neurology area can achieve better diagnosis by using higher-resolution Magnetic Resonance Imaging (MRI) images [92]. However, the high-resolution images cannot be obtained in many scenes because of the poor imaging sensor or acquisition device as well as the unsuitable environment. As a result, the images and scenes captured by such equipment are typically low resolutions, with sensor noise and unexpected artifacts.

There are several ways of increasing the spatial resolution of an image. From a hardware perspective, the direct method is to reduce the pixel size and thereby fit more pixel sensors onto the chip area. A reduced pixel size means that a reduced number of photons will hit the sensor element and that shot noise will occur resulting in degraded quality. Hence, there is a physical limit to reducing the size of the image sensor [90]. It would also be possible to make larger image sensors, but that would lead to larger and more expensive cameras, which might not be desirable. Other solutions would be increasing the size of the sensor chip at the cost of the increased capacitance of the system, which will result in a slower transfer rate or increasing the focal length of the camera lens, which would also result in larger and heavier cameras. Therefore, it is preferable to apply an algorithmic approach

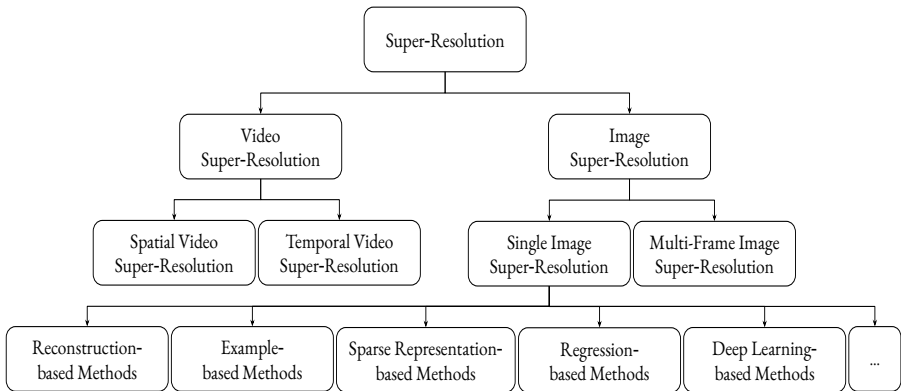


Figure 1.1 – The taxonomy of existing super-resolution techniques [17].

toward the increasing spatial resolution that doesn't depend on the development of new sensors and allows for the use of already existing image capture systems.

An alternative approach to hardware-based solutions for spatial resolution enhancement is to apply signal processing techniques to recover fine image details degraded or almost lost during image capture. These approaches are often referred to as super-resolution (SR) image reconstruction techniques. The field of super-resolution was established in the 80s and initial work was done in the frequency domain, using several under-sampled low-resolution images to reduce aliasing in fused high-resolution images. SR techniques attempt to recover high-resolution (HR) images from low-resolution (LR) images which remains an important yet challenging topic in image processing.

In general, as presented in Figure 1.1, existing SR techniques can be grouped into two categories according to the LR input and the reconstructed HR output, *i.e.*, video super-resolution (VSR) and image super-resolution (ISR). On the whole, VSR aims to improve the spatial resolution (known as spatial VSR) [114] or the frame rate (known as temporal VSR) [76] of the observed video. ISR can be further classified into multi-frame image super-resolution (MISR) [56, 74] and single image super-resolution (SISR) [6, 86]. MISR refers to reconstructing an HR image via fusing the complementary information in a series of correlated images of the same scene, while SISR generates an HR image from one LR observation. In terms of application scenarios, SISR is more practical than MISR and VSR because it is much less demanding on the input, which is one reason why SISR attracts wider attention. In this thesis, we particularly focus on the SISR problem which is further detailed in the next section.

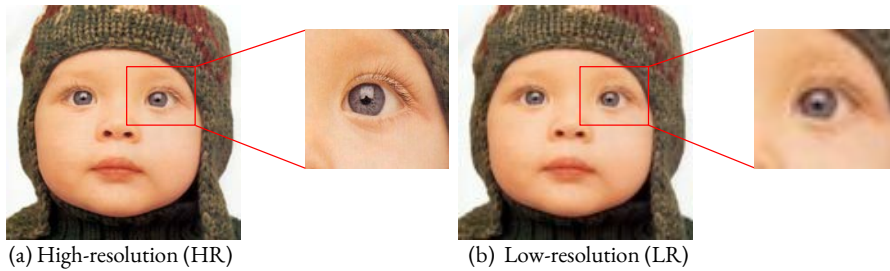


Figure 1.2 – Example of a low-resolution image compared to a high-resolution sample of the same scene.

### 1.1 Single Image Super-Resolution via Deep Learning

The basic principle of single image super-resolution (SISR) is to use one LR image of a scene, in order to create an image with a higher spatial resolution that conveys finer detail or content with higher frequencies than the LR image (see Figure 1.2). This offers an opportunity for overcoming resolution limitations in various computer vision and image understanding tasks, such as computer graphics [52, 112], medical imaging [4, 5, 28, 33, 44, 51, 113], security and surveillance [61, 102, 143]. Super-resolution techniques not only improve image perceptual quality but also help to improve the final accuracy of many computer vision tasks such as detection [3, 35, 101] and recognition [8, 97, 129], which shows the importance of this topic.

The problem of SISR is a highly ill-posed procedure since there are multiple different HR images with slight variations in camera angle, color, brightness, and other variables that may correspond to an identical LR image. Furthermore, there are fundamental uncertainties among the LR and HR data since the downsampling of different HR images may lead to a similar LR image, making this conversion a many-to-one process. To address this problem, a variety of classical SR methods have been proposed, such as reconstruction-based [15, 64, 108], example-based [45, 46], sparse representation [63, 68], and regression-based approaches [94, 142].

Since the great success at the Imagenet Large Scale Visual Recognition Competition (ILSVRC12) [55], Convolutional Neural Networks (CNNs) have become the main workhorse for most computer vision tasks such as motion analysis [26], image generation [31], and 3D recognition [78]. The powerful feature representation and end-to-end training paradigm of CNNs make them a promising approach to SISR.

The SISR task can generally be divided into three stages depicted in Figure 1.3:

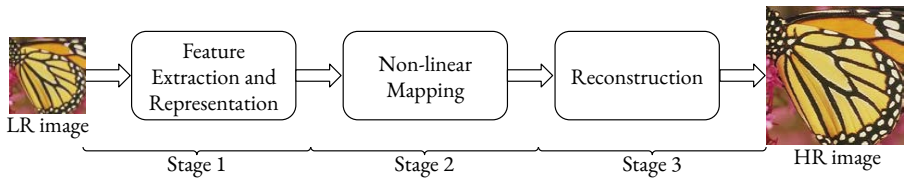


Figure 1.3 – Example of single image super-resolution (SISR) pipeline.

feature extraction and representation, non-linear mapping, and image reconstruction [23]. In classical models, it is time-consuming and inefficient to design an algorithm satisfying all these processes. On the contrary, CNNs can transfer the SISR task to an almost end-to-end framework incorporating all these three processes, which can greatly decrease manual and computing expenses [25]. Additionally, given the ill-posed nature of SISR which can lead to unstable and hard convergence on the results, CNNs can alleviate this issue through efficient network architectures and loss functions design. Moreover, modern GPU enables deeper and more complex CNN-based SR models to train fast, which shows greater representation power than traditional models. Therefore, Dong et al. [23] pioneered the field of SISR with neural networks, proposing *Super-Resolution Convolutional Neural Network* (SRCNN), a three-layer CNN to learn an end-to-end mapping function from an interpolated LR input to its corresponding HR output, which outperformed classical SR algorithms. Since SRCNN successfully applied CNN to SISR task, a great number of CNN-based SR methods have been proposed and shown promising results [6, 83, 84, 85, 86].

## 1.2 Objectives

Single image super-resolution is a notoriously challenging ill-posed problem which aims at restoring the lost structures and textures from LR images. The main objective of this thesis is to produce algorithms for modeling the process of SISR that can obtain images with resolution beyond the limit of imaging systems, thereby benefiting the subsequent analysis and understanding tasks such as detection and recognition.

Recently, with the rapid development of deep learning techniques, CNNs have been widely explored in SISR and obtained significant performance. Despite their remarkable performance, most existing CNN-based SR methods still have some drawbacks. Throughout this thesis, we develop different algorithms to tackle the following problems:

- **High-frequency enhancement.** The output feature maps of a convolutional layer can be seen as a mixture of information at lower and higher frequencies each of which contains structures and textures of different complexity. For example, the lower frequency information is composed of global structures and textures that can directly be forwarded to the final HR output without substantial computations. The higher frequency information consists of fine details where more complex restoring functions are expected. At this point, most existing CNN-based SR methods overlook the fact that most of the low-frequency information is already contained in the LR input. As a result, these models spend the same amount of computation treating low- and high-frequency information and lack flexible modulation ability in dealing with them, which ends up the representational ability of the network.
- **Feature degradation and model complexity.** Thanks to the increase in capacity of CNNs in depth and width, their performance has greatly improved. The increase of depth brings benefits in terms of representation power, but at the same time may result in a loss of low-level feature information, since the features gradually disappear as the network depth increases. Although authors in [145] introduce various skip connections and concatenation operations between intermediate layers and deep layers to fuse different levels of features, the extreme connectivity pattern in their network not only hinders their scalability when using large widths or depths but also increases computational demands and memory consumption dramatically, hence limiting the use of modern architectures in real-world scenarios. Therefore, it is of crucial importance to design a lightweight network architecture that effectively computes multi-level feature representations for restoring high-quality HR images within the network.
- **Scale arbitrary SISR.** SISR has seen its applications in diverse real-life scenarios and users. Therefore, it is necessary to develop a flexible and universal scale arbitrary SR model that can be adapted to any scale, including non-integer scale factors. Currently, most CNN-based SR models can only be applied to one or a limited number of upsampling factors. Although a few scales arbitrary SR methods have been proposed, they tend to lack the flexibility to be used and the simplicity to be implemented. Thus, exploring a CNN-based accurate scale arbitrary SR model as simple and flexible as bicubic is crucial to the spread of SISR technology.
- **Attention mechanism.** Attention mechanisms have demonstrated great benefits at improving the performance of deep models for computer vision tasks. Recently, researchers have devoted great efforts to expand the application

of attention mechanisms to SISR. Taking efficiency into account, the most popular attention mechanism for SR networks is squeeze-and-excitation (SE) attention [41] used for high-level vision problems. It computes channel attention with the help of 2D global pooling and provides notable performance gains at a considerably low computational cost. However, the SE attention only considers encoding inter-channel information but neglects the importance of spatial information, which is essential for enhancing image details in low-level vision tasks.

### 1.3 Thesis Outline and Contributions

In this thesis, we analyze various exiting CNN-based SR methods, identifying issues that hinder their performance, and propose new solutions to them in each chapter. Hence, each chapter corresponds to an article either published or submitted in a journal or conference:

- **Chapter 2: Background.** In this chapter, we first give the problem definition and review the mainstream datasets and evaluation metrics used for performance comparison in this thesis. Then, we provide a literature review including state-of-the-art models related to the proposed methods.
- **Chapter 3: Frequency-based Enhancement Network for Efficient Super-Resolution.** In this chapter, we focus on high-frequency enhancement. Specifically, we introduce a novel frequency-based enhancement block (FEB) which is able to separate features into low and high frequencies while also enabling efficient communication among them. Since low frequencies are preserved by downsampling operations and thus can be recovered directly from the input, FEB assigns more computational capacity to high frequencies, resulting in more accurate features that improve reconstruction quality. The proposed block design is simple and generic and can be used as a direct replacement of commonly used SR blocks with no need to change network architectures. We experimentally show that when replacing SR blocks with the FEB we consistently improve the reconstruction error, while reducing the number of parameters in the model. Moreover, we propose a lightweight SR model named frequency-based enhancement network based on FEB that performs favorably against the state-of-the-art SR algorithms in terms of visual quality, memory footprint, and inference time.

*This work has been submitted to IEEE Access (Parichehr Behjati, Pau Rodriguez, Carles Fernandez, Armin Mehri, F. Xavier Roca, Seiichi Ozawa, and Jordi Gonzalez, 2022)*

- **Chapter 4: OverNet: Lightweight Multi-Scale Super-Resolution with Overscaling Network.** In this chapter, we introduce OverNet, a deep but lightweight convolutional network to solve SISR at arbitrary scale factors with a single model. OverNet consists of two main parts: a lightweight feature extractor and an overscaling module for reconstruction. The feature extractor follows a novel recursive framework of skip and dense connections to reduce low-level feature degradation. The overscaling module is a new inductive bias that generates an accurate SR image by internally constructing an overscaled intermediate representation of the output features. Finally, to solve the problem of reconstruction at arbitrary scale factors, we introduce a novel multi-scale loss by downsampling the output at multiple super-resolution factors and we minimize the reconstruction error in all of them. Experiments show that our proposal outperforms previous state-of-the-art approaches while maintaining relatively low computation and memory requirements.

*This work has been published at WACV (Parichehr Behjati, Pau Rodriguez, Armin Mehri, Carles Fernandez, Isabelle Hupont, and Jordi Gonzalez, 2021)*

- **Chapter 5: Single Image Super-Resolution Based on Directional Variance Attention Networks.** This chapter presents a directional variance attention network, a computationally efficient yet accurate network for SISR. This network leverages a novel directional variance attention specifically optimized for SR, to enhance features in different channels and spatial regions. Such a mechanism allows the network to focus on more informative features and improve discriminative capabilities. Moreover, we introduce a novel procedure for using attention mechanisms together with residual blocks, following two independent but parallel computational paths. The idea is to hierarchically aggregate their respective contributions across the network to facilitate the preservation of finer details. Extensive experiments on a variety of public datasets demonstrate the superiority of the proposed architecture over state-of-the-art models, in terms of both quantitative and visual quality.

*This work has been submitted to Pattern Recognition (Parichehr Behjati, Pau Rodriguez, Carles Fernandez, Isabelle Hupont, Armin Mehri, and Jordi Gonzalez, 2022)*

- **Chapter 6: Conclusion.** The last chapter concludes the work developed in this thesis and proposes further directions for research in SISR problem.





## 2 Background

This thesis mainly focuses on addressing and solving single image super-resolution problem with deep learning. In this chapter, we first detail the problem definition. Next, we introduce some related works, including benchmark datasets, assessment methods, SISR frameworks, upsampling methods, and optimization objectives. Then, we provide a literature review including state-of-the-art methods related to the proposed methods.

### 2.1 Problem Definition

Single image super-resolution refers to the process of reconstructing an HR image from its LR version. Generally, the LR image  $I_x$  is modeled as the output of the following degradation:

$$I_x = \mathcal{D}(I_y; \delta), \quad (2.1)$$

where  $\mathcal{D}$  denotes a degradation mapping function,  $I_y$  is the corresponding HR image and  $\delta$  is the parameters of the degradation process (*e.g.*, the scaling factor or noise). In practice, the degradation process (*i.e.*,  $\mathcal{D}$  and  $\delta$ ) is unknown and only LR images are provided. In this case, we are required to recover an HR approximation  $\hat{I}_y$  of the ground truth HR image  $I_y$  from the LR image  $I_x$ , following:

$$\hat{I}_y = \mathcal{F}(I_x; \theta), \quad (2.2)$$

where  $\mathcal{F}$  is the super-resolution model and  $\theta$  denotes the parameters of  $\mathcal{F}$ .

Although the degradation process is unknown and can be affected by various factors (*e.g.*, compression artifacts, anisotropic degradations, sensor noise, and speckle noise), we are trying to model the degradation mapping. Most works directly model the degradation as a single downsampling operation, as follows:

$$\mathcal{D}(I_y; \delta) = (I_y) \downarrow_s, \{s\} \subset \delta, \quad (2.3)$$

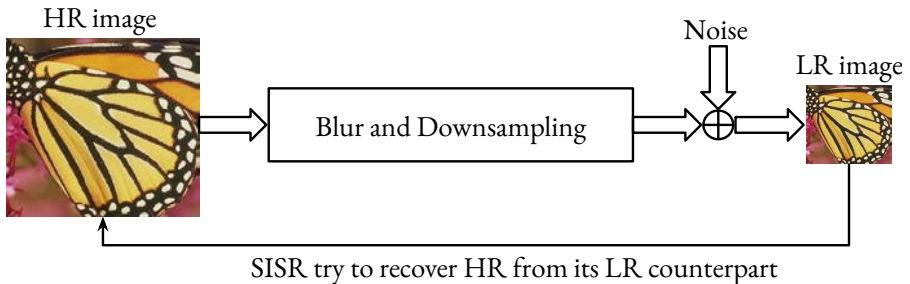


Figure 2.1 – Downsampling and upsampling in super-resolution. Noise is added to simulate realistic degradation within an image.

where  $\downarrow_s, \{s\}$  is a downsampling operation with the scaling factor  $s$ . As a matter of fact, most datasets for generic SR are built based on this pattern, and the most commonly used downsampling operation is bicubic interpolation with antialiasing. However, there are other works [145] modeling the degradation as a combination of several operations:

$$\mathcal{D}(I_y; \delta) = (I_y \otimes \kappa) \downarrow_s + n_\zeta, \{\kappa, s, \zeta\} \subset \delta, \quad (2.4)$$

where  $I_y \otimes \kappa$  represents the convolution between a blur kernel  $\kappa$  and the HR image  $I_y$ , and  $n_\zeta$  is some additive white gaussian noise with standard deviation  $\zeta$ . Compared to the naive definition of equation 2.3, the combinative degradation pattern defined in equation 2.4 and Figure 2.1 is closer to real-world cases and has been shown to be more beneficial for SR. To this end, the objective of SR is as follows:

$$\hat{\theta} = \operatorname{argmin}_\theta \mathcal{L}(\hat{I}_y, I_y) + \lambda \Phi(\theta), \quad (2.5)$$

where  $\mathcal{L}(\hat{I}_y, I_y)$  represents the loss function between the generated HR image  $\hat{I}_y$  and the ground truth image  $I_y$ ,  $\Phi(\theta)$  is the regularization term and  $\lambda$  is the trade-off parameter.

## 2.2 Benchmark Datasets

Data is always essential for data-driven models, especially the deep learning-based SR models, to achieve promising reconstruction performance. Nowadays, industry and academia have launched several available datasets for SISR.



Figure 2.2 – Representative test images from six super-resolution datasets used for comparing and evaluating algorithms.

### 2.2.1 Training and Test Datasets

Today there are a variety of datasets available for image SR, which greatly differ in image amounts, quality, resolution, diversity, etc. Among them, DIV2K [117] is the most widely used dataset for model training, which is a high-quality dataset that contains 800 training images, 100 validation images, and 100 test images. Meanwhile, there are also many test datasets than can be used to effectively test the performance of the models. The representative image from all the datasets is shown in Figure 2.2.

- **Set5** [9] is a classical dataset and only contains five test images of a baby, bird, butterfly, head, and a woman.
- **Set14** [138] consists of more categories as compared to Set5. however, the number of images are still low *i.e.* 14 test images.
- **B100** [2] is another classical dataset having 100 test images. The dataset is composed of a large variety of images ranging from natural images to object-specific such as plants, people, food, etc.
- **Urban100** [45] is a relatively more recent dataset. The number of images is the same as B100. however, the composition is entirely different. The focus of the photographs is on human-made structures *i.e.* urban scenes.

- **Manga109** [82] is the latest addition for evaluating super-resolution algorithms. The dataset is a collection of 109 test images of a manga volume. These mangas were professionally drawn by Japanese artists and were available only for commercial use between the 1970s and 2010s.

### 2.3 Assessment Methods

The image quality assessment (IQA) can be generally divided into objective methods and subjective methods. Objective methods commonly use a specific formulation to compute the results, which are simple and fair, thus becoming the mainstream assessment method in SISR. However, they can only reflect the recovery of image pixels from a numerical point of view and are difficult to accurately measure the true visual effect of the image. In contrast, subjective methods are always based on human subjective judgments and more related to evaluating the perceptual quality of the image. Based on the pros and cons of the two types of methods mentioned above, several assessment methods are briefly introduced in the following with respect to the aspects of image reconstruction accuracy, image perceptual quality, and reconstruction efficiency.

#### 2.3.1 Image Reconstruction Accuracy

The assessment methods applied to evaluate image reconstruction accuracy are also called *Distortion measures*, which are full-reference. Specifically, given a distorted image  $\hat{x}$  and a ground-truth reference image  $x$ , full-reference distortion quantifies the quality of  $\hat{x}$  by measuring its discrepancy to  $x$  using different algorithms.

##### Peak Signal-to-Noise Ratio

Peak Signal-to-Noise Ratio (PSNR) [126] is the most widely used IQA method in the SISR field, which can be easily defined via the mean squared error (MSE) between the ground truth image  $I_y \in \mathbb{R}^{H \times W}$  and the reconstructed image  $\hat{I}_y \in \mathbb{R}^{H \times W}$ :

$$MSE = \frac{1}{HW} \sum_{i=0}^{H-1} \sum_{j=0}^{W-1} (I_y(i, j) - \hat{I}_y(i, j))^2, \quad (2.6)$$

$$PSNR = 10 \cdot \log_{10} \left( \frac{MAX^2}{MSE} \right), \quad (2.7)$$

where  $MAX$  is the maximum possible pixel of the image. Since PSNR is highly related to MSE, a model trained with the MSE loss will be expected to have high

PSNR scores. Although higher PSNR generally indicates that the construction is of higher quality, it just considers the per pixel MSE, which makes it fails to capture the perceptual differences [125].

### Structural Similarity Index Measure

Structural Similarity Index Measure (SSIM) [126] is another popular assessment method that measures the similarity between two images on perceptual basis, including structures, luminance, and contrast. Different from PSNR, which calculates absolute errors on the pixel-level, SSIM suggests that there exists strong inter-dependencies among the pixels that are spatially close. These dependencies carry important information related to the structures perceptually. Thus the SSIM can be expressed as a weighted combination of three comparative measures:

$$\begin{aligned}
 SSIM(\hat{I}_y, I_y) &= (l(\hat{I}_y, I_y))^\alpha \cdot c(\hat{I}_y, I_y)^\beta \cdot s(\hat{I}_y, I_y)^\gamma \\
 &= \frac{(2\mu_{\hat{I}_y} \mu_{I_y} + c_1)(2\sigma_{\hat{I}_y I_y} + c_2)}{(\mu_{\hat{I}_y}^2 + \mu_{I_y}^2 + c_1)(\sigma_{\hat{I}_y}^2 + \sigma_{I_y}^2 + c_1)},
 \end{aligned} \tag{2.8}$$

where  $l$ ,  $c$ , and  $s$  represents luminance, contrast, and structure between  $\hat{I}_y$  and  $I_y$ , respectively,  $\mu_{\hat{I}_y}$ ,  $\mu_{I_y}$ ,  $\sigma_{\hat{I}_y}^2$ ,  $\sigma_{I_y}^2$ , and  $\sigma_{\hat{I}_y I_y}$  are the average( $\mu$ ) / variance ( $\sigma^2$ ) / covariance ( $\sigma$ ) of the corresponding items. A higher SSIM indicates higher similarity between two images, which has been widely used due to its convenience and stable performance on evaluating the perceptual quality.

### 2.3.2 Image Perceptual Quality

Since the visual system of humans is complex and concerns many aspects to judge the differences between two images, *i.e.*, the textures and flow inside the images, methods which pursue absolutely similarity differences (PSNR/SSIM) will not always perform well. Although distortion measures have been widely used, the improvement in reconstruction accuracy is not always accompanied by an improvement in visual quality. In fact, researchers have shown that the distortion and perceptual quality are at odds with each other in some cases [10]. The image perceptual quality of an image  $\hat{x}$  is defined as the degree to which it looks like a natural image, which has nothing to do with its similarity to any reference image.

#### Natural Image Quality Evaluator

Natural Image Quality Evaluator (NIQE) [87] is a completely blind image quality assessment method. Without the requirement of knowledge about anticipated

distortions in the form of training examples and corresponding human opinion scores, NIQE only makes use of measurable deviations from statistical regularities observed in natural images. It extracts a set of local (quality-aware) features from images based on a natural scene statistic (NSS) model, then fits the feature vectors to a multivariate Gaussian (MVG) model. The quality of a test image is then predicted by the distance between its MVG model and the MVG model learned from a natural image:

$$D(v_1, v_2, \Sigma_1, \Sigma_2) = \sqrt{((v_1 - v_2)^T (\frac{\Sigma_1 + \Sigma_2}{2})^{-1} (v_1 - v_2))}, \quad (2.9)$$

where  $v_1, v_2$  and  $\Sigma_1, \Sigma_2$  are the mean vectors and covariance matrices of the HR and SR image's MVG model. Notice that, a higher NIQE index indicates lower image perceptual quality.

### Ma

Ma et al. [79] proposed a learning-based no-reference image quality assessment. It is designed to focus on SR images, while other learning-based methods are applied to images degraded by noise, compression, or fast fading rather than SR. It learns from perceptual scores based on human subject studies involving a large number of SR images. And then it quantifies the SR artifacts through three types of statistical properties, *i.e.*, local/global frequency variations and spatial discontinuity. Then these features are modeled by three independent learnable regression forests respectively to fit the perceptual scores of SR images,  $\hat{y}_n (n = 1, 2, 3)$ . The final predicted quality score is  $\hat{y}_n = \sum_n \lambda_n \cdot \hat{y}_n$ , and the weight  $\lambda$  is learned by minimizing  $\lambda^* = \operatorname{argmin}_y (\sum_n \lambda_n \cdot \hat{y}_n - y)^2$ .

Ma performs well on matching the perceptual scores of SR images, but it is still limited compared with other learning-based no-reference methods since it can only assess the quality degradation arising from the distortion types on which they have been trained.

### Perceptual Index

In the 2018 PIRM Challenge on Perceptual Image Super-Resolution [11], perception index (PI) is first proposed to evaluate the perceptual quality. It is a combination of the no-reference image quality measures Ma and NIQE:

$$PI = \frac{1}{2}((10 - Ma) + NIQE). \quad (2.10)$$

A lower PI indicates better perceptual quality. This is a new image quality evaluation standard, which has been greatly promoted and used in recent years.

### 2.3.3 Reconstruction Efficiency

Although designing deeper networks is the easiest way to obtain better reconstruction performance, it cannot be ignored that these models will also bring more parameters, execution time, and computational costs. In order to broaden the practical application of SISR, we need to consider the trade-off between the model performance and model complexity. Therefore, it is important to evaluate the reconstruction efficiency by the following basic assessments.

- **Model size:** The model size is related to the storage that the devices need to store the data. A model containing more parameters is harder for the device with limited hardware to run it. Therefore, building lightweight models is conducive to the promotion and application of the algorithm. Among all the indicators, the parameter quantity of the model is the most intuitive indicator to measure the model size.
- **Execution Time:** Usually, a lightweight model tends to require a short execution time, but the emergence of complex strategies such as the attention mechanism has broken this balance. In other words, when some complex operations are introduced into the model, a lightweight network may also require a long execution time. Therefore, it is critically important to evaluate the execution time of the model.
- **Multi-Adds:** The number of multiply-accumulate operations, or Multi-Adds, is always used to measure the model computation since operations in the CNN model are mainly multiplications and additions. The value of Multi-Adds is related to the speed or the time needed to run the model. In summary, the trade-off between the model performance and model complexity is still need to be concerned.

## 2.4 Super-resolution Frameworks

Since image super-resolution is an ill-posed problem, how to perform upsampling (*i.e.*, generating HR output from LR input) is the key problem. Although the architectures of existing models vary widely, they can be attributed to four model frameworks (as depicted in Figure 2.3), based on the employed upsampling operations and their locations in the model.



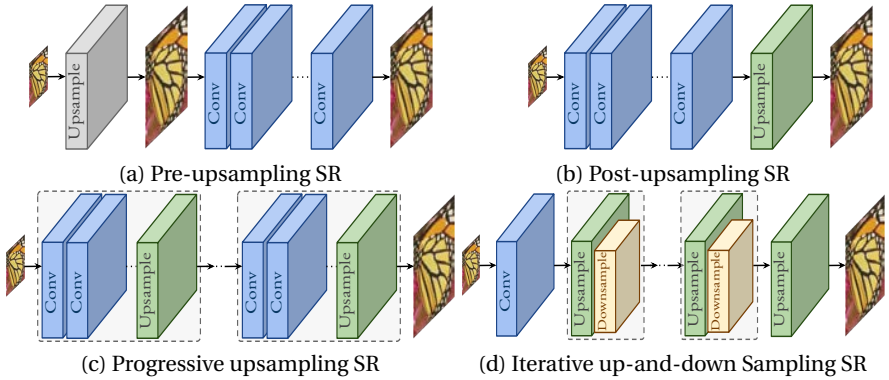


Figure 2.3 – Super-resolution model frameworks based on deep learning. The cube size represents the output size. The gray ones denote predefined upsampling, while the green, yellow and blue ones indicate learnable upsampling, downsampling and convolutional layers, respectively. And the blocks enclosed by dashed boxes represent stackable modules [127].

### 2.4.1 Pre-upsampling Super-resolution

On account of the difficulty of directly learning the mapping from low-dimensional space to high-dimensional space, utilizing traditional upsampling algorithms to obtain higher-resolution images and then refining them using deep neural networks is a straightforward solution. Thus, Dong et al. [23] firstly adopt the pre-upsampling SR framework (as Figure 2.3(a) shows) and proposed SRCNN to learn an end-to-end mapping from interpolated LR images to HR images. Specifically, the LR images are upsampled to coarse HR images with the desired size using traditional methods (*e.g.*, bicubic interpolation), then deep CNNs are applied on these images for reconstructing high-quality details.

Since the most difficult upsampling operation has been completed, CNNs only need to refine the coarse images, which significantly reduces the learning difficulty. In addition, these models can take interpolated images with arbitrary sizes and scaling factors as input, and give refined results with comparable performance to single-scale SR models [52]. Thus it has gradually become one of the most popular frameworks [53, 105, 110, 111]. However, the predefined upsampling often introduces side effects (*e.g.*, noise amplification and blurring), and since most operations are performed in high-dimensional space, the cost of time and space is much higher than other frameworks [24].

### 2.4.2 Post-upsampling Super-resolution

In order to improve the computational efficiency and make full use of deep learning technology to increase resolution automatically, researchers proposed to perform most computation in low-dimensional space by replacing the predefined upsampling with end-to-end learnable layers integrated at the end of the models. In the pioneer works [24, 104] of this framework (see Figure 2.3(b)), namely post-upsampling, the LR input images are fed into deep CNNs without increasing resolution, and end-to-end learnable upsampling layers are applied at the end of the network.

Since the feature extraction process with huge computational cost only occurs in low-dimensional space and the resolution increases only at the end, the computation and spatial complexity are much reduced. Therefore, this framework also has become one of the most mainstream frameworks [60, 71, 118]. These models differ mainly in the learnable upsampling layers, CNN structures, learning strategies, etc.

### 2.4.3 Progressive Upsampling Super-resolution

Although the post-upsampling SR framework has immensely reduced the computational cost, it still has some shortcomings. On the one hand, the upsampling is performed in only one step, which greatly increases the learning difficulty for large scaling factors. On the other hand, each scaling factor requires training an individual SR model, which cannot cope with the need for multi-scale SR. To address these drawbacks, a progressive upsampling framework is adopted by the Laplacian pyramid SR network (LapSRN) [57] (see Figure 2.3(c)). Specifically, the models under this framework are based on a cascade of CNNs and progressively reconstruct higher-resolution images. At each stage, the images are upsampled to higher resolution and refined by CNNs. Other works such as MS-LapSRN [58] and progressive SR (ProSR) [124] also adopt this framework and achieve relatively high performance. In contrast to the LapSRN and MS-LapSRN using the intermediate reconstructed images as the base images for subsequent modules, the ProSR keeps the main information stream and reconstructs intermediate-resolution images by individual heads.

By decomposing a difficult task into simple tasks, the models under this framework greatly reduce the learning difficulty, especially with large factors, and also cope with the multi-scale SR without introducing overmuch spatial and temporal cost. However, the models under this framework also encounter some problems such as complicated model designing for multiple stages and the training stability.

### 2.4.4 Up-and-down Sampling Super-resolution

In order to better capture the mutual dependency of LR-HR image pairs, an efficient iterative procedure named back-projection [48] is incorporated into SISR [116]. This SR framework, namely iterative up-and-down sampling SR (depicted in Figure 2.3(d)), tries to iteratively apply back-projection refinement, *i.e.*, computing the reconstruction error then fusing it back to tune the HR image intensity. Specifically, Haris et al. [34] exploit iterative up-and-down sampling layers and propose DBPN, which connects upsampling and downsampling layers alternately and reconstructs the final HR result using all of the intermediate reconstructions. Similarly, the SRFBN [69] employs an iterative up-and-down sampling feedback block with more dense skip connections and learns better representations.

The models under this framework can better exploit the deep relationships between LR-HR image pairs and thus provide higher-quality reconstruction results. Nevertheless, the design criteria of the back-projection modules are still unclear. Since this mechanism has just been introduced into deep learning-based SR, the framework has great potential and needs further exploration.

## 2.5 Upsampling Methods

In addition to the upsampling positions in the model, how to perform upsampling is of great importance. Although there have been various traditional upsampling methods [115, 132], making use of CNNs to learn end-to-end upsampling has gradually become a trend. In this section, we will introduce some traditional interpolation-based algorithms and deep learning-based upsampling layers.

### 2.5.1 Interpolation-based Upsampling

Image interpolation, a.k.a. image scaling, refers to resizing digital images and is widely used by image-related applications. The traditional interpolation methods include nearest-neighbor interpolation, bilinear, bicubic interpolation, etc. Since these methods are interpretable and easy to implement, some of them are still widely used in CNN-based SR models.

- **Nearest-neighbor interpolation:** The nearest-neighbor interpolation is a simple and intuitive algorithm. It selects the value of the nearest pixel for each position to be interpolated regardless of any other pixels. Thus, this method is very fast but usually produces blocky results of low quality.
- **Bilinear interpolation:** The bilinear interpolation first performs linear interpolation on one axis of the image and then performs on the other axis.

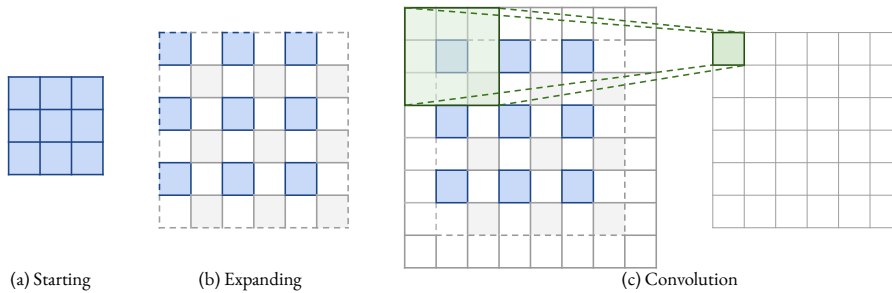


Figure 2.4 – Transposed convolution layer. The blue boxes denote the input, and the green boxes indicate the kernel and the convolution output.

Since it results in a quadratic interpolation with a receptive field-sized  $2 \times 2$ , it shows better performance than nearest-neighbor interpolation while keeping a relatively fast speed.

- **Bicubic interpolation:** Similarly, the bicubic interpolation performs cubic interpolation on each of the two axes. Compared to bilinear interpolation, the bicubic interpolation takes  $4 \times 4$  pixels into account and results in smoother results with fewer artifacts but much lower speed. In fact, the bicubic interpolation with anti-aliasing is the mainstream method for building SR datasets (*i.e.*, degrading HR images to LR images), and is also widely used in the pre-upsampling SISR framework.

As a matter of fact, the interpolation-based upsampling methods improve the image resolution only based on its own image signals, without bringing any more information. Instead, they often introduce some side effects, such as computational complexity, noise amplification, blurring results. Therefore, the current trend is to replace the interpolation-based methods with learnable upsampling layers.

### 2.5.2 Learning-based Upsampling

In order to overcome the shortcomings of interpolation-based methods and learn upsampling in an end-to-end manner, transposed convolution layer, and sub-pixel layer are introduced into the SISR field.

#### Transposed Convolutional Layers

Transposed convolution layer, a.k.a. deconvolution layer [136, 137], tries to perform transformation opposite a normal convolution, *i.e.*, predicting the possible input

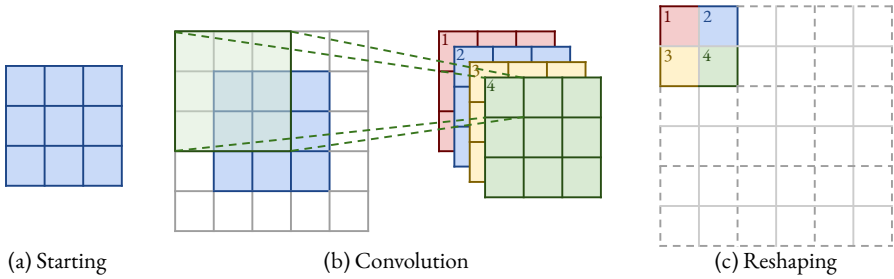


Figure 2.5 – Sub-pixel layer. The blue boxes denote the input, and the boxes with other colors indicate different convolution operations and different output feature maps.

based on feature maps sized like convolution output. Specifically, it increases the image resolution by expanding the image by inserting zeros and performing convolution. Taking  $\times 2$  SR with  $3 \times 3$  kernel as an example as depicted in Figure 2.4, the input is firstly expanded twice the original size, where the added pixel values are set to 0 (Figure 2.4(b)). Then a convolution with kernel sized  $3 \times 3$ , stride 1, and padding 1 is applied (Figure 2.4(c)). In this way, the input is upsampled by a factor of 2, in which case the receptive field is at most  $2 \times 2$ . Since the transposed convolution enlarges the image size in an end-to-end manner while maintaining a connectivity pattern compatible with vanilla convolution, it is widely used as an upsampling layer in SR models [34, 80, 118]. However, this layer can easily cause *uneven overlapping* on each axis [89], and the multiplied results on both axes further create a checkerboard-like pattern of varying magnitudes and thus hurt the SISR performance.

### Sub-Pixel Layer

The sub-pixel layer [104], another end-to-end learnable upsampling layer, performs upsampling by generating a plurality of channels by convolution and then reshaping them, as depicted in Figure 2.5. Within this layer, a convolution is firstly applied for producing outputs with  $s^2$  times channels, where  $s$  is the scaling factor (Figure 2.5(b)). Assuming the input size is  $h \times w \times c$ , the output size will be  $h \times w \times s^2 c$ . After that, the reshaping operation (a.k.a. shuffle [104]) is performed to produce outputs with size  $sh \times sw \times c$  (Figure 2.5(c)). In this case, the receptive field can be up to  $3 \times 3$ . Due to the end-to-end upsampling manner, this layer is also widely used by SR models [1, 6, 60]. Compared with transposed convolution layer, the sub-pixel layer has a larger receptive field, which provides more contextual information to

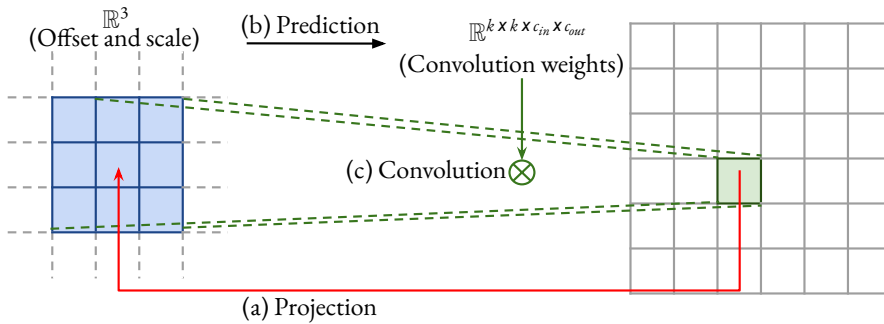


Figure 2.6 – Meta upscale module. The blue boxes denote the projection patch, and the green boxes and lines indicate the convolution operation with predicted weights.

help generate more realistic details. However, since the distribution of the receptive fields is uneven and blocky regions actually share the same receptive field, it may result in some artifacts near the boundaries of different blocks. On the other hand, independently predicting adjacent pixels in a blocky region may cause unsmooth outputs.

### Meta Upscale Module

The previous methods need to predefine the scaling factors, *i.e.*, training different upsampling modules for different factors, which is inefficient and not in line with real needs. Thus Hu et al. [42] propose meta upscale module (Figure 2.6), which firstly solves SR of arbitrary scaling factors based on meta-learning. Specifically, for each target position on the HR images, this module projects to a small patch on the LR feature maps (*i.e.*,  $k \times k \times c_{in}$ ), predicts convolution weights (*i.e.*,  $k \times k \times c_{in} \times c_{out}$ ) according to the projection offsets and the scaling factor by dense layers and perform convolution. In this way, the meta upscale module can continuously zoom in it with arbitrary factors by a single model. And due to the large amount of training data (multiple factors are simultaneously trained), the module can exhibit comparable or even better performance on fixed factors. Although this module needs to predict weights during inference, the execution time of the upsampling module only accounts for about 1% of the time of feature extraction [42]. However, this method predicts a large number of convolution weights for each target pixel based on several values independent of the image contents, so the prediction result may be unstable and less efficient when faced with larger magnifications.

To summarize, these learning-based layers have become the most widely used

upsampling methods. Especially in the post-upsampling framework, these layers are usually used in the final upsampling phase for reconstructing HR images based on high-level representations extracted in low-dimensional space, and thus achieve end-to-end SR while avoiding overwhelming operations in high-dimensional space.

## 2.6 Optimization Objective

Evaluation and parameter up-gradation are the important steps in all deep learning-based models. In this section, we will introduce the necessary procedures during the model training.

### 2.6.1 Learning Strategy

According to different strategies, the deep learning-based SR models can be mainly divided into supervised learning methods and unsupervised learning methods.

#### Supervised Learning

In supervised learning SISR, researchers compute the reconstruction error between the ground-truth image  $I_y$  and the reconstructed image  $\hat{I}_y$ :

$$\hat{\theta}_F = \operatorname{argmin}_{\mathcal{F}} \mathcal{L}(\hat{I}_y, I_y). \quad (2.11)$$

Alternatively, researchers may sometimes search for a mapping  $\phi$ , such as a pre-trained neural network, to transform the images or image feature maps to some other space and then compute the error:

$$\hat{\theta}_F = \operatorname{argmin}_{\mathcal{F}} \mathcal{L}(\Phi(\hat{I}_y, \phi(I_y))). \quad (2.12)$$

Among them,  $\mathcal{L}$  is the loss function which is used to minimize the gap between the reconstructed image and ground-truth image. According to different loss functions, the model can achieve different performances. Therefore, an effective loss function is also crucial for SISR.

#### Unsupervised Learning

In unsupervised learning SISR, the way of evaluation and parameter up-gradation is changing by different unsupervised learning algorithms. For example, ZSSR[105] uses the test image and its downscaling images with the data augmentation methods to build the training dataset and then applies the loss function to optimize the model. In CinCGAN[135], a model consists of two CycleGAN [149], where pa-

rameters are upgraded through optimizing the generator-adversarial loss, the cycle consistency loss, the identity loss, and the total variation loss together in each cycle.

### 2.6.2 Loss Functions

The loss function plays a critical role in the model's performance. Choosing the correct loss function for the problem in question helps the model learn the right set of features for optimal and faster convergence. Various loss functions have been proposed in the SISR problem, each targeting a specific problem and penalize a different aspect for the purpose of enhancing the resolution of image results. Deep learning models often use a weighted sum of more than one loss function to help the model focus on the different problems contributed by multiple loss functions simultaneously. In this section, we'll take a closer look at the loss functions used widely.

#### Pixel Loss

Pixel loss is the simplest and most popular type among loss functions in SISR, which aims to measure the difference between two images on pixel basis so that these two images can converge as close as possible. It mainly includes the L1 loss (*i.e.*, mean absolute error) and L2 loss (*i.e.*, mean square error):

$$\mathcal{L}_{L1}(\hat{I}_y, I_y) = \frac{1}{hwc} \sum_{i,j,k} \left| \hat{I}_y^{i,j,k} - I_y^{i,j,k} \right|, \quad (2.13)$$

$$\mathcal{L}_{L2}(\hat{I}_y, I_y) = \frac{1}{hwc} \sum_{i,j,k} \left( \hat{I}_y^{i,j,k} - I_y^{i,j,k} \right)^2, \quad (2.14)$$

where  $h$ ,  $w$  and  $c$  are the height, width, and the number of channels of the image.

While L2 loss favors a high PSNR, L1 loss is believed to be more robust against outliers. Also, LAPSRN [57] used Charbonnier Loss as a loss function rather than L2 loss which deals better with outliers. It produced better and sharper images compared to images created with L2 loss, which generally have blur effects.

#### Content Loss

In order to evaluate the perceptual quality of images, the content loss is introduced into SR [50]. Specifically, it measures the semantic differences between images using a pre-trained image classification network. Denoting this network as  $\phi$  and the extracted high-level representations on  $l$ -th layer as  $\phi^{(l)}(I)$ , the content loss



is indicated as the Euclidean distance between high-level representations of two images, as follows:

$$L_{content} = (I_y, \hat{I}_y; \phi, l) = \frac{1}{h_l w_l c_l} \sqrt{\sum_{i,j,k} (\phi_{i,j,k}^{(l)}(\hat{I}_y) - \phi_{i,j,k}^{(l)}(I_y))^2}, \quad (2.15)$$

where  $h_l$ ,  $w_l$  and  $c_l$  are the height, width and number of channels of the representations on layer  $l$ , respectively.

Essentially the content loss transfers the learned knowledge of hierarchical image features from the classification network  $\phi$  to the SR network. In contrast to the pixel loss, the content loss encourages the output image  $\hat{I}_y$  to be perceptually similar to the target image  $I_y$  instead of forcing them to match pixels exactly. Thus it produces visually more perceptible results and is also widely used in this field [12, 50, 60, 101, 122], where the VGG [106] and ResNet [38] are the most commonly used pre-trained CNNs.

### Texture Los

Texture loss is introduced by Gatys et al. [30] as an improvement over the content loss to capture the image style for the purpose of image style transfer. The texture loss helps to generate images that have the same style (*e.g.*, texture, color, contrast) as the desired HR image. It can be defined as the spatial correlation between different feature maps extracted from a pre-trained network  $\phi$ . The correlation between the feature maps is represented by the Gram matrix, which is obtained by calculating the inner product of the vectorized feature maps. The Gram matrix captures the tendency of features to co-occur in different spatial locations of the image.

### Adversarial Loss

In recent years, due to their powerful learning ability, the GANs [31] receive more and more attention and are introduced to various vision tasks. To be concrete, the GAN consists of a generator performing generation (*e.g.*, text generation, image transformation), and a discriminator which takes the generated results and instances sampled from the target distribution as input and discriminates whether each input comes from the target distribution. During training, two steps are alternately performed: (a) fix the generator and train the discriminator to better discriminate, (b) fix the discriminator and train the generator to fool the discriminator. Through adequate iterative adversarial training, the resulting generator can produce outputs consistent with the distribution of real data, while the discriminator cannot distinguish between the generated data and real data.

In terms of super-resolution, it is straightforward to adopt adversarial learning, in which case we only need to treat the SR model as a generator and define an extra discriminator to judge whether the input image is generated or not. Therefore, Ledig et al. [60] firstly propose SRGAN using adversarial loss based on cross-entropy, as follows:

$$L_{gan\_ce\_g}(\hat{I}_y; D) = -\log D(\hat{I}_y), \quad (2.16)$$

$$L_{gan\_ce\_d}(\hat{I}_y, I_s; D) = -\log D(I_s) - \log(1 - D(\hat{I}_y)), \quad (2.17)$$

where  $L_{gan\_ce\_g}$  and  $L_{gan\_ce\_d}$  denote the adversarial loss of the generator (*i.e.*, the SR model) and the discriminator  $D$  (*i.e.*, a binary classifier), respectively, and  $I_s$  represents images randomly sampled from the ground truths. Besides, Sajjadi et al. [101] also adopts the similar adversarial loss in Enhancenet.

Furthermore, Wang et al. [124] and Yuan et al. [135] use adversarial loss based on least square error for more stable training process and higher quality results [81], given by:

$$L_{gan\_ls\_g}(\hat{I}_y; D) = (D(\hat{I}_y) - 1)^2, \quad (2.18)$$

$$L_{gan\_ls\_d}(\hat{I}_y, I_s; D) = (D(\hat{I}_y))^2 + (D(I_s) - 1)^2. \quad (2.19)$$

### 2.6.3 Other Improvements

In addition to the learning strategies, there are other techniques further improving SR models such as:

- **Context-wise network fusion.** Context-wise network fusion (CNF) [98] refers to a stacking technique fusing predictions from multiple SR networks. To be concrete, they train individual SR models with different architectures separately, feed the prediction of each model into individual convolutional layers, and finally sum the outputs up to be the final prediction result. Within this CNF framework, the final model constructed by the lightweight SRCNN [23] achieves comparable performance with state-of-the-art models with acceptable efficiency [98].
- **Data augmentation.** Data augmentation is one of the most widely used techniques for boosting performance with deep learning. For image super-resolution, some useful augmentation options include cropping, flipping, scaling, rotation and color jittering [6, 57, 71, 84, 110]. In addition, Bei et al. [7] also randomly shuffle RGB channels, which not only augments data but also alleviates color bias caused by the dataset with color unbalance.

- **Network interpolation.** PSNR-based models produce images closer to ground truths but introduce blurring problems, while GAN-based models bring better perceptual quality but introduce unpleasant artifacts (*e.g.*, meaningless noise-making images more realistic). In order to better balance the distortion and perception, Wang et al. [123] propose a network interpolation strategy. Specifically, they train a PSNR-based model and train a GAN-based model by fine-tuning, then interpolate all the corresponding parameters of both networks to derive intermediate models. By tuning the interpolation weights without retraining networks, they produce meaningful results with much fewer artifacts.
- **Self-ensemble.** Self-ensemble, a.k.a. enhanced prediction [116], is an inference technique commonly used by SR models. Specifically, rotations with different angles ( $0^\circ, 90^\circ, 180^\circ, 270^\circ$ ) and horizontal flipping are applied on the LR images to get a set of 8 images. Then these images are fed into the SR model and the corresponding inverse transformation is applied to the reconstructed HR images to get the outputs. The final prediction result is conducted by the mean [71, 116, 124, 144] or the median [105] of these outputs. In this way, these models further improve performance.

## 2.7 Most Related CNN-based Frameworks for SISR

In recent years, image super-resolution models based on deep learning have received more and more attention and achieved state-of-the-art performance. In previous sections, we decompose SR models into specific components, including model frameworks (Section 2.4), upsampling methods (Section 2.5), and optimization objective (Section 2.6). As a matter of fact, most of the state-of-the-art SR models today can basically be attributed to a combination of multiple strategies we summarize above. Here, we focus our discussion on the deep learning-based SR approaches that are most related to our works.

### 2.7.1 Evolution of Architectures

Recently, CNN-based methods have been extensively studied in image SR, due to their strong nonlinear representational power. Generally, such methods cast SR as an image-to-image regression problem, and learn an end-to-end mapping from LR to HR directly. Dong et al. [23] pioneered the field of SISR with neural networks, proposing SRCNN, a three-layer CNN which outperformed traditional algorithms. In SRCNN, researchers found that better reconstruction performance can be obtained by adding more convolutional layers to increase the receptive field.

However, directly stacking the layers will cause vanishing/exploding gradients and degradation problem [36].

In ResNet, He et al. [38] proposed a residual learning framework, where a residual mapping is desired instead of fitting the whole underlying mapping. In SISR, as LR image and HR image share most of the same information, it is easy to explicitly model the residual image between LR and HR images. Residual learning enables deeper networks and remits the problem of gradient vanishing and degradation. With the help of residual learning, Kim et al. [52] first pushed the depth of SR network to 20, outperforming SRCNN by a large margin. For the convenience of network design, the residual block [38] has gradually become the basic unit in the network structure. Therefore, Ledig et al. [60] employed residual blocks to construct a deeper network (SRResNet) for image SR, which was further improved by EDSR [71] and MDSR [71] by removing unnecessary modules (*e.g.*, batch normalization) from the residual blocks. By using effective building modules, image SR networks became deeper and yielded better performance. Among them, Zhang et al. [144] proposed a residual in residual structure to form a very deep network (over 400 convolutional layers), and achieved state-of-the-art performance. Later, in order to employ hierarchical features from all the convolutional layers in deep networks, dense blocks started being employed in several SR architectures [111, 118]. More recently, Zhang et al. [145] and Liu et al. [72] also used dense and residual connections in RDN and RFANet to utilize information from the whole feature hierarchy. Although these existing deep learning-based approaches have made considerable progress to improve SISR performance, they demand substantial memory and computational resources. This makes modern architectures less applicable in practice.

Numerous lightweight models have been proposed to alleviate the aforementioned computational burden. For example, DRCN [53] was the first to apply recursive algorithm to SISR to reduce the number of parameters by reusing them multiple times. Tai et al. [110] and Ahn et al. [1] improved DRCN by combining the recursive and residual network schemes in order to achieve better performance with even fewer parameters. Likewise, Behjati et al. [6] and Jiang et al. [49] also joined residual connections and recursive layers to reduce the computational cost. On the other hand, LapSRN [57] employed a pyramidal framework to increase the image size gradually. By doing so, LapSRN effectively performed SISR on extremely low-resolution cases. Chu et al. [20] introduced Neural Architecture Search (NAS) strategies to automatically build an SR model given certain constraints. Meanwhile, Hui et al. [47] proposed an information multi-distillation block that extracted features at a granular level with the channel splitting strategy which was further improved in [72]. More recently, Luo et al. [77] proposed lattice blocks that applied so-called butterfly structures to combine residual blocks. Later, Xuehui Wang and

Chen. [130] proposed an attentive feature block to utilize auxiliary features of previous layers for facilitating features learning of the current layer. Li et al. [67] proposed a linearly-assembled pixel-adaptive regression network, which casts the direct LR to HR mapping learning into a linear coefficient regression task. Recently, to simplify the challenges of directly super-resolving details, some authors adopted the progressive structure to reconstruct HR images in a stage-by-stage upscaling manner [70, 148]. Although all the aforementioned works demonstrate that lightweight SR networks are capable of providing good trade-offs between performance and number of parameters, there is still room for improvement in terms of performance.

### 2.7.2 Frequency-based Networks

It is well-known that high-frequency information (*e.g.* textures, edges) is significant for SISR. Although significant progress has been made, texture details of the LR images often tend to be smoothed in the super-resolved results since most existing CNN-based SR methods do not pay enough attention to the limited high-frequency information in the LR images. In SISR, the LR inputs and extracted features contain different types of information across channels, spaces, and layers, such as low- and high-frequency information each of which with different complexity. Lower-frequency information is composed of simpler structures and textures where simpler functions are required for reconstruction; higher-frequency information consists of complex structures and textures where more complex restoring functions are expected. At this point, most existing CNN-based SR methods spend the same amount of computation treating low- and high-frequency information and lack flexible modulation ability in dealing with them, which ends up the representational ability of the network as well as leads to blurry super-resolved results. To address this problem, Li et al. [69] proposed a feedback network (SRFBN) based on a recurrent architecture design, in which the LR input is recursively refined to obtain a corresponding HR output. The main architecture is based on a feedback block that consists of several projection groups. Each projection group first finds high-resolution features (via deconvolution) and then generates low-resolution features (via convolution). Later, Haris et al. [34] proposed dense deep back-projection network (D-DBPN) that iteratively perform back-projections to learn the feedback error signal between LR and HR images. The motivation is that only a feed-forward approach is not optimal for modeling the mapping from LR to HR images, and a feedback mechanism can greatly help in achieving better results. For this purpose, the proposed architecture comprised of a series of up and downsampling layers that are densely connected with each other to combine HR images from multiple depths in the network. Recently, Qiu et al. [96] and Yang and Lu [131] proposed multi-branch architectures. In these methods, one branch is responsible for cap-

turing high-frequency features such as texture and edge, and another is dedicated to learn low-frequency features such as image outline and contour. Similarly, Li et al. [66] introduced the octave convolution to image SR which uses two branches to perform information update and frequency communication between low- and high-frequency features. Although these methods delivered impressive results, they tend to increase the amount of computation on high-frequency information by increasing the overall number of operations of the model, without paying attention to model complexity. The increase in complexity due to the independent treatment of multiple frequencies is a key issue that limits the performance of these deep CNN-based SR methods. Therefore, the efficient reconstruction of high-frequency details in SISR is still a challenge today.

### 2.7.3 Attention Mechanisms

The aim of introducing attention mechanisms to neural networks is to re-calibrate the feature responses towards the most informative and important components of the inputs [41]. Attention mechanisms have been successfully applied to deep CNN-based image enhancement methods and, more particularly, to SISR. Zhang et al. [144] first incorporated an existing squeeze-and-excitation (SE) channel attention mechanism [41] into SISR and proposed RCAN, which markedly improved the representation ability of the model and SISR performance. In the SE block, each input channel is squeezed into a channel descriptor (*i.e.*, a constant) using global average pooling, then these descriptors are fed into two dense layers to produce channel-wise scaling factors for input channels. Hu et al. [43] combined the SE attention and a spatial attention mechanism to adaptively recalibrate the feature responses by explicitly modeling channel-wise and spatial feature interdependencies. More recent works extend this idea by either adopting different spatial attention mechanisms or designing advanced attention blocks [84, 88]. Meanwhile, Liu et al. [73] further proposed a second-order channel attention (SOCA) module to better learn the feature correlations. The SOCA adaptively rescales the channel-wise features by using second-order feature statistics instead of global average pooling, and enables extracting more informative and discriminative representations.

Non-local or self-attention modules are also popular due to their capability of building spatial or channel-wise attention. When CNN-based methods conduct convolution in a local receptive field, the contextual information outside this field is ignored, while the features in distant regions may have a high correlation and can provide effective information. Given this issue, non-local attention has been proposed as a filtering algorithm to compute a weighted mean of all pixels of an image. In this way, distant pixels can also contribute to the response of a position in concern. For example, Mei et al. [85] proposed local and non-local attention

blocks to extract features that capture the long-range dependencies between pixels and pay attention to more challenging parts. Similarly NLSA [86] and NAAN [121] exploit non-local attention mechanisms to capture long-distance spatial contextual information. In CSNLN [85], a cross-scale non-local attention module is proposed to mine long-range dependencies between LR features and large-scale HR patches within the same feature map. Nevertheless, these methods notoriously consume large amounts of memory to compute large affinity matrices at each spatial position and are often adopted only in large models, thus not being suitable for real-world scenarios.

### 2.7.4 Reconstruction Methods

One of the most important stages of SISR is reconstruction, which consists of generating HR images based on high-level features extracted from a low-dimensional space. Interpolation is a commonly used method in SR networks, such as SRCNN [23], VDSR [52] and DRRN [110], to resize the LR image to the target size as the input of a CNN model for SR reconstruction. However, computational operations are greatly increased due to the large input image size. Thus, FSRCNN [24] and SRDenseNet [60] directly adopted the LR image as input, in which a transposed convolution layer was added to implement the final upsampling reconstruction [118]. This method greatly reduces unnecessary computational overhead. Furthermore, EPSCN [104] proposed a method called sub-pixel layer to overcome the problem of the checkerboard effect in transposed convolution. Sub-pixel layer has been widely used in recent SR models, such as EDSR [71], WDSR [134] and RCAN [144]. However, these methods cannot manage multi-scale training.

Few works tackle SISR at different scale factors, and those that do treat the problem as independent tasks, *i.e.* a model is trained for each scale. Lim et al. [71] proposed the first multi-scale SR model, which has different image processing blocks and upsampling modules for each integer scale factor. Later, Li et al. [65] proposed a multi-scale residual network. They use multi-path convolution layers with different kernel sizes to extract multi-scale spatial features. Later, Grm et al. [32] proposed to upsample the image progressively by  $\times 2$  using a series of so-called SR modules and compute the loss of generated SR results by each module. Thus, these methods require vast amounts of computational resources. Recently, Meta-SR [42] introduced an upsampling module based on meta-learning to solve SR at arbitrary scale factors with a single model through a weight prediction technique. However, this method must predict a large number of convolution weights for each target pixel, the prediction is inefficient, and the results may be unstable [127].

## 2.8 Summary

In this chapter, we firstly detailed the problem definition. Then, we presented some related works, including benchmark datasets, assessment methods, SISR frameworks, upsampling methods, and optimization objectives. Finally, we briefly introduced the different aspects of the image SR problems that will be treated in this thesis.





## 3 Frequency-based Enhancement Network for Efficient Super-Resolution

---

Single image super resolution (SISR) has witnessed great strides with the rapid development of deep learning. Recent advances on SISR are mostly devoted to designing deeper and wider networks to enhance their representation learning capacity. However, as the networks increase in depth and width, deep learning SR methods are faced with the challenge of computational complexity in practice. A promising and under-explored solution is to adapt the amount of computation based on the different frequency bands of the input. To this end, we present a novel frequency-based enhancement block (FEB) which explicitly enhances the information of high frequencies while forwarding low-frequencies to the output. In particular, this block efficiently decomposes features into low- and high-frequency and assigns more computation to high-frequency ones. Thus, it can help the network generate more discriminative representations by explicitly recovering finer details. Our FEB design is simple and generic and can be used as a direct replacement of commonly used SR blocks with no need to change network architectures. It is also orthogonal and complementary to attention-based SR methods. We experimentally show that when replacing SR blocks with FEB we consistently improve the reconstruction error, while reducing the number of parameters in the model. Moreover, we propose a lightweight SR model — Frequency-based Enhancement Network (FENet) — based on FEB that matches the performance of larger models. Extensive experiments demonstrate that our proposal performs favorably against the state-of-the-art SR algorithms in terms of visual quality, memory footprint, and inference time.

---

### 3.1 Motivation

Convolutional neural networks (CNNs) have recently achieved unprecedented success in various problems [38, 128]. The powerful feature representation and end-to-end training paradigm of CNNs make them a promising approach to single image super-resolution (SISR). Recently, most CNN-based SR methods focus on

elaborate architecture designs such as residual learning [1, 6, 60, 65] and dense connections [49, 145]. Although significant progress has been made, as discussed in [43, 96], texture details of the LR images often tend to be smoothed in the super-resolved results since most existing CNN-based SR methods do not pay enough attention to the limited high-frequency information in the LR images. In natural images, information is conveyed at different frequencies. The output feature maps of a convolutional layer can also be seen as a mixture of information at lower and higher frequencies. The lower frequency information is composed of global structures and textures that can directly be forwarded to the final HR output without substantial computations. The higher frequency information consists of fine details where more complex restoring functions are expected. At this point, leading CNN-based methods such as EDSR [71] and RDN [145] overlook the fact that most of the low-frequency information is already contained in the input. As a result, these models spend the same amount of computation treating low- and high-frequency information and lack flexible modulation ability in dealing with them, which ends up the representational ability of the network.

Previous works address this problem by incorporating attention mechanisms [18, 133, 144] into the networks to model interdependencies among spatial locations, channels, or both. The common idea behind attention-based SR methods is to adjust network architectures so that they produce rich feature representations. However, as SR networks are so diverse, the attention module is usually designed solely for a specific network structure [120]. Recently, various SR methods such as multi-branch networks [66, 131] and progressive reconstruction methods [69, 148] mainly focus on refining the high-frequency texture details. Although these methods delivered impressive results, they tend to increase the amount of computation on high-frequency information by increasing the overall number of operations of the model, without paying attention to model complexity. The increase in complexity due to the independent treatment of multiple frequencies is a key issue that limits the performance of these deep CNN-based methods. Therefore, the efficient reconstruction of high-frequency details in SISR is still a challenge today.

In this paper, we address the aforementioned problems from a different perspective. Instead of designing deep and complex networks or adding various shortcut connections to strengthen feature representations, we introduce a novel frequency-based enhancement block (FEB) which is able to separate features into low and high frequencies while also enabling efficient communication among them. Since low frequencies are preserved by downsampling operations and thus can be recovered directly from the input, FEB assigns more computational capacity to high frequencies. The proposed FEB gradually and iteratively enhances high-frequency feature maps during training while preserving low-frequency information, resulting in more accurate features that improve reconstruction quality.



Figure 3.1 – The visual comparison of SR results by the networks with different building modules for scale factor  $\times 4$ . The residual block is used as building module for EDSR. In EDSR-FEB, we replace residual block with proposed FEB.

The proposed FEB offers the following advantages. First, it is generic and can be easily applied to existing SR models without the need of modifying network architectures or requiring hyper-parameters tuning. Second, FEB reduces model parameters in the baseline SR models while simultaneously obtaining better SR performance. In Figure 3.1, we provide an example of visual quality of EDSR [71], which uses residual blocks [38] as its building module. It can be observed that, when we replace residual blocks with our blocks (EDSR-FEB), the network obtains better visual quality while reducing the number of parameters.

Based on FEB, we build a lightweight SR network named frequency-based enhancement network (FENet). Our network leads to significant improvements for single image SR, surpassing SR networks with complicated skip connections and concatenations. Furthermore, to demonstrate the effectiveness of the proposed FEB, we take current state-of-the-art SR methods as baselines, and replace their building blocks with our proposed block. Extensive experiments conducted on SR benchmark datasets demonstrate that such baseline results can be greatly improved by using the proposed FEB, which additionally reduces model parameters and computational complexity.

## 3.2 Frequency-based Enhancement Network

In this section, we first describe the overall network architecture. Next, we detail the proposed frequency-based enhancement block (FEB). Finally, we discuss the differences between the proposed method and similar related works.

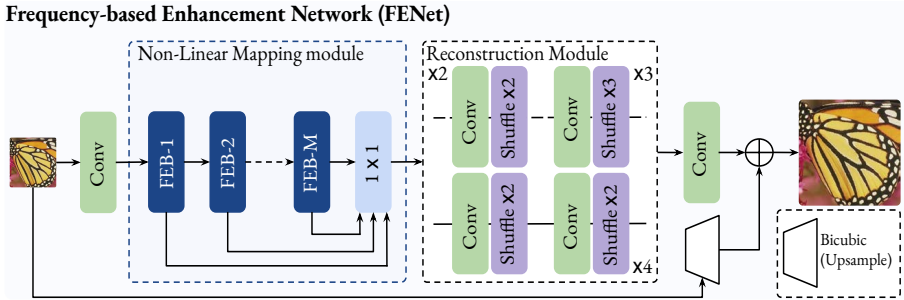


Figure 3.2 – Proposed frequency-based enhancement network (FENet) for SISR, which consists of non-linear mapping and reconstruction modules.

### 3.2.1 Network overview

As shown in Figure 3.2, the overall network architecture of frequency-based enhancement network (FENet) consists of a non-linear mapping module and a reconstruction module. Let's denote as  $I_{LR}$  and  $I_{SR}$  the input and output of FENet, respectively. We apply only one  $3 \times 3$  convolutional layer ( $\mathcal{H}$ ) to extract the initial features  $H_0$  from the LR input image:

$$H_0 = \mathcal{H}(I_{LR}). \quad (3.1)$$

It is worth noting that only one convolutional layer is used here for lightweight design.

Then, we use the non-linear mapping module, which consists of several stacked FEBs to generate new powerful representations, which can be formulated as

$$H_k = \mathcal{B}_k(H_{k-1}), \quad k = 1, \dots, M, \quad (3.2)$$

where  $\mathcal{B}_k$  denotes mapping function of the  $k$ -th FEB.  $H_{k-1}$  represents the features from the previous adjacent FEB, and  $M$  is the total number of FEBs.

Inspired by [65], we apply a feature fusion strategy to integrate the features from all the FEB, which helps to extract more hierarchical contextual information. The fusion operation is formulated as

$$H = \mathcal{F}([H_1, H_2, \dots, H_M]) \quad (3.3)$$

where  $[H_1, H_2, \dots, H_M]$  refers to the concatenation of feature maps produced by FEBs and  $\mathcal{F}$  is a  $1 \times 1$  convolutional operation.

Finally, we utilize the reconstruction module that contains convolutional layers and pixelshuffle layers [104] to upsample the features to the HR size. In addition, we incorporate a global connection path to grant access to the original LR information and facilitate the back-propagation of the gradients, in which only a bicubic interpolation is applied to the input  $I_{LR}$ . Therefore, we obtain:

$$I_{SR} = \mathcal{R}(H) + \text{Bicubic}^\dagger(I_{LR}) \quad (3.4)$$

where  $\mathcal{R}$  is the reconstruction module, and  $I_{SR}$  is the final output of the network.

To optimize the network parameters, we adopt  $L_1$  loss as a cost function for training. Given a training set with  $N$  pairs of LR images and HR counterparts, denoted by  $\{I_{LR}^i, I_{HR}^i\}_{i=1}^N$ , the network is optimized to minimize the  $L_1$  loss function:

$$L_1(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \|I_{SR} - I_{HR}\|_1, \quad (3.5)$$

where  $\boldsymbol{\theta}$  denotes the parameter set.

### 3.2.2 Frequency-based Enhancement Block (FEB)

A natural image can be decomposed into a low frequency component that describes smoothly changing structures and a high-frequency component that describes the rapidly changing fine details [14, 100]. Similarly, we argue that the output feature maps of a convolutional layer can also be decomposed into features of different frequencies, and propose an efficient frequency-based enhancement block (FEB) which naturally decomposes low and high frequencies at feature level. The high-frequency information part is processed by higher-complexity operations (in number of parameters and non-linearities), whereas the lower-frequency part is processed by lower-complexity operations to compensate for the increase of computation. As a result, the proposed approach learns discriminative representations in order to efficiently achieve more accurate reconstructions.

As demonstrated in Figure 3.3, the proposed FEB contains two pathways, each of which is responsible for a different functionality. Each pathway has a  $1 \times 1$  convolutional layer at the beginning. Given the input  $\mathbf{X} \in \mathbb{R}^{C \times H \times W}$ , where  $C$  denotes the number of channels and  $H \times W$  the spatial dimensions, we have

$$\mathbf{X}_1 = \mathcal{F}'_{split}(\mathbf{X}) \quad (3.6)$$

$$\mathbf{X}_2 = \mathcal{F}''_{split}(\mathbf{X}) \quad (3.7)$$

where  $\{\mathbf{X}_1, \mathbf{X}_2\}$  only have half of the channel number of  $\mathbf{X}$ .  $\mathcal{F}'_{split}$  and  $\mathcal{F}''_{split}$  are two

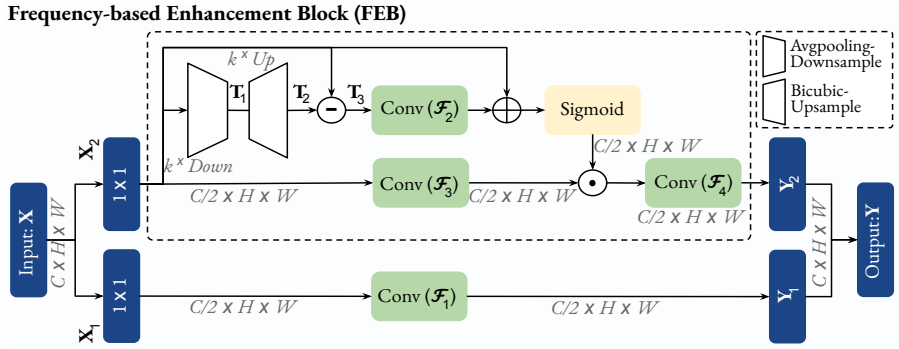


Figure 3.3 – Schematic illustration of the proposed Frequency-based Enhancement Network (FEB). As it can be seen, the original filters are separated into two processing lines, each of which is in charge of a different functionality.

$1 \times 1$  convolutional operations, respectively. Then, the described operations are separately sent into a dedicated pathway for collecting different types of information (*i.e.* low- and high-frequency information). The first pathway targets at retaining the original information (low-frequency). To save computation, we perform only a simple  $3 \times 3$  convolutional operation to capture the global layout and coarse details as follows:

$$\mathbf{Y}_1 = \mathcal{F}_1(\mathbf{X}_1), \quad (3.8)$$

where  $\mathbf{Y}_1$  is the output of the  $3 \times 3$  convolutional layer ( $\mathcal{F}_1$ ).

In the second pathway, we first apply an average pooling layer upon  $\mathbf{X}_2$ , yielding  $\mathbf{T}_1$ :

$$\mathbf{T}_1 = \text{AvgPool}^\dagger(\mathbf{X}_2, k), \quad (3.9)$$

where  $k$  denotes the kernel size of the pooling layer and the size of the intermediate feature map  $\mathbf{T}_1$  is  $\frac{C}{2} \times \frac{H}{k} \times \frac{W}{k}$ . Each value in  $\mathbf{T}_1$  can be viewed as the average intensity of each specified small area of  $\mathbf{X}_2$ . After that,  $\mathbf{T}_1$  is upsampled via a bicubic interpolation operator to produce a new tensor  $\mathbf{T}_2$  of the same size as  $\mathbf{X}_2$

$$\mathbf{T}_2 = \text{Bicubic}^\dagger(\mathbf{T}_1, k), \quad (3.10)$$

where  $\mathbf{T}_2$  contains averaged information and it can be regarded as a smoother version of the original  $\mathbf{X}_2$ . Then, in order to obtain the high-frequency information,

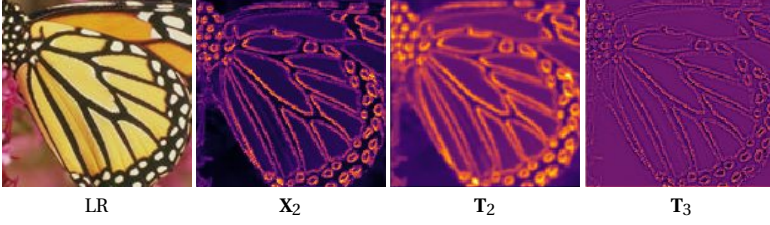


Figure 3.4 – Visual activation feature maps of input  $\mathbf{X}_2$ ,  $\mathbf{T}_2$ , and obtained high-frequency information ( $\mathbf{T}_3$ ).

$\mathbf{T}_2$  is element-wise subtracted from  $\mathbf{X}_2$ :

$$\mathbf{T}_3 = \mathbf{X}_2 - \mathbf{T}_2, \quad (3.11)$$

The visual activation maps of  $\mathbf{X}_2$ ,  $\mathbf{T}_2$  and high-frequency information ( $\mathbf{T}_3$ ) are also shown in Figure 3.4. It can be observed that  $\mathbf{T}_2$  is smoother than  $\mathbf{X}_2$  as it is the average information of  $\mathbf{X}_2$ . Meanwhile,  $\mathbf{T}_3$  retains the details and edges. Now, the high-frequency enhancement operation can be formulated as follows:

$$\mathbf{Y}'_2 = \sigma(\mathcal{F}_2(\mathbf{T}_3) + \mathbf{X}_2) \cdot \mathcal{F}_3(\mathbf{X}_2), \quad (3.12)$$

where  $\sigma$  is the sigmoid function, and  $\mathcal{F}_2$  and  $\mathcal{F}_3$  are two  $3 \times 3$  convolutional layers, respectively. As shown in Equation 3.12, we use  $\mathbf{X}_2$  as residuals to form the weights, which is found beneficial. Then the output of the second pathway can be written as

$$\mathbf{Y}_2 = \mathcal{F}_4(\mathbf{Y}'_2), \quad (3.13)$$

where  $\mathcal{F}_4$  is a  $3 \times 3$  convolutional operation. Finally, both intermediate outputs of the first and second pathways  $\{\mathbf{Y}_1, \mathbf{Y}_2\}$  are concatenated together as the output  $\mathbf{Y} \in \mathbb{R}^{C \times H \times W}$  to obtain a rich feature representation.

Compared to other works such as [66, 131], which require a considerably large amount of computations for decomposing features of different frequencies, FEB can separate the low- and high-frequency feature representations in an efficient way and focus on reconstructing the high-frequency ones.

### 3.2.3 Discussion

**Difference to prominent SR blocks.** Prominent SR blocks such as residual blocks [71] or dense blocks [118] process low- and high-frequency information simultaneously



by the same convolution operations and do not discriminate the computation of features by their frequencial components. Therefore, some local details of LR images cannot be effectively utilized for HR reconstruction, leading to blurry super-resolved results [66]. In contrast, our proposal treats different frequencies in a heterogeneous way and also models inter-channel dependencies, which consequently enrich the output feature. Moreover, FEB benefits SR approaches by reducing the number of parameters while achieving superior SR performance.

**Difference to attention-based methods.** Our work is quite different from existing methods such as [21, 43, 84, 144] which rely on supplementary attention blocks and require additional learnable parameters. In contrast our approach internally changes the way of exploiting convolutional filters of convolutional layers, and hence require no additional learnable parameters. In the following experiment section, we will demonstrate without any extra learnable parameters, FEB can yield significant improvements over baselines and other attention-based SR approaches. Moreover, it is complementary to attention mechanisms, and also benefit from their inclusion into the pipeline.

### 3.3 Experimental Results

In this section, we first conduct an ablation study to validate the effectiveness of the proposed FEB. Then, we systematically compare FENet with state-of-the-art SISR algorithms on five commonly used benchmark datasets.

#### 3.3.1 Settings

**Datasets and metrics.** Following [18], we use 800 high-quality images from the DIV2K dataset [117] for training. We evaluate our models on several benchmark datasets: Set5 [9], Set14 [138], B100 [2], and Urban100 [45], and, Manga109 [82], each with diverse characteristics. All results are evaluated with two commonly used metrics: PSNR and SSIM. To keep the consistency with previous works, quantitative results are evaluated on the luminance channel (Y). Furthermore, we adopt the Perceptual Index (PI) [11], which can avoid the situation where over-smoothed images may present a higher PSNR or SSIM when the performances of two methods are similar.

**Degradation models.** To comprehensively illustrate the efficacy of the proposed method, three degradation models are used to simulate LR images, following [145]. The first one, denoted by **BI**, consists of generating LR images by bicubic downsam-

pling ground truth HR images with  $\times 2$ ,  $\times 3$ ,  $\times 4$ . The second one, denoted by **BD**, first performs bicubic downsampling on HR images with  $\times 3$ , and then blurs the images with a Gaussian kernel of size  $7 \times 7$  and standard deviation 1.6. Finally, we further produce LR images in a third challenging way, denoted by **DN**, by carrying out bicubic downsampling followed by additive Gaussian noise, with noise level of 30.

**Implementation details.** During training, data augmentation is carried out by means of random horizontal flips and  $90^\circ$  rotation. At each training mini-batch, 64 LR RGB patches of size  $64 \times 64$  are provided as inputs. We train FENet using an ADAM optimizer with learning rate  $10^{-3}$ . The learning rate is halved every  $2 \times 10^5$  iterations. We set the number of FEB to 12 in our FENet. Our network has been implemented using PyTorch, and trained on a NVIDIA RTX 3090 GPU.

### 3.3.2 Ablation Study

#### Comparing Pooling Methods

In this section, we investigate the influence of different pooling types on the performance. In our experiments, we use FENet as the basic network and then replace average pooling operators (*Avg*) in all FEBs with maximum pooling operators (*Max*).

As shown in Table 3.1, using the average pooling operator while keeping the rest of configurations unchanged yields a performance increase of about 0.07dB in average. We argue that this may be due to the fact that, unlike maximum pooling, average pooling builds connections among locations within the whole pooling window, which can better capture local contextual information.

Table 3.1 – Average PSNR obtained when FEB using different pooling methods on five benchmark datasets for scale factor  $\times 4$ .

Scale	Dataset	+ Max	+ Avg
$\times 4$	Set5	32.17	<b>32.24(+0.07dB)</b>
	Set14	28.53	<b>28.61(+0.08dB)</b>
	B100	27.54	<b>27.61(+0.07dB)</b>
	Urban100	26.09	<b>26.15(+0.06dB)</b>
	Manga109	30.38	<b>30.43(+0.05dB)</b>

Table 3.2 – Average PSNR to show the effect of downsampling rate on the performance on Set5 dataset. We record the results in  $10 \times 10^4$  iterations.

Downsampling Rate	Scales		
	$\times 2$	$\times 3$	$\times 4$
2	37.89	34.22	32.08
3	37.91	34.24	32.10
4	37.94	34.29	32.14
5	<b>37.95</b>	<b>34.31</b>	<b>32.15</b>

#### The Effect of Downsampling Rate

We also investigate how the downsampling rate in FEB influences the image SR performance. In Table 3.2, we show the performance with different downsampling rates used in FEB. It can be observed that as the downsampling rate increases, slightly better performance is achieved. However, we do not use larger downsampling rates due to two reasons: (1) the resolution of the input features is already very small; (2) higher downsampling rates lead to performance improvements at the expense of more computations due to bicubic operation. Therefore, for the rest of experiments, we set the downsampling rate to 4 for all scale factors, as it still provides significant improvements with a lower computational cost than  $\times 5$ .

#### The Effect of Increasing the Number of FEBs

As discussed in [71], increasing the depth of the network can effectively improve the performance. In this work, adding the number of FEBs is the simplest way to gain excellent results. For better balancing the model size and performance, we compare the proposed model with the different numbers of FEBs, *i.e.*, 6, 8, 10, and 12.

Table 3.3 – Average PSNR obtained with FENet when using different number of FEBs on five benchmark datasets for scale factor  $\times 4$ .

Blocks	6	8	10	12
Params	379K	477K	572K	675k
Set5	31.98	32.15	32.19	<b>32.24</b>
Set14	28.44	28.54	28.57	<b>28.61</b>
B100	27.42	27.54	27.58	<b>27.61</b>
Urban100	25.90	25.97	26.05	<b>26.15</b>
Manga109	30.07	30.20	30.35	<b>30.43</b>

Table 3.4 – Average PSNR obtained with FENet when using different SR blocks on five benchmark datasets for scale factor  $\times 4$ .

Name	+ RB	+ DB	+ IMDB	+ MSRB	+ FEB
Params	707K	714K	727K	739K	675K
Set5	32.02	32.06	32.07	32.10	<b>32.24</b>
Set14	28.46	28.53	28.53	28.57	<b>28.61</b>
B100	27.42	27.51	27.54	27.55	<b>27.61</b>
Urban100	25.84	25.87	25.89	25.97	<b>26.15</b>
Manga109	30.12	32.20	30.21	30.17	<b>30.43</b>

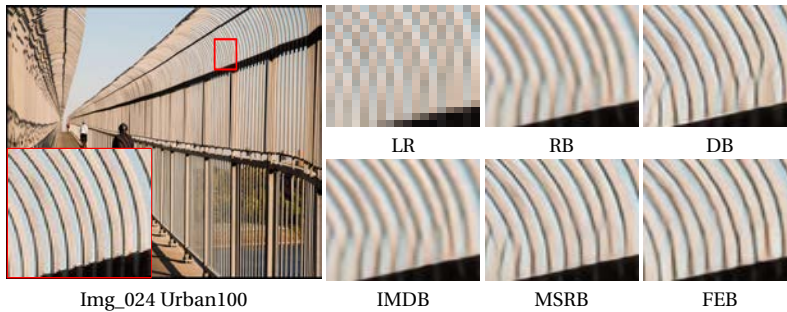


Figure 3.5 – Visual comparisons of SR results using FENet with different SR blocks for scale factor  $\times 4$ .

As shown in Table 3.3, our FENet performance improves rapidly with the growth in number of FEBs. Although the performance of the network would further improve by using more FEBs, we found it leads to diminishing returns with respect to the number of parameters. Therefore, we use 12 FEBs in our experiments.

### The Effectiveness of FEBs

To demonstrate the effectiveness of our proposed FEB scheme, we use FENet as the basic network. To keep the number of parameters similar, we replace the 12 FEBs with 8 residual blocks (RB) [71], 5 dense blocks (DB) [118], 6 information multi-distillation blocks (IMDB) [47], or 4 multi-scale residual blocks (MSRB) [65]. In Table 3.4, we compare the number of parameters and the performance in PSNR for all methods for scale factor  $\times 4$ .

As reported in Table 3.4, method with FEB outperforms all the methods with different SR blocks with fewer number of parameters. These experiments justify

that the proposed FEB results are more helpful for image SR. Moreover, we provide visual comparisons (Figure 3.5) of FENet using different SR blocks for scale factor  $\times 4$ . It can be observed that the FENet using FEB obtains better visual quality and represents more diverse structure patterns.

#### Attention mechanisms vs FEB

To further verify the effectiveness of FEB, we use a ResNet architecture, *i.e.*, a regular architecture composed of 8 stacked residual blocks. Then, we integrate two commonly used attention mechanism namely CA [144] (*ResNet-CA*) and CSAR [43] (*ResNet-CSAR*) into residual blocks as done in [144], respectively. Furthermore, we replace 8 residual blocks with 12 FEBs (*FENet*) and integrated the two mentioned attention mechanisms into FEBs and named them as *FENet-CA* and *FENet-CSAR*.

As reported in Table 3.5, ResNet-CSAR and ResNet-CA obtain better performance than ResNet but they require additional learnable parameters. Quite differently, FENet does not rely on any extra learnable parameters since it heterogeneously exploits the convolutional filters and thus achieves better performance than ResNet-CSAR and ResNet-CA. It should also be mentioned that the proposed FEB is also compatible with the above mentioned attention mechanisms. For example, when adding CA blocks to each FEB of FENet (*FENet-CA*), we can further gain another 0.07dB in average. This also indicates that our approach is orthogonal to this kind of supplementary attention modules.

Table 3.5 – Average PSNR obtained with FENet when using different attention mechanisms on five benchmark datasets scale factor  $\times 4$ .

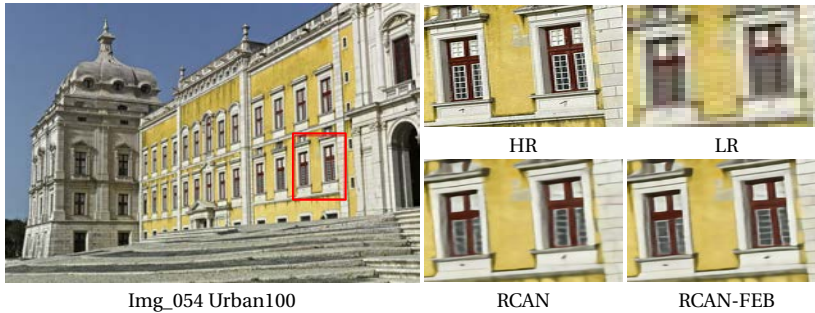
Methods	Params	Set14	B100	Urban100
ResNet	707K	28.46	27.42	25.84
ResNet-CA	733k	28.50	27.46	25.89
ResNet-CSAR	782k	28.53	27.50	25.93
FENet	675K	28.61	27.61	26.18
FENet-CA	701k	28.68	27.69	26.25
FENet-CSAR	750k	28.70	27.72	26.28

#### Generalization ability

To demonstrate the generalization ability of the proposed structure, we select two state-of-the-art SR networks with different model sizes, called EDSR [71] and RCAN [144]. The EDSR contains 32 stacked residual blocks with  $256 \times 256$  filters. The RCAN consists of 200 residual channel attention blocks with  $64 \times 64$  filter sizes. We replace their building blocks with FEBs. The corresponding networks with FEB are

Table 3.6 – Average PSNR obtained with state-of-the-art SR methods when using FEB on five benchmark datasets for scale factor  $\times 4$ .

Name	EDSR	EDSR-FEB	RCAN	RCAN-FEB
Params	43M	28M	16M	9M
Set5	32.50	<b>32.58(+0.08dB)</b>	32.63	<b>32.70(+0.07dB)</b>
Set14	28.72	<b>28.80(+0.08dB)</b>	28.87	<b>28.96(+0.06dB)</b>
B100	27.72	<b>27.81(+0.09dB)</b>	27.77	<b>27.85(+0.08dB)</b>
Urban100	26.67	<b>26.76(+0.09dB)</b>	26.82	<b>26.89(+0.07dB)</b>
Manga109	31.02	<b>31.09(+0.07dB)</b>	31.22	<b>31.30(+0.08dB)</b>

Figure 3.6 – The visual comparison of SR results by the networks with different building modules for  $\times 4$  scale factor. The residual blocks followed by channel attentions are used as building modules for RCAN. In RCAN-FEB, we replace the blocks with proposed FEBs.

named as *EDSR-FEB* and *RCAN-FEB*, respectively. For fair comparison, all networks are trained on their default settings.

As shown in Table 3.6, EDSR-FEB has an improvement of 0.08dB in average with almost  $\times 2$  fewer number of parameters (parameters: 28M) compared to the original EDSR (parameters: 43M). Moreover, the improvement by RCAN-FEB is also higher than RCAN with approximately half amount of parameters. From these comparisons, we can easily find that (1) the proposed FEB performs much better than channel attention, (2) for deeper networks, a similar phenomenon can also be observed, (3) FEB reduces the number of parameters by half while achieving better performance. Figure 3.1 and 3.6 additionally show visual comparisons for scale factor  $\times 4$ . It can be observed that EDSR-FEB and RCAN-FEB can reconstruct sharper and more natural-looking images. This is mainly because FEB can extract high-frequency features and use them for reconstruction.

### 3.3.3 Comparison With state-of-the-art Methods

In this section, FENet is compared to other light- and heavy-weight state-of-the-art SR methods. A self-ensemble method [116] is also used to further improve the performance of the FENet (denoted as FENet+).

#### Results with BI degradation models

In this section, we compare the proposed FENet and FENet+ with state-of-the-art lightweight models: VDSR [52], DRCN [53], SRDenseNet [118], SEINet [19], SRResNet [60], CARN [1], IMDN [47], SRFBN-S [69], A2F-S [130], CBPN [148], LAPAR-A [67], MADNet [59], FALSAR-A [20], DPN [70], HDRN [49], and OISR-RK2 [40].

Table 3.7 shows quantitative results when evaluating PSNR and SSIM on five benchmark dataset with different algorithms for scale factors  $\times 2$ ,  $\times 3$ , and  $\times 4$ . For a more informative comparison, the number of parameters is also given. From Table 3.7, we find that FENet only has less than 0.7M parameters but performs favorably against other compared approaches on most datasets. For example, in comparison with SRDenseNet [118] and OISR-RK2 [40], FENet achieves better or competitive results, while only needing 30% and 40% of their parameters, respectively. On the other hand, thanks to the FEB, FENet achieves competitive or better results when compared to the large SR methods. Specifically, FENet outperforms FSN [66] by a large margin at all scales in all datasets with  $18\times$  fewer parameters. Furthermore, it can be seen that FENet+ achieves further improvements through the use of self-ensembles [116] and it is the best performing one, at all scales and in all datasets.

In Figure 3.7, we present some qualitative visual comparisons for the  $\times 4$  scale factor. It can be observed that SR images reconstructed by FENet have more refined details, especially in the edges and lines. This further validates the effectiveness of the proposed FEB.

#### Results with BD and DN degradation models

Following [145], we also show the SR results with **BD** degradation model and further introduce **DN** degradation model. The proposed FENet and FENet+ are compared with state-of-the-art methods including SPMSR [93], SRCNN [23], FSRCNN [24], VDSR [52], IRCNN\_G [140], IRCNN\_C [140], and SRMD(NF) [141]. We included the RDN [145] high-capacity model for reference.

As shown in Table 3.8, our methods perform the best on all datasets with **BD** and **DN** degradation models. The significantly better results of FENet and FENet+ indicate that our methods adapt well to scenarios with multiple degradation models. Moreover, our methods achieve comparable results to the RDN. It is worth noting

### 3.3. Experimental Results

Table 3.7 – Average PSNR/SSIM values for models with the same order of magnitude of parameters. Performance is shown for scale factors  $\times 2$ ,  $\times 3$  and  $\times 4$  with BI degradation model. The best and second best results are highlighted in red and blue respectively.

Scale	Method	Params	Set5		Set14		B100		Urban100		Manga109	
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
$\times 2$	VDSR [52]	0.7M	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.22	0.9729
	DRCN [53]	1.8M	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.63	0.9723
	SEINet [19]	1M	37.89	0.9598	33.61	0.9160	32.08	0.8984	–	–	–	–
	CARN [1]	1.6M	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
	SRFBN-S [69]	0.3M	37.78	0.9597	33.35	0.9156	32.00	0.8970	31.41	0.9207	38.06	0.9757
	A2F-S [130]	0.3M	37.79	0.9597	33.32	0.9152	31.99	0.8972	31.44	0.9211	38.11	0.9757
	CBPN [148]	1M	37.90	0.9590	33.60	0.9171	32.17	0.8989	32.14	0.9279	–	–
	MADNet [59]	0.9M	37.94	0.9604	33.46	0.9167	32.10	0.8988	31.74	0.9246	–	–
	FALSR-A[20]	1M	37.82	0.9595	33.55	0.9168	32.12	0.8987	31.93	0.9256	–	–
	HDRN [49]	0.9M	37.75	0.9590	33.49	0.9150	32.03	0.8980	31.87	0.9250	38.07	0.9770
	DPN [70]	0.8M	37.52	0.9586	33.08	0.9129	31.89	0.8958	30.82	0.9144	–	–
	LAPAR-A [67]	0.5M	38.01	0.9605	33.62	0.9183	32.19	0.8999	32.10	0.9283	38.67	0.9772
	IMDN [47]	0.7M	38.00	0.9605	33.63	0.9177	32.19	0.8996	32.17	0.9283	38.88	0.9774
	OISR-RK2 [40]	1.4M	38.02	0.9605	33.62	0.9178	32.20	0.9000	<b>32.21</b>	<b>0.9290</b>	–	–
FENet	0.6M	<b>38.08</b>	<b>0.9608</b>	<b>33.70</b>	<b>0.9184</b>	<b>32.20</b>	<b>0.9001</b>	32.18	0.9287	<b>38.89</b>	<b>0.9775</b>	
FENet+	0.6M	<b>38.14</b>	<b>0.9610</b>	<b>33.77</b>	<b>0.9190</b>	<b>32.27</b>	<b>0.9006</b>	<b>32.24</b>	<b>0.9292</b>	<b>38.94</b>	<b>0.9779</b>	
	EDSR [71]	43M	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
	CASGCN [133]	14M	38.26	0.9615	34.02	0.9213	32.36	0.9020	33.17	0.9377	39.41	0.9785
	FSN [66]	7.3M	37.68	0.9605	33.51	0.9180	32.09	0.9015	31.68	0.9248	–	–
$\times 3$	VDSR [52]	0.7M	33.66	0.9213	29.77	0.8314	28.82	0.7976	27.14	0.8279	37.22	0.9750
	DRCN [53]	1.7M	33.82	0.9226	29.76	0.8311	28.80	0.7963	27.15	0.8276	32.24	0.9343
	CARN [1]	1.6M	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
	SRFBN-S [69]	0.4M	34.20	0.9255	30.10	0.8372	28.96	0.8010	27.66	0.8415	33.02	0.9404
	A2F-S [130]	0.3M	34.06	0.9241	30.08	0.8370	28.92	0.8006	27.57	0.8392	32.86	0.9394
	MADNet [59]	0.9M	34.26	0.9262	30.29	0.8410	29.04	0.8033	27.91	0.8464	–	–
	HDRN [49]	0.9M	34.24	0.9240	30.23	0.8400	28.96	0.8040	27.93	0.8490	33.17	0.9420
	DPN [70]	0.8M	33.71	0.9222	29.80	0.8320	28.84	0.7981	27.17	0.8282	–	–
	LAPAR-A [67]	0.5M	34.36	0.9267	30.34	0.8421	29.11	0.8054	28.15	0.8523	33.51	0.9441
	IMDN [47]	0.7M	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519	<b>33.61</b>	<b>0.9445</b>
	OISR-RK2 [40]	1.6M	34.39	0.9272	30.35	0.8420	29.11	0.8058	<b>28.24</b>	<b>0.8544</b>	–	–
	FENet	0.6M	<b>34.40</b>	<b>0.9273</b>	<b>30.36</b>	<b>0.8422</b>	<b>29.12</b>	<b>0.8060</b>	28.20	0.8539	33.57	0.9444
	FENet+	0.6M	<b>34.47</b>	<b>0.9279</b>	<b>30.41</b>	<b>0.8426</b>	<b>29.17</b>	<b>0.8065</b>	<b>28.28</b>	<b>0.8545</b>	<b>33.63</b>	<b>0.9450</b>
		EDSR [71]	43M	34.65	0.9280	30.52	0.8462	29.25	0.8093	28.80	0.8653	34.17
CASGCN [133]		14M	34.75	0.9300	30.59	0.8476	29.33	0.8114	28.93	0.8671	34.36	0.9494
$\times 4$	VDSR [52]	0.7M	31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524	28.83	0.8809
	DRCN [53]	1.8M	31.54	0.8850	29.19	0.7720	27.32	0.7280	25.12	0.7560	29.09	0.8845
	SEINet [19]	1.4M	32.05	0.8934	28.49	0.7783	27.44	0.7325	–	–	–	–
	SRDenseNet [118]	2M	32.00	0.8931	28.50	0.7782	27.53	0.7337	26.05	0.7819	30.41	0.9071
	SRResNet [60]	1.5M	32.05	0.8910	28.53	0.7804	27.57	0.7354	26.07	0.7839	–	–
	CARN [1]	1.6M	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	<b>30.47</b>	<b>0.9084</b>
	SRFBN-S [69]	0.5M	31.98	0.8923	28.45	0.7779	27.44	0.7313	25.71	0.7719	29.91	0.9008
	A2F-S [130]	0.3M	31.87	0.8900	28.36	0.7760	27.41	0.7305	25.58	0.7685	29.77	0.8987
	CBPN [148]	1.2M	32.21	0.8944	<b>28.63</b>	0.7813	27.58	0.7356	26.14	0.7869	–	–
	MADNet [59]	1M	32.11	0.8939	28.52	0.7799	27.52	0.7340	25.89	0.7782	–	–
	HDRN [49]	0.9M	32.23	0.8960	28.58	0.7810	27.53	0.7370	26.09	0.7870	30.43	0.9080
	DPN [70]	0.8M	31.42	0.8849	28.07	0.7688	27.30	0.7256	25.25	0.7546	–	–
	LAPAR-A [67]	0.7M	32.15	0.8944	28.61	0.7818	27.61	0.7366	26.14	0.7871	30.42	0.9074
	IMDN [47]	0.7M	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838	30.45	0.9075
OISR-RK2 [40]	1.5M	32.14	0.8947	<b>28.63</b>	<b>0.7819</b>	27.60	0.7369	26.17	0.7888	–	–	
FENet	0.6M	<b>32.24</b>	<b>0.8961</b>	28.61	0.7818	<b>27.63</b>	<b>0.7371</b>	<b>26.20</b>	<b>0.7890</b>	30.46	0.9083	
FENet+	0.6M	<b>32.29</b>	<b>0.8966</b>	<b>28.67</b>	<b>0.7823</b>	<b>27.69</b>	<b>0.7377</b>	<b>26.28</b>	<b>0.7898</b>	<b>30.52</b>	<b>0.9089</b>	
	EDSR [71]	43M	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148
	CASGCN [133]	14M	32.60	0.9002	28.88	0.7890	27.70	0.7416	26.79	0.8086	31.18	0.9169
	FSN [66]	8M	32.10	0.8959	28.57	0.7874	27.53	0.7438	25.76	0.7817	–	–



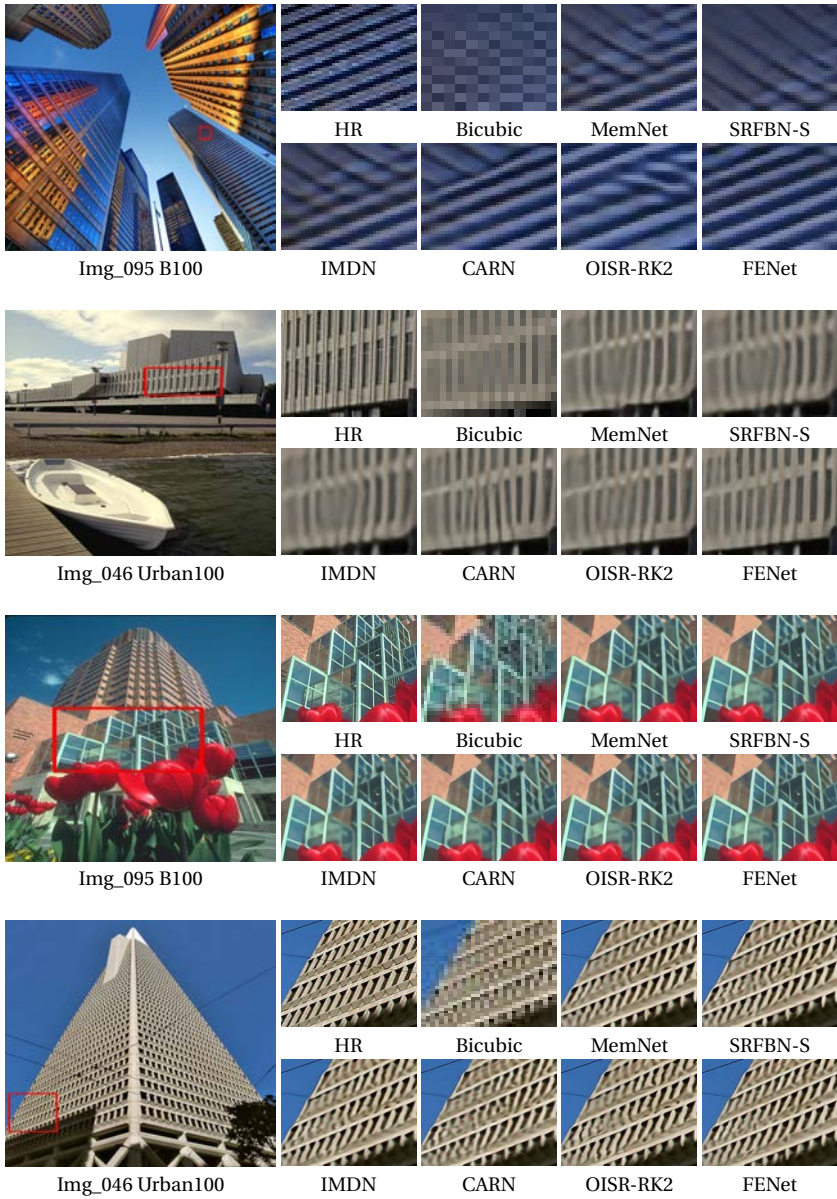


Figure 3.7 – Visual results of BI degradation model ( $\times 4$ ).

### 3.3. Experimental Results

Table 3.8 – Quantitative results with **BD** and **DN** degradation models. The best and second best results are highlighted in **red** and **blue** respectively.

Methods	Degrad.	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	BD	28.78	0.8308	26.38	0.7271	26.33	0.6918	23.52	0.6862	25.46	0.8149
	DN	24.01	0.5369	22.87	0.4724	22.92	0.4449	21.63	0.4687	23.01	0.5381
SPMSR [93]	BD	32.21	0.9001	28.89	0.8105	28.13	0.7740	25.84	0.7856	29.64	0.9003
	DN	–	–	–	–	–	–	–	–	–	–
SRCNN [23]	BD	32.05	0.8944	28.80	0.8074	28.13	0.7736	25.70	0.7770	29.47	0.8924
	DN	25.01	0.6950	23.78	0.5898	23.76	0.5538	21.19	0.5737	23.75	0.7148
FSRCNN [24]	BD	26.23	0.8124	24.44	0.7106	24.86	0.6832	22.04	0.6745	23.04	0.7927
	DN	24.18	0.6932	32.02	0.5856	23.41	0.5556	21.15	0.5682	22.39	0.7111
VDSR [52]	BD	33.25	0.9150	29.46	0.8244	28.57	0.7893	26.61	0.8136	31.06	0.9234
	DN	25.20	0.7183	24.00	0.6112	24.00	0.5749	22.22	0.6096	24.20	0.7525
IRCNN_G [140]	BD	33.38	0.9182	29.63	0.8281	28.65	0.7922	26.77	0.8154	31.15	0.9245
	DN	25.70	0.7379	24.45	0.6305	24.28	0.5900	22.90	0.6429	24.88	0.7765
IRCNN_C [140]	BD	29.55	0.8246	27.33	0.7135	26.46	0.6572	24.89	0.7172	28.68	0.7701
	DN	26.18	0.7430	24.68	0.6300	24.52	0.5850	22.63	0.6205	24.74	0.7701
SRMD(NF) [141]	BD	34.09	0.9242	30.11	0.8364	28.98	0.8009	27.50	0.8370	32.97	0.9391
	DN	27.74	0.8026	26.13	0.6924	25.64	0.6495	24.28	0.7092	26.72	0.8590
FENet	BD	<b>34.46</b>	<b>0.9265</b>	<b>30.42</b>	<b>0.8423</b>	<b>29.14</b>	<b>0.8050</b>	<b>28.29</b>	<b>0.8532</b>	<b>33.93</b>	<b>0.9453</b>
	DN	<b>28.40</b>	<b>0.8150</b>	<b>26.17</b>	<b>0.6930</b>	<b>25.82</b>	<b>0.6597</b>	<b>24.71</b>	<b>0.7347</b>	<b>27.73</b>	<b>0.8599</b>
FENet+	BD	<b>34.52</b>	<b>0.9270</b>	<b>30.50</b>	<b>0.8429</b>	<b>29.20</b>	<b>0.8056</b>	<b>28.36</b>	<b>0.8539</b>	<b>33.99</b>	<b>0.9459</b>
	DN	<b>28.47</b>	<b>0.8157</b>	<b>26.24</b>	<b>0.6937</b>	<b>25.89</b>	<b>0.6602</b>	<b>24.78</b>	<b>0.7353</b>	<b>27.80</b>	<b>0.8606</b>
RDN [145]	BD	34.57	0.9280	30.53	0.8447	29.23	0.8079	28.46	0.8581	33.97	0.9465
	DN	28.46	0.8151	26.60	0.7101	25.96	0.6573	24.92	0.7362	28.00	0.8590

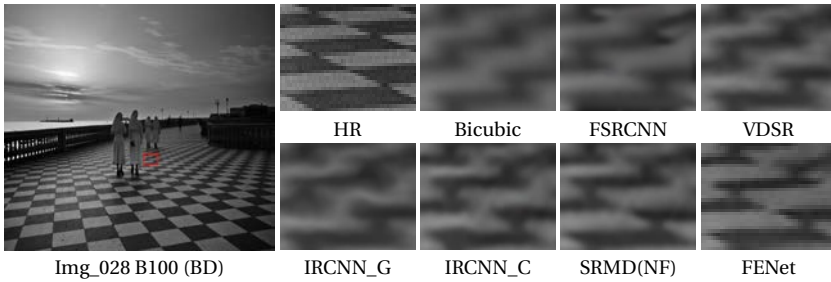


Figure 3.8 – Visual results of **BD** degradation model ( $\times 3$ ).

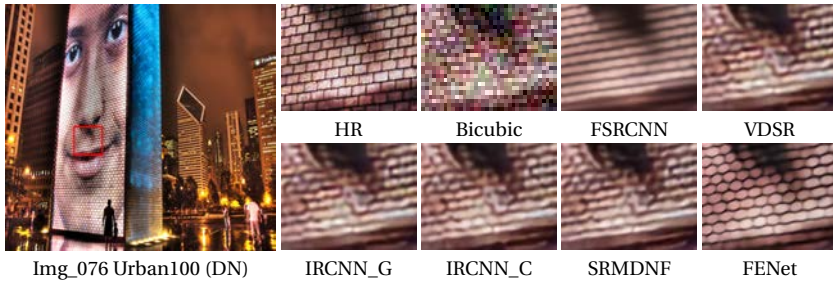


Figure 3.9 – Visual results of **DN** degradation model ( $\times 3$ ).

that while RDN has 22M parameters, OverNet only has 0.9M parameters.

In Figure 3.8 and 3.9, we show two sets of visual results with **BD** and **DN** degradation models from the standard benchmark datasets. For **BD** degradation model, the proposed FENet suppresses the blurring artifacts and recovers sharper edges. For **DN** degradation model, FENet can not only handle the noise efficiently, but also recover details more accurately. These comparisons further showcase the robustness and effectiveness of our method in handling **BD** and **DN** degradation models.

### Model complexity analysis

In this section, we compare the trade-off between performance, number of parameters and the number of multiplications and additions (Multi-Adds) for our methods (FENet and FENet+) and existing lightweight SR networks. The Multi-Adds are calculated corresponding to a  $1280 \times 720$  HR image.

Figure 3.10 shows the PSNR performances of several existing lightweight models, namely VDSR [52], DRCN [53], SRDenseNet [118], SEINet [19], SRResNet [60], CARN [1], IMDN [47], SRFBN-S [69], A2F-S [130], CBPN [148], LAPAR-A [67], MADNet [59], FALSAR-A [20], DPN [70], HDRN [49], and OISR-RK2 [40] versus the number of parameters and Multi-Adds with results evaluated on Urban100 for scale factor  $\times 4$ . As shown in Figure 3.10, our models achieve state-of-the-art results with less parameters and Multi-Adds operations. This demonstrates that our proposals achieve a better trade-off between model size and reconstruction performance.

### Memory Complexity and running time analysis

Table 3.9 illustrates the superiority of the proposed FENet in terms of Inference Time (s) and Memory Consumption (MB), when compared to recent light- and heavy-weight state-of-the-art approaches on Urban100 for  $\times 4$ . For a fair comparison, we

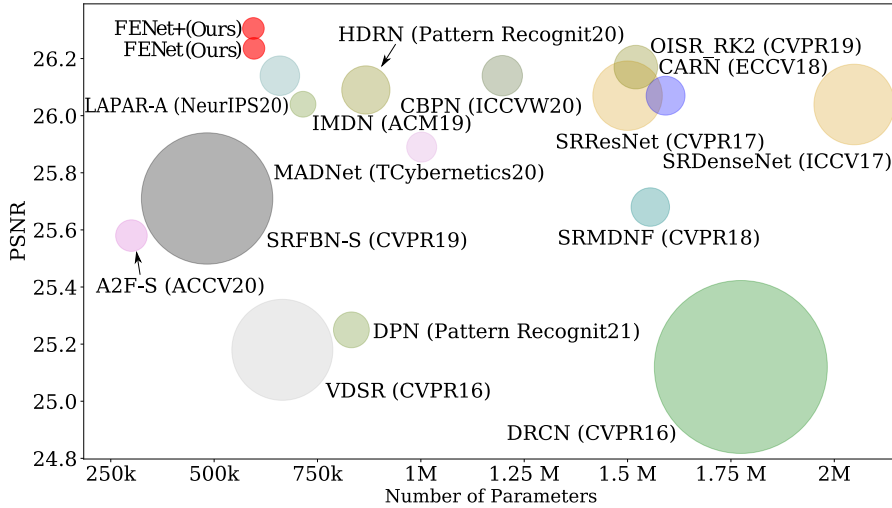


Figure 3.10 – Comparing capacity vs performance for lightweight state-of-the-art SISR models on Urban100 ( $\times 4$ ). Circle sizes are set proportional to the number of multiplications and additions (Multi-Adds).

Table 3.9 – Average running time (s) and memory consumption (MB) comparison on Urban100 for  $\times 4$  scale factor.

Methods	Params	Memory	Running Time(s)	PSNR
CARN [1]	1.5M	1,116	0.032	26.07
SRFBN-S [69]	0.5M	2,154	0.102	25.71
SRDenseNet [118]	2M	5,531	0.221	26.05
IMDN [47]	0.7M	871	0.028	26.04
A2F-S [130]	0.3M	915	0.032	25.58
LAPAR-A [67]	0.7M	1,240	0.053	26.14
MSRN [65]	8M	2,731	0.070	26.04
RCAN [144]	16M	1,531	0.087	26.82
EDSR [71]	43M	2,731	0.035	26.64
FENet	0.6M	850	0.009	26.20

use a single NVIDIA RTX 3090 GPU for evaluation, and their official source code implementations. It can be observed that our model achieves dominant performance in terms of memory usage and time consumption, reflecting its efficiency.

### Perceptual Metrics

Perceptual metrics better reflect the human judgment of image quality. In this paper, Perceptual Index (PI) [11] is chosen as the perceptual metric. Table 3.10 shows the PI for those works with publicly available source code, and the same order of magnitude in terms of parameters. We observe that our proposed model obtains better results than all the compared baselines. This demonstrates the ability of the proposed FENet for generating realistic images.

Table 3.10 – Perceptual index comparison of the proposed method with recent lightweight state-of-the-art methods on five datasets for  $\times 4$ . The lower is better. All of the output SR images are provided officially.

Methods	Params	Set5	Set14	B100	Urban100	Manga109
DRCN [53]	1.7M	6.451	5.945	5.897	5.791	5.563
CARN [1]	1.5M	6.297	5.775	5.700	5.540	5.132
SRFBN-S [52]	0.6M	6.451	5.775	5.702	5.549	5.010
SRDenseNet [118]	2M	6.128	5.615	5.653	5.526	4.762
IMDN [47]	0.7M	6.124	5.644	5.659	5.531	4.810
LAPAR-A [67]	0.7M	6.084	5.499	5.532	5.179	4.771
FENet	0.6M	5.598	5.495	5.447	5.175	4.761

## 3.4 Summary

In this chapter, we have presented a lightweight network for accurate image SR. Specifically, we have proposed a frequency-based enhancement block (FEB) which efficiently decomposes features into low and high frequencies and treats them differently. The proposed FEB allows the network to explicitly allocate more computational capacity to high-frequency features hence improving discriminative capabilities of the network. Experimental results on several benchmark datasets demonstrate that our method can achieve superior performance with a low number of parameters. We proved that the FEB can be flexibly embedded into other SR models by simply replacing their building modules, thus improving their original performance while reducing the number of parameters. The provided evidence suggests that the proposed FEB may help with other low-level image restoration tasks, such as denoising and dehazing.

## 4 Lightweight Multi-Scale Super-Resolution with Overscaling Network

---

Single image super-resolution (SISR) has achieved great success due to the development of deep convolutional neural networks (CNNs). However, as the depth and width of the networks increase, CNN-based SR methods have been faced with the challenge of computational complexity in practice. Moreover, most SR methods train a dedicated model for each target resolution, losing generality and increasing memory requirements. To address these limitations, we introduce OverNet, a deep but lightweight convolutional network to solve SISR at arbitrary scale factors with a single model. We make the following contributions: first, we introduce a lightweight feature extractor that enforces efficient reuse of information through a novel recursive structure of skip and dense connections. Second, to maximize the performance of the feature extractor, we propose a model agnostic reconstruction module that generates accurate high-resolution images from overscaled feature maps obtained from any SR architecture. Third, we introduce a multi-scale loss function to achieve generalization across scales. Experiments show that our proposal outperforms previous state-of-the-art approaches in standard benchmarks while maintaining relatively low computation and memory requirements.

---

### 4.1 Motivation

Single image super-resolution (SISR) is the task of reconstructing an HR from its LR version. As obtaining an HR image from LR is an ill-posed problem, the model needs to *learn* the original data distribution to produce the most likely solutions.

Convolutional neural networks (CNNs) have recently become the main workhorse to tackle SISR [23]. Thanks to the increase in capacity of CNNs in depth and width [71], their performance has greatly improved. Despite their remarkable performance, most deep networks still have some drawbacks. Firstly, increase in depth and width has also raised computational demands and memory consumption. This makes modern architectures less applicable in practice, such as in mobile

and embedded applications. Secondly, as the network depth increases, low-level feature information gradually disappears in the successive non-linear operations to produce the output. However, these low-level features are crucial for the network to reconstruct high-quality images.

Aside from the aforementioned problems, another desired ability is to upsample images to arbitrary scales using a single model. Current state-of-the-art SISR models such as RDN [145], ESPCNN [104] and EDSR [71], only consider SR at certain integer scale factors ( $\times 2$ ,  $\times 3$ ,  $\times 4$ ) and treat each super-resolution scale as an independent task. They then train a different specialized model for each, which is not practical for mobile applications.

To address these problems, we propose overscaling network (OverNet), a novel lightweight method for SISR. OverNet consists of two main parts: a lightweight feature extractor and an overscaling module (OSM) for reconstruction. The feature extractor follows a novel recursive framework of skip and dense connections to reduce low-level feature degradation. The OSM is a new inductive bias which generates accurate SR images by internally constructing an overscaled intermediate representation of the output features. Finally, to solve the problem of reconstruction at arbitrary scale factors, we introduce a novel multi-scale loss by downsampling the output at multiple super-resolution factors and we minimize the reconstruction error in all of them. Our main contributions can be summarized as follows:

- A lightweight recursive feature extractor, which results in improved performance over state-of-the-art models, even those having an order of magnitude more parameters.
- An overscaling module (OSM) that generates overscaled maps from which HR images can be accurately recovered at arbitrary scales. This module boosts the reconstruction accuracy efficiently with respect to its number of parameters. Additionally, we demonstrate that integrating this module into existing state-of-the-art models improves their original performance.
- A novel multi-scale loss function for SISR, that allows the simultaneous training of all scale factors using a single model. As a result, the model is able to maintain accurate reconstruction results across scales.

## 4.2 Proposed Overscaling Network

This section describes the main components of our architecture as shown in Figure 4.1, and the novel loss function.

**Algorithm 1** Overscaling network forward step. Given an LR image and a set of output scales, OverNet produces an HR reconstruction for each scale. Learnable parameters are omitted to improve readability.

---

```

function OVERNET(LR image  $I^{LR}$ , target scales S)
  # Compute features with the CNN
   $\mathbf{h} = \mathcal{H}(I^{LR})$ 
  # Overscaling module
   $\hat{I}^{HR} = \mathcal{O}(\mathbf{h})$ 
  # Output
  for  $s$  in S do
     $\hat{I}_s^{HR} = \text{bicubic}_\downarrow(\hat{I}^{HR}, \text{scale} = s)$ 
  end for
  return  $\{\hat{I}_s^{HR}, s \in S\}$ 
end function

```

---

**Problem formulation.** Algorithm 1 formulates the main pipeline steps. Given a set of HR images and their downscaled versions  $\{I^{HR}, I^{LR}\}$ , the goal of SISR is to find a function  $\mathcal{F} : LR \rightarrow HR$  that maps LR images to their original HR version. The problem is ill-posed since there are multiple possible HR images corresponding to a single LR image. However, it is possible to *learn* the most likely reconstruction by parametrizing  $\mathcal{F}$  over a set of parameters  $\theta$ , and finding the most likely  $\theta$  given some criterion  $\mathcal{L}$ :

$$\theta^* = \arg \min_{\theta} \sum \mathcal{L}(\mathcal{F}(I^{LR}, \theta), I^{HR}) \quad (4.1)$$

We choose  $\mathcal{L}$  to be the L1 distance, since we empirically obtained superior PSNR results compared to L2. In this work  $\mathcal{F}$  is composed of two parts: (i) a feature extractor  $\mathcal{H}$ :

$$\mathbf{h} = \mathcal{H}(I^{LR}, \theta_h) \quad (4.2)$$

with parameters  $\theta_h$ , and (ii) the overscaling module  $\mathcal{O}$ :

$$\hat{I}^{HR} = \mathcal{O}(\mathbf{h}, \theta_o) \quad (4.3)$$

with  $\theta_o$  the parameters used in this operation, and  $\hat{I}^{HR}$  the reconstructed image. These two parts are described next.



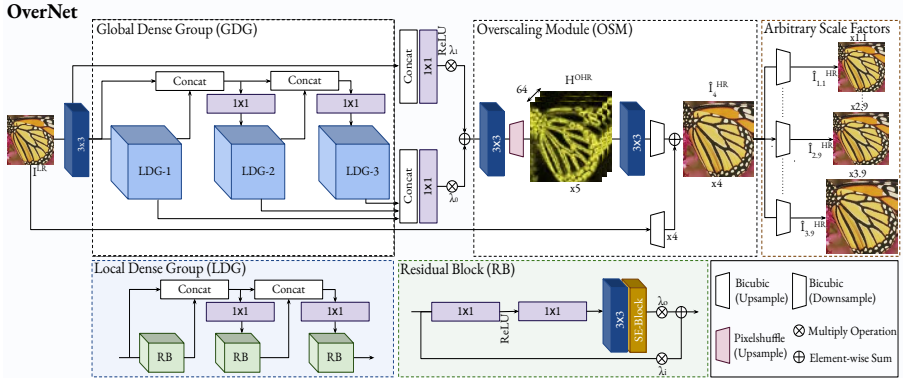


Figure 4.1 – Demonstration of our proposed overscaling network with short and long skip connections. As the maximum scale factor in this particular example is set to  $N = 4$ , the required overscaling is  $\times 5$ .

### 4.2.1 Feature Extractor

The feature extractor computes useful representations of the LR patch in order to infer its HR version. Concretely, we propose a recursive structure based on residual blocks (RBs) assembled into local dense groups (LDGs) and LDGs into global dense group (GDG), see Figure 4.1.

**Residual blocks.** We use a modified version of WDSR [134] with wide low-rank convolutions instead of using standard residual blocks [145]. These convolutions widen the activation space before the non-linearity to let more information pass through it and lose less detail while using the same amount of computation as standard  $3 \times 3$  residual blocks. In order to make the network focus on more informative features, we exploit the inter-dependencies among feature channels using squeeze-and-excitation (SE) operations [41] after these convolutions, see Figure 4.1.

Inspired by [109], the model learns a scalar multiplier  $\lambda$  to balance the amount of information that should be carried by the identity and activation operations within the residual blocks (RBs) of the network.

Let  $\mathbf{x}_i$  and  $\mathbf{x}_o$  be the input and output vectors of the  $k$ -th RB, and  $WA$  the wide activation operation [134]. Then, the RB proceeds as:

$$\mathbf{x}_o = \lambda_o SE(WA(\mathbf{x}_i)) + \lambda_i \mathbf{x}_i. \quad (4.4)$$

**Local and global dense groups.** RBs are grouped into the so-called local dense

groups (LDGs). The input of an RB is concatenated with the output of all the previous RBs in the group and merged with a  $1 \times 1$  convolution. This recursion is repeated for all RBs within the LDG. In this way, we gather all local information progressively by  $1 \times 1$  convolution layers.

To increase the network capacity, a similar recursion is applied to the global dense group (GDG), but this time incorporating skip connections between LDGs. We repeat this procedure while integrating the recursive concatenations through the LDGs into a single output. The output of each LDG is concatenated to the input of the next one. In order to facilitate access to local information, the final output of the network receives the concatenation of the outputs of all the LDGs. Therefore, the model incorporates features from multiple layers. This strategy makes information propagation efficient due to the multi-level representation and many shortcut connections. Inspired by MemNet [111], we then introduce a  $1 \times 1$  convolutional layer to adaptively merge the output information, as directly using these concatenated features would greatly increase computational complexity. The output of these hierarchical features can be formulated as

$$\mathbf{f}_D = \text{conv}_{1 \times 1}([\mathbf{f}_0, \dots, \mathbf{f}_{D-1}]), \quad (4.5)$$

where  $[\mathbf{f}_0, \dots, \mathbf{f}_{D-1}]$  refers to the concatenation of feature maps produced by LDGs.

To make sure that no information is lost before the reconstruction step, we incorporate a long-range skip connection to grant access to the original information and encourage back-propagation of gradients from the output of the feature extractor to the first  $3 \times 3$  convolution layer. We also include a global average pooling followed by a  $1 \times 1$  convolution, to fully capture channel-wise dependencies from the aggregated information. The final output before the reconstruction step is then,

$$\mathbf{h} = \lambda_0 \mathbf{f}_D + \lambda_1 \sigma(\text{conv}_{1 \times 1}(\text{GAP}(\text{conv}_{3 \times 3}(I^{LR})))), \quad (4.6)$$

where  $\sigma$  denotes the ReLU activation, GAP denotes global average pooling, and  $\lambda_0$  and  $\lambda_1$  are learned parameters.

### 4.2.2 Overscaling Module

In this work, we introduce a new inductive bias in SISR architectures so as to generate images that are more accurate and present fewer artifacts. We hypothesize that, since overscaling produces multiple values for the same pixel, these values act as an ensemble of predictions thus reducing noise when combined to produce the final image.

Let us consider  $N$  the maximum scale factor addressed by the network. We first generate an intermediate representation of the final image consisting of overscaled maps  $H^{OHR}$ , with an overscale factor  $(N + 1)$  times larger. Thus, given the features  $\mathbf{h}$  extracted from  $I^{LR}$ , we use a  $3 \times 3$  convolutional layer followed by the strided sub-pixel convolutional layer proposed in [104] to upscale the features  $\mathbf{h}$  to  $H^{OHR}$ :

$$H^{OHR} = \text{pixelshuffle}(\text{conv}_{3 \times 3}(\mathbf{h})). \quad (4.7)$$

To obtain the final output of the overscaling module, we further include a second long-range skip connection from the original  $I^{LR}$  image. The final HR image is obtained by adjusting the overscaled maps and incorporating them into the naïve upscaling of the original LR image:

$$\hat{I}^{HR} = \text{bicubic}_1(\text{conv}_{3 \times 3}(H^{OHR})) + \text{bicubic}^1(I^{LR}). \quad (4.8)$$

Hence, we could think of the whole network as learning how to *refine* or *correct* a naïve bicubic upscaling of the low-resolution input, in order to bring it closer to the actual high-resolution counterpart. Since the final  $\hat{I}^{HR}$  images are obtained with an efficient non-parametric interpolation, we are able to produce multiple scales with negligible computational cost, and only using differentiable operations.

### 4.2.3 Multi-Scale Loss

We propose the minimization of a multi-scale loss to optimize the network. We choose a finite set of scale factors  $S = \{s_1 \dots s_n\}$ , all within the interval of scales targeted by the network. The training process is conducted as follows: Once the network has reconstructed the HR image, images at the target scales are obtained through a bank of bicubic interpolators,  $\hat{I}_s^{HR} = \text{bicubic}_1(\hat{I}^{HR}, s)$ . Then, we minimize the following loss function:

$$\mathcal{L} = \sum_{s \in S} |\hat{I}_s^{HR} - \text{bicubic}_1(I^{HR}, s)|. \quad (4.9)$$

Training with this multi-scale loss at different target scales simultaneously provides additional supervision to the model, compared to a single-scale training. As a result, the model is enforced to learn how to generate highly representative overscaled maps, from which HR images at arbitrary scales can be recovered accurately, hence enforcing the generalization capability of the network across scales.

#### 4.2.4 Difference with Other SR Methods

**Difference with MemNet.** MemNet stands for the very deep persistent memory network proposed in [111]. The most crucial part of MemNet is the stacked memory blocks. Inside the memory blocks of MemNet, the output features of each recursive unit are concatenated at the end of the network and then fused with a  $1 \times 1$  convolution. The motivation of MemNet and ours is similar. The key difference is that we fuse the features at every possible point inside the local and global dense groups (LDGs, GDG), which boosts the representation power via the additional convolution layers and non-linearity. On the other hand, MemNet takes upsampled images as input. Hence, the number of multi-adds of MemNet is larger than ours. The input of our model is an LR image and we upsample it at the end of the network in order to achieve computational efficiency.

**Difference with SRDenseNet.** SRDenseNet [60] adopts dense blocks and skip connections. In this method, all feature levels are combined at the end of the final dense block. Differently, we connect all the RBs at the end of each local dense group (LDG) and do the same strategy inside the global dense group (GDG). Therefore, the model incorporates features from multiple layers. This strategy makes information propagation efficient due to the multi-level representation and facilitates the model to restore the details and context of the image simultaneously. Moreover, we gather local information progressively with the  $1 \times 1$  convolution layer, but SRDenseNet preserves these dense block features via concatenation operations.

### 4.3 Experimental Results

In this section, we evaluate the performance of our models on series of standard benchmark datasets. In addition, we provide comparison with state-of-the-art algorithms.

#### 4.3.1 Settings

**Datasets and metrics.** We use the high-quality DIV2K dataset for training. Several benchmark datasets are used for testing, namely Set5 [9], Set14 [138], B100 [2], Urban100 [45] and Manga109 [82]. SR results are evaluated with two commonly used metrics: PSNR and SSIM, on the Y channel of the YCbCr space. Furthermore, we adopt the Perceptual Index (PI) [11], which can avoid the situation where over-smoothed images may present a higher PSNR or SSIM when the performances of two methods are similar.

**Degradation models.** To comprehensively illustrate the efficacy of the proposed method, three degradation models are used to simulate LR images, following [145]. The first one, denoted by **BI**, consists of generating LR images by bicubic downsampling ground truth HR images with  $\times 2$ ,  $\times 3$ ,  $\times 4$ . The second one, denoted by **BD**, first performs bicubic downsampling on HR images with  $\times 3$  and then blurs the images with a Gaussian kernel of size  $7 \times 7$  and standard deviation 1.6. Finally, we further produce LR images in a third challenging way, denoted by **DN**, by carrying out bicubic downsampling followed by additive Gaussian noise, with the noise level of 30.

**Implementation details.** We denote our original model as OverNet and further introduce OverNet w/o OSM (OverNet without overscaling module). We used  $64 \times 64$  RGB input patches from the LR images for training. LR patches were sampled randomly and augmented with random horizontal flips and  $90^\circ$  rotation. The number of LDGs and RBs was set to 3 in all experiments. We trained our models with the ADAM optimizer [54]. The mini-batch size was set to 64, and the learning rate to the maximum convergent value  $10^{-3}$ , applying weight normalization in all convolutional layers [134]. The learning rate was decreased by half every  $2 \times 10^5$  back-propagation iterations. We implemented our networks using the PyTorch framework [91] and trained them on a NVIDIA RTX 3090 GPU.

### 4.3.2 Ablation Studies

To further investigate the performance behavior of the proposed methods, we analyze their effect on model training via an ablation study. We first show how skip connections inside the proposed local and global dense groups affect the performance of OverNet. Next, we conduct an ablation experiment to analyze the effect of OSM and the multi-scale loss.

#### Feature Extractor Ablation

In this section, we investigate the effect of skip connections (SCs) inside the local and global dense groups (LDG, GDG). In this work, SCs contain concatenation and  $1 \times 1$  convolutions. The small changes in the number of parameters between columns are due to the removal of SCs with  $1 \times 1$  convolutions. Table 4.1 presents the results of experiments conducted on Urban100 dataset at scale  $\times 3$ .

It can be observed that the model which used SCs only in GDG attains better performance than the one without SCs (config 1 which is ResNet+OSM) because the short connections inside the GDG effectively carry the information from intermediate to higher layers. Furthermore, by gathering all features before the upscaling

Table 4.1 – Effects of skip connections (SCs) in local and global dense groups (LDG, GDG) measured on Urban100 with  $\times 3$ . The best result is **highlighted**.

Configurations	1	2	3	4
SCs in LDGs	×	✓	×	✓
SCs in GDG	×	×	✓	✓
#Params	695K	806K	732K	943K
PSNR	28.20	28.19	28.24	<b>28.29</b>

module, the model can better leverage multi-level representations.

On the other hand, as discussed in [39], multiplicative manipulations such as  $1 \times 1$  convolutions on the shortcut connection can hamper information propagation, and complicate optimization. Similarly, SCs in LDGs behave as shortcut connections inside the residual blocks. Thus, it is natural to expect performance degradation when the global SCs are deactivated. This is because the global SCs ease the information propagation while the local connections are being learned. Therefore, when OverNet uses SCs in both LDGs and GDG, it outperforms all the three models.

In detail, information propagates globally via SCs used in GDG, and information flows in the LDGs are fused with the ones that come through global connections. By doing so, information is transmitted by multiple shortcuts and thus mitigates the vanishing gradient problem: the advantage of multi-level representation is leveraged by the SCs in GDG, which help the information to propagate to higher layers.

### Effect of the OSM Across Scales

Here we analyze the benefits of incorporating the OSM module and also explore the influence of different interpolation methods on the reconstruction. We run the following experiments: (i) directly using pixelshuffle to generate the images without overscaling feature maps, followed by a bicubic interpolation to downscale to arbitrary scales; (ii) downscaling with bilinear interpolation the overscaled feature maps produced by pixelshuffle and (iii) doing the same as (ii) with bicubic interpolation. As shown in Table 4.2, superior results are achieved by a large margin when the proposed overscaling method is applied. These experiments suggest that, contrary to common practice in the field, the addition of overscaling strongly increases reconstruction accuracy. Best results are achieved using OSM with bicubic interpolation, which in turn yields better results than bilinear.

Table 4.2 – PSNR results of different OSM upscaling methods trained for arbitrary scales. The test dataset is B100. Best results are **highlighted**, second best underlined.

Experiment	Scale									
	×1.1	×1.2	×1.3	×1.4	×1.5	×1.6	×1.7	×1.8	×1.9	×2.0
Pixelshuffle	42.40	39.71	38.10	36.75	35.60	34.70	33.96	33.30	33.65	32.22
OSM-bilinear	<u>42.63</u>	<u>39.89</u>	<u>38.15</u>	<u>36.83</u>	<u>35.70</u>	<u>34.78</u>	<u>34.05</u>	<u>33.37</u>	<u>32.76</u>	<u>32.31</u>
OSM-bicubic	<b>42.74</b>	<b>39.95</b>	<b>38.19</b>	<b>36.87</b>	<b>35.74</b>	<b>34.80</b>	<b>34.10</b>	<b>33.42</b>	<b>32.81</b>	<b>32.34</b>
Meta-RDN	<u>42.82</u>	<u>40.40</u>	<u>38.28</u>	<u>36.95</u>	<u>35.86</u>	<u>34.90</u>	<u>34.13</u>	<u>33.45</u>	<u>32.86</u>	<u>32.35</u>
OSM-RDN	<b>42.93</b>	<b>40.48</b>	<b>38.42</b>	<b>37.06</b>	<b>36.01</b>	<b>35.02</b>	<b>34.25</b>	<b>35.53</b>	<b>32.95</b>	<b>32.46</b>
	×2.1	×2.2	×2.3	×2.4	×2.5	×2.6	×2.7	×2.8	×2.9	×3.0
Pixelshuffle	31.60	31.22	30.75	30.50	30.27	29.95	29.73	29.42	29.17	29.14
OSM-bilinear	<u>31.71</u>	<u>31.29</u>	<u>30.84</u>	<u>30.55</u>	<u>30.37</u>	<u>30.02</u>	<u>29.77</u>	<u>29.52</u>	<u>29.30</u>	<u>29.26</u>
OSM-bicubic	<b>31.75</b>	<b>31.34</b>	<b>30.86</b>	<b>30.65</b>	<b>30.42</b>	<b>30.11</b>	<b>29.83</b>	<b>29.64</b>	<b>29.36</b>	<b>29.30</b>
Meta-RDN	<b>31.82</b>	<u>31.41</u>	<u>31.06</u>	<b>30.62</b>	<u>30.45</u>	<u>30.13</u>	<b>29.82</b>	<u>29.67</u>	<b>29.40</b>	<u>29.30</u>
RDN-OSM	<u>31.75</u>	<b>31.46</b>	<b>31.10</b>	<u>30.60</u>	<b>30.48</b>	<b>30.15</b>	<u>29.79</u>	<b>29.71</b>	<u>29.35</u>	<b>29.38</b>
	×3.1	×3.2	×3.3	×3.4	×3.5	×3.6	×3.7	×3.8	×3.9	×4.0
Pixelshuffle	28.78	28.70	28.50	28.30	28.14	28.10	28.72	27.74	27.60	27.65
OSM-bilinear	<u>28.81</u>	<u>28.77</u>	<u>28.62</u>	<u>28.49</u>	<u>28.23</u>	<u>28.22</u>	<u>28.90</u>	<u>27.82</u>	<u>27.79</u>	<u>27.75</u>
OSM-bicubic	<b>28.90</b>	<b>28.81</b>	<b>28.66</b>	<b>28.51</b>	<b>28.26</b>	<b>28.25</b>	<b>28.96</b>	<b>27.84</b>	<b>27.83</b>	<b>27.80</b>
Meta-RDN	<u>28.87</u>	<b>28.79</b>	<u>28.68</u>	<u>28.54</u>	<u>28.32</u>	<b>28.27</b>	<b>28.04</b>	<u>27.92</u>	<u>27.82</u>	<u>27.75</u>
RDN-OSM	<b>28.96</b>	<u>28.70</u>	<b>28.80</b>	<b>28.64</b>	<b>28.41</b>	<u>28.23</u>	<u>28.00</u>	<b>27.97</b>	<b>27.89</b>	<b>27.83</b>

In addition, we compare our results with Meta-RDN [42], the only method in the literature (to our knowledge) able to carry out SISR at non-integer scales. Meta-RDN is a heavier state-of-the-art model with 22M parameters. For a fair comparison, we trained Meta-RDN by replacing its meta-upscale module with OSM (RDN-OSM), while applying their original training settings. RDN-OSM achieves better or comparable performance.

### OSM Across Architectures

The aim of this section is to demonstrate the benefits of our OSM hold across architectures. To this end, we use state-of-the-art networks including CARN [1],

Table 4.3 – Average PSNR of state-of-the-art methods using OSM instead of their typical upsampling module. The best results are **highlighted**.

Dataset	Scale	CARN	CARN-OSM	EDSR	EDSR-OSM	RDN	Meta-RDN	RDN-OSM	RCAN	RCAN-OSM
Set5	×2	37.76	<b>37.81</b>	38.20	<b>38.26</b>	38.24	-	<b>38.31</b>	38.27	<b>38.36</b>
	×3	34.29	<b>34.35</b>	34.76	<b>34.80</b>	34.71	-	<b>34.77</b>	34.74	<b>34.81</b>
	×4	32.13	<b>32.15</b>	32.62	<b>32.66</b>	32.47	-	<b>32.58</b>	32.63	<b>32.70</b>
Set14	×2	33.52	<b>33.60</b>	34.02	<b>34.08</b>	34.01	34.04	<b>34.11</b>	34.12	<b>34.19</b>
	×3	30.29	<b>30.36</b>	30.66	<b>30.71</b>	30.57	30.55	<b>30.63</b>	30.65	<b>30.74</b>
	×4	28.60	<b>28.68</b>	28.94	<b>29.01</b>	28.81	28.84	<b>28.91</b>	28.87	<b>28.93</b>
Urban100	×2	31.92	<b>32.01</b>	33.10	<b>33.15</b>	32.89	-	<b>32.96</b>	33.34	<b>33.40</b>
	×3	28.06	<b>28.12</b>	29.02	<b>29.09</b>	28.80	-	<b>28.91</b>	29.09	<b>29.15</b>
	×4	26.07	<b>26.13</b>	26.86	<b>26.91</b>	26.61	-	<b>26.70</b>	26.82	<b>26.90</b>

EDSR[71], RDN[145], Meta-RDN [42], and RCAN[144] as references. We replaced their typical upsample modules with our overscaling module (CARN-OSM, EDSR-OSM, RDN-OSM, and RCAN-OSM in Table 4.3) and trained them on DIV2K for all scale factors while applying their original training settings.

The results of this experiment are listed in Table 4.3. It can be observed that all the methods with OSM have higher PSNR than the corresponding baselines at all scale factors. This indicates that OSM is robust and orthogonal to the feature extractor chosen, and it moderately improves the SR performance (PSNR: +0.07dB in average).

### Generalization Across Scales

By construction, the overscaling factor in our architecture is always  $(N + 1)$  when targeting a maximum scale of  $N$ , c.f. Section 4.2.2. The following experiments investigate the generalization capability of models that target a maximum scale  $N$  across lower scales  $M \leq N$ . To this end, we trained models for  $N \in \{2, 3, 4\}$  and evaluated them across scales. Table 4.4 illustrates the experimental results. It can be observed that models trained to target larger scales yield better PSNR scores for all scale factors. This demonstrates the generalization capabilities of the proposed architecture across scales, as it is not necessary to train a dedicated model for each scale. Instead, training a larger scale seems to be always beneficial for lower scales. Moreover, the cost to pay in terms of additional parameters is low. Note that ×4 and ×8 are composed of multiple consecutive ×2 operations, thus introducing fewer



Table 4.4 – Average PSNR to show the performance of OverNet across scales. The test dataset is Set5. Best results are **highlighted**.

Overscaling factor	Parameters	Scales		
		×2	×3	×4
×3	927K	38.11	–	–
×4	943K	38.12 (+0.01dB)	34.49	–
×5	1079K	38.14 (+0.03dB)	34.51 (+0.02dB)	32.32
×8	955K	<b>38.15 (+0.04dB)</b>	<b>34.52 (+0.03dB)</b>	<b>32.36(+0.04dB)</b>

parameters. Overscaling to higher scales slightly improves the PSNR at the expense of more computation. For the rest of the experiments, we overscale to  $N + 1$  since it still provides significant improvement at a slightly higher computational cost.

### Effect of Multi-Scale Loss

Multi-scale learning can process multiple scales with a single trained model, while most state-of-the-art algorithms require training separate models for each supported scale. This property targets real-world applications, where the output size is usually fixed but the input LR scale can vary. Moreover, the multi-scale loss acts as a regularizer, enforcing the generalization of the network across scales and improving performance. As a result, the model is able to maintain accurate reconstruction results across scales. Table 4.5 shows experimental results, where the model trained with multi-scale loss achieves better performance with a large margin.

Table 4.5 – Effect of multi-scale loss. OverNet-S uses single-scale loss, OverNet-M multi-scale loss. Best results are **highlighted**.

Methods	Scale	Set5	B100	Urban100
OverNet-S	×2	38.11	32.23	32.39
	×3	34.49	29.13	28.29
	×4	32.32	27.59	26.23
OverNet-M	×2	<b>38.19 (+0.08dB)</b>	<b>32.34 (+0.11dB)</b>	<b>32.47 (+0.08dB)</b>
	×3	<b>34.56 (+0.07dB)</b>	<b>29.26 (+0.13dB)</b>	<b>28.36 (+0.07dB)</b>
	×4	<b>32.40 (+0.08dB)</b>	<b>27.66 (+0.07dB)</b>	<b>26.30 (+0.07dB)</b>

### 4.3.3 Comparison with State-of-the-art Methods

In this section, OverNet w/o OSM and OverNet are compared to other lightweight and heavy state-of-the-art SR methods. A self-ensemble method [116] is also used to further improve the performance of the OverNet (denoted as OverNet+).

#### Results with BI Degradation Models

We compare the proposed OverNet and OverNet+ with several lightweight state-of-the-art SR methods: VDSR [52], DRCN [53], SRDenseNet [118], SEINet [19], SRResNet [60], CARN [1], IMDN [47], SRFBN-S [69], A2F-S [130], CBPN [148], LAPAR-A [67], MADNet [59], FALSAR-A [20], DPN [70], HDRN [49], and OISR-RK2 [40]. We also train OverNet by replacing its OSM with the typical pixelshuffle upsampling (OverNet w/o OSM). For fair comparison, we train our models individually for each scale factor, including  $\times 2$ ,  $\times 3$  and  $\times 4$ . We test our models on different benchmarks with PSNR and SSIM.

Table 4.6 shows quantitative evaluation results, including the number of parameters and the number of multiplications and additions (Multi-Adds), for a more informative comparison. Multi-Adds were calculated with  $1280 \times 720$  SR images at all scales. Note that, in this table, we only compare models that have a roughly similar number of parameters as ours. OverNet and OverNet+ exceeds all the previous methods on numerous benchmark datasets. OverNet w/o OSM also achieves comparable or better results. Results show that both OSM and the proposed feature extractor independently increase PSNR when compared to other SR methods. Finally, combining the proposed feature extractor and OSM together further increases performance.

In addition, we present qualitative results in Figure 4.2. It can be observed that most of the compared methods would produce noticeable artifacts and produce blurred edges. In contrast, our method can recover sharper and clearer edges, more faithful to the ground truth. For example in images `Img_073` and `Img_099`, we see that, unlike OverNet, most of the compared methods fail to recover the definition and orientation of the lines of the blue buildings. For image `Img_076`, the texture of the predicted SR images for all compared methods contains blur or aliasing. In contrast, our proposal partially recovers the brick pattern, resulting in a more faithful SR image.

#### Results with BD and DN Degradation Models

Following [145], we show the results obtained after applying **BD** and **DN** degradation models, and compare to eight SR methods: Bicubic, SPMSR [93], SRCNN [23], FSRCNN [24], VDSR [52], IRCNN\_G [140], IRCNN\_C [140], and SRMD(NF) [141].

## Chapter 4. Lightweight Multi-Scale Super-Resolution with Overscaling Network

Table 4.6 – Average PSNR/SSIM values for models with the same order of magnitude of parameters. Performance is shown for scale factors  $\times 2$ ,  $\times 3$  and  $\times 4$  with BI degradation model. The best and second best results are highlighted in red and blue respectively.

Scale Method	Params Multi-Adds		Set5		Set14		B100		Urban100		Manga109		
			PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
$\times 2$	VDSR [52]	0.7M	613G	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.22	0.9729
	DRCN [53]	1.8M	17,974G	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.63	0.9723
	SEINet [19]	1M	226G	37.89	0.9598	33.61	0.9160	32.08	0.8984	-	-	-	-
	CARN [1]	1.6M	223G	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765
	SRFBN-S [69]	0.3M	680G	37.78	0.9597	33.35	0.9156	32.00	0.8970	31.41	0.9207	38.06	0.9757
	A2F-S [130]	0.3M	306.1G	37.79	0.9597	33.32	0.9152	31.99	0.8972	31.44	0.9211	38.11	0.9757
	CBPN [148]	1M	240.7G	37.90	0.9590	33.60	0.9171	32.17	0.8989	32.14	0.9279	-	-
	MADNet [59]	0.9M	187.1G	37.94	0.9604	33.46	0.9167	32.10	0.8988	31.74	0.9246	-	-
	FALSR-A[20]	1M	234.7G	37.82	0.9595	33.55	0.9168	32.12	0.8987	31.93	0.9256	-	-
	HDRN [49]	0.9M	316.2G	37.75	0.9590	33.49	0.9150	32.03	0.8980	31.87	0.9250	38.07	0.9770
	DPN [70]	0.8M	140G	37.52	0.9586	33.08	0.9129	31.89	0.8958	30.82	0.9144	-	-
	LAPAR-A [67]	0.5M	171G	38.01	0.9605	33.62	0.9183	32.19	0.8999	32.10	0.9283	38.67	0.9772
	IMDN [47]	0.7M	158.8G	38.00	0.9605	33.63	0.9177	32.19	0.8996	32.17	0.9283	38.88	0.9774
	OISR-RK2 [40]	1.4M	316.2G	38.02	0.9605	33.62	0.9178	32.20	0.9000	32.21	0.9290	-	-
	OverNet w/o OSM	0.9M	180G	38.08	0.9607	33.69	0.9179	32.18	0.8999	32.35	0.9305	38.91	0.9779
OverNet	0.9M	189G	<b>38.11</b>	<b>0.9609</b>	<b>33.73</b>	<b>0.9186</b>	<b>32.23</b>	<b>0.9004</b>	<b>32.39</b>	<b>0.9309</b>	<b>38.95</b>	<b>0.9781</b>	
OverNet+	0.9M	189G	<b>38.17</b>	<b>0.9613</b>	<b>33.79</b>	<b>0.9190</b>	<b>32.29</b>	<b>0.9009</b>	<b>32.46</b>	<b>0.9315</b>	<b>38.99</b>	<b>0.9787</b>	
$\times 3$	VDSR [52]	0.7M	613G	33.66	0.9213	29.77	0.8314	28.82	0.7976	27.14	0.8279	37.22	0.9750
	DRCN [53]	1.7M	17,974G	33.82	0.9226	29.76	0.8311	28.80	0.7963	27.15	0.8276	32.24	0.9343
	CARN [1]	1.6M	119G	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440
	SRFBN-S [69]	0.4M	832G	34.20	0.9255	30.10	0.8372	28.96	0.8010	27.66	0.8415	33.02	0.9404
	A2F-S [130]	0.3M	136.1G	34.06	0.9241	30.08	0.8370	28.92	0.8006	27.57	0.8392	32.86	0.9394
	MADNet [59]	0.9M	88.4G	34.26	0.9262	30.29	0.8410	29.04	0.8033	27.91	0.8464	-	-
	HDRN [49]	0.9M	187.1G	34.24	0.9240	30.23	0.8400	28.96	0.8040	27.93	0.8490	33.17	0.9420
	DPN [70]	0.8M	114.2G	33.71	0.9222	29.80	0.8320	28.84	0.7981	27.17	0.8282	-	-
	LAPAR-A [67]	0.5M	114G	34.36	0.9267	30.34	0.8421	29.11	0.8054	28.15	0.8523	33.51	0.9441
	IMDN [47]	0.7M	71.5G	34.36	0.9270	30.32	0.8417	29.09	0.8046	28.17	0.8519	33.61	0.9445
	OISR-RK2 [40]	1.6M	160.1G	34.39	0.9272	30.35	0.8420	29.11	0.8058	28.24	0.8544	-	-
	OverNet w/o OSM	0.9M	111.5G	34.45	0.9277	30.39	0.8431	29.12	0.8059	28.24	0.8544	33.74	0.9446
	OverNet	0.9M	118.8G	<b>34.49</b>	<b>0.9279</b>	<b>30.43</b>	<b>0.8436</b>	<b>29.15</b>	<b>0.8063</b>	<b>28.29</b>	<b>0.8546</b>	<b>33.78</b>	<b>0.9451</b>
	OverNet+	0.9M	118.8G	<b>34.54</b>	<b>0.9284</b>	<b>30.49</b>	<b>0.8442</b>	<b>29.21</b>	<b>0.8069</b>	<b>28.35</b>	<b>0.8552</b>	<b>33.84</b>	<b>0.9456</b>
	$\times 4$	VDSR [52]	0.7M	613G	31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524	28.83
DRCN [53]		1.8M	17,974G	31.54	0.8850	29.19	0.7720	27.32	0.7280	25.12	0.7560	29.09	0.8845
SEINet [19]		1.4M	83G	32.05	0.8934	28.49	0.7783	27.44	0.7325	-	-	-	-
SRDenseNet [118]		2M	390G	32.00	0.8931	28.50	0.7782	27.53	0.7337	26.05	0.7819	30.41	0.9071
SRResNet [60]		1.5M	-	32.05	0.8910	28.53	0.7804	27.57	0.7354	26.07	0.7839	-	-
CARN [1]		1.6M	91G	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084
SRFBN-S [69]		0.5M	1,037G	31.98	0.8923	28.45	0.7779	27.44	0.7313	25.71	0.7719	29.91	0.9008
A2F-S [130]		0.3M	77.2G	31.87	0.8900	28.36	0.7760	27.41	0.7305	25.58	0.7685	29.77	0.8987
CBPN [148]		1.2M	97.9G	32.21	0.8944	28.63	0.7813	27.58	0.7356	26.14	0.7869	-	-
MADNet [59]		1M	54.1G	32.11	0.8939	28.52	0.7799	27.52	0.7340	25.89	0.7782	-	-
HDRN [49]		0.9M	316.2G	32.23	0.8960	28.58	0.7810	27.53	0.7370	26.09	0.7870	30.43	0.9080
DPN [70]		0.8M	140G	31.42	0.8849	28.07	0.7688	27.30	0.7256	25.25	0.7546	-	-
LAPAR-A [67]		0.7M	94G	32.15	0.8944	28.61	0.7818	27.61	0.7366	26.14	0.7871	30.42	0.9074
IMDN [47]		0.7M	40.9G	32.21	0.8948	28.58	0.7811	27.56	0.7353	26.04	0.7838	30.45	0.9075
OISR-RK2 [40]		1.5M	114.2G	32.14	0.8947	28.63	0.7819	27.60	0.7369	26.17	0.7888	-	-
OverNet w/o OSM	0.9M	82.3G	32.28	0.8963	28.64	0.9729	27.64	0.7372	26.21	0.7891	30.48	0.9106	
OverNet	0.9M	89G	<b>32.32</b>	<b>0.8965</b>	<b>28.69</b>	<b>0.9733</b>	<b>27.69</b>	<b>0.7374</b>	<b>26.23</b>	<b>0.7895</b>	<b>30.53</b>	<b>0.9110</b>	
OverNet+	0.9M	89G	<b>32.38</b>	<b>0.8970</b>	<b>28.75</b>	<b>0.9739</b>	<b>27.76</b>	<b>0.7380</b>	<b>26.28</b>	<b>0.7901</b>	<b>30.59</b>	<b>0.9115</b>	

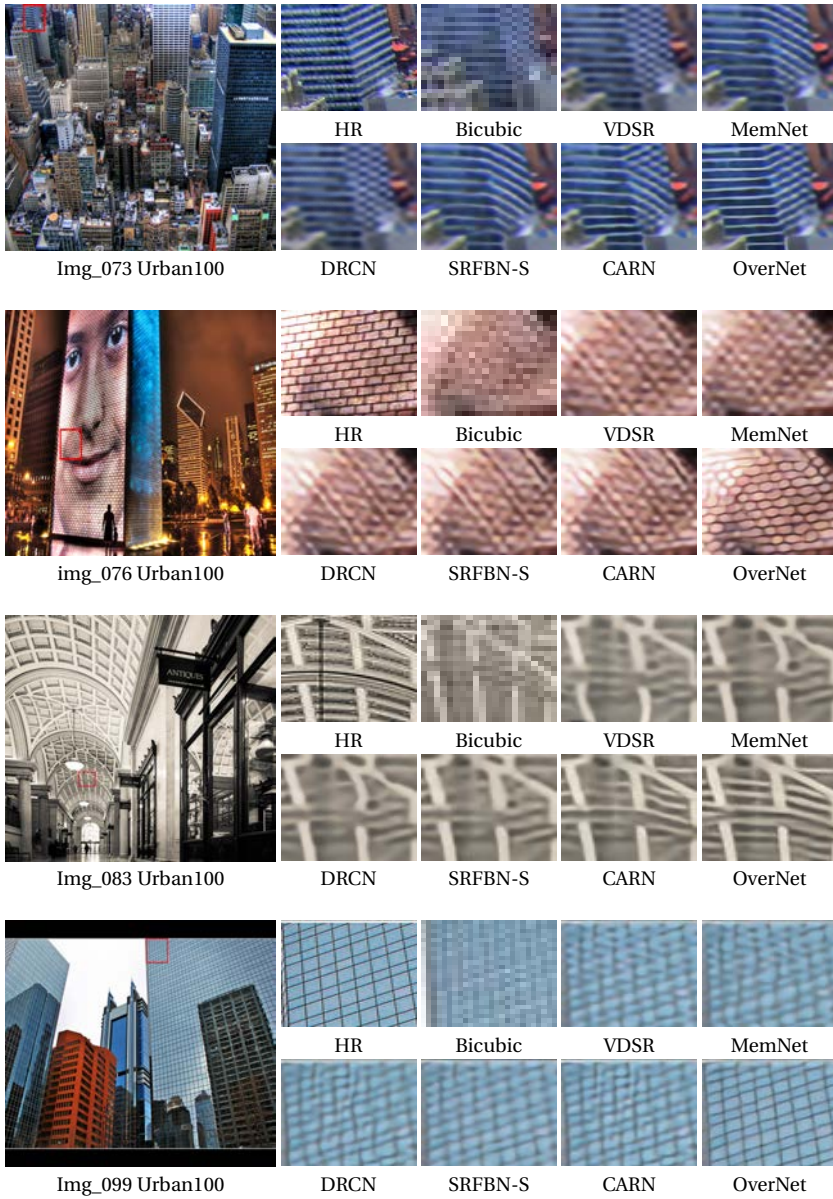


Figure 4.2 – Visual results of **BI** degradation model for scale factor  $\times 4$ .

Table 4.7 – Quantitative results with **BD** and **DN** degradation models. The best and second best results are highlighted in **red** and **blue** respectively.

Methods	Degrad.	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	BD	28.78	0.8308	26.38	0.7271	26.33	0.6918	23.52	0.6862	25.46	0.8149
	DN	24.01	0.5369	22.87	0.4724	22.92	0.4449	21.63	0.4687	23.01	0.5381
SPMSR [93]	BD	32.21	0.9001	28.89	0.8105	28.13	0.7740	25.84	0.7856	29.64	0.9003
	DN	–	–	–	–	–	–	–	–	–	–
SRCNN [23]	BD	32.05	0.8944	28.80	0.8074	28.13	0.7736	25.70	0.7770	29.47	0.8924
	DN	25.01	0.6950	23.78	0.5898	23.76	0.5538	21.19	0.5737	23.75	0.7148
FSRCNN [24]	BD	26.23	0.8124	24.44	0.7106	24.86	0.6832	22.04	0.6745	23.04	0.7927
	DN	24.18	0.6932	32.02	0.5856	23.41	0.5556	21.15	0.5682	22.39	0.7111
VDSR [52]	BD	33.25	0.9150	29.46	0.8244	28.57	0.7893	26.61	0.8136	31.06	0.9234
	DN	25.20	0.7183	24.00	0.6112	24.00	0.5749	22.22	0.6096	24.20	0.7525
IRCNN_G [140]	BD	33.38	0.9182	29.63	0.8281	28.65	0.7922	26.77	0.8154	31.15	0.9245
	DN	25.70	0.7379	24.45	0.6305	24.28	0.5900	22.90	0.6429	24.88	0.7765
IRCNN_C [140]	BD	29.55	0.8246	27.33	0.7135	26.46	0.6572	24.89	0.7172	28.68	0.7701
	DN	26.18	0.7430	24.68	0.6300	24.52	0.5850	22.63	0.6205	24.74	0.7701
SRMDNF [141]	BD	34.09	0.9242	30.11	0.8364	28.98	0.8009	27.50	0.8370	32.97	0.9391
	DN	27.74	0.8026	26.13	0.6924	25.64	0.6495	24.28	0.7092	26.72	0.8590
OverNet w/o OSM	BD	34.45	0.9274	30.40	0.8432	29.02	0.8068	28.19	0.8537	33.99	0.9460
	DN	28.41	0.8145	26.54	0.6933	25.87	0.7069	24.88	0.7356	27.88	0.8573
OverNet	BD	<b>34.59</b>	<b>0.9281</b>	30.46	0.8441	29.13	0.8074	28.24	0.8543	<b>34.04</b>	<b>0.9467</b>
	DN	<b>28.49</b>	<b>0.8153</b>	<b>26.62</b>	<b>0.7106</b>	<b>25.95</b>	<b>0.6578</b>	<b>24.93</b>	<b>0.7365</b>	<b>28.04</b>	<b>0.8593</b>
OverNet+	BD	<b>34.65</b>	<b>0.9287</b>	<b>30.52</b>	<b>0.8446</b>	<b>29.19</b>	<b>0.8080</b>	<b>28.30</b>	<b>0.8549</b>	<b>34.10</b>	<b>0.9472</b>
	DN	<b>28.55</b>	<b>0.8159</b>	<b>26.69</b>	<b>0.7111</b>	<b>26.01</b>	<b>0.6585</b>	<b>24.98</b>	<b>0.7371</b>	<b>28.10</b>	<b>0.8599</b>
RDN [145]	BD	34.58	0.9280	<b>30.53</b>	<b>0.8447</b>	<b>29.23</b>	<b>0.8079</b>	<b>28.46</b>	<b>0.8582</b>	33.97	0.9465
	DN	28.47	0.8151	26.60	0.7101	25.93	0.6573	24.92	0.7364	28.00	0.8590

We also included the RDN [145] high-capacity model for reference.

As shown in Table 4.7, our models achieve the best PSNR and SSIM scores over other SR methods with similar capacity. It can be observed that RDN performs slightly better in some BD datasets but not in DN datasets. Thanks to OSM, OverNet is able to reduce the DN degradation to obtain better results when compared to RDN. It is worth noting that while RDN has 22M parameters, OverNet only has 0.9M parameters. The performance gains over other state-of-the-art methods are consistent with the visual results in Figure 4.3 and 4.4.

For **BD** degradation model (Figure 4.3), other methods were unable to remove blurring artifacts. In contrast, OverNet could suppress the blurring artifacts, recover sharper edges, and generate more accurate details in the SR images. Regarding

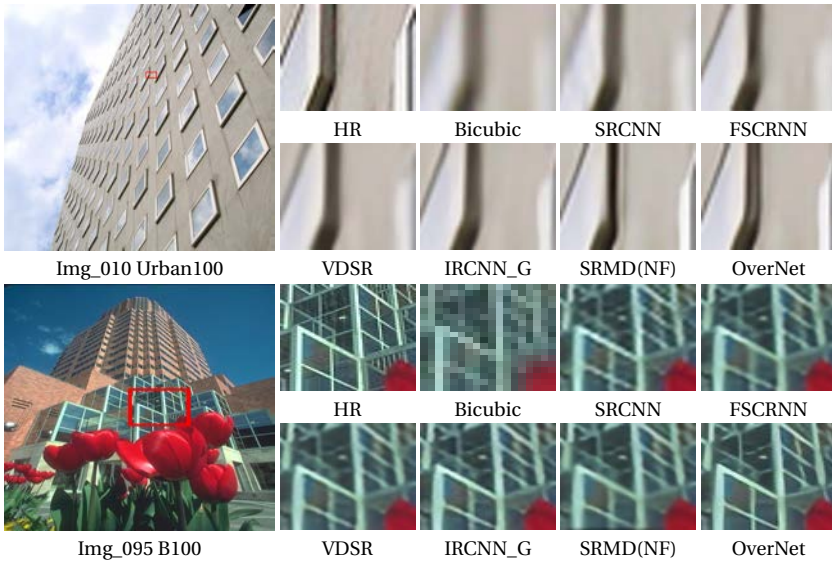


Figure 4.3 – Visual results of **BD** degradation model for scale factor  $\times 3$ .

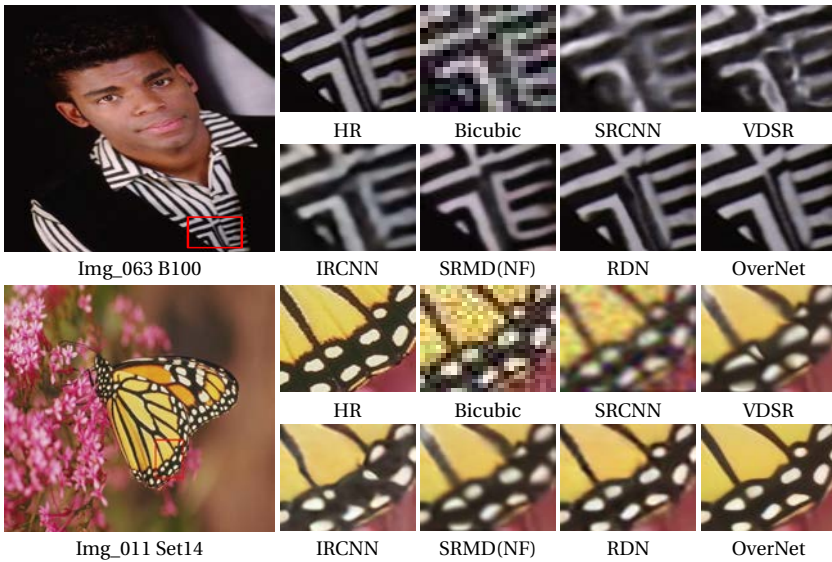


Figure 4.4 – Visual results of **DN** degradation models for scale factor  $\times 3$ .



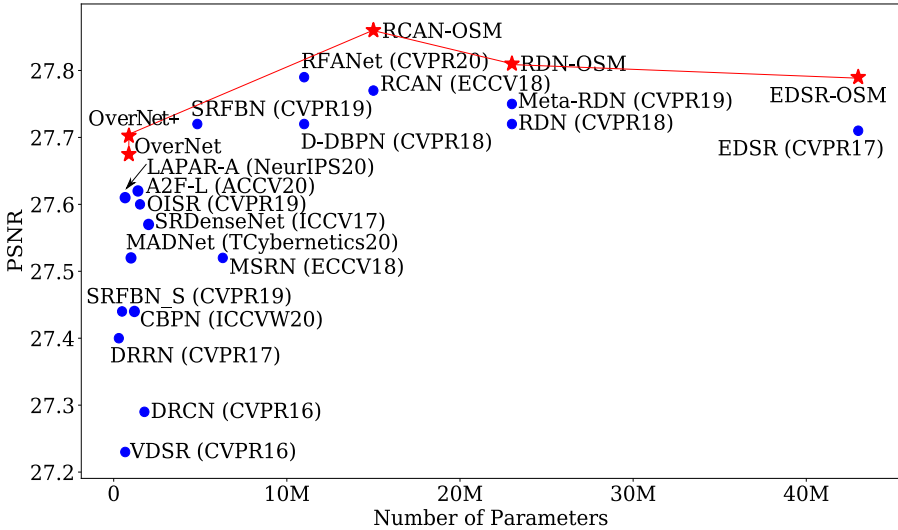


Figure 4.5 – Comparative capacity and performance of state-of-the-art SISR models. The red stars represents our methods.

DN degradation model (Figure 4.4), we observe that the noised details are difficult to recover by other methods. However, OverNet can not only handle the noise efficiently but also recover more details. This comparison indicates that OverNet is applicable for jointly image denoising and SR. These results with BD and DN degradation models demonstrate the effectiveness and robustness of our proposal.

**Memory Complexity and Running Time Analysis**

In Figure 4.5, we compare OverNet and OverNet+ against various benchmark algorithms in terms of network parameters and reconstruction PSNR, using the B100 dataset with a scale of  $\times 4$ . Our methods achieve the best SR results among all the lightweight SR networks with fewer parameters. In comparison with the networks with a large number of parameters, the proposed OverNet and OverNet+ achieve better or competitive results. This demonstrates our method can well balance the number of parameters and the reconstruction performance. We also replace the original upsample modules from different SR methods with OSM: RDN, EDSR, and RCAN (RDN+OSM, EDSR+OSM, and RCAN+OSM). It can be observed that all the methods with OSM have higher PSNR than the corresponding baselines.

We compare the running time of OverNet with recent light- and heavy-weight

Table 4.8 – Average running time (s) and memory consumption (MB) comparison on Urban100 for  $\times 4$  scale factor.

Methods	Params	Memory	Running Time(s)	PSNR
CARN [1]	1.5M	1,116	0.032	26.07
SRFBN-S [69]	0.5M	2,154	0.102	25.71
SRDenseNet [118]	2M	5,531	0.221	26.05
LAPAR-A [67]	0.7M	1,240	0.053	26.14
MSRN [65]	8M	2,731	0.070	26.04
RCAN [144]	16M	1,531	0.087	26.82
EDSR [71]	43M	2,731	0.035	26.64
RDN [145]	22M	2,480	1.268	26.61
D-DBPN [34]	10M	3,241	1.595	26.38
Meta-RDN [42]	22M	3,350	2.579	26.65
OverNet	0.9M	914	0.043	26.23

state-of-the-art approaches on Urban100, using a scale factor  $\times 4$ . The running time of each network is evaluated using its official code, on the same machine with an NVIDIA RTX 3090 GPU. OverNet is the fastest (see Table 4.8), reflecting its efficiency.

### Perceptual Metrics

Perceptual metrics better reflect the human judgment of image quality. In this paper, Perceptual Index (PI) [11] is chosen as the perceptual metric. Table 4.9 shows the PI for those works with publicly available source code, and the same order of magnitude in terms of parameters. We observe that our proposed model obtains

Table 4.9 – Perceptual index comparison of the proposed method with recent lightweight state-of-the-art methods on five datasets for  $\times 4$ . The lower is better. All of the output SR images are provided officially.

Methods	Params	Set5	Set14	B100	Urban100	Manga109
DRCN [53]	1.7M	6.451	5.945	5.897	5.791	5.563
CARN [1]	1.5M	6.297	5.775	5.700	5.540	5.132
SRFBN-S [52]	0.6M	6.451	5.775	5.702	5.549	5.010
SRDenseNet [118]	2M	6.128	5.615	5.653	5.526	4.762
IMDN [47]	0.7M	6.124	5.644	5.659	5.531	4.810
LAPAR-A [67]	0.7M	6.084	5.499	5.532	5.179	4.771
OverNet	0.9M	5.610	5.513	5.459	5.187	4.766



better results than all the compared baselines. This demonstrates the ability of the proposed OverNet for generating realistic images.

### 4.4 Summary

In this chapter, we have introduced OverNet, a novel efficient architecture for image super-resolution at arbitrary scales using a single model. OverNet consists of: (i) a lightweight feature extractor that enhances the flow of information to preserve details; (ii) an overscaling module that helps to generate accurate SR images at different scaling factors, and (iii) a multi-scale loss that improves training compared to dedicated single-scale models. Thanks to the OSM, we can train a single model for super-resolution at arbitrary scale factors. We have proved that the overscaling head can be flexibly applied to other SR models by simply replacing their upsampling module, thus improving their original performance. OverNet outperforms state-of-the-art algorithms with a reduced number of parameters and low computational requirements. The provided evidence suggests that the proposed overscaling method may help with other low-level image restoration tasks, such as denoising and dehazing.

## 5 Directional Variance Attention Networks

---

Recent advances in single-image super-resolution (SISR) explore the power of deep convolutional neural networks (CNNs) to achieve better performance. However, most of the progress has been made by scaling CNN architectures, which usually raise computational demands and memory consumption. This makes modern architectures less applicable in practice. In addition, most CNN-based SR methods rarely leverage the intermediate features that are helpful for final image recovery. In order to address these issues, we propose directional variance attention network (DiVANet), a computationally efficient yet accurate network for SISR. DiVANet leverages a novel directional variance attention (DiVA) to capture long-range spatial dependencies and exploit inter-channel dependencies simultaneously. Additionally, in order to make an efficient use of features in early layers, these are hierarchically aggregated into feature banks for posterior use at the network output. In parallel, DiVA extracts most relevant features from the network into attention banks for improving the final output and preventing information loss along with the successive operations inside the network. The processing is split into two independent paths of computation that can be simultaneously carried out, resulting in a highly efficient model for reconstructing fine details. Experimental results demonstrate the superiority of DiVANet over the state of the art in several datasets while maintaining a relatively low computation and memory footprint.

---

### 5.1 Motivation

Recently, convolutional neural networks (CNNs) have achieved great success for single image super-resolution (SISR). From SRCNN [23] (with only three convolutional layers) to MDSR [71] (with more than 160), network depth and overall performance have been dramatically growing over time. The increase of depth brings benefits in terms of representation power [43], but at the same time does not take into account the hierarchy of features and their interrelations across the whole

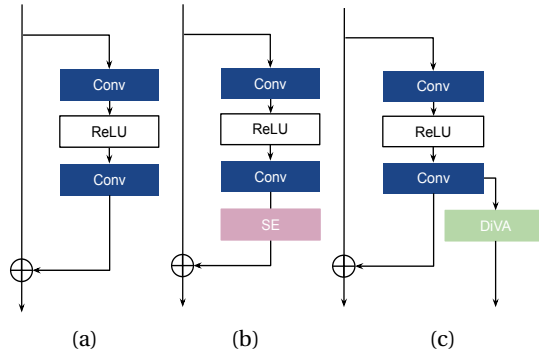


Figure 5.1 – (a) Basic residual block without attention mechanisms. (b) Residual channel attention block proposed in previous works. (c) Our proposed directional variance attention (DiVA), which has its own dedicated computational path.

architecture. Although SRDenseNet [118] and RDN [145] employ residual dense blocks to fuse different levels of features, the extreme connectivity pattern in their networks not only hinders their scalability when using large widths or depths but also increases computational demands and memory consumption dramatically, hence limiting the use of modern architectures in real-world scenarios. To tackle this issue, some SISR methods focus on lightweight architecture designs such as recursive operations with weight sharing [1, 6] and neural architecture search [20] in order to reduce the number of network parameters. Although these methods achieve good performance at moderate sizes, they do not fully utilize the features in early layers, which limits their performance. Therefore, it is of crucial importance to design a good lightweight network architecture which effectively computes multi-level feature representations for restoring high-quality HR images within the network, yet this remains to be explored.

Attention mechanisms have demonstrated great benefits at improving the performance of deep models for computer vision tasks. Recently, researchers have devoted great efforts to expand the application of attention mechanisms to SISR. Taking efficiency into account, the most popular attention mechanism for SR networks is squeeze-and-excitation (SE) attention [41] used for high-level vision problems. It provides notable performance gains at a considerably low computational cost. However, the SE attention encodes the whole feature map to a single value and hence ignores the spatial relationship between features, which is essential for capturing spatial structures in low-level vision tasks. Moreover, all the previous attention-based approaches performed *in-place* attention within the residual

blocks, as in Figure 5.1 (b). To the best of our knowledge, this work is the first to identify that such *in-place* attention mechanisms may discard relevant details that will no longer be available at deeper levels of the architecture.

To address the problem of feature degradation, we propose the use of a collection of stacked residual blocks which outputs are linearly fused at different stages of the feature hierarchy, to minimize the information loss during processing through the network and ease the gradient flow for optimization. Under this design, the network can aggregate more representative features, thus generating more accurate SR results. Moreover, we present the concept of *directional variance pooling*, which enables the network to attend to larger regions and facilitates capturing longer-range dependencies. Based on the directional variance pooling, we propose a novel and efficient directional variance attention mechanism (DiVA) specifically related to low-level vision tasks. DiVA leverages spatial relationships between features by exploiting higher-order feature statistics, in order to enhance features in different channels and spatial regions without incurring significant computation overhead. Furthermore, in order to alleviate the loss of information caused by commonly used in-place attention mechanisms, we propose to combine DiVA modules with residual blocks by keeping a dedicated computational path for attention modules. Figure 5.1(c) illustrates our approach, where attention modules are independent of the purely residual path and parallel to it.

To verify the effectiveness of the proposed approaches, we build a deep but lightweight architecture for SISR named directional variance attention network (DiVANet), illustrated in Figure 5.2. In summary, these are the main contributions of the paper:

- We propose a lightweight and efficient directional variance attention network (DiVANet) for high-quality image SR. Extensive experiments on a variety of public datasets demonstrate the superiority of the proposed architecture over state-of-the-art models, in terms of both quantitative and visual quality.
- We propose a directional variance attention mechanism (DiVA), specifically optimized for SR, to enhance features in different channels and spatial regions. Such a mechanism allows the network to focus on more informative features and improve discriminative capabilities.
- We introduce a novel procedure for using attention mechanisms together with residual blocks, following two independent but parallel computational paths. The idea is to hierarchically aggregate their respective contributions across the network to facilitate the preservation of finer details.

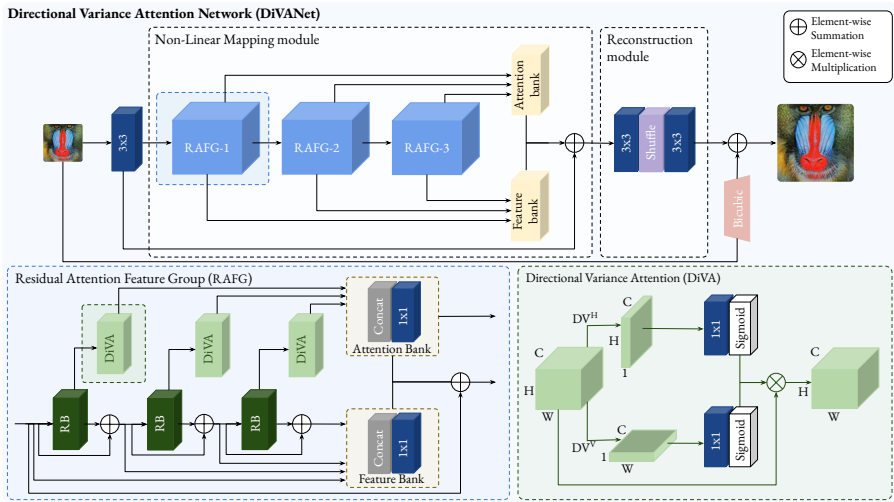


Figure 5.2 – **Top:** Proposed directional variance attention network (DiVANet) architecture for SISR. **Bottom:** residual attention feature group (RAFG), containing residual blocks (RB) and the proposed directional variance attention (DiVA).

## 5.2 Directional Variance Attention Network (DiVANet)

In this section, we first provide an overview of the proposed directional variance attention network (DiVANet) for SISR. Then, we present the detailed configuration of its two main components: the directional variance attention blocks (DiVA) and the residual attention feature groups (RAFGs).

### 5.2.1 Network Overview

As shown in Figure 5.2, the overall architecture of DiVANet consists of a non-linear mapping module and a final reconstruction module. Let's denote  $I_{LR}$  and  $I_{SR}$  the input and output of DiVANet, respectively. As recommended in [71], we apply only one  $3 \times 3$  convolutional layer to extract the initial features  $F_0$  from the LR input image:

$$F_0 = \text{Conv}_{3 \times 3}(I_{LR}). \quad (5.1)$$

Next, extracted features  $F_0$  are sent to the non-linear mapping module (NLM) which computes useful representations of the LR patch in order to infer its HR

version:

$$F = H_{NLM}(F_0), \quad (5.2)$$

where  $F$  is the output of the non-linear mapping module  $H_{NLM}$  (further detailed in Section 5.2.3), containing high resolution features.

Finally, a reconstruction module with two convolutional layers and a pixel-shuffle layer upsamples the features to the HR size. In addition, we incorporate a global connection path  $H_{UP}$  to grant access to the original LR information and facilitate the back-propagation of the gradients, in which only a bicubic interpolation is applied to the input  $I_{LR}$ . Therefore, we obtain:

$$I_{SR} = H_{REC}(F) + H_{UP}(I_{LR}). \quad (5.3)$$

where  $H_{REC}(\cdot)$  is the reconstruction module, and  $I_{SR}$  is the final output of the network.

To optimize DiVANet, we adopt  $L_1$  loss as a cost function for training. Given a training set with  $N$  pairs of LR images and HR counterparts, denoted by  $\{I_{LR}^i, I_{HR}^i\}_{i=1}^N$ , the network is optimized to minimize the  $L_1$  loss function:

$$L_1(\boldsymbol{\theta}) = \frac{1}{N} \sum_{i=1}^N \|I_{SR} - I_{HR}\|_1, \quad (5.4)$$

where  $\boldsymbol{\theta}$  denotes the parameter set.

### 5.2.2 Directional Variance Attention (DiVA)

To provide a clear description of the proposed DiVA mechanism, we first revisit the SE attention, which is widely used in SR networks.

#### Background: Squeeze-and-Excitation Attention

The well-known Squeeze-and-Excitation attention mechanism (SE) is employed in many image classification tasks. Structurally, an SE block is divided into two processes: *Squeeze* is designed to embed global information, and *excitation* aims at adaptive recalibration of channel relationships. Let  $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_C] \in \mathbb{R}^{C \times H \times W}$  be an input. Then, the squeeze step for the  $c$ -th channel can be formulated as follows:

$$z_c = \frac{1}{H \times W} \sum_{i=1}^H \sum_{j=1}^W x_c(i, j), \quad (5.5)$$

where  $x_c(i, j)$  is the value at position  $(i, j)$  of the  $c$ -th channel, and  $z_c$  is the value obtained for channel  $c$  after average pooling. The main purpose of the squeeze operation is to extract and condense global information per channel.

The second step, excitation, aims to leverage inter-channel dependencies to enhance or decrease the response of individual channels. This operation can be formulated as:

$$\hat{\mathbf{X}} = \mathbf{X} \cdot \sigma(\hat{\mathbf{z}}), \quad (5.6)$$

where  $\cdot$  refers to channel-wise multiplication,  $\sigma$  is the sigmoid function, and  $\hat{\mathbf{z}}$  is the result generated by a transformation function, which is formulated as follows:

$$\hat{\mathbf{z}} = \mathbf{W}_2(\delta(\mathbf{W}_1(\mathbf{z}))). \quad (5.7)$$

Here,  $\delta$  denotes the ReLU function.  $\mathbf{W}_1$  and  $\mathbf{W}_2$  are fully connected layers that set the channel dimension of features to  $\frac{C}{r}$  and  $C$ , respectively.

The SE block has been widely used in various SR networks [59, 77, 144], and proven to be a key component for achieving state-of-the-art performance. However, SE attention generally suffers from two basic problems, both stemming from the global average pooling operation. First, it re-weights the importance of each channel by only modeling channel relationships, but neglects spatial information which would be advantageous to enhance image details. Second, it only exploits channel-wise statistics of features by global average pooling, while ignoring higher-order statistics of channels, thus hindering the discriminative ability of the network [21].

Inspired by the above observations, we propose the use of directional variance attention (DiVA) module that captures not only cross-channel but also spatial information, while considering higher-order feature statistics.

### Directional Variance Attention Block

Figure 5.2 (bottom) depicts the proposed DiVA block. In order to encourage attention blocks to capture long-range interactions spatially while keeping a low computational footprint, we factorize two-dimensional global average pooling as formulated in Equation (5.5) into a pair of 1D feature encoding operations. We first feed  $\mathbf{X}$  into two parallel pathways, to encode each channel along either the horizontal or the vertical dimension. We define the horizontal directional average pooling  $DA^h$  of the  $c$ -th channel at height  $h$  as:

$$z_c^h = DA^h(x_c) = \frac{1}{W} \sum_{0 < i \leq W} x_c(i, h) \quad (5.8)$$

Similarly, the vertical directional average pooling  $DA^w$  of the  $c$ -th channel at width  $w$  is defined as:

$$z_c^w = DA^w(x_c) = \frac{1}{H} \sum_{0 < j \leq H} x_c(w, j) \quad (5.9)$$

The described operations process features along two spatial directions individually. This is different from the squeeze operation in channel attention methods (Eq. 5.5), which produces a single feature vector and dismissing the spatial relationship between features. These two transformations also enable the attention block to build relationships among multiple spatial positions within the input feature. However, since image SR ultimately aims at restoring high-frequency components of images, it is important to extract statistics that can effectively represent the characteristics of each channel. To this end, we replace directional average pooling with directional variance pooling, a higher-order feature statistic. Thus, we define the horizontal directional variance pooling  $DV^h$  of the  $c$ -th channel at height  $h$  as:

$$z_c^h = DV^h(x_c) = \frac{1}{W} \sum_{0 \leq i < W} (x_c(i, h) - DA^h(x_c))^2 \quad (5.10)$$

Similarly, the vertical directional variance pooling  $DV^w$  of the  $c$ -th channel at width  $w$  is defined as:

$$z_c^w = DV^w(x_c) = \frac{1}{H} \sum_{0 \leq j < H} (x_c(w, j) - DA^w(x_c))^2 \quad (5.11)$$

As described above, Equations 5.10 and 5.11 facilitate a global receptive field and encode spatial information by exploiting directional variance pooling. These two feature maps with spatial information are then separately encoded into two attention maps that can be complementarily applied to the input feature map to enhance features in different channels and spatial regions.

Specifically, given the aggregated feature maps produced by Equations 5.10 and 5.11, two  $1 \times 1$  convolutional  $F_h$  and  $F_w$  are utilized to separately transform  $z_h$  and  $z_w$ , yielding:

$$\mathbf{a}^h = \sigma(F_h(\mathbf{z}^h)), \quad (5.12)$$

$$\mathbf{a}^w = \sigma(F_w(\mathbf{z}^w)). \quad (5.13)$$

Recall that  $\sigma$  is the sigmoid function.  $\mathbf{a}^h \in \mathbb{R}^{C \times H}$  and  $\mathbf{a}^w \in \mathbb{R}^{C \times W}$  are used as



attention weights, respectively. Finally, the recalibrated output can be written as:

$$y_c(i, j) = x_c(i, j) \cdot y_c^h(i) \cdot y_c^w(j). \quad (5.14)$$

Compared to other works like [86, 121], which require a considerably large amount of computation to build relationships between each pair of locations, DiVA is substantially lightweight and can capture long-range spatial dependencies and exploit inter-channel dependencies simultaneously. Furthermore, unlike SE, which relies on global average pooling to exploit first-order statistics, the proposed attention mechanism adaptively learns feature inter-dependencies by exploiting higher-order statistics that represent the characteristics of each channel. The DiVA mechanism helps to emphasize informative representations and improve discriminative learning ability. Section 5.3.2 provides a more detailed analysis on the performance of our approach against existing attention-based methods.

### 5.2.3 Residual Attention Feature Group (RAFG)

The residual attention feature group (RAFG) is the core of the non-linear mapping module. It is designed to attend and preserve higher frequency details across the entire network. As shown in Figure 5.2, it is composed of two dedicated computational paths: (i) residual path and (ii) attention path. We detail each of these below.

#### Residual Path

It has been demonstrated that stacked residual blocks can be useful to construct deep CNNs [71]. However, in image SR, very deep networks built in such a way would suffer from training difficulty and hardly gain performance [21]. This is because the residual features from initial blocks need to traverse a long path to propagate until the final blocks, as these features are repeatedly merged with identity features to form more complex ones during transmission. Therefore, highly representative features are mostly computed locally and lost in residuals during network propagation.

In this work, we address this issue from a different perspective. Instead of designing a complex architecture with various skip and dense connections, we propose to linearly combine the residual features at a feature bank which is built by aggregating all the features from previous blocks. Figure 5.2 (bottom) shows the details of the proposed RAFG. It contains three residual blocks, the output of which are respectively sent to the end of the RAFG, and then concatenated together. However, aggregating residual features from different residual blocks directly by systematic concatenation is problematic. Thus, we incorporate a  $1 \times 1$  convolutional layer to project them into a common space after feature aggregation. In this

way, information from preceding residual blocks can be hierarchically propagated bottom-up without degradations or interference, leading to a more discriminative feature representation.

Using hierarchical feature banks enables us to exploit residual features non-locally. In other words, these feature banks capture detailed information from features across the whole architecture, thus reducing feature degradation and boosting the network’s overall representational ability.

### Attention Path

The features extracted by a deep neural network contain different types of information at each channel. If we are able to increase the network’s sensitivity to specific channels that contain useful information for image reconstruction, the performance of the network will be improved.

Previous approaches performed channel attention *in-place* within the residual blocks to further boost the representational ability of the network [41]. This usually implied an element-wise product between the attention output and the residual block output. However, such in-place channel attention may discard relevant details which will no longer be available at deeper levels of the architecture, so we propose to keep a separate computational path to aggregate computations resulting of attention operations, independent from the aggregation of residual features, and parallel to it.

As shown in Figure 5.2 (bottom), the output of each residual block is directly sent to a DiVA block before element-wise addition. Specifically, the attention outputs are then aggregated to an attention bank followed by a  $1 \times 1$  convolutional layer. Finally, the outputs of feature and attention banks are combined together, so that they are able to attend to relevant features while preserving higher frequency details across the whole network, further improving the representational ability.

## 5.3 Experimental Results

In this section, we first conduct an ablation study to validate the effectiveness of each proposed component. Then, we systematically compare DiVANet with state-of-the-art SISR algorithms on five commonly used benchmark datasets.

### 5.3.1 Settings

**Datasets and metrics.** Following [18], we use 800 high-quality images from the DIV2K dataset [117] for training. We evaluate our models on several benchmark datasets: Set5 [9], Set14 [138], B100 [2], and Urban100 [45], and, Manga109 [82],

each with diverse characteristics. All results are evaluated with two commonly used metrics: PSNR and SSIM. To keep the consistency with previous works, quantitative results are evaluated on the luminance channel (Y). Furthermore, we also adopt the Perceptual Index (PI) [11], which can avoid the situation where over-smoothed images may present a higher PSNR or SSIM when the performances of two methods are similar.

**Degradation models.** To fairly compare against existing works, we adopt bicubic downsampling (denoted as **BI**) as our standard degradation model for generating LR images from ground truth HR images at  $\times 2$ ,  $\times 3$  and  $\times 4$  scales. Moreover, to comprehensively illustrate the efficacy of the proposed method, we further adopt two other multi-degradation models as in [145]. We define **BD** as a degradation model that performs bicubic downsampling on HR images at  $\times 3$  scale, and then blurs them with a Gaussian kernel of size  $7 \times 7$  and standard deviation 1.6. Additionally, we further produce LR images in a more challenging way: we first bicubic downsample HR images with scaling factor  $\times 3$  and then add Gaussian noise with noise level 30 (denoted as **DN**).

**Implementation details.** During training, data augmentation is carried out by means of random horizontal flips and  $90^\circ$  rotation. At each training mini-batch, 64 LR RGB patches of size  $64 \times 64$  are provided as inputs. We train our models using an ADAM optimizer with learning rate  $10^{-3}$ . The learning rate is decreased by half every  $2 \times 10^5$  iterations. Our network has been implemented using PyTorch, and trained on a NVIDIA RTX 3090 GPU. We implement two lightweight models in this paper, namely DiVANet and DiVANet-S. DiVANet consists of 3 RAFGs, each with three residual blocks and three DiVA modules. In this implementation of DiVANet, all convolutional layers have 64 filters with kernel size  $3 \times 3$ , except for the  $1 \times 1$  convolutional layers in the feature and attention banks. DiVANet-S has a similar structure as DiVANet, except the parameters of the residual blocks within each RAFG are shared.

### 5.3.2 Ablation Study

To further investigate the behavior of the proposed methods, we analyze their effect on model training via an ablation study. We first demonstrate the effectiveness of the proposed DiVA mechanism. Then, we conduct an ablation experiment to study the effect of the essential components of our architecture.

### Comparing Pooling Methods

To demonstrate the advantages of the proposed directional variance pooling (*D-Var*) over other pooling methods, we attempt to replace it with: global average pooling (*Avg*), global variance pooling (*Var*), and directional average pooling (*D-Avg*). We do not employ maximum pooling in this experiment, since Mehri et al. [84] already demonstrated that it degrades SR performance. Additionally, we will also compare to a baseline implementation, identical to the proposed method except for the absence of the attention path (*Baseline*).

The results of this experiment are listed in Table 5.1. It can be seen that exploiting higher-order statistics (global variance pooling) is more effective than first-order ones (global average pooling). Furthermore, when we change from global average pooling to directional average pooling the performance increases by up to +0.14dB in average, with a negligible increase in the number of parameters. This is mainly because the proposed attention with directional average pooling simultaneously captures longer-range spatial interactions and exploits inter-channel dependencies, further improving the representational ability of the network. However, it only leverages first-order statistics of the features. Finally, when directional variance pooling is applied, the attention mechanism enhances features in different channel and spatial regions by exploiting higher-order feature statistics and attains the best performance in all datasets (PSNR: +0.20dB in average). This improvement is more prominent for the B100 and Urban100 datasets. Since B100 and Urban100 present contents with higher structural complexity, it can be interpreted that the attention with directional variance pooling can help the network to exploit more informative features and enhance its discriminative learning ability. These results demonstrate the superiority of using directional variance pooling over other pooling strategies.

Table 5.1 – Effect of different pooling methods for DiVA. Average PSNR on five benchmark datasets with scale factor  $\times 4$  are shown.

Settings	Baseline	1st	2nd	3rd	
Methods	Baseline	+ Avg	+ Variance	+ D-Avg	+ D-Var
Params	815K	902K	902K	939K	939K
Set5	32.18	32.33(+0.15dB)	32.35(+0.17dB)	32.37(+0.19dB)	<b>32.41(+0.23dB)</b>
Set14	28.59	28.62(+0.03dB)	28.64(+0.05dB)	28.66(+0.07dB)	<b>28.70(+0.11dB)</b>
B100	27.56	27.59(+0.03dB)	27.60(+0.04dB)	27.61(+0.05dB)	<b>27.65(+0.09dB)</b>
Urban100	26.09	26.30(+0.11dB)	26.34(+0.15dB)	26.35(+0.26dB)	<b>26.42(+0.33dB)</b>
Manga109	30.50	30.60(+0.10dB)	30.63(+0.13dB)	30.65(+0.15dB)	<b>30.73(+0.23dB)</b>

Table 5.2 – Average PSNR obtained with DiVANet when using different attention mechanisms on five benchmark datasets (scale factor  $\times 4$ ).

Name	Baseline	+ SE	+ SA	+ CSAR	+ DiVA
Params	815K	902K	902K	940K	939K
Set5	32.18	32.33	32.29	32.35(+0.17dB)	<b>32.41(+0.23dB)</b>
Set14	28.59	28.63	38.60	28.66(+0.07dB)	<b>28.70(+0.11dB)</b>
B100	27.56	27.58	27.56	27.60(+0.04dB)	<b>27.63(+0.07dB)</b>
Urban120	26.09	26.31	26.30	26.33(+0.24dB)	<b>26.42(+0.33dB)</b>
Manga109	30.50	30.62	30.60	30.64(+0.14dB)	<b>30.73(+0.23dB)</b>

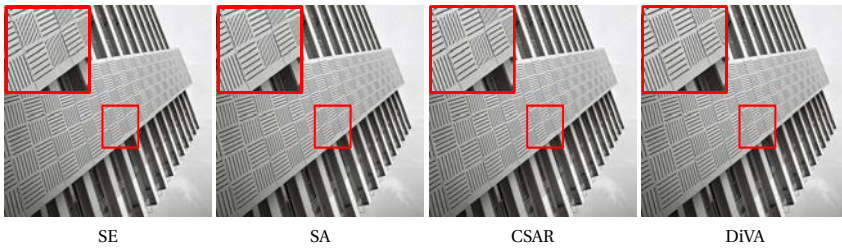


Figure 5.3 – Visual comparison of SR results using DiVANet with different attention mechanisms ( $\times 4$  scale factor).

### Comparing Attention Schemes

To demonstrate the effectiveness of our proposed attention mechanism, we use DiVANet as the basic network, and then replace our attention scheme with Squeeze-and-Excitation channel attention (SE) [41], spatial attention (SA) [43] and channel-wise spatial attention residual (CSAR) [43]. Note that we only compare the DiVA scheme with equally lightweight attention mechanisms. As shown in Figure 5.1 (c), the channel attention module feeds from the residual block but splits out from it through a dedicated computational path; we have trained all the aforementioned variations using this same architectural pattern.

Table 5.2 compares the performance of these attention methods in terms of PSNR. We see that all the methods with an attention mechanism obtain better performance than the one without it (*Baseline*). This indicates that attention contributes importantly in terms of performance. As reported in Table 5.2, integrating SE or SA attention into DiVANet moderately improves the SR performance. Moreover, when CSAR [43] is utilized the performance is further boosted (+0.13dB in average), demonstrating the effectiveness of combining channel-wise and spatial

attention. On the other hand, the model using the proposed DiVA yields the best performance (PSNR: +0.20dB in average). Compared to CSAR, DiVA efficiently encodes both cross-channel and spatial information, attaining better performance with fewer parameters. These experiments justify that with comparable learnable parameters, the proposed DiVA attention is more helpful for image SR. Figure 5.3 shows a visual comparison of networks with different attention mechanisms. It can be observed that the network with our proposed attention obtains better visual quality and restores more image details than other methods.

### Influence of Model Size

We also investigate the effectiveness of DiVA attention in networks with different model sizes. For comparison, we select two state-of-the-art networks, SRDenseNet [118] and RCAN [144], whose number of parameters are 2,015K and 15,592K, respectively. Then, DiVA is performed *in-place*, either at the end of the SRDenseNet blocks (SRDenseNet+DiVA) or replacing RCAN’s channel attention (RCAN+DiVA). For fair comparison, all networks are trained on their default settings. Table 5.3 shows the results of experiments conducted on five datasets at scale  $\times 4$ . It can be observed that SRDenseNet+DiVA and RCAN+DiVA respectively achieve better performance than the original SRDenseNet and RCAN networks. These experimental results indicate that DiVA is also effective in heavier models, increasing the performance by 0.07dB in average.

Table 5.3 – The results of adding DiVA in different networks. Average PSNR on five benchmark datasets with scale factor  $\times 4$  are shown.

Methods	SRDensNet	SRDensNet+DiVA	RCAN	RCAN+DiVA
Multi-Adds	390G	392G	916.9G	964.1G
Params	2,015K	2,250	15,592K	16,410K
Set5	32.02	<b>32.08(+0.06dB)</b>	32.68	<b>32.76(+0.08dB)</b>
Set14	28.50	<b>28.57(+0.07dB)</b>	28.95	<b>29.01(+0.06dB)</b>
B100	27.53	<b>27.59(+0.06dB)</b>	27.55	<b>27.61(+0.06dB)</b>
Urban100	26.05	<b>26.13(+0.08dB)</b>	27.05	<b>27.11(+0.06dB)</b>
Manga109	30.41	<b>30.52(+0.11dB)</b>	31.62	<b>31.66(+0.04dB)</b>

### Effect of the RAFG

This section discusses the effect of each of the two dedicated computational paths in the proposed RAFG: residual path and attention path.

Table 5.4 – Average PSNR for a regular ResNet architecture (Baseline) vs one using the proposed feature banks on five benchmark dataset with  $\times 4$  scale factor.

Methods	Baseline	Baseline+FB	SRDenseNet [118]
Multi-Adds	50G	54G	390G
Params	749K	815K	2,015K
Set5	31.85	<b>32.18(+0.33dB)</b>	32.02
Set14	32.36	<b>32.59(+0.23dB)</b>	28.50
B100	27.30	<b>27.55(+0.22dB)</b>	27.53
Urban100	25.73	<b>26.09(+0.36dB)</b>	26.05
Manga109	30.29	<b>30.49(+0.20dB)</b>	30.41

**Residual path.** In this experiment, we use a ResNet architecture (*Baseline*) without the RAFG computational path, i.e., a regular architecture composed of several stacked residual blocks. Then, we add hierarchical feature banks to this baseline, denoting it as *Baseline+FB*. Table 5.4 shows the results of the experiments conducted on the five datasets with scale  $\times 4$ . The small change in number of parameters between *Baseline* and *Baseline+FB* is due to adding feature banks, which contain  $1 \times 1$  convolutions.

As reported in Table 5.4, the PSNR of *Baseline* is 25.73dB on Urban100, which is a strong baseline for lightweight SISR methods. When deploying our hierarchical bank of residuals, the PSNR increases to 26.09dB. In addition, we compare our method with SRDenseNet [118]. This model combines residual skip connections with dense connections to utilize all the hierarchical features from all the convolutional layers, hence being very computationally intensive due to this dense feature fusion strategy. In contrast, we preserve the local information progressively by placing a  $1 \times 1$  convolution every three residual blocks. From Table 5.4, we find that our network achieves better performance with significantly lower computational cost and number of parameters. We attribute this considerable improvement to the effectiveness of the proposed connectivity pattern, where the features in each residual block can be better utilized by the network.

**Attention path.** Previous SISR approaches perform channel attention in-place within the residual blocks, whereas this work takes the attention out of the main computational path, and computes it in parallel. To verify the effectiveness of this approach, in this experiment, we use a baseline which is identical to the proposed method except for the absence of the attention path (*Baseline*). We then place the DiVA attention mechanism both inside (*Baseline\_in*) and outside (*Baseline\_out*) of the residual blocks, comparing their performance in Table 5.5. As can be observed,

Table 5.5 – Average PSNR obtained on the ResNet baseline network, when placing the DiVA attention mechanism within (*Baseline\_in*) or outside (*Baseline\_out*) the residual blocks. Results are shown on five benchmark datasets and with a  $\times 4$  scale factor.

Methods	Baseline	Baseline_in	Baseline_out
Params	815K	890K	939K
Set5	32.18	32.32(+0.14dB)	<b>32.41(+0.23dB)</b>
Set14	28.59	28.64(+0.05dB)	<b>28.70(+0.11dB)</b>
B100	27.55	27.58(+0.03dB)	<b>27.65(+0.10dB)</b>
Urban100	26.09	26.31(+0.02dB)	<b>26.41(+0.12dB)</b>
Manga109	30.49	30.65(+0.16dB)	<b>30.73(+0.24dB)</b>

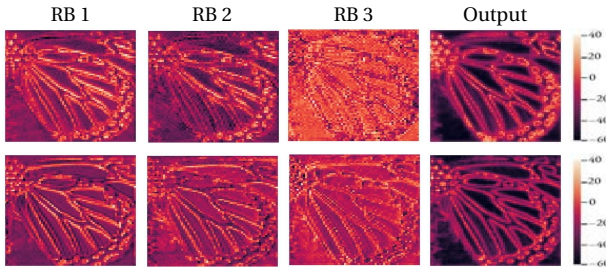


Figure 5.4 – Average feature maps of residual blocks (RBs). **Top:** Attention is applied within the residual (classic approach). **Bottom:** Attention is applied outside the residual (our approach).

*Baseline\_out* leads to performance improvement, having just a few more parameters due to the aggregation operation inside the attention feature bank. These results prove that moving attention operations outside of the residual blocks is beneficial to prevent the loss of information caused by commonly used in-place attention. This justifies our choice for keeping a separate computational path to aggregate computations coming from attention operations.

Figure 5.4 additionally shows average feature maps in residual blocks, when attention mechanisms are applied inside (in-place, top row) or outside (as in our RAFG, bottom row). This visualization shows how RAFGs are able to learn sharper representations than those obtained with in-place attention. In essence, each RAFG directs computations towards edges and details, thus obtaining a more defined representation at the output. In contrast, when using in-place attention, feature maps vary significantly from the first residual block to the last. As a result, edges



and contours are outlined at the first layer, and smooth areas within the original image become suppressed at subsequent blocks.

### 5.3.3 Comparison with State-of-the-art Lightweight Methods

In this section, DiVANet and DiVANet-S are compared to other lightweight state-of-the-art SR methods. A self-ensemble method [116] is also used to further improve the performance of the DiVANet (denoted as DiVANet+).

#### Results with BI Degradation Model

Simulating LR images with the **BI** degradation model is widely used in the context of image SR. For the **BI** degradation model, we compare our proposed DiVANet-S, DiVANet and DiVANet+ with state-of-the-art SR frameworks, including VDSR [52], DRCN [53], SRDenseNet [118], CARN [1], SRFBN-S [69], CBPN [148], FALSAR-A[20], SRMDNF [141], LAPAR-A [67], MAFFSRN [88], LatticeNet [77], MPRNet [84], RFDN-L [72], MADNet [59], HDRN [49], DPN [70], and A2F-L [130].

Table 5.6 shows quantitative results when evaluating PSNR and SSIM on five benchmark datasets with different algorithms. For a more informative comparison, the number of parameters and the number of multiplications and additions (Multi-Adds) are also given. It can be observed that the proposed DiVANet-S has only less than 500K parameters, but its performance is superior to many state-of-the-art methods. For example, in comparison with CARN and CBPN, DiVANet-S attains significantly better performance while only needing 30% and 40% of their parameters, respectively. Furthermore, DiVANet is the best performing one, at all scales and in all datasets. Especially on the challenging dataset Urban100, which contains rich structural contents, the proposed DiVANet advances the state-of-the-art with improvement margins of 0.14dB, 0.18dB and 0.10dB for scale factors  $\times 2$ ,  $\times 3$  and  $\times 4$ , respectively. In addition, more significant improvements are shown in the Manga109 dataset, where the proposed DiVANet model outperforms A<sup>2</sup>F-L (with the highest performance amongst the aforementioned methods), by PSNR gains of 0.13dB and 0.11dB for  $\times 2$  and  $\times 3$  enlargement. The advantage of our method can also be verified via SSIM scores. The SSIM score focuses on the visible structures in the image. The proposed DiVANet also achieves the best SSIM score, which indicates that DiVANet can better recover visible structures. These results validate the superiority of the proposed method, particularly on super-resolving the images with fine structures such as those in Urban100 and Manga109. Furthermore, it can be seen that DiVANet+ achieves further improvements through the use of self-ensembles [116].

In Figure 5.5, we present some qualitative visual comparisons for the  $\times 4$  scale

### 5.3. Experimental Results

Table 5.6 – Average PSNR/SSIM values for models with the same order of magnitude of parameters. Performance is shown for scale factors  $\times 2$ ,  $\times 3$  and  $\times 4$  with BI degradation model. The Multi-Adds is calculated corresponding to a  $1280 \times 720$  HR image. The best and second best results are highlighted in red and blue respectively.

Scale	Method	Params	Multi-Adds	Set5		Set14		B100		Urban100		Manga109		
				PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	
$\times 2$	VDSR [52]	665K	613G	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.22	0.9729	
	DRCN [53]	1,774K	17,974G	37.53	0.9587	33.03	0.9124	31.90	0.8960	30.76	0.9140	37.63	0.9723	
	CARN [1]	1,592K	223G	37.76	0.9590	33.52	0.9166	32.09	0.8978	31.92	0.9256	38.36	0.9765	
	SRFBN-S [69]	282K	680G	37.78	0.9597	33.35	0.9156	32.00	0.8970	31.41	0.9207	38.06	0.9757	
	CBPN [148]	1,036K	240.7G	37.90	0.9590	33.60	0.9171	32.17	0.8989	32.14	0.9279	-	-	
	FALSR-A[20]	1,021K	234.7G	37.82	0.9595	33.55	0.9168	32.12	0.8987	31.93	0.9256	-	-	
	SRMDNF [141]	1,513K	348G	37.79	0.9600	33.32	0.9150	32.05	0.8980	31.33	0.9200	-	-	
	LAPAR-A [67]	548K	171G	38.01	0.9605	33.62	0.9183	32.19	0.8999	32.10	0.9283	38.67	0.9772	
	MAFFSRN [88]	790K	154.4G	38.07	0.9607	33.59	0.9177	32.23	0.9005	32.38	0.9308	-	-	
	LatticeNet [77]	756K	169.5G	38.15	0.9610	33.78	0.9193	32.25	0.9005	32.24	0.9302	-	-	
	MPRNet [84]	538K	163.3G	38.08	0.9608	33.79	0.9196	32.25	0.9004	32.25	0.9317	-	-	
	RFDN-L [72]	626K	38G	38.08	0.9606	33.67	0.9190	32.18	0.8996	32.24	0.9290	38.95	0.9773	
	MADNet [59]	878K	187.1G	37.94	0.9604	33.46	0.9167	32.10	0.8988	31.74	0.9246	-	-	
	HDRN [49]	878K	316.2G	37.75	0.9590	33.49	0.9150	32.03	0.8980	31.87	0.9250	38.07	0.9770	
	DPN [70]	832K	140G	37.52	0.9586	33.08	0.9129	31.89	0.8958	30.82	0.9144	-	-	
	A <sup>2</sup> F-L [130]	1,363K	306.1G	38.09	0.9607	33.78	0.9192	32.23	0.9002	32.46	0.9313	38.95	0.9772	
	DiVANet-S	405K	75G	38.10	0.9605	33.76	0.9189	32.22	0.8999	32.40	0.9305	38.88	0.9771	
	DiVANet	902K	189G	<b>38.16</b>	<b>0.9612</b>	<b>33.80</b>	<b>0.9195</b>	<b>32.29</b>	<b>0.9008</b>	<b>32.60</b>	<b>0.9325</b>	<b>39.08</b>	<b>0.9755</b>	
	DiVANet+	902K	189G	<b>38.23</b>	<b>0.9618</b>	<b>33.88</b>	<b>0.9201</b>	<b>32.36</b>	<b>0.9011</b>	<b>32.67</b>	<b>0.9330</b>	<b>39.15</b>	<b>0.9790</b>	
	$\times 3$	VDSR [52]	665K	613G	33.66	0.9213	29.77	0.8314	28.82	0.7976	27.14	0.8279	37.22	0.9750
DRCN [53]		1,774K	17,974G	33.82	0.9226	29.76	0.8311	28.80	0.7963	27.15	0.8276	32.24	0.9343	
CARN [1]		1,592K	119G	34.29	0.9255	30.29	0.8407	29.06	0.8034	28.06	0.8493	33.50	0.9440	
SRFBN-S [69]		376K	832G	34.20	0.9255	30.10	0.8372	28.96	0.8010	27.66	0.8415	33.02	0.9404	
SRMDNF [141]		1,530K	156G	34.12	0.9250	30.04	0.8370	28.97	0.8030	27.57	0.8400	-	-	
LAPAR-A [67]		544K	114G	34.36	0.9267	30.34	0.8421	29.11	0.8054	28.15	0.8523	33.51	0.9441	
MAFFSRN [88]		807K	68.5G	34.45	0.9277	30.40	0.8432	29.13	0.8061	28.26	0.8552	-	-	
LatticeNet [77]		765K	76.3G	34.53	0.9281	30.39	0.8424	29.15	0.8059	28.33	0.8538	-	-	
MPRNet [84]		538K	63.1G	34.57	0.9285	30.42	0.8441	29.17	0.8073	28.42	0.8578	-	-	
RFDN-L [72]		633K	38G	34.47	0.9280	30.35	0.8421	29.11	0.8053	28.32	0.8547	33.78	0.9458	
MADNet [59]		930K	88.4G	34.26	0.9262	30.29	0.8410	29.04	0.8033	27.91	0.8464	-	-	
HDRN [49]		878K	187.1G	34.24	0.9240	30.23	0.8400	28.96	0.8040	27.93	0.8490	33.17	0.9420	
DPN [70]		832K	114.2G	33.71	0.9222	29.80	0.8320	28.84	0.7981	27.17	0.8282	-	-	
A <sup>2</sup> F-L [130]		1,367K	136.1G	34.54	0.9283	30.41	0.8436	29.14	0.8062	28.40	0.8574	33.83	0.9463	
DiVANet-S		451K	38G	34.48	0.9275	30.43	0.8431	29.13	0.8055	28.42	0.8568	33.80	0.9455	
DiVANet		949K	89G	<b>34.60</b>	<b>0.9285</b>	<b>30.47</b>	<b>0.8447</b>	<b>29.19</b>	<b>0.8073</b>	<b>28.58</b>	<b>0.8603</b>	<b>33.94</b>	<b>0.9468</b>	
DiVANet+		949K	89G	<b>34.66</b>	<b>0.9289</b>	<b>30.53</b>	<b>0.8452</b>	<b>29.26</b>	<b>0.8077</b>	<b>28.66</b>	<b>0.8610</b>	<b>34.02</b>	<b>0.9473</b>	
$\times 4$		VDSR [52]	665K	613G	31.35	0.8838	28.01	0.7674	27.29	0.7251	25.18	0.7524	28.83	0.8809
		DRCN [53]	1,774K	17,974G	31.54	0.8850	29.19	0.7720	27.32	0.7280	25.12	0.7560	29.09	0.8845
		SRDenseNet [118]	2,015K	390G	32.00	0.8931	28.50	0.7782	27.53	0.7337	26.05	0.7819	30.41	0.9071
	CARN [1]	1,592K	91G	32.13	0.8937	28.60	0.7806	27.58	0.7349	26.07	0.7837	30.47	0.9084	
	SRFBN-S [69]	483K	1,037G	31.98	0.8923	28.45	0.7779	27.44	0.7313	25.71	0.7719	29.91	0.9008	
	CBPN [148]	1,197K	97.9G	32.21	0.8944	28.63	0.7813	27.58	0.7356	26.14	0.7869	-	-	
	SRMDNF [141]	1,555K	89G	31.96	0.8930	28.35	0.7770	27.49	0.7340	25.68	0.7730	-	-	
	LAPAR-A [67]	659K	94G	32.15	0.8944	28.61	0.7818	27.61	0.7366	26.14	0.7871	30.42	0.9074	
	MAFFSRN [88]	830K	38.6G	32.20	0.8953	28.62	0.7822	27.59	0.7370	26.16	0.7887	-	-	
	LatticeNet [77]	777K	43.6G	32.30	0.8962	28.68	0.7830	27.62	0.7367	26.25	0.7873	-	-	
	MPRNet [84]	538K	31.3G	32.38	0.8969	28.69	0.7841	27.63	0.7385	26.31	0.7921	-	-	
	RFDN-L [72]	643K	38G	32.28	0.8957	28.61	0.7818	27.58	0.7363	26.20	0.7883	30.61	0.9096	
	MADNet [59]	1,002K	54.1G	32.11	0.8939	28.52	0.7799	27.52	0.7340	25.89	0.7782	-	-	
	HDRN [49]	867K	316.2G	32.23	0.8960	28.58	0.7810	27.53	0.7370	26.09	0.7870	30.43	0.9080	
	DPN [70]	832K	140G	31.42	0.8849	28.07	0.7688	27.30	0.7256	25.25	0.7546	-	-	
	A <sup>2</sup> F-L [130]	1,374K	77.2G	32.32	0.8964	28.67	0.7839	27.62	0.7379	26.32	0.7931	30.72	0.9115	
	DiVANet-S	442K	28G	32.32	0.8958	28.63	0.7827	27.61	0.7377	26.35	0.7926	30.68	0.9105	
	DiVANet	939K	57G	<b>32.41</b>	<b>0.8973</b>	<b>28.70</b>	<b>0.7844</b>	<b>27.65</b>	<b>0.7388</b>	<b>26.42</b>	<b>0.7958</b>	<b>30.73</b>	<b>0.9119</b>	
	DiVANet+	939K	57G	<b>32.48</b>	<b>0.8978</b>	<b>28.78</b>	<b>0.7848</b>	<b>27.73</b>	<b>0.7390</b>	<b>26.49</b>	<b>0.7963</b>	<b>30.78</b>	<b>0.9124</b>	

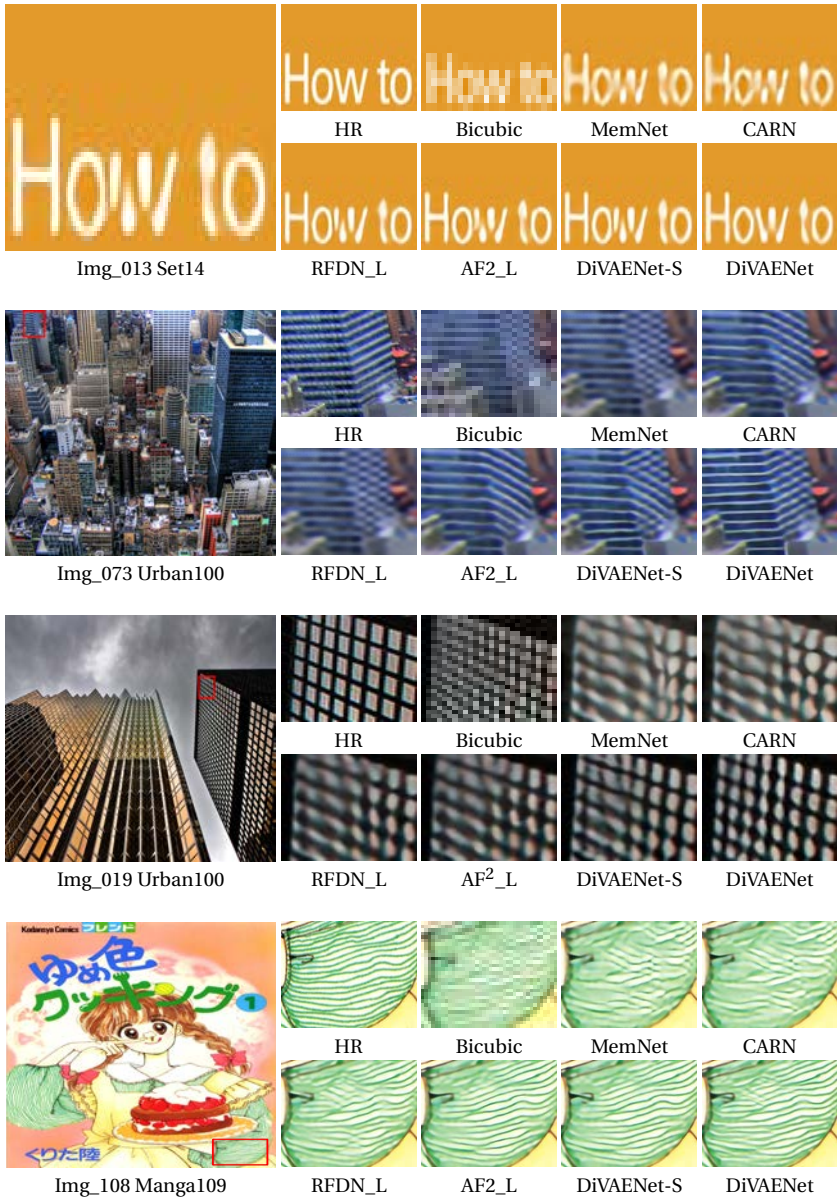


Figure 5.5 – Visual results of BI degradation model for  $\times 4$  scale factor.

Table 5.7 – Quantitative results with **BD** and **DN** degradation models. Performance is shown for scale factor  $\times 3$ . The best and second best results are highlighted in **red** and **blue** respectively.

Methods	Degrad.	Set5		Set14		B100		Urban100		Manga109	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
SRCNN [23]	BD	32.05	0.8944	28.80	0.8074	28.13	0.7736	25.70	0.7770	29.47	0.8924
	DN	25.01	0.6950	23.78	0.5898	23.76	0.5538	21.19	0.5737	23.75	0.7148
VDSR [52]	BD	33.25	0.9150	29.46	0.8244	28.57	0.7893	26.61	0.8136	31.06	0.9234
	DN	25.20	0.7183	24.00	0.6112	24.00	0.5749	22.22	0.6096	24.20	0.7525
IRCNN_G [140]	BD	33.38	0.9182	29.63	0.8281	28.65	0.7922	26.77	0.8154	31.15	0.9245
	DN	25.70	0.7379	24.45	0.6305	24.28	0.5900	22.90	0.6429	24.88	0.7765
IRCNN_C [140]	BD	29.55	0.8246	27.33	0.7135	26.46	0.6572	24.89	0.7172	28.68	0.7701
	DN	26.18	0.7430	24.68	0.6300	24.52	0.5850	22.63	0.6205	24.74	0.7701
SRMDNF [141]	BD	34.09	0.9242	30.11	0.8364	28.98	0.8009	27.50	0.8370	32.97	0.9391
	DN	27.74	0.8026	26.13	0.6924	25.64	0.6495	24.28	0.7092	26.72	0.8590
RDN [145]	BD	34.57	0.9280	30.53	0.8447	29.23	0.8079	28.46	0.8581	33.97	0.9465
	DN	28.46	0.8151	26.60	0.7101	25.96	0.6573	24.92	0.7362	28.00	0.8590
CASGCN [133]	BD	34.62	0.9283	30.60	0.8458	29.30	0.8196	28.68	0.8611	34.27	0.9476
	DN	-	-	-	-	-	-	-	-	-	-
DiVANet-S	BD	34.45	0.9263	30.40	0.8420	29.11	0.8048	28.26	0.8529	33.90	0.9448
	DN	28.41	0.8154	26.16	0.6933	25.87	0.6599	24.88	0.7356	28.13	0.8600
DiVANet	BD	<b>34.64</b>	<b>0.9286</b>	<b>30.63</b>	<b>0.8460</b>	<b>29.31</b>	<b>0.8198</b>	<b>28.70</b>	<b>0.8613</b>	<b>34.30</b>	<b>0.9479</b>
	DN	<b>28.49</b>	<b>0.8159</b>	<b>26.22</b>	<b>0.6939</b>	<b>25.93</b>	<b>0.6605</b>	<b>24.94</b>	<b>0.7361</b>	<b>28.18</b>	<b>0.8605</b>
DiVANet+	BD	<b>34.70</b>	<b>0.9291</b>	<b>30.69</b>	<b>0.8469</b>	<b>29.39</b>	<b>0.8206</b>	<b>28.78</b>	<b>0.8621</b>	<b>34.38</b>	<b>0.9486</b>
	DN	<b>28.57</b>	<b>0.8164</b>	<b>26.29</b>	<b>0.6945</b>	<b>26.01</b>	<b>0.6611</b>	<b>24.99</b>	<b>0.7369</b>	<b>28.26</b>	<b>0.8611</b>

factor. It can be observed that DiVANet-S and DiVANet prevent distortions, suppress artifacts and generate more faithful results. These visual comparisons also demonstrate the powerful representational ability of our methods.

### Results with BD and DN Degradation Models

Following [69, 145], we also provide the results after applying **BD** and **DN** degradation models. The proposed DiVANet-S, DiVANet, and DiVANet+ are compared with state-of-the-art methods including SRCNN [23], VDSR [52], IRCNN\_G [140], IRCNN\_C [140], SRMDNF [141], RDN [145], and CASGCN [133]. As shown in Table 5.7, our methods achieve better PSNR and SSIM scores compared to other SR methods, in all datasets. The consistently better results of our methods indicate that they adapt well to scenarios with multiple degradation models.

In Figures 5.6 and 5.7 we provide some visual results for the **BD** and **DN** degra-

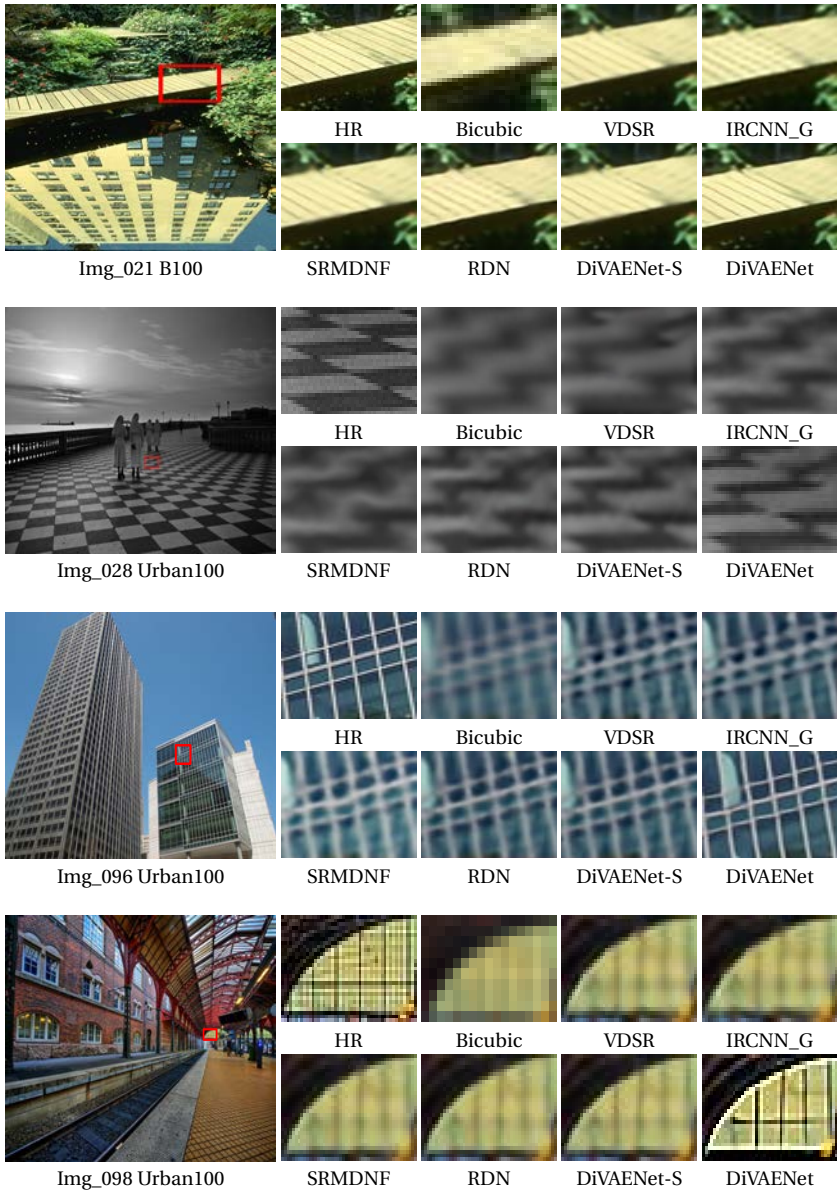


Figure 5.6 – Visual results of **BD** degradation model for  $\times 3$  scale factor.



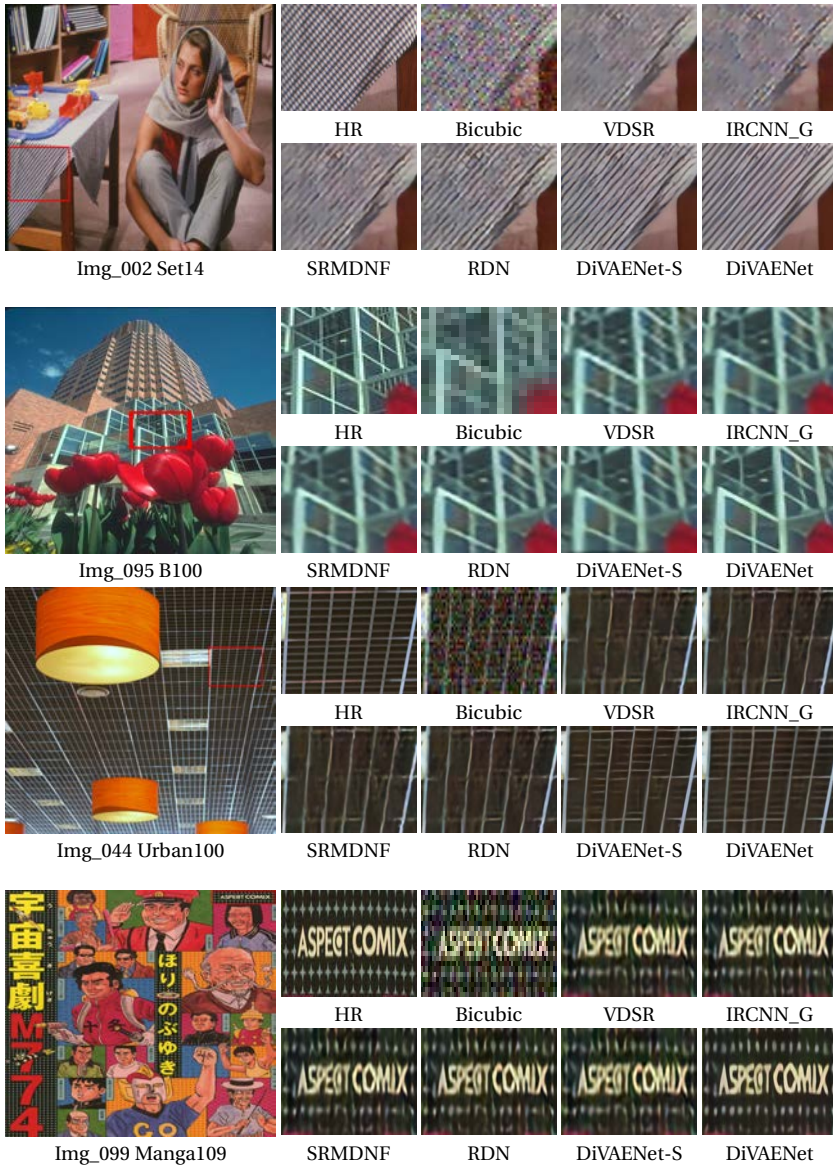


Figure 5.7 – Visual results of **DN** degradation model for  $\times 3$  scale factor.

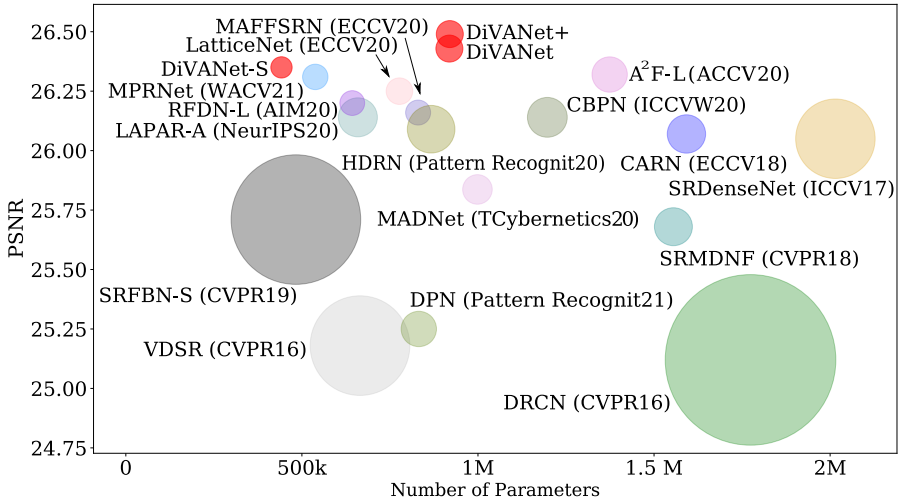


Figure 5.8 – Comparing capacity vs performance for lightweight state-of-the-art SISR models on Urban100 ( $\times 4$ ). Circle sizes are set proportional to the number of multiplications and additions (Multi-Adds).

dation models for  $\times 4$  scale factor from the standard benchmark datasets. For **BD** degradation, other methods were unable to remove blurring artifacts. In contrast, DiVANet-S and DiVANet are able to recover structured details that were missing in the LR image, by efficiently exploiting the feature hierarchy. Regarding the **DN** degradation, we observe that recovering details becomes difficult with other methods. However, ours deliver good results by removing additional noise and enhancing the details. From these comparisons, we further indicate the robustness and effectiveness of our methods in handling **BD** and **DN** degradation models.

### Model Complexity Analysis

In this section, we compare the trade-off between performance and number of parameters for our methods (DiVANet-S, DiVANet and DiVANet+) and existing lightweight networks. Figure 5.8 shows the PSNR performances of several lightweight models, namely VDSR [52], DRCN [53], SRDenseNet [118], CARN [1], SRFBN-S [69], CBPN [148], SRMDNF [141], LAPAR-A [67], MAFFSRN [88], LatticeNet [77], MPRNet [84], RFDN-L [72], MADNet [59], HDRN [49], DPN [70], and A<sup>2</sup>F-L [130]. versus their number of parameters, with results evaluated on Urban100 for  $\times 4$ . As shown in Figure 5.8, our models achieve state-of-the-art results with less parameters and

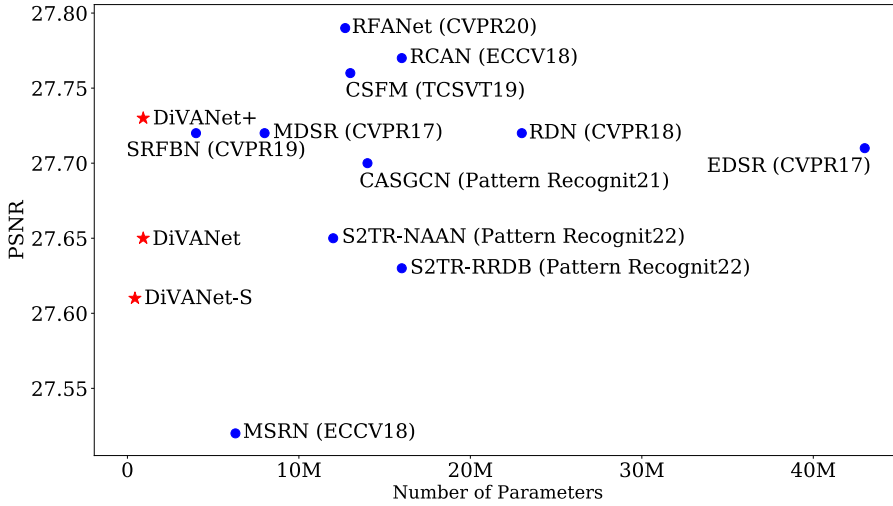


Figure 5.9 – Comparing capacity vs performance for non-lightweight state-of-the-art SISR models in the B100 dataset ( $\times 4$ ). The red stars represent our proposed methods.

Multi-Add operations. This demonstrates that our proposals achieve a better trade-off between model size and reconstruction performance.

In addition, we compare our models with large networks such as EDSR [71], MDSR [71], MSRN [65], RDN [145], RCAN [144], SRFBN [69], CSFM [43], RFANet [72],  $S^2$ TSR-NAAN [121],  $S^2$ TSR-RRDB [121], and CASGCN [133]. The results are given in Figure 5.9 in terms of network parameters and reconstruction effects (PSNR). For example,  $S^2$ TSR-RRDB, CASGCN, and RFANet respectively have parameters/PSNR ratios of 16M/27.63dB, 14M/27.70dB, and 12M/27.76, under  $\times 4$  setting on the B100 dataset. On the other hand, the proposed DiVANet (0.9M/27.65) and DiVANet+ (0.9M/27.73) achieve competitive or better results, while only needing the 5%, 6% and 7% parameters of  $S^2$ TSR-NAAN,  $S^2$ TSR-RRDB and CASGCN, respectively. The DiVANet-S model also shows comparable results to the heavy models. In particular, the DiVANet-S model outperforms MSRN by a large margin of 0.10dB. It is worth noting that while MSRN has 8M parameters, DiVANet-S only has 0.4M parameters. Thus, the proposed networks are lightweight and more efficient than other state-of-the-art methods.



### Memory Complexity and Running Time Analysis

Table 5.8 illustrates the superiority of the proposed DiVANet-S and DiVANet architectures in terms of Inference Time (s) and Memory Consumption (MB), when compared to recent light- and heavy-weight state-of-the-art approaches on Urban100  $\times 4$ . For a fair comparison, we use a single NVIDIA RTX 3090 GPU for evaluation, and their official source code implementations. It can be observed that our models have the fastest running time, while also using the least memory per image. Especially, our networks are highly efficient than RCAN [144], EDSR[71], RFANet [72], and RDN [145] which are  $16\times$ ,  $5\times$ ,  $12\times$ , and  $10\times$  slower than DiVANet, respectively. These networks mainly leverage much deeper network designs to achieve more accurate SR results. This comparison demonstrates that our methods effectively balance performance and running time.

Table 5.8 – Average running time (s) and memory consumption (MB) comparison on Urban100 for  $\times 4$ .

Methods	Params	Memory	Running Time(s)	PSNR
CARN[1]	1.5M	1,116	0.032	26.07
SRFBN-S[69]	0.5M	2,154	0.031	25.71
SRDenseNet[118]	2M	5,531	0.221	26.05
RFDN-L[72]	0.6M	3,015	0.033	26.22
IMDN [47]	0.7M	1,113	0.028	26.04
A <sup>2</sup> F-L[130]	1.3M	3,015	0.032	26.32
RCAN[144]	16M	1,531	0.297	26.82
EDSR[71]	43M	2,731	0.085	26.64
SAN[69]	16M	3,015	0.224	26.79
RDN[145]	23M	5,015	0.172	26.82
DiVANet-S	0.4M	671	0.004	26.35
DiVANet	0.9M	875	0.007	26.42

### Perceptual Metrics

Perceptual metrics better reflect the human judgment of image quality. In this paper, Perceptual Index (PI) [11] is chosen as the perceptual metric. Table 5.9 shows the PI for those works with publicly available source code, and the same order of magnitude in terms of parameters. We observe that our proposed models obtains better results than all the compared baselines. This demonstrates the ability of the proposed DiVANet and DiVANet-S for generating realistic images.

Table 5.9 – Perceptual index comparison of the proposed methods with recent lightweight state-of-the-art methods on five datasets for  $\times 4$ . The lower is better. All of the output SR images are provided officially.

Methods	Params	Set5	Set14	B100	Urban100	Manga109
CARN[1]	1.5M	6.297	5.775	5.700	5.540	5.132
SRFBN-S[52]	0.6M	6.451	5.775	5.702	5.549	5.010
SRDenseNet[118]	2M	6.128	5.615	5.653	5.526	4.762
RFDN_L[72]	0.6M	6.124	5.644	5.659	5.531	4.810
A <sup>2</sup> F_L[130]	1.3M	6.084	5.499	5.532	5.179	4.771
DiVANet-S	0.4M	5.550	5.490	5.430	5.168	4.676
DiVANet	0.9M	<b>5.511</b>	<b>5.361</b>	<b>5.163</b>	<b>5.149</b>	<b>4.480</b>

## 5.4 Summary

In this chapter, we have proposed a novel and efficient architecture called directional variance attention network (DiVANet) for modeling the process of single image super-resolution. We propose a directional variance attention mechanism (DiVA), specifically related to SR, which encodes spatial and inter-channel information simultaneously by considering higher-order feature statistics. Moreover, we present a novel residual attention feature group (RAFG) which combines an efficient connectivity pattern with a DiVA module that is processed in parallel to the main residual computational path. Through a series of ablation experiments, we have demonstrated the effectiveness of the proposed DiVA and RAFG schemes. We use the same DiVANet structure to handle three degradation models. We have empirically shown that our proposal attains better PSNR, SSIM, and perceptual scores than previous lightweight state-of-the-art models on all benchmarks while having a similar or fewer amount of parameters.



## 6 Conclusions and Future work

### 6.1 Conclusions

Single image super-resolution (SISR) is a classical problem that is still a challenging and open research problem in computer vision. The intention behind studying and developing methods for SISR, is to obtain images with resolution beyond the limit of imaging systems. In the last two decades, SISR has witnessed great strides with the rapid development of deep learning. Although considerable progress has been made, existing deep learning-based SR methods still face some limitations.

In this thesis, we studied various challenging problems in SISR and proposed solutions to them in each chapter. In chapter 3, we have focused on the high-frequency enhancement since most current SR networks do not discriminate the computation of features by their frequencial components. Therefore, we have proposed a novel frequency-based enhancement block (FEB) which is able to separate features into low and high frequencies and explicitly allocates computation to high-frequency features, thus improving the discriminative capabilities of the network. Unlike prominent SR blocks, FEB treats different frequencies in a heterogeneous way and also models inter-channel dependencies, which consequently enrich the output feature. We have proved that the proposed block can be flexibly embedded into other SR models by simply replacing their building modules, thus improving their original performance (PSNR: +0.08dB in average) while reducing the number of parameters by half. Furthermore, based on FEB, we have built a lightweight SR network by simply stacking several FEBs which leads to significant improvements for single image SR, surpassing deep SR networks with complicated skip connections and concatenations.

In chapter 4, we have introduced an overscaling module and multi-scale loss to solve SISR at arbitrary scale factors. Overscaling module helps to generate accurate SR images at different scaling factors while multi-scale loss allows the simultaneous training of all scale factors using a single model. Furthermore, in order to reduce low-level feature degradation, we have proposed a lightweight recursive feature extractor. We have shown that both overscaling module and the proposed feature ex-

tractor independently increase PSNR when compared to other SR methods. Finally, combining the proposed feature extractor and overscaling module together (OverNet) further increases performance. Moreover, we have experimentally shown that OverNet outperforms previous state-of-the-art approaches in standard benchmarks while maintaining relatively low computation and memory requirements.

Finally, In chapter 5, we have presented the concept of *directional variance pooling*. Based on the directional variance pooling, we have proposed a novel and efficient directional variance attention mechanism specifically related to low-level vision tasks. This mechanism leverages spatial relationships between features by exploiting higher-order feature statistics, in order to enhance features in different channels and spatial regions. Moreover, we have proved that moving attention operations outside of the residual blocks is beneficial to prevent the loss of information caused by commonly used in-place attention. Therefore, we have proposed to keep a dedicated computational path for attention mechanisms to aggregate computations coming from attention operations. Finally, to verify the effectiveness of the proposed approaches, we have built a computationally efficient yet accurate network for SISR. The proposed network outperforms all state-of-the-art SR models that have less than 5M parameters in terms of both quantitative and visual quality.

The take-home message from this work is that by using effective building modules and loss functions, we are able to design fast, accurate, and lightweight SR networks which can be easily applied to real-world applications. We hope that the idea of decomposing low- and high-frequency information at feature level for adaptive computation can provide the computer vision community with a different perspective on network architecture design. We believe that the proposed approaches would have an important impact on the practical deployment of SISR.

## 6.2 Future Perspective

Despite the great success achieved by CNN-based models in the SISR problem, there are still many unsolved problems. Thus in this section, we will point out some of these problems explicitly and introduce some promising trends for future evolution.

- **Towards real-world scenarios.** Image SR is greatly limited in real-world scenarios since real-world images tend to suffer degradation like blurring, additive noise, and compression artifacts. Thus the models trained on datasets conducted manually often perform poorly in real-world scenes. Therefore, it is important to improve the SR quality of natural images by using the new datasets obtained by different resolution cameras with real-world scenarios. Moreover, SR cannot only be used in domain-specific data and scenes directly

but can also help other vision tasks greatly. Therefore, it is also a promising direction to apply SR to more specific domains, such as video surveillance, object tracking, medical imaging, and scene rendering.

- **Evaluation metrics.** Evaluation metrics are one of the most fundamental components for machine learning. If the performance cannot be measured accurately, researchers will have great difficulty verifying improvements. Nowadays the PSNR and SSIM have been the most widely used metrics for SR. However, the PSNR tends to result in excessive smoothness and the results can vary wildly between almost indistinguishable images. The SSIM performs evaluation in terms of brightness, contrast and structure, but still cannot measure perceptual quality accurately. Although researchers have proposed various metrics, but currently there is no unified and admitted evaluation metrics for SR quality. Thus more accurate metrics for evaluating reconstruction quality are urgently needed.
- **Mutual promotion with high-level tasks.** As we all know, high-level computer vision tasks (*e.g.*, image classification, image segmentation, and image analysis) are highly dependent on the quality of the input image, so SISR technology is usually used for pre-processing. Meanwhile, the quality of the SR images will greatly affect the accuracy of these tasks. Therefore, we recommend using the accuracy of high-level computer vision tasks as an evaluation indicator to measure the quality of the SR image. For the time being, we can design some loss functions related to high-level tasks, thus we can combine the feedback from other tasks to further improve the quality of SR images.

## 6.3 Scientific Articles

This dissertation has led to the following publications:

### 6.3.1 Submitted Journals

- **Parichehr Behjati**, Pau Rodriguez, Carles Fernández, Isabelle Hupont, Armin Mehri, and Jordi González. Single Image Super-Resolution Based on Directional Variance Attention Networks. *Pattern Recognition*, 2022
- **Parichehr Behjati**, Pau Rodriguez, Carles Fernández, Armin Mehri, F. Xavier Roca, Seiichi Ozawa, and Jordi González. Frequency-based Enhancement Network for Efficient Super-Resolution. *IEEE ACCESS*, 2022
- Armin Mehri, **Parichehr Behjati**, and Angel D.Sappa. SRFormer: Efficient

super-resolution based network for single image super resolution. *IEEE Transactions on Image Processing*, 2022.

### 6.3.2 International Conferences and Workshops

- **Parichehr Behjati**, David Vazquez, Josep M. Gonfaus, Josep M. Gonfaus, Pau Rodriguez, and Jordi Gonzalez. Catastrophic interference in Disguised Face Recognition. In *IbPRIA*, pages 64-75, 2019
- **Parichehr Behjati**, Pau rodriguez, Carles Fernandez Tena, Isabelle Hupont, Armin Mehri, and Jordi Gonzalez. OverNet: Lightweight multi-scale super-resolution with overscaling network. In *WACV*, pages 2704–2713, 2021.
- Armin Mehri, **Parichehr Behjati**, and Angel D.Sappa. MPRNet: Multi-path residual network for lightweight image super resolution. In *WACV*, pages 2704–2713, 2021.
- Armin Mehri, **Parichehr Behjati**, and Angel D.Sappa. Linet: A lightweight network for image super resolution. In *ICPR*, pages 7196–7202, 2021.

### 6.4 Contributed Code

- **OverNet**: code to reproduce the results presented in [6] within the PyTorch framework. <https://github.com/pbehjatii/OverNet>
- **DiVANet**: code to reproduce the results presented in chapter 4 using PyTorch. <https://github.com/pbehjatii/DiVANet>
- **FENet**: code to reproduce the results presented in chapter 5 within the PyTorch framework. <https://github.com/pbehjatii/FENet>

# A Experiments on Other Image Enhancement Tasks

Image enhancement technology has become one of the most important applications in computer vision and computer graphics and attracted increasing attention in the field of digital image processing, such as image dehazing, image super-resolution, image deblurring, and image denoising.

In this thesis, we have presented a novel set of lightweight CNN-based algorithms to generate state-of-the-art (both in terms of performance and inference time) super-resolution proposals. We have shown the robustness and effectiveness of our methods in handling three degradation models (BI, BD, and DN) and the provided evidence suggests that the proposed methods (FENet, OverNet, and DiVANet) may help with other image enhancement tasks.

In this chapter, in order to comprehensively illustrate the efficacy of the proposed methods, we further apply our methods to other image enhancement tasks. We show that the proposed algorithms presented in chapter 3, 4, and 5, and can also achieve state-of-the-art performance on:

- **Image dehazing.** Almost every visible light photon reflects from surfaces and is captured by the camera, but with the presence of aerosols such as dust, mist, and fumes, the light reflected in the matter is scattered and deviated from its original propagation path before it reaches the camera sensor. This has a substantial effect on the captured image and creates a so-called haze phenomenon, which reduces visibility of the scene content. A hazy image is a degraded image with poor contrast and faint surface color intensity (see Figure A.1(b)), therefore, estimating these values is of great interest in getting enhanced quality haze-free images. Image dehazing aims to recover the clean image from the corrupted input which is needed in many real-world applications when a high-quality image is required and also in areas where fog and haze are very common (*e.g.*, satellite imaging, archaeology, and traffic detection) [75].
- **Image compression artifacts reduction.** With the rapid development of consumer devices (*e.g.*, digital cameras and smartphones) and wireless network, the number of images and videos has achieved explosive growth, which has



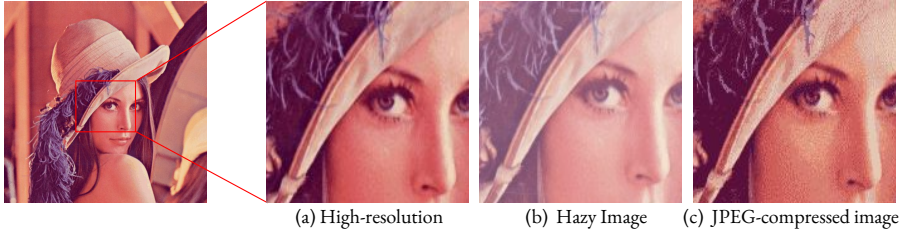


Figure A.1 – (a): High-resolution image (HR). (b): hazy image. (c): the JPEG-compressed image, where we could see blocking artifacts, ringing effects and blurring on the eyes, abrupt intensity changes on the face.

brought more pressure and challenges to storage and transmission systems. To save the storage capacity and transmission bandwidth, captured images and videos are usually compressed to reduce information redundancy. Lossy compression algorithms, *e.g.*, Joint Photographic Experts Group (JPEG) [119] and High Efficiency Video Coding (HEVC) [107], have been widely explored to achieve this goal. However, due to the inevitable signal loss during compression, these compression algorithms usually generate visually unpleasing compression artifacts. These artifacts not only decrease the visual quality, but also degrade the performance of downstream computer vision systems, especially at high compression ratios. Therefore, removing compression artifacts is an important postprocessing task and has attracted more attention in recent years (*e.g.* Figure A.1(c)).

In the next sections, we systematically compare the proposed methods in chapter 3, 4, and 5 (FENet, OverNet, and DiVANet) with state-of-the-art image dehazing and JPEG compression artifact reduction methods respectively.

### A.1 Experimental Settings

**Datasets and metrics.** For image dehazing, we used RESIDE dataset, which contains synthetic hazy images in both indoor and outdoor scenarios. The Indoor training set of RESIDE contains 1399 clean image and 13990 hazy images generated by corresponding clean images. The global atmosphere light ranges from 0.8 to 1.0, the scatter parameters change from 0.04 to 0.2. We evaluate our models on SOTS dataset, which contains 500 indoor images and 500 outdoors ones. We also evaluate our methods on realistic hazy images. For image compression artifacts reduction, we use DIV2K dataset for training. We evaluate our methods on two benchmark

datasets (LIVE1 [103] and Classic5 [29]) for JPEG quality factors 10, 20, 30, and 40. All results are evaluated with a commonly used metric: PSNR and SSIM.

**Implementation details.** During training, data augmentation is carried out by means of random horizontal flips and  $90^\circ$  rotation. At each training mini-batch, 64 RGB patches of size  $64 \times 64$  are provided as inputs. We train our models using an ADAM optimizer with learning rate  $10^{-3}$ . The learning rate is decreased by half every  $2 \times 10^5$  iterations. Our networks have been implemented using PyTorch, and trained on a NVIDIA RTX 3090 GPU.

## A.2 Results on Image Dehazing

In this section, we will compare the proposed FENet, OverNet, and DiVANet methods with previous state-of-the-art image dehazing algorithms both quantitatively and qualitatively including DCP [37], AODNet [62], DehazeNet [13], GFN [99], GCANet [16], and FFA-Net [95].

As shown in Table A.1, it can be observed that our proposed models outperform all state-of-the-art methods by a large margin. In Figure A.2, A.3, and A.4, we additionally provide some visual results. From the indoor and outdoor results (Figure A.2 and A.3), it can be observed that our methods prevent distortions, suppress artifacts and generate more faithful results. Moreover, our networks are clearly superior in the realistic performance of image details and color fidelity (Figure A.4).

Table A.1 – Quantitative comparisons (average PSNR and SSIM) on SOTS for different methods. Best and second best performance are in red and blue colors, respectively.

Methods	Indoor		Outdoor	
	PSNR	SSIM	PSNR	SSIM
DCP [37]	16.62	0.8179	19.13	0.8148
AOD-Net [139]	19.06	0.8504	320.29	0.8765
DehazeNet [139]	21.14	0.8472	22.46	0.8514
GFN [99]	22.30	0.8800	21.55	0.8444
GCANet [139]	30.23	0.9800	–	–
FFA-Net [95]	36.39	0.9886	33.57	0.9840
FENet (Ours)	36.45	0.9892	33.64	0.9849
OverNet (Ours)	<b>36.56</b>	<b>0.9899</b>	<b>33.72</b>	<b>0.9855</b>
DiVANet (Ours)	<b>36.61</b>	<b>0.9906</b>	<b>33.82</b>	<b>0.9860</b>

## Appendix A. Experiments on Other Image Enhancement Tasks

---

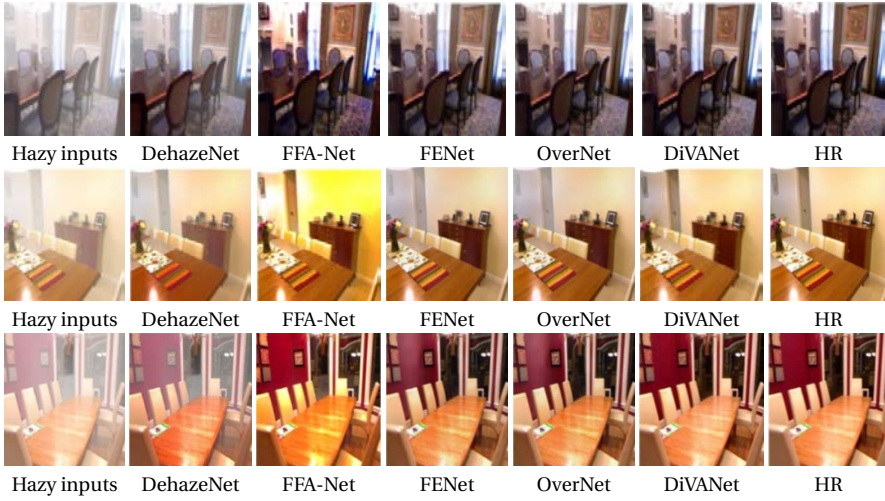


Figure A.2 – Qualitative comparisons on SOTS dataset (indoor).

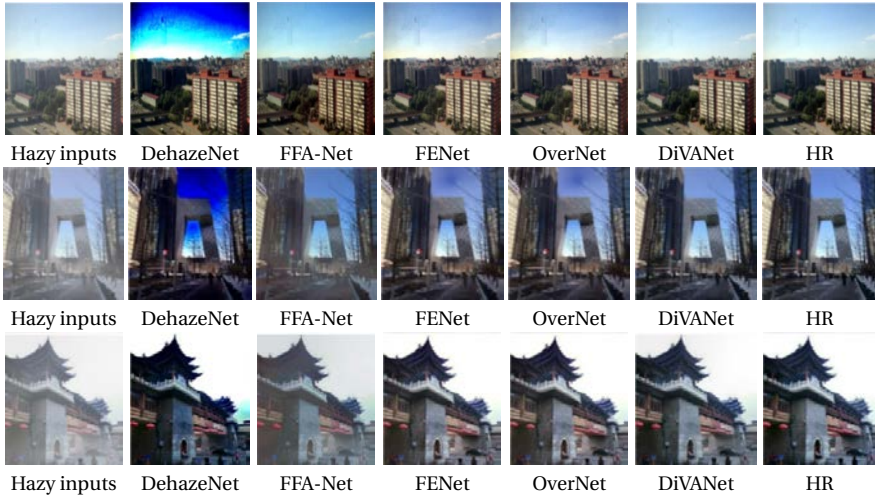


Figure A.3 – Qualitative comparisons on SOTS dataset (outdoor).

### A.3. Results on JPEG Compression Artifacts Reduction

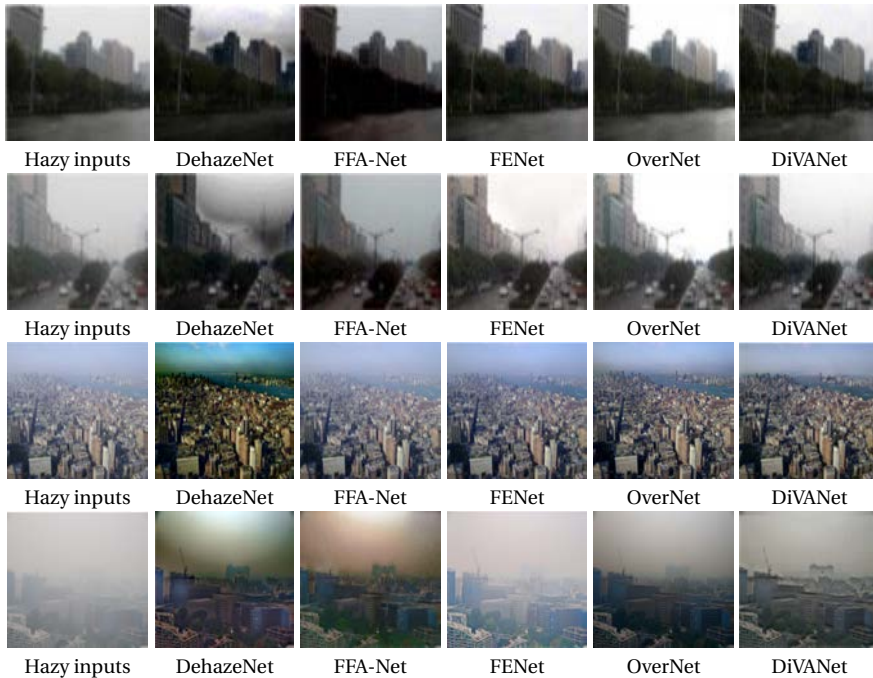


Figure A.4 – Qualitative comparisons on realistic hazy images.

### A.3 Results on JPEG Compression Artifacts Reduction

In this section, we further apply the proposed methods to reduce image compression reduction. We compare our methods with state-of-the-art methods, including ARCNN [22], DnCNN-3 [139], QGAC [27], RNAN [146], and RDN [147].

As shown in Table A.2, our methods achieves the best performance on LIVE1 and Classic5 with all JPEG qualities. Besides, compared with the previous best model such as RDN, our methods have only less than 1M parameters, while RDN is a large model that has 22M parameters.

In Figure A.5, we provide some visual comparisons for JPEG compression artifacts reduction. The blocking artifacts can be removed to some degree, but RDN would also suffers from over-smoothness, and cannot recover rich textures. By contrast, our methods can remove heavy noise corruption and preserve high-frequency image details, resulting in sharper edges and more natural textures.

## Appendix A. Experiments on Other Image Enhancement Tasks

Table A.2 – Quantitative comparisons (average PSNR and SSIM) with state-of-the-art methods for JPEG compression artifact reduction on benchmark datasets. Best and second best performance are in red and blue colors, respectively.

Methods	$\sigma$	Classic5		LIVE1	
		PSNR	SSIM	PSNR	SSIM
ARCNN [22]	10	29.03	0.7929	28.96	0.8076
	20	31.15	0.8517	31.29	.8733
	30	32.51	0.8806	32.67	0.9043
	40	33.32	0.8953	33.63	0.9198
DnCNN-3 [139]	10	29.40	0.8026	29.19	0.8123
	20	31.63	0.8610	31.59	0.8802
	30	32.91	0.8861	32.98	0.9090
	40	33.77	0.9003	33.96	0.9247
QGAC [27]	10	29.84	0.8370	29.53	0.8400
	20	31.98	0.8850	31.86	0.9010
	30	33.22	0.9070	33.23	0.9250
	40	–	–	–	–
RNAN[146]	10	29.96	0.8178	29.63	0.8239
	20	32.11	0.8693	32.03	0.8877
	30	33.38	0.8924	33.45	0.9149
	40	34.27	0.9061	34.47	0.9299
RDN [147]	10	30.00	0.8188	29.67	0.8247
	20	32.15	0.8699	32.07	0.8882
	30	33.43	0.8930	33.51	0.9153
	40	34.27	0.9061	34.51	0.9302
FENet (Ours)	10	30.16	0.8234	29.79	0.8278
	20	32.39	0.8734	32.17	0.8899
	30	33.59	0.8949	33.59	0.9166
	40	34.41	0.9075	34.58	0.9312
OverNet (Ours)	10	30.20	0.8245	29.83	0.8284
	20	32.43	0.8744	32.24	0.8903
	30	33.65	0.8966	33.63	0.9170
	40	34.46	0.9079	34.64	0.9314
DIVANet (Ours)	10	30.23	0.8249	29.86	0.8287
	20	32.47	0.8748	32.26	0.8909
	30	33.67	0.8961	33.67	0.9174
	40	34.50	0.9082	34.69	0.9317

### A.3. Results on JPEG Compression Artifacts Reduction



Figure A.5 – Image compression artifacts reduction results with JPEG quality  $\sigma = 10$ .

### A.4 Summary

In this thesis, we have demonstrated that the proposed methods (FENet, OverNet, and DiVANet) achieve state-of-the-art performance on five image restoration tasks: lightweight image SR, image denoising, image deblurring, image dehazing, and JPEG compression artifact reduction, which demonstrates the effectiveness and generalizability of the proposed methods. In the future, we will extend the proposed methods to other restoration tasks such as image inpainting and deraining.

## Bibliography

- [1] N. Ahn, B. Kang, and K.-A. Sohn. Fast, accurate, and lightweight super-resolution with cascading residual network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 252–268, 2018.
- [2] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 33(5):898–916, 2010.
- [3] Y. Bai, Y. Zhang, M. Ding, and B. Ghanem. Sod-mtgan: Small object detection via multi-task generative adversarial network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 206–221, 2018.
- [4] S. M. A. Bashir and Y. Wang. Deep learning for the assisted diagnosis of movement disorders, including isolated dystonia. *Frontiers in neurology*, 12, 2021.
- [5] M. Bates, B. Huang, G. T. Dempsey, and X. Zhuang. Multicolor super-resolution imaging with photo-switchable fluorescent probes. *Science*, 317(5845):1749–1753, 2007.
- [6] P. Behjati, P. Rodriguez, A. Mehri, I. Hupont, C. F. Tena, and J. Gonzalez. Overnet: Lightweight multi-scale super-resolution with overscaling network. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2694–2703, 2021.
- [7] Y. Bei, A. Damian, S. Hu, S. Menon, N. Ravi, and C. Rudin. New techniques for preserving global structure and denoising with low information loss in single-image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 874–881, 2018.
- [8] G. Berger, C. Peyrard, and M. Baccouche. Boosting face recognition via neural super-resolution. In *ESANN*, 2016.
- [9] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel. Low-complexity single-image super-resolution based on nonnegative neighbor embedding. 2012.



- [10] Y. Blau and T. Michaeli. The perception-distortion tradeoff. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6228–6237, 2018.
- [11] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor. The 2018 pirm challenge on perceptual image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV) Workshops*, pages 0–0, 2018.
- [12] A. Bulat and G. Tzimiropoulos. Super-fan: Integrated facial landmark localization and super-resolution of real-world low resolution faces in arbitrary poses with gans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 109–117, 2018.
- [13] B. Cai, X. Xu, K. Jia, C. Qing, and D. Tao. Dehazenet: An end-to-end system for single image haze removal. *IEEE Transactions on Image Processing*, 25(11): 5187–5198, 2016.
- [14] F. W. Campbell and J. G. Robson. Application of fourier analysis to the visibility of gratings. *The Journal of physiology*, 197(3):551–566, 1968.
- [15] K. Chang, X. Zhang, P. L. K. Ding, and B. Li. Data-adaptive low-rank modeling and external gradient prior for single image super-resolution. *Signal Processing*, 161:36–49, 2019.
- [16] D. Chen, M. He, Q. Fan, J. Liao, L. Zhang, D. Hou, L. Yuan, and G. Hua. Gated context aggregation network for image dehazing and deraining. In *2019 IEEE winter conference on applications of computer vision (WACV)*, pages 1375–1383. IEEE, 2019.
- [17] H. Chen, X. He, L. Qing, Y. Wu, C. Ren, R. E. Sheriff, and C. Zhu. Real-world single image super-resolution: A brief review. *Information Fusion*, 79:124–145, 2022.
- [18] R. Chen, H. Zhang, and J. Liu. Multi-attention augmented network for single image super-resolution. *Pattern Recognition*, 122:108349, 2022.
- [19] J.-S. Choi and M. Kim. A deep convolutional neural network with selection units for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 154–160, 2017.
- [20] X. Chu, B. Zhang, H. Ma, R. Xu, and Q. Li. Fast, accurate and lightweight super-resolution with neural architecture search. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 59–64. IEEE, 2021.

- 
- [21] T. Dai, J. Cai, Y. Zhang, S.-T. Xia, and L. Zhang. Second-order attention network for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11065–11074, 2019.
- [22] C. Dong, Y. Deng, C. C. Loy, and X. Tang. Compression artifacts reduction by a deep convolutional network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 576–584, 2015.
- [23] C. Dong, C. C. Loy, K. He, and X. Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015.
- [24] C. Dong, C. C. Loy, and X. Tang. Accelerating the super-resolution convolutional neural network. In *European conference on computer vision*, pages 391–407. Springer, 2016.
- [25] W. Dong, L. Zhang, G. Shi, and X. Wu. Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization. *IEEE Transactions on image processing*, 20(7):1838–1857, 2011.
- [26] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox. Flownet: Learning optical flow with convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2758–2766, 2015.
- [27] M. Ehrlich, L. Davis, S.-N. Lim, and A. Shrivastava. Quantization guided jpeg artifact correction. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part VIII 16*, pages 293–309. Springer, 2020.
- [28] M. Fernández-Suárez and A. Y. Ting. Fluorescent probes for super-resolution imaging in living cells. *Nature reviews Molecular cell biology*, 9(12):929–943, 2008.
- [29] A. Foi, V. Katkovnik, and K. Egiazarian. Pointwise shape-adaptive dct for high-quality denoising and deblocking of grayscale and color images. *IEEE transactions on image processing*, 16(5):1395–1411, 2007.
- [30] L. A. Gatys, A. S. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

- [31] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.
- [32] K. Grm, W. J. Scheirer, and V. Štruc. Face hallucination using cascaded super-resolution and identity priors. *IEEE Transactions on Image Processing*, 29(1): 2150–2165, 2019.
- [33] J. Hamaide, G. De Groof, G. Van Steenkiste, B. Jeurissen, J. Van Audekerke, M. Naeyaert, L. Van Ruijssevelt, C. Cornil, J. Sijbers, M. Verhoye, et al. Exploring sex differences in the adult zebra finch brain: in vivo diffusion tensor imaging and ex vivo super-resolution track density imaging. *Neuroimage*, 146:789–803, 2017.
- [34] M. Haris, G. Shakhnarovich, and N. Ukita. Deep back-projection networks for super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1664–1673, 2018.
- [35] M. Haris, G. Shakhnarovich, and N. Ukita. Task-driven super resolution: Object detection in low-resolution images. In *International Conference on Neural Information Processing*, pages 387–395. Springer, 2021.
- [36] K. He and J. Sun. Convolutional neural networks at constrained time cost. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5353–5360, 2015.
- [37] K. He, J. Sun, and X. Tang. Single image haze removal using dark channel prior. *IEEE transactions on pattern analysis and machine intelligence*, 33(12): 2341–2353, 2010.
- [38] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.
- [39] K. He, X. Zhang, S. Ren, and J. Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pages 630–645. Springer, 2016.
- [40] X. He, Z. Mo, P. Wang, Y. Liu, M. Yang, and J. Cheng. Ode-inspired network design for single image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1732–1741, 2019.

- 
- [41] J. Hu, L. Shen, and G. Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.
- [42] X. Hu, H. Mu, X. Zhang, Z. Wang, T. Tan, and J. Sun. Meta-sr: A magnification-arbitrary network for super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1575–1584, 2019.
- [43] Y. Hu, J. Li, Y. Huang, and X. Gao. Channel-wise and spatial feature modulation network for single image super-resolution. *IEEE Transactions on Circuits and Systems for Video Technology*, 30(11):3911–3927, 2019.
- [44] B. Huang, W. Wang, M. Bates, and X. Zhuang. Three-dimensional super-resolution imaging by stochastic optical reconstruction microscopy. *Science*, 319(5864):810–813, 2008.
- [45] J.-B. Huang, A. Singh, and N. Ahuja. Single image super-resolution from transformed self-exemplars. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5197–5206, 2015.
- [46] J.-J. Huang, T. Liu, P. Luigi Dragotti, and T. Stathaki. Srrhf+: Self-example enhanced single image super-resolution using hierarchical random forests. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 71–79, 2017.
- [47] Z. Hui, X. Gao, Y. Yang, and X. Wang. Lightweight image super-resolution with information multi-distillation network. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 2024–2032, 2019.
- [48] M. Irani and S. Peleg. Improving resolution by image registration. *CVGIP: Graphical models and image processing*, 53(3):231–239, 1991.
- [49] K. Jiang, Z. Wang, P. Yi, and J. Jiang. Hierarchical dense recursive network for image super-resolution. *Pattern Recognition*, 107:107475, 2020.
- [50] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *European conference on computer vision*, pages 694–711. Springer, 2016.
- [51] J. Jurek, M. Kociński, A. Materka, M. Elgalal, and A. Majos. Cnn-based super-resolution reconstruction of 3d mr images using thick-slice scans. *Biocybernetics and Biomedical Engineering*, 40(1):111–125, 2020.

- [52] J. Kim, J. K. Lee, and K. M. Lee. Accurate image super-resolution using very deep convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1646–1654, 2016.
- [53] J. Kim, J. K. Lee, and K. M. Lee. Deeply-recursive convolutional network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1637–1645, 2016.
- [54] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [55] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.
- [56] A. Laghrib, A. Hadri, A. Hakim, and S. Raghay. A new multiframe super-resolution based on nonlinear registration and a spatially weighted regularization. *Information Sciences*, 493:34–56, 2019.
- [57] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Deep laplacian pyramid networks for fast and accurate super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 624–632, 2017.
- [58] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang. Fast and accurate image super-resolution with deep laplacian pyramid networks. *IEEE transactions on pattern analysis and machine intelligence*, 41(11):2599–2613, 2018.
- [59] R. Lan, L. Sun, Z. Liu, H. Lu, C. Pang, and X. Luo. Madnet: A fast and lightweight network for single-image super resolution. *IEEE transactions on cybernetics*, 51(3):1443–1453, 2020.
- [60] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, et al. Photo-realistic single image super-resolution using a generative adversarial network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4681–4690, 2017.
- [61] S. Lee, J.-H. Kim, and J.-P. Heo. Super-resolution of license plate images via character-based perceptual loss. In *2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pages 560–563. IEEE, 2020.
- [62] B. Li, X. Peng, Z. Wang, J. Xu, and D. Feng. Aod-net: All-in-one dehazing network. In *Proceedings of the IEEE international conference on computer vision*, pages 4770–4778, 2017.

- [63] B. Li, Y. Zhou, Y. Zhang, and A. Wang. Depth image super-resolution based on joint sparse coding. *Pattern Recognition Letters*, 130:21–29, 2020.
- [64] J. Li and W. Guan. Adaptive lq-norm constrained general nonlocal self-similarity regularizer based sparse representation for single image super-resolution. *Information Fusion*, 53:88–102, 2020.
- [65] J. Li, F. Fang, K. Mei, and G. Zhang. Multi-scale residual network for image super-resolution. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 517–532, 2018.
- [66] S. Li, Q. Cai, H. Li, J. Cao, L. Wang, and Z. Li. Frequency separation network for image super-resolution. *IEEE Access*, 8:33768–33777, 2020.
- [67] W. Li, K. Zhou, L. Qi, N. Jiang, J. Lu, and J. Jia. Lpar: Linearly-assembled pixel-adaptive regression network for single image super-resolution and beyond. *Advances in Neural Information Processing Systems*, 33, 2020.
- [68] X. Li, G. Cao, Y. Zhang, A. Shafique, and P. Fu. Combining synthesis sparse with analysis sparse for single image super-resolution. *Signal Processing: Image Communication*, 83:115805, 2020.
- [69] Z. Li, J. Yang, Z. Liu, X. Yang, G. Jeon, and W. Wu. Feedback network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3867–3876, 2019.
- [70] Y. Liang, R. Timofte, J. Wang, S. Zhou, Y. Gong, and N. Zheng. Single-image super-resolution-when model adaptation matters. *Pattern Recognition*, 116:107931, 2021.
- [71] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee. Enhanced deep residual networks for single image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 136–144, 2017.
- [72] J. Liu, W. Zhang, Y. Tang, J. Tang, and G. Wu. Residual feature aggregation network for image super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2359–2368, 2020.
- [73] R. Liu, Z. Su, G. Lin, and F. Zhou. Second-order attention network for magnification-arbitrary single image super-resolution. In *2020 8th International Conference on Digital Home (ICDH)*, pages 127–134. IEEE, 2020.

- [74] X. Liu, L. Chen, W. Wang, and J. Zhao. Robust multi-frame super-resolution based on spatially weighted half-quadratic estimation and adaptive btv regularization. *IEEE Transactions on Image Processing*, 27(10):4971–4986, 2018.
- [75] Y. Liu, J. Pan, J. Ren, and Z. Su. Learning deep priors for image dehazing. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 2492–2500, 2019.
- [76] A. Lucas, S. Lopez-Tapia, R. Molina, and A. K. Katsaggelos. Generative adversarial networks and perceptual losses for video super-resolution. *IEEE Transactions on Image Processing*, 28(7):3312–3327, 2019.
- [77] X. Luo, Y. Xie, Y. Zhang, Y. Qu, C. Li, and Y. Fu. Latticenet: Towards lightweight image super-resolution with lattice block. In *Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXII 16*, pages 272–289. Springer, 2020.
- [78] H. Lyu, N. Sha, S. Qin, M. Yan, Y. Xie, and R. Wang. Advances in neural information processing systems. *Advances in neural information processing systems*, 32, 2019.
- [79] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang. Learning a no-reference quality metric for single-image super-resolution. *Computer Vision and Image Understanding*, 158:1–16, 2017.
- [80] X. Mao, C. Shen, and Y.-B. Yang. Image restoration using very deep convolutional encoder-decoder networks with symmetric skip connections. *Advances in neural information processing systems*, 29:2802–2810, 2016.
- [81] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017.
- [82] Y. Matsui, K. Ito, Y. Aramaki, A. Fujimoto, T. Ogawa, T. Yamasaki, and K. Aizawa. Sketch-based manga retrieval using manga109 dataset. *Multimedia Tools and Applications*, 76(20):21811–21838, 2017.
- [83] A. Mehri, P. B. Ardakani, and A. D. Sappa. Linet: A lightweight network for image super resolution. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 7196–7202. IEEE, 2021.
- [84] A. Mehri, P. B. Ardakani, and A. D. Sappa. Mprnet: Multi-path residual network for lightweight image super resolution. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2704–2713, 2021.

- [85] Y. Mei, Y. Fan, Y. Zhou, L. Huang, T. S. Huang, and H. Shi. Image super-resolution with cross-scale non-local attention and exhaustive self-exemplars mining. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5690–5699, 2020.
- [86] Y. Mei, Y. Fan, and Y. Zhou. Image super-resolution with non-local sparse attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3517–3526, 2021.
- [87] A. Mittal, R. Soundararajan, and A. C. Bovik. Making a “completely blind” image quality analyzer. *IEEE Signal processing letters*, 20(3):209–212, 2012.
- [88] A. Muqet, J. Hwang, S. Yang, J. Kang, Y. Kim, and S.-H. Bae. Multi-attention based ultra lightweight image super-resolution. In *European Conference on Computer Vision*, pages 103–118. Springer, 2020.
- [89] A. Odena, V. Dumoulin, and C. Olah. Deconvolution and checkerboard artifacts. *Distill*, 1(10):e3, 2016.
- [90] S. C. Park, M. K. Park, and M. G. Kang. Super-resolution image reconstruction: a technical overview. *IEEE signal processing magazine*, 20(3):21–36, 2003.
- [91] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.
- [92] S. Peled and Y. Yeshurun. Superresolution in mri: application to human white matter fiber tract visualization by diffusion tensor imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, 45(1):29–35, 2001.
- [93] T. Peleg and M. Elad. A statistical prediction model based on sparse representations for single image super-resolution. *IEEE transactions on image processing*, 23(6):2569–2582, 2014.
- [94] E. Pérez-Pellitero, J. Salvador, J. Ruiz-Hidalgo, and B. Rosenhahn. Antipodally invariant metrics for fast regression-based super-resolution. *IEEE Transactions on Image Processing*, 25(6):2456–2468, 2016.
- [95] X. Qin, Z. Wang, Y. Bai, X. Xie, and H. Jia. Ffa-net: Feature fusion attention network for single image dehazing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11908–11915, 2020.



- [96] Y. Qiu, R. Wang, D. Tao, and J. Cheng. Embedded block residual network: A recursive restoration model for single-image super-resolution. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4180–4189, 2019.
- [97] M. Rajnoha, A. Mezina, and R. Burget. Multi-frame labeled faces database: Towards face super-resolution from realistic video sequences. *Applied Sciences*, 10(20):7213, 2020.
- [98] H. Ren, M. El-Khamy, and J. Lee. Image super resolution based on fusing multiple convolution neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 54–61, 2017.
- [99] W. Ren, L. Ma, J. Zhang, J. Pan, X. Cao, W. Liu, and M.-H. Yang. Gated fusion network for single image dehazing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3253–3261, 2018.
- [100] L. Russell. De valois and karen k. de valois. *Spatial Vision*, (14), 1988.
- [101] M. S. Sajjadi, B. Scholkopf, and M. Hirsch. Enhancenet: Single image super-resolution through automated texture synthesis. In *Proceedings of the IEEE international conference on computer vision*, pages 4491–4500, 2017.
- [102] P. Shamsolmoali, M. Zareapoor, D. K. Jain, V. K. Jain, and J. Yang. Deep convolution network for surveillance records super-resolution. *Multimedia Tools and Applications*, 78(17):23815–23829, 2019.
- [103] H. Sheikh. Live image quality assessment database release 2. <http://live.ece.utexas.edu/research/quality>, 2005.
- [104] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016.
- [105] A. Shocher, N. Cohen, and M. Irani. “zero-shot” super-resolution using deep internal learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3118–3126, 2018.
- [106] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

- 
- [107] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand. Overview of the high efficiency video coding (hevc) standard. *IEEE Transactions on circuits and systems for video technology*, 22(12):1649–1668, 2012.
- [108] J. Sun, Z. Xu, and H.-Y. Shum. Image super-resolution using gradient profile prior. In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.
- [109] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi. Inception-v4, inception-resnet and the impact of residual connections on learning. In *Thirty-first AAAI conference on artificial intelligence*, 2017.
- [110] Y. Tai, J. Yang, and X. Liu. Image super-resolution via deep recursive residual network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3147–3155, 2017.
- [111] Y. Tai, J. Yang, X. Liu, and C. Xu. Memnet: A persistent memory network for image restoration. In *Proceedings of the IEEE international conference on computer vision*, pages 4539–4547, 2017.
- [112] X. Tao, H. Gao, R. Liao, J. Wang, and J. Jia. Detail-revealing deep video super-resolution. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4472–4480, 2017.
- [113] I. Teh, D. McClymont, E. Carruth, J. Omens, A. McCulloch, and J. E. Schneider. Improved compressed sensing and super-resolution of cardiac diffusion mri with structure-guided total variation. *Magnetic resonance in medicine*, 84(4): 1868–1880, 2020.
- [114] Y. Tian, Y. Zhang, Y. Fu, and C. Xu. Tdan: Temporally-deformable alignment network for video super-resolution. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3360–3369, 2020.
- [115] R. Timofte, V. De Smet, and L. Van Gool. A+: Adjusted anchored neighborhood regression for fast super-resolution. In *Asian conference on computer vision*, pages 111–126. Springer, 2014.
- [116] R. Timofte, R. Rothe, and L. Van Gool. Seven ways to improve example-based single image super resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1865–1873, 2016.
- [117] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, and L. Zhang. Ntire 2017 challenge on single image super-resolution: Methods and results. In

- Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 114–125, 2017.
- [118] T. Tong, G. Li, X. Liu, and Q. Gao. Image super-resolution using dense skip connections. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4799–4807, 2017.
- [119] G. K. Wallace. The jpeg still picture compression standard. *IEEE transactions on consumer electronics*, 38(1):xviii–xxxiv, 1992.
- [120] F. Wang, H. Hu, and C. Shen. Bam: A lightweight and efficient balanced attention mechanism for single image super resolution. *arXiv preprint arXiv:2104.07566*, 2021.
- [121] L. Wang and K.-J. Yoon. Semi-supervised student-teacher learning for single image super-resolution. *Pattern Recognition*, 121:108206, 2022.
- [122] X. Wang, K. Yu, C. Dong, and C. C. Loy. Recovering realistic texture in image super-resolution by deep spatial feature transform. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 606–615, 2018.
- [123] X. Wang, K. Yu, S. Wu, J. Gu, Y. Liu, C. Dong, Y. Qiao, and C. Change Loy. Esrgan: Enhanced super-resolution generative adversarial networks. In *Proceedings of the European conference on computer vision (ECCV) workshops*, pages 0–0, 2018.
- [124] Y. Wang, F. Perazzi, B. McWilliams, A. Sorkine-Hornung, O. Sorkine-Hornung, and C. Schroers. A fully progressive approach to single-image super-resolution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 864–873, 2018.
- [125] Z. Wang and A. C. Bovik. Mean squared error: Love it or leave it? a new look at signal fidelity measures. *IEEE signal processing magazine*, 26(1):98–117, 2009.
- [126] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [127] Z. Wang, J. Chen, and S. C. Hoi. Deep learning for image super-resolution: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 2020.
- [128] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

- 
- [129] L.-Y. Xu and Z. Gajic. Improved network for face recognition based on feature super resolution method. *International Journal of Automation and Computing*, 18(6):915–925, 2021.
- [130] Y. Z. J. Y. L. F. Xuehui Wang, Qing Wang and L. Chen. Lightweight single-image super-resolution network with attentive auxiliary feature learning. 2020.
- [131] C. Yang and G. Lu. Deeply recursive low-and high-frequency fusing networks for single image super-resolution. *Sensors*, 20(24):7268, 2020.
- [132] J. Yang, J. Wright, T. S. Huang, and Y. Ma. Image super-resolution via sparse representation. *IEEE transactions on image processing*, 19(11):2861–2873, 2010.
- [133] Y. Yang and Y. Qi. Image super-resolution via channel attention and spatial graph convolutional network. *Pattern Recognition*, 112:107798, 2021.
- [134] J. Yu, Y. Fan, J. Yang, N. Xu, Z. Wang, X. Wang, and T. Huang. Wide activation for efficient and accurate image super-resolution. *arXiv preprint arXiv:1808.08718*, 2018.
- [135] Y. Yuan, S. Liu, J. Zhang, Y. Zhang, C. Dong, and L. Lin. Unsupervised image super-resolution using cycle-in-cycle generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 701–710, 2018.
- [136] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.
- [137] M. D. Zeiler, D. Krishnan, G. W. Taylor, and R. Fergus. Deconvolutional networks. In *2010 IEEE Computer Society Conference on computer vision and pattern recognition*, pages 2528–2535. IEEE, 2010.
- [138] R. Zeyde, M. Elad, and M. Protter. On single image scale-up using sparse-representations. In *International conference on curves and surfaces*, pages 711–730. Springer, 2010.
- [139] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang. Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising. *IEEE transactions on image processing*, 26(7):3142–3155, 2017.

- [140] K. Zhang, W. Zuo, S. Gu, and L. Zhang. Learning deep cnn denoiser prior for image restoration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3929–3938, 2017.
- [141] K. Zhang, W. Zuo, and L. Zhang. Learning a single convolutional super-resolution network for multiple degradations. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3262–3271, 2018.
- [142] K. Zhang, Z. Wang, J. Li, X. Gao, and Z. Xiong. Learning recurrent residual regressors for single image super-resolution. *Signal Processing*, 154:324–337, 2019.
- [143] L. Zhang, H. Zhang, H. Shen, and P. Li. A super-resolution reconstruction algorithm for surveillance images. *Signal Processing*, 90(3):848–859, 2010.
- [144] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu. Image super-resolution using very deep residual channel attention networks. In *Proceedings of the European conference on computer vision (ECCV)*, pages 286–301, 2018.
- [145] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image super-resolution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2472–2481, 2018.
- [146] Y. Zhang, K. Li, K. Li, B. Zhong, and Y. Fu. Residual non-local attention networks for image restoration. *arXiv preprint arXiv:1903.10082*, 2019.
- [147] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu. Residual dense network for image restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2480–2495, 2020.
- [148] F. Zhu and Q. Zhao. Efficient single image super-resolution via hybrid residual feature learning with compact back-projection network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.
- [149] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.