Single Image Super-Resolution Using Capsule Generative Adversarial Network

Mr. Amir Hajian

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

A Dissertation Submitted in Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy in Electrical Engineering
Department of Electrical Engineering
FACULTY OF ENGINEERING
Chulalongkorn University
Academic Year 2021

Copyright of Chulalongkorn University

การสร้างคืนภาพความละเอียดสูงยิ่งยวดโดยใช้โครงข่ายปรปักษ์ก่อกำเนิดแบบแคปซูล

นายอาเมอร์ ฮาเจียน

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรดุษฎีบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย
ปีการศึกษา 2564

ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title	Single Image Super-Resolution Using Capsule Generative Adversarial Network
By	Mr. Amir Hajian
Field of Study	Electrical Engineering
Thesis Advisor	Associate Professor SUPAVADEE ARAMVITH, Ph.D.

Accepted by the FACULTY OF ENGINEERING, Chulalongkorn University in Partial Fulfillment of the Requirement for the Doctor of Philosophy

.... Dean of the FACULTY OF ENGINEERING
(Professor SUPOT TEACHAVORASINSKUN, D.Eng.)

DISSERTATION COMMITTEE

.... Chairman
(Professor Kosin Chamnongthai, Ph.D.)
.... Thesis Advisor
(Associate Professor SUPAVADEE ARAMVITH, Ph.D.)
.... Examiner
(Associate Professor NISACHON TANGSANGIUMVISAI, Ph.D.)
.... Examiner
(Associate Professor CHARNCHAI PLUEMPITIWIRIYAWEJ, Ph.D.)
.... Examiner
(Assistant Professor SUREE PUMRIN, Ph.D.)
.... Examiner
(PUNNARAI SIRICHAROEN, Ph.D.)

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

อาเมอร์ ฮาเจียน : การสร้างคืนภาพความละเอียดสูงยิ่งยวดโดยใช้โครงข่ายปรปักษ์ก่อกำเนิดแบบแคปซูล. (Single Image Super-Resolution Using Capsule Generative Adversarial Network) อ.ที่ปรึกษาหลัก : รศ. ดร.สุภาวดี อร่ามวิทย์

การวิจัยนี้ มีจุดมุ่งหมายเพื่อตรวจสอบและนำเสนอ ภายใต้สถาปัตยกรรม โครงข่ายแบบเจเนอเรทีฟแอดเวอเซอเรียล (Generative Adversarial Network : GAN) [53] โดยใช้สถาปัตยกรรมโครงข่ายแคปซูล [76] ในโมดูลแยกแยะของแบบจำลองแบบ Caps-GAN สำหรับการสร้างคืนภาพความละเอียดสูงยิ่งยวด นอกจากนี้ การศึกษานี้มีวัตถุประสงค์ เพื่อพัฒนากรอบการทำงานการสร้างคืนภาพความละเอียดสูงยิ่งยวด ในอัตราขยายภาพ 3 ขนาด และมีการวัดประสิทธิภาพของ Caps-GAN ก็ถูกนำมาเปรียบเทียบกับวิธีการอื่นๆ อีกด้วย โมเดล Caps-GAN ของเราประกอบด้วยส่วนประกอบพื้นฐาน 3 ส่วน คือ โมดูลตัวสร้าง โมดูลแยกแคปซูล และการรวมกันของฟังก์ชันการสูญเสียตามแนวคิด GAN โมดูลตัวสร้างที่นำเสนอ ใช้ส่วนข้อมูลที่เหลือในสถาปัตยกรรมบล็อกหนาแน่นตกค้าง (RRDB) [28] ภายใต้กรอบการสุ่มตัวอย่างแบบก้าวหน้า [30] ในขณะที่แนวคิดการประมาณการคอขวดเชิงลึก [38] ใช้เพื่อถ่ายโอนข้อมูลรายละเอียดความถี่สูงของชั้นข้อมูลช่วงต้น ไปยังแต่ละขั้นตอนการสุ่มตัวอย่าง เพื่อป้องกันไม่ให้เกรเดียนต์หายไป และฟังก์ชันวัตถุประสงค์ฟิวชั่นแบบใหม่ที่รวมการสูญเสีย SSIM หลายระดับและการสูญเสีย L2 (MS-SSIM + L2) เพื่อปรับปรุงผลลัพธ์เชิงปริมาณและเชิงคุณภาพตลอดจนการสร้างรายละเอียดที่ซับซ้อนขึ้นใหม่ ในโมเดล Caps-GAN ของเรามี โมดูลแยกแยะภายใต้โครงข่ายประสาทเทียม ซึ่งถูกแทนที่ด้วยสถาปัตยกรรมเครือข่ายแคปซูล ในการดึงความสัมพันธ์ของคุณลักษณะแบบลำดับชั้น ตัวแยกแยะแคปซูลของเราแสดงให้เห็นถึงประสิทธิภาพที่เหนือกว่าในการสกัดข้อมูล ที่ข้อมูลมีซับซ้อนและความยากเรียนรู้ ส่วนในการฝึกอบรมแบบจำลองของเรา แสดงให้เห็นว่ามีความสามารถในการฝึกอบรมโมเดล GAN ของได้ดีขึ้นและรวดเร็วขึ้น เมื่อเทียบกับผู้จำแนกตาม ภายใต้โครงข่ายประสาทเทียมอื่น โมดูลแยกแยะแบบแคปซูลที่ได้รับการฝึกฝนด้วยการสูญเสีย GAN [28] และโมดูลแยกแยะได้รับการฝึกฝนด้วยการสูญเสียการรับรู้ [8] การสูญเสียการรับรู้ของเราประกอบด้วยการสูญเสีย 2 ประเภท ได้แก่ การสูญเสียเนื้อหาที่ใช้แบบจำลองที่ได้รับการฝึกอบรมล่วงหน้า สำหรับการสร้างลักษณะโดยรวมของภาพ และการสูญเสียที่เป็นปฏิปักษ์สำหรับการสร้างรายละเอียดของพื้นผิวความถี่สูง การประเมินเชิงปริมาณและคุณภาพนั้น ใช้ชุดข้อมูลภาพมาตรฐานห้าชุด ได้แก่ Set5, Set14, BSDS100, Urban100, Manag109 และ DIV2K สำหรับการเปรียบเทียบเชิงปริมาณ เมตริกคุณภาพ ได้แก่ PSNR และ SSIM ตลอดจนการทดสอบ MOS สำหรับการเปรียบเทียบด้วยคุณภาพกับอัตราขยาย 2 ขนาด

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

สาขาวิชา	วิศวกรรมไฟฟ้า	ลายมือชื่อนิสิต ....
ปีการศึกษา	2564	ลายมือชื่อ อ.ที่ปรึกษาหลัก ....

6171461021 : MAJOR ELECTRICAL ENGINEERING

KEYWORD: Image super-resolution, Generative Adversarial Network, Capsule Network, Fusion Objective Function

Amir Hajian : Single Image Super-Resolution Using Capsule Generative Adversarial Network. Advisor: Assoc. Prof. SUPAVADEE ARAMVITH, Ph.D.

The current research aims to investigate and propose a Generative Adversarial Network (GAN) architecture [53] using capsule network architecture [76] in the discriminator module of the proposed model (Caps-GAN) for Single Image Super-Resolution. Besides, the study aims to develop the proposed SR framework in three scale factors. Finally, the performance of Caps-GAN is compared with other state-of-the-art models. Our Caps-GAN model consists of three fundamental components: the generator module, capsule discriminator module, and combinations of loss functions based on the GAN concept. The proposed generator utilizes the residual in residual dense blocks (RRDB) architecture [28] under a progressively up-sampling framework [30]. At the same time, the depth-wise bottleneck projections concept [38] is employed to transfer the high-frequency details of the early layer to each up-sampling stage to prevent gradient vanishing. Additionally, a novel fusion objective function that combines Multi-level SSIM loss and L2 loss (MS-SSIM + L2) is introduced to improve the quantitative and qualitative results and reconstruct the sophisticated details. In our Caps-GAN model, the CNN-based discriminator has been replaced with the capsule network architecture. Duo to the capability of the capsule network to extract the hierarchical feature relationships, our capsule discriminator demonstrates superior performance in extracting difficult-to-learn patterns in training our model. This capability leads to training our GAN model much better and faster than the CNN-based discriminator. The capsule discriminator is trained with GAN loss [28], and the generator is trained with a perceptual loss [8]. Our perceptual loss consists of two types of losses including a content loss (pre-trained model) for producing the overall appearance of the image, and an adversarial loss for producing high-frequency details of texture. The quantitative and visual evaluations are based on five benchmark datasets including, Set5, Set14, BSDS100, Urban100, Manag109, and DIV2K. For quantitative comparison, the quality metrics including PSNR and SSIM, and the MOS test for visual comparison at two scales.

Field of Study:	Electrical Engineering	Student's Signature ....
Academic Year:	2021	Advisor's Signature ....

CHULALONGKORN UNIVERSITY

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my supervisor Assoc. Prof. Dr. Supavadee Aramvith for her continuous support during the past three years. This thesis would have never been completed without her assistance.

I would also like to thank the members of the committee, Prof. Dr.Kosin Chamnongthai, Asst. Prof. Dr. Suree Pumrin, Asst. Prof. Dr. Charnchai Pluempittiwiriyawej, Lecturer. Dr. Punnarai Siricharoen and the chairperson, Assoc. Prof. Dr. Nisachon Tangsangiumvisai, for their critical reviews and advice on my dissertation.

I would like to thank “The 100th Anniversary Chulalongkorn University Fund for Doctoral Scholarship”, "The 90th Anniversary Chulalongkorn University Fund (Ratchadaphiseksomphot Endowment Fund), and “The Electrical Engineering Department, Chulalongkorn University Ph.D. (EECU-PhD) Honors Program Scholarship” scholarship for which this dissertation has been supported.

Also, I appreciate all members of the Video Technology Research Group (VTRG), Mr. Sovan Chen, Mr. Watchara Ruangsang, and Mr. Wazir Muhammad. They have spent their valuable time giving me suggestions and guidance.

Finally, I would like to express my deepest gratitude to my friend Dr. Hamid Rreza Sadeghpour for his constant support and affectionate encouragement during this journey.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Amir Hajian

TABLE OF CONTENTS

Page

.... iii

ABSTRACT (THAI) iii

.... iv

ABSTRACT (ENGLISH) iv

ACKNOWLEDGEMENTS v

TABLE OF CONTENTS vi

LIST OF TABLES ix

LIST OF FIGURES x

1. CHAPTER ONE 1

Introduction 1

  1.1 Motivation and Problem Statement 1

  1.2 Objectives 4

  1.3 Scope of Research 5

  1.4 Expected Output 5

  1.5 Research Procedure 5

  1.6 Thesis Layout 6

2. CHAPTER TWO 7

Background and Literature Review 7

  2.1 Introduction 7

  2.2 Super-resolution Frameworks 10

    2.2.1 Pre-upsampling Super-resolution 10

    2.2.2 Post-upsampling Super-resolution 11

    2.2.3 Progressive Up-sampling Framework 12

    2.2.4 Iterative Up-and-down Sampling Super-resolution 14

  2.3 Network Architecture 15

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.3.1 Residual Learning......15
2.3.2 Recursive Learning......16
2.3.3 Multi-path Learning......18
2.3.4 Dense Connections......20
2.3.5 Attention Mechanism......21
2.3.6 Region-recursive Learning......23
2.3.7 Pyramid Pooling......23
2.3.8 Generative Adversarial Network (GAN)......24
2.4 Loss Function......29
2.4.1 L2 Loss (MSE)......30
2.4.2 L1 Loss (MAE)......30
2.4.3 Charbonnier loss......31
2.4.4 Content Loss......32
2.4.5 Adversarial Loss......33
2.5 Other Improvement......35
2.5.1 Context-wise Network Fusion......35
2.5.2 Data Augmentation......35
2.5.3 Multitask Learning......36
2.5.4 Network Interpolation......37
2.5.5 Self-ensemble......37
2.6 Capsule Neural Network......37
3. CHAPTER THREE......40
Proposed Network Architecture......40
3.1 Introduction......40
3.2 Generator Architecture......41
3.2.1 The Architecture of Generator Network......43
3.2.2 Residual Bottleneck Projection......48
3.2.3 Objective Function......49
3.3 Capsule Discriminator Architecture......54

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.4 Total Loss of Caps-GAN 57
    3.4.1 GAN Loss 58
    3.4.2 Perceptual Loss 59
4. CHAPTER FOUR 63
Experimental Result and Discussion 63
4.1 Introduction 63
4.2 Training Details 63
4.3 Testing Details 64
4.4. Generator Module 66
    4.4.1 Effect of Different Projections on Training 66
    4.4.2 Comparison of Different Content Losses 68
    4.4.3 Comparison Results with Other Non-GAN models 71
    4.4.4 Generator Network Size Analysis 78
4.5 Discriminator Module Performance 80
4.6 Discussion of Caps-GAN Experimental Results 82
    4.6.1 Network Parameters, Execution Time, and Training Iterations 83
    4.6.2 Quantitative Comparison of Caps-GAN model 85
    4.6.3 Visual Comparison of Caps-GAN model 87
5. CHAPTER FIVE 102
Conclusions and Future Works 102
5.1 Introduction 102
5.2 Contributions 104
5.3 Future Work 104
REFERENCES 105
VITA 113

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

LIST OF TABLES

Page

Table 4.1: Quantitative comparison among different content losses on scale ×2.......70

Table 4.2: Quantitative comparison among different content losses on scale ×4.......70

Table 4.3: Quantitative comparison among different content losses on scale ×8.......71

Table 4.4: Quantitative benchmark test results on a scale ×2....................................76

Table 4.5: Quantitative benchmark test results on a scale ×4....................................77

Table 4.6: Quantitative benchmark test results on a scale of ×8. 77

Table 4.7: Comparison of execution time on the Set5 on a scale of ×4, network parameters, and training iterations of GAN-based super-resolution models...............83

Table 4.8: Quantitative benchmark GAN results on a scale ×2..................................85

Table 4.9: Quantitative benchmark GAN results on a scale of ×4. 86

Table 4.10:Quantitative benchmark GAN results on a scale of ×8. 86

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

LIST OF FIGURES

Page

Figure 1.1: The original image (a) and its down-sampled LR versions (c)-(d) with same height and width 1

Figure 2.1: The basic architecture of the SRCNN network. 8

Figure 2.2: The fundamental components of the super-resolution models. 9

Figure 2.3: The Pre-upsampling framework 11

Figure 2.4: The post-upsampling framework architecture. 12

Figure 2.5: The progressive up-sampling framework architecture. 13

Figure 2.6: The iterative up-and-down sampling framework. 14

Figure 2.7: The residual learning architecture. 15

Figure 2.8: The recursive learning architecture. 17

Figure 2.9: The global multi-path learning architecture. 19

Figure 2.10: The scale-specific multi-path learning architecture. 20

Figure 2.11: The dense connection learning architecture 21

Figure 2.12: The channel attention learning architecture. 22

Figure 2.13: The pyramid polling learning architecture. 24

Figure 2.14: The GAN architecture. 25

Figure 2.15: The CapsNet architecture. 38

Figure 3.1: The proposed architecture of our Caps-GAN model 40

Figure 3.2: The proposed architecture of our generator using a Progressive Multi Residual Dense block structure 43

Figure 3.3: Simplified residual block by removing batch normalization layer. 45

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Figure 3.4: The dense block and the simplified residual in residual dense 46

Figure 3.5: The pixel-shuffle module and transformation feature maps from the LR domain to the HR image on a scale of ×2. 47

Figure 3.6: (a) The Residual Projection, (b) Bottleneck Projection, (c) Depth-wise (DW) Bottleneck Projection. 48

Figure 3.7: Block diagram of Multiscale SSIM (MS-SSIM).....................................52

Figure 3.8: The architecture of Discriminator with Capsule layers in the Caps-GAN model..55

Figure 3.9: The GAN loss and perceptual loss of the Caps-GAN model. 58

Figure 3.10: The effect of content loss and adversarial loss to train our Caps-GAN model..62

Figure 4.1: Testing Datasets used to evaluate the performance of the model. 64

Figure 4.2: The assessment criteria of the Mean Option Score (MOS)......................65

Figure 4.3: Average training loss per epoch for training our model with different projection approaches on the DIV2K dataset. 67

Figure 4.4: Average PSNR (dB) per epoch for training our model with different projection approach on the DIV2K dataset. 67

Figure 4.5: Visual comparison of different content losses on the Monarch image of Set14 dataset on a scale of ×8 68

Figure 4.6: Visual comparison of different content loss on Img-092 image of the Urban100 dataset on a scale of ×4. 69

Figure 4.7: Visual comparison of the image "3096" on a scale of ×4 on the BSD100 dataset. 72

Figure 4.8: Visual comparison of the image "Gakuen Noise" on a scale of ×4 on the Manga109 dataset. 72

Figure 4.9: Visual comparison of the image "Img-012" on a scale of ×4 on the Urban100 dataset. 73

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 4.10: Visual comparison of the image "302008" on a scale of ×8 on the BSD100 dataset. 74

Figure 4.11: Visual comparison of the image "Karappo Highschool" on a scale of ×8 on the Manga109 dataset. 75

Figure 4.12: Visual comparison of the image "Img-004" on a scale of ×8 on the Urban100 dataset. 75

Figure 4.13: Performance and number of parameters evaluated on the Set5 dataset on a scale of ×4. 79

Figure 4.14: Performance and execution time evaluated on the Set5 dataset on a scale of ×4. 79

Figure 4.15: Discriminator loss during training model. 80

Figure 4.16: Visual comparison of our GAN model with capsule and CNN discriminator on a scale of ×8. 81

Figure 4.17: Visual comparison of our GAN model with capsule and CNN discriminator on a scale of ×4. 82

Figure 4.18: The GAN-based models' performances and parameters evaluated on the Set5 dataset on a scale of ×4. 84

Figure 4.19: The GAN-based models' performances and execution time evaluated on the Set5 dataset on a scale of ×4. 84

Figure 4.20: Visual comparison of the image " Comic " on a scale of ×4 on the Set14 dataset. 88

Figure 4.21: Visual comparison of the image "8023 " on a scale of ×4 on the BSD100 dataset. 89

Figure 4.22: Visual comparison of the image "img_077" on a scale of ×4 on the Urban100 dataset. 90

Figure 4.23: Visual comparison of the image "Aisazu NihaIrarenai " on a scale of ×4 on the Manga109 dataset. 91

Figure 4.24: Visual comparison of the image "0801 " on a scale of ×4 on the DIV2K dataset. 92

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 4.25: Visual comparison of the image "Lenna" on a scale of ×8 on the Set14 dataset. 95

Figure 4.26: Visual comparison of the image "189080 " on a scale of ×8 on the BSD100 dataset...96

Figure 4.27: Visual comparison of the image "img_037 " on a scale of ×8 on the Urban100 dataset. 97

Figure 4.28: Visual comparison of the image "BurariTessen" on a scale of ×8 on the Manga109 dataset. 98

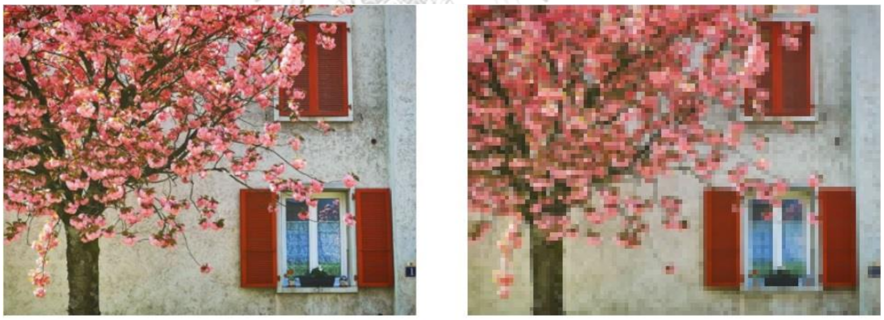
Figure 4.29: Visual comparison of the image "0869" on a scale of ×8 on the DIV2K dataset. 99

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

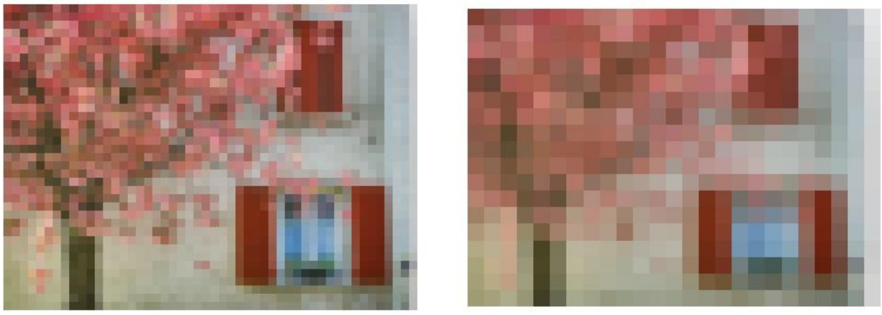
CHAPTER ONE
Introduction

1.1 Motivation and Problem Statement

Recent years have seen a growing interest in the use of images in many applications such as medical imaging [1, 2] traffic control [3], video applications [4], and satellite imaging [5]. One of the most essential attributes of an image that affects the visual quality is the resolution of an image. The presented image with higher resolution contains more detailed information. The detailed information is beneficial for further image processing tasks such as image segmentation, image recognition, and image analysis by either human eyes or machines. Figure 1.1 demonstrates an image of the same scene presented at different resolutions.



(a) Image size: 200 × 150 (b) Image size: 100 × 75



(c) Image size: 50 × 37 (d) Image size: 25 × 18

Figure 1.1: The original image (a) and its down-sampled LR versions (c)-(d) with same height and width

As shown in Figure 1.1, the highest resolution belongs to Figure 1.1 (a), and the image's resolution is gradually decreased by a factor of two, respectively. The higher-resolution image (200 ×150) provides more details, such as the texture of the wall's texture, flowers' texture, and even the curtain behind the windows. These important details gradually disappear with the reduction of image resolution, and even the identification of image contents with a resolution of 25×18 becomes too difficult as shown in Figure 1.1 (d).

In our society, low-resolution (LR) images are more often than not collected. It is because of deficiencies in imaging equipment or limitation of storage. Thus, methods for enhancing the resolution of images are in great demand. The process of retrieving or reconstructing the high-resolution (HR) image from one or more low-resolution (LR) images is known as super-resolution (SR).

There are two common approaches including hardware-based and software-based methods for obtaining HR images. The hardware-based approach is based on hardware improvement. Increasing the pixels in the complementary metal-oxide-semiconductor (CMOS) camera will result in, a higher resolution image. Thus, one technique to increase the spatial resolution is to enlarge the chip size so that more CMOS sensors can be included on the chip [6]. An alternative method for enlarging the chip size is to reduce the pixel size per unit area. That is to say; more pixels are implemented on a fixed-size chip [7]. Nevertheless, smaller pixel sizes might not result in a higher resolution due to the diffraction limit of the optics and the selection of maximum sampling frequency. It implies that both hardware-based solutions suffer the deficiency of cost and sophisticated engineering technology.

The other method, called the software-based approach, requires designing more accurate and faster algorithms to improve the resolution of LR images. Due to the highly developed computing units such as image signal processors (ISP) and graphic processing units (GPU), the software-based approach is a feasible solution. The computationally intensive tasks of the Deep Learning (DL) technique attempt to improve the visual quality of LR images. Deep Learning (DL) is a branch of machine learning methods based on artificial neural networks. The prominent superiority of DL over other machine learning algorithms leads to utilizing it in various fields such as computer vision, translation machines, natural language processing, and audio recognition. Because of DL capacity in extracting effective high-level abstractions between LR and HR images, it is used widely for SR purposes. The recent SR algorithms based on DL have achieved considerable improvements over the conventional image processing-based methods [8], [9], and [10].

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Recently proposed SR algorithms based on DL attempt to improve SR models' accuracy and operation speed by using faster and deeper CNNs. Despite their breakthroughs in designing faster and more accurate models, one important aspect of recovering the acceptable texture details in the reconstructed images in the SR algorithm remains challenging. This problem seems to be more crucial in larger-scale factors. Recent studies have widely focused on minimizing mean square error (MSE) value in the SR image reconstruction. It means that the SR results of these models show high peak signal-to-noise rates (PSNR) while the results lack high-frequency details. Hence, despite the high PSNR value, the resulting image is perceptually unsatisfying, especially in producing higher resolution. To solve this important obstacle, Mathieu et al. [11] and Denton et al. [12] suggested Generative Adversarial

Networks (GAN) [53] for reconstructing the SR images containing more detailed information similar to the original image known as the realistic SR image. Li and Wand [13] proposed an idea of learning procedure from one manifold to another in their GAN algorithm of style transfer. This approach attempts to minimize the squared error in the pre-trained feature space of VGG [59] or any scattering networks.

Although these GAN-based SR models show acceptable SR results on a scale of ×4, these models require more improvement in the perceptual quality as well as PSNR and SSIM. Since the CNNs are incapable of recognizing hierarchical feature information. Therefore, the CNN-based discriminator cannot train the generator module efficiently.

To solve this problem, utilizing Capsule network architecture (CapsNet) [83] in the discriminator module of the GAN model is a good solution for optimizing the training process of the GAN-based SR model. The current research aims to utilize the CapsNet in the discriminator module of the GAN network to improve the perceptual quality of the previous GAN-based models while improving the PSNR and SSIM.

CHULALONGKORN UNIVERSITY

1.2 Objectives

1. Investigate and propose fusing CNN architecture for single image super-resolution.
2. Develop a super-resolution framework in 3 different scale factors.
3. Evaluate the performance of the proposed algorithm compared with state-of-the-art super-resolution convolutional neural networks.

1.3 Scope of Research

a. Propose an algorithm for image super-resolution technique based on a reconstruction model.

b. The upscale factor is at least two times.

c. Evaluate the performance of the proposed algorithm with state-of-the-art models in terms of the objective and subjective scores.

d. The performance is measured by PSNR, SSIM, and MOS criteria.

1.4 Expected Output

a. Produce better quality images compared to previous SR models.

b. The algorithm can be operated in real-time scenarios.

1.5 Research Procedure

1. Review the literature on image super-resolution based on deep convolutional neural networks.
2. Review python programming language, Tensorflow, and Keras libraries.
3. Review and collect the image datasets for training purposes.
4. Python coding of my Caps-GAN model and training on the Graphics Processing Unit (GPU) at different scale factors.
5. Modify my Caps-GAN model's combination of various objective functions (pre-train model, adversarial loss, content loss, discriminator loss) and adjust them.
6. Final train our model on the Graphics Processing Units (GPU).
7. Calculate the complexity and validate the effectiveness of our Caps-GAN model.

8. Evaluate the performance of our Caps-GAN model with existing state-of-the-art methods, and take the MOS test of my Caps-GAN results.
9. Submit the ISI/Scopus indexed journal paper.
10. Final thesis defense.

1.6 Thesis Layout

The rest of the thesis is organized into four chapters.

Chapter Two: This chapter describes the fundamental components of the image super-resolution algorithm and reviews the literature on improvement techniques for each fundamental component.

Chapter Three: In this chapter, we will explain the proposed Caps-GAN framework methodology, including the generator module, discriminator module, and total loss of the proposed model.

Chapter Four: In this chapter, the training datasets, testing datasets, experimental results, and finally evaluation of the performance in terms of subjective and objective evaluations will be explained.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Chapter Five: In the final chapter, the conclusion part, contribution, and suggestions for future work will be discussed.

CHAPTER TWO
Background and Literature Review

2.1 Introduction

The image super-resolution (SR) has received noticeable attention from scholars and artificial intelligence (AI) companies in recent decades. The SR algorithm is utilized in numerous computer vision applications including object classification [14], object detection [15], image segmentation [16], medical imaging [1], satellite and aerial imaging [4], and surveillance [2].

The SR model aims to reconstruct a high-resolution (HR) image from a given low-resolution image while refining small details and maintaining the image's visual quality. Other names such as enlargement, scaling, zooming, and up-sampling refer to the image SR.

The single image super-resolution (SISR) is considered an ill-posed inverse problem in which multiple solutions exist for reconstructing the HR image from the LR. The SR models can be categorized into two main groups including traditional SR methods and deep learning SR models [17].

The Interpolation-based [18], Reconstruction-based [19], Self-example based [20], and multi-image SR based [5] are considered as the traditional SR model. Since these classical SR models suffer over-smoothing results, pixelization degradation, jagged contour artifacts, and time-consuming problems, the traditional methods are not considered effective.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Deep learning is a branch of machine learning, and the SR models based on this concept in recent years have proven impressive performance compared to the conventional SR algorithms. The deep learning-based model reconstructs the image by learning the LR-to-HR mapping in a supervised manner. Specifically, any learning-based SR algorithm's convolution neural network (CNN) is trained to learn the feature mapping from the LR patch to the corresponding HR patch. Dong et al. [21] represented the first deep learning-based SR model based on the deep learning concept, as shown in Figure 2.1.

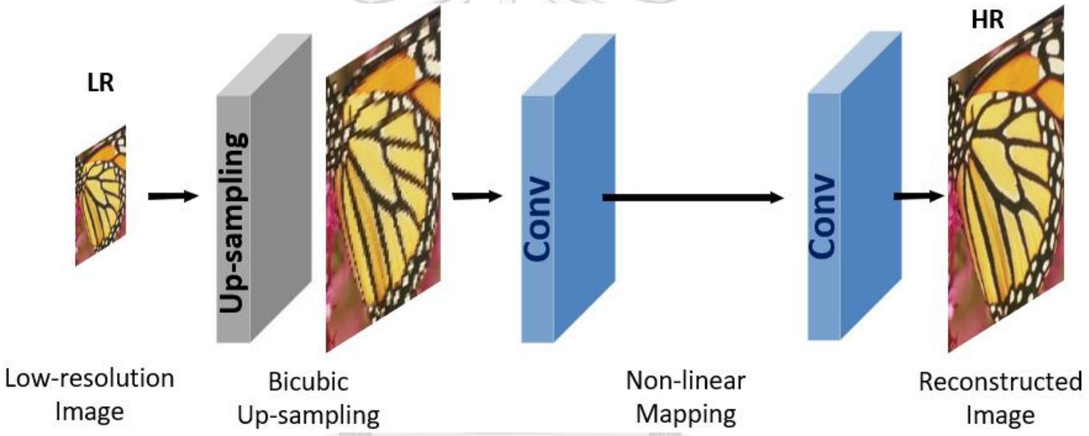


Figure 2.1: The basic architecture of the SRCNN network.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

The Super-resolution Convolutional Neural Network (SRCNN) [20] architecture is a shallow three-layer network that learns the mapping from the interpolated LR patch to the HR patch based on an end-to-end nonlinear algorithm.

In this SR model, the LR patch is upsampled (pre-up-sampling framework) by the bicubic interpolation and extracting the features of the interpolated patch. The second layer is utilized for operating the nonlinear mapping and the final layer reconstructs the HR image. The Mean Square Error loss function (MSE-loss) is employed as the training objective function of SRCNN [20]. The reconstructed image at the output is an image at high-resolution (HR) known as a super-resolution (SR) image.

Although this shallow single path SR algorithm with the pre-up-sampling framework and without any skip connections shows superior visual quality and peak signal-to-noise ratio (PSNR) compared to the conventional SR model, the SRCNN [20] result suffers lacking details and over smoothing degradation.

Followed by the SRCNN [20] model, several deep learning-based SR algorithms have been proposed to improve the SR model capability to represent better results. A variety of SR models is made by combining and improving a set of components. Figure 2.2 demonstrates the fundamental components of the deep learning SR model.

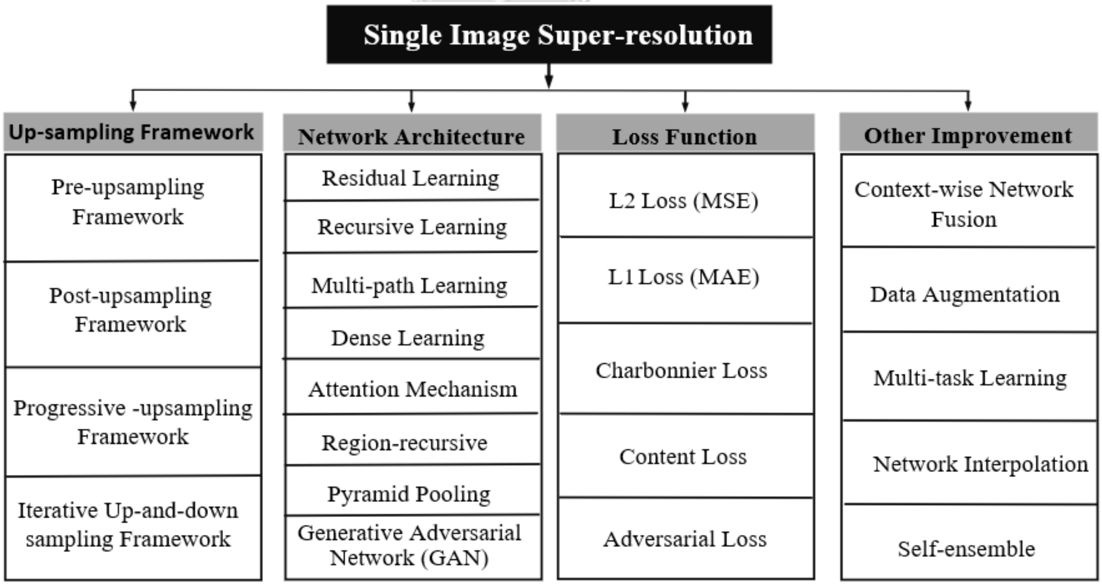


Figure 2.2: The fundamental components of the super-resolution models.

As shown in Fig 2.2, the Up-sampling Framework, Network Architecture, and the Loss Function of the SR model are considered the fundamental components of any deep learning SR model. In addition to these fundamental components, some other improvements such as Context-wise Network Fusion, Data Augmentation, Multi-task Learning, Network Interpolation, and Self-ensemble methods are combined with some SR models to improve the efficiency.

In this section, the literature on these fundamental components in the existing SR models is reviewed and the advantages and disadvantages of each one are summarized. Next, the literature on the Capsule Network is reviewed, and the advantages compared to the Convolutional neural network (CNN) concept are discussed.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2 Super-resolution Frameworks

The basic concept of image super-resolution is performing the up-sampling operation on an ill-posed image. The ill-posed problem creates a critical challenge in generating acceptable HR output from the LR input. Although the existing SR models based on their architectures vary widely, based on their up-sampling models, they attributed to four frameworks including Pre-upsampling, Post-upsampling, Progressive up-sampling, and Iterative up-and-down sampling Framework.

2.2.1 Pre-upsampling Super-resolution

Direct learning of mapping from the low to high-dimensional space is considered a difficult task. Therefore, the best and most straightforward solution is to apply conventional up-sampling algorithms to the low-dimensional image in the first stage, and later enhance it with a deep neural networks model. Figure 2.3 demonstrates the architecture of the pre-upsampling method.

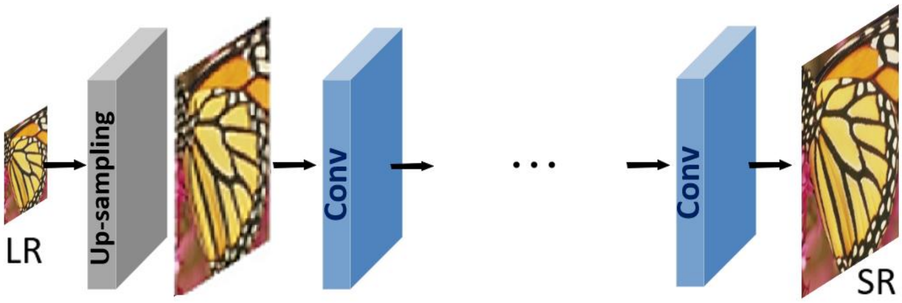


Figure 2.3: The Pre-upsampling framework

Based on this idea, Dong et al. [16], [20] propose the SRCNN algorithm by implementing the pre-upsampling framework to learn an end-to-end mapping from interpolated LR image to the HR. The coarse HR image is particularly obtained from the LR image with traditional methods such as bicubic interpolation. Then the deep convolutional neural network (CNN) is used to reconstruct the high-quality details of the HR image. Although the pre-upsampling approach has become one of the most common frameworks [22], [23], [24], [25] in the SR field, it has high computational cost due to convolution operations on the high dimensional features. Moreover, this up-sampling framework leads to degradation effects such as blurring and noise amplification in the resulting image.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.2.2 Post-upsampling Super-resolution

To improve the computational cost and create an efficient deep learning algorithm for increasing the image resolution, scholars attempted to perform most of the computation in low-dimensional space as shown in Figure 2.4. Specifically, they replaced the predefined up-sampling module with end-to-end learnable CNN layers in a low-dimensional space. Then the up-sampling operation is applied at the last stage of the model structure. This idea is known as the post-upsampling strategy in the SR field.

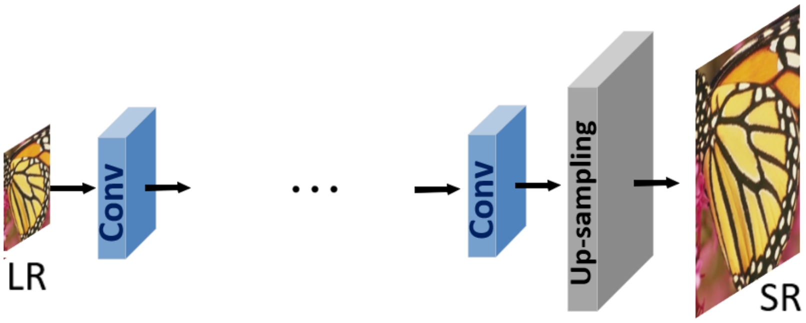


Figure 2.4: The post-upsampling framework architecture.

As shown in Figure.2.4, the LR image is fed into deep CNNs without increasing size, and the end-to-end learnable up-sampling layers are utilized at the end of the network structure. This approach aims to reduce the model's computation cost and spatial complexity by performing the feature extraction process in a low-dimensional space and increasing the resolution only at the end of the structure. Thus, the post-upsampling framework is considered the most mainstream framework in the SR [8], [26], [27], [28], [29]. The new SR models based on the post-upsampling framework vary by applying different deep learning network architectures, and the learning objective functions.

CHULALONGKORN UNIVERSITY

2.2.3 Progressive Up-sampling Framework

Even though the post-upsampling SR framework has reduced the computational cost, this up-sampling framework has some limitations. Since the up-sampling is applied in only one step, it increases the learning difficulty, especially in large scaling factors. That is to say, the quality of the SR images at large scale factors is not desirable.

Moreover, this framework is not desirable if we require training an individual SR model for multiscale purposes. To tackle these two essential drawbacks, a progressive up-sampling framework was introduced by the Laplacian pyramid SR network (LapSRN) [30], as shown in Figure 2.5.

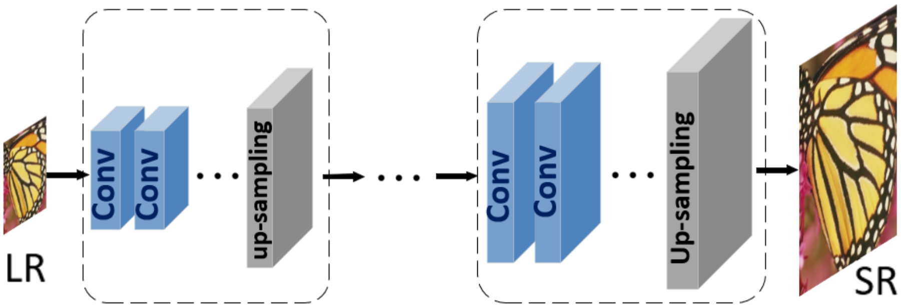


Figure 2.5: The progressive up-sampling framework architecture.

The architecture under this framework is based on a cascade of CNNs and progressively produces higher-resolution images. In contrast to the post-up-sampling framework with very deep architecture that applies numerous convolution layers in a low-dimensional space and then suddenly upsamples at the end of the model, the post-upsampling framework applies fewer convolution layers in a low-dimensional space. It progressively applies the predefined up-sampling modules in several stages as demonstrated in Figure 2.5. Other studies such as MS-LapSRN [31] and ProSR [32] also utilized a progressive framework model and achieved high performances.

This framework has more significant benefits, such as reducing the learning difficulties and achieving better results on larger scales. However, this model has some drawbacks, such as the training stability and multiple-stage design complexity, which force us to design more advanced training strategies. It is noticeable that integrating some specific learning strategies such as residual connections [33] or residual projection [34] to the different stages of the progressive up-sampling framework can improve the training performance of the SR model. Since the progressive up-sampling approach can reconstruct the minor details of the image, it has considerable potential for further research.

2.2.4 Iterative Up-and-down Sampling Super-resolution

The idea behind the iterative up-and-down sampling is to improve the capture of the mutual dependencies of LR and HR images. The most successful iterative procedure, back-projection [32], was incorporated into the SR algorithm [33]. This SR framework as shown in Figure 2.6, attempts to iteratively perform back-projection modification such as computing the reconstruction error and then fusing it back to tune the intensity of the HR image.

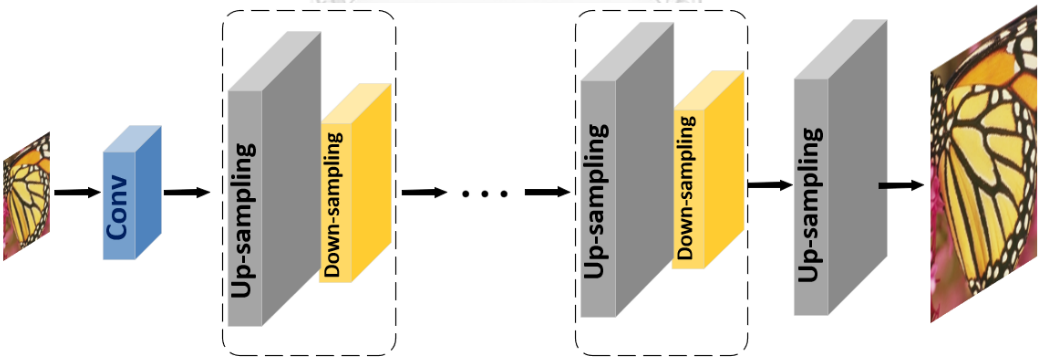


Figure 2.6: The iterative up-and-down sampling framework.

Following this idea, Haris et al. [33] demonstrated some iterative up-and-down sampling layers and proposed a Deep Back Propagation Network (DBPN) algorithm. DBPN [33] connects up-and-down sampling layers alternately and reconstructs the final HR image by employing intermediate reconstructions. Correspondingly, the

Image Super-resolution Feedback Network (SRFBN) [34] utilized dense skip connections in the iterative up-and-down sampling structure and yielded better representations; since this approach applies up-sampling and down-sampling iteratively, the computational cost and the network complexity increase dramatically.

2.3 Network Architecture

One of the most significant parts of any deep learning model is the network architecture of CNN. Researchers attempt to apply a variety of architectures such as Residual, Recursive, Multi-path, Dense Connection, and Group convolution in their SR networks. In this section, the different architectures of SR networks are reviewed, and their advantages and disadvantages are discussed.

2.3.1 Residual Learning

In 2016, He et al. [35] introduced the ResNet in image convolution recognition algorithm but before them, the residual learning strategy had been applied in the SR models [36], [37], and [28]. Figure 2.7 demonstrates the basic architecture of the residual connection. The residual strategy in the SR field can be categorized into two branches namely, global and local residual.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

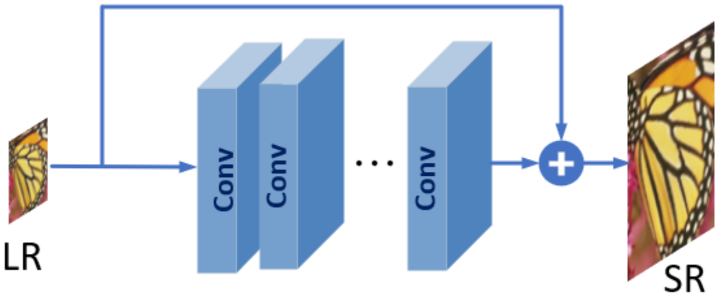


Figure 2.7: The residual learning architecture.

Global Residual Learning: Due to the image-to-image translation concept in the SR model, the input image and the target image are highly correlated. Hence, the global residual only needs a residual map to restore the missed high-frequency details. This idea leads to reducing the complexity and learning difficulty in the SR model [22], [24], [30], [31].

Local Residual Learning: This approach is inspired by the residual concept of ResNet [35]. And it is used to solve the degradation problem caused by increasing the depth of the CNN network. Therefore, this type of network architecture improves the learning ability and reduces the training difficulties [28], [36], [37], [29], [38].

The implementation of both local and global residual strategies by shortcut connections and element-wise addition in network structure is shown in Figure 2.7. In Global Residual, the shortcut connections are directly from input to output image, while Local Residual, uses multiple shortcut connections between network layers in different depths. In addition, there are some variants of the residual block, including residual projection [35], bottleneck residual projection [33], and depth-wise bottleneck projection [39], that are used in different CNN architectures.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.3.2 Recursive Learning

The idea of recursive learning was proposed for getting the higher-level features in the training stage of the SR algorithm. As shown in Figure 2.8, the same module is recursively applied several times.

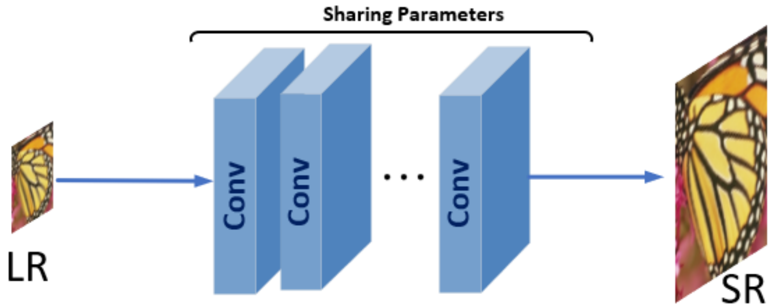


Figure 2.8: The recursive learning architecture.

For example, the 16-recursive in Deeply Recursive Convolution Network [24] (DRCN) algorithm used a single convolutional layer as its recursive unit, and it successfully gained a receptive field of 41 × 41 without over parameters. This receptive field of 41 × 41 is considerably larger than the 13×13 field of another SR algorithm known as SRCNN [21]. The Deeply Recursive Residual Network [23] (DRRN) algorithm applied the ResBlock [35] as the recursive unit for 25 recursions. It is noticeable that the DRRN architecture performed better than the 17-ResBlock baseline.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Lately, Li et al. [34] utilized an iterative up-and-down sampling framework and suggested a feedback network based on recursive learning. It means the feedback network shares the weights of the whole network across all recursions. Some recursive structures utilize recursive modules in different parts of the SR network. For instance, Han et al. [28] used Dual-state Recurrent Network (DSRN) to signal exchange between the LR and HR samples. The idea of exchanging signals at each recursion leads to a better exploration relationship between LR and HR images.

Correspondingly, Lai et al. [30] used the combination of both embedding and up-sampling modules as a recursive unit. This approach can reduce the model size and improve the loss performance. Although the recursive approach learns more advanced features of the image at the same range of network parameters, its computational cost is still high. Hence the high computation cost leads to a vanishing gradient. To resist gradient problems, scholars often integrate recursive learning with multi-supervision or residual learning approaches [22], [23], [24], [24].

2.3.3 Multi-path Learning

The aim of proposing multi-path learning is to provide a better modeling capability bypassing the features through multiple paths with different operations and later fusion these features. In general, this multi-path approach can be categorized into three branches that are: global, local, and scale-specific multi-path learning.

Global Multi-path Learning: In this model, multiple paths are responsible for extracting features of different aspects of input. Figure 2.9 demonstrates the basic architecture of global multi-pass. Since these paths cross each other in the propagation, it significantly enhances the SR algorithm's learning ability. LapSRN [30] proposed two paths, which are feature extraction and reconstruction path. The former path predicts the sub-band residuals in a coarse-to-fine fashion and the latter path reconstructs the SR image based on the signals from both paths. Inspired by LapSRN [30] idea, the Enhanced Deep Residual Network [26] (EDSR) uses two paths to extract features in high-dimensional and low-dimensional space, and it exchanges information to improve the learning ability of the SR network respectively and continuously. Following this idea, Ren et al. [40] applied multiple unbalanced paths'

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

structures for the up-sampling of images, and concatenated them at the end of this model.

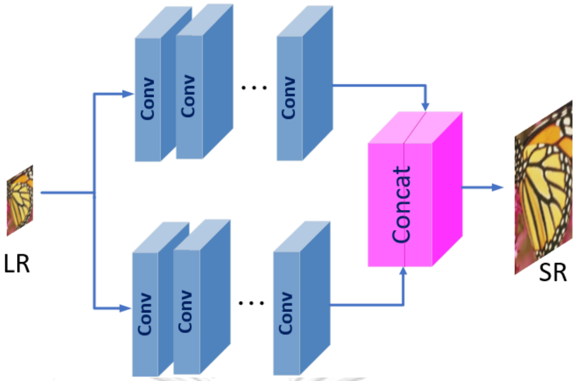


Figure 2.9: The global multi-path learning architecture.

Local Multi-path Learning: Inspired by the inception model [41], the Multiscale Residual Network [36] (MSRN) designed a new block for multiscale feature extraction (see Figure 2.9). In this structure, two convolution layers with kernel size 3 × 3 and 5 × 5 are to extract features simultaneously, and later concatenate both features and send them through the same operations again. This local multi-path model shows a better feature extraction performance and finally improves the quality of the reconstructed HR image.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Scale-specific Multi-path Learning: Lim et al. [26] introduced a scale-specific multi-path network structure to reconstruct multiscale SR results with a single network. The model first shares the principal components of intermediate layers and upsamples them by different scale factors at the end of the structure as shown in Figure 2.10.

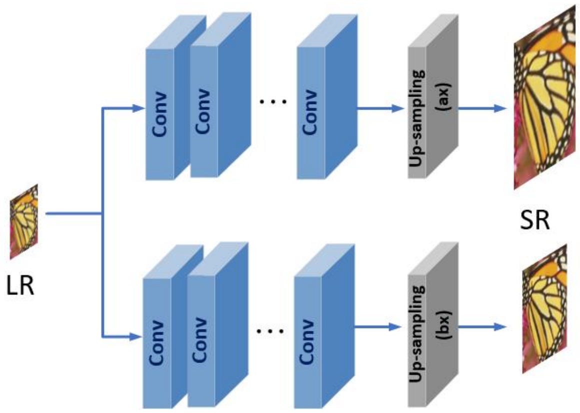


Figure 2.10: The scale-specific multi-path learning architecture.

Chulalongkorn University

In this model, only the desired selected scale of the network is trained. The weights for other scales' updates are disabled. According to this idea, MDSR [26] decreases the model size by sharing the network parameters for different scales and shows a better performance than the single-scale model. The equivalent multi-path specific scale learning is also adopted by CARN [42] and ProSR [31].

2.3.4 Dense Connections

Utilizing the Dense connection concept in the SR field inspired by Huang et al. DenseNet [43]. According to this approach, for each layer of the dense blocks, the feature maps of all previous layers are utilized as the input of the current layer as illustrated in Figure 2.11. The Dense connection architecture dramatically increases the performance of our deep learning network by squeezing channels after concatenating all feature maps. This architecture aims to reduce the model size and preserve our network against gradient vanishing. Because of the successful fusing of low-level and high-level features, dense connections are more popular in the SR tasks.

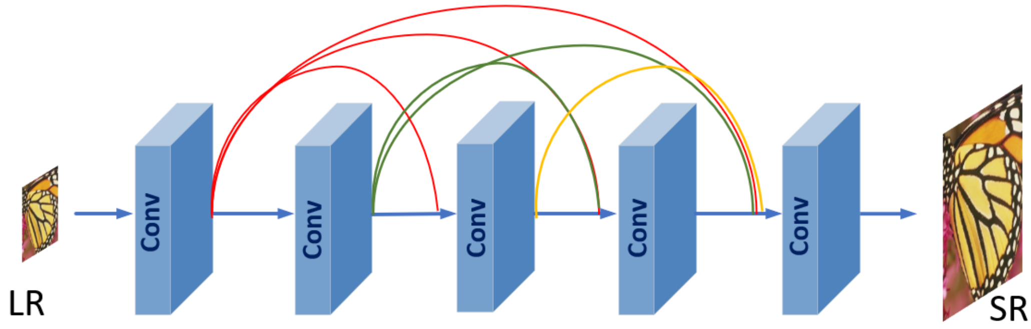


Figure 2.11: The dense connection learning architecture

Following this approach, Tong et al. [27] designed a 69-layers SRDenseNet and inserted dense connections between different dense blocks (block-level dense connections). That is to say, for every dense block, the feature maps of all previous blocks are utilized as inputs, and also the output feature maps of each layer are used as inputs into all subsequent blocks. The idea of block-level dense connections structure was also used by MemNet [22], CARN [42], RDN [44] and ESRGAN [29].

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.3.5 Attention Mechanism

The attention mechanism approach can be categorized into two models Channel Attention and Non-local Attention.

Channel Attention: To enhance learning ability by explicitly modeling channel interdependence, Hu et al. [45] introduced a “squeeze-and-excitation” block as shown in Figure 2.12. This idea was inspired by paying attention to the interdependence and interaction of the feature representations between different channels. In this architecture, every input channel was squeezed into a channel descriptor with the help of global average pooling (GAP). Then these descriptors are fed into two dense layers to generate channel-wise scaling factors for input channels.

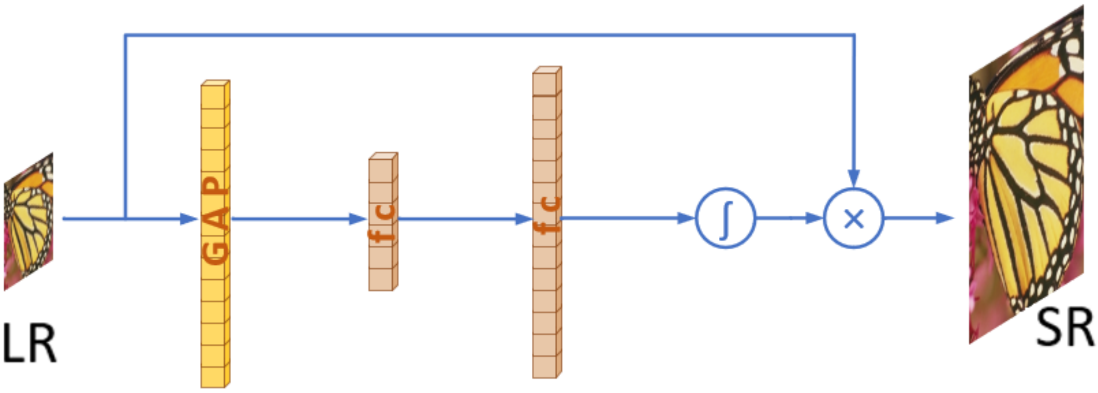


Figure 2.12: The channel attention learning architecture.

Zhang et al. [38] integrated the channel attention technique with the SR based on this concept. They introduced Residual Channel Attention Networks [38] (RCAN) which significantly increases the representation ability of the model and the SR performance. Dai et al. [46] further designed a second-order channel attention (SOCA) structure to improve the learning of feature correlations. The channel-wise features are rescaled by applying second-order feature statistics rather than global average pooling.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Non-local Attention: Most existing SR architectures do not use local receptive fields. Nevertheless, some distant textures or objects may be crucial for local patch generation. Zhang et al. [38] combined local and non-local attention blocks to extract features that capture the long-range dependencies between pixels. The non-local branch assists the embedded Gaussian function in estimateing pairwise relationships between every two positions in the feature maps to anticipate the scaling weights. The local branch utilizes an encoder-decoder module to learn the local attention. Therefore, this structure can capture spatial attention properly and enhance representation simultaneously. Based on this idea, Dai et al. [46] combined the local and non-local attention mechanisms to capture long-distance spatial contextual information.

2.3.6 Region-recursive Learning

In the field of SR, most learning procedures are considered pixel-independent tasks and consequently cannot accurately maintain the interdependence between generated pixels. Motivated by PixelCNN [47], Dahl et al. [48] utilized two convolutional networks. In this structure, the first network is used to capture global contextual information, and another network is used for serial generation dependence. This recursive pixel concept executes pixel-by-pixel generation using these two networks.

Inspired by the human attention shifting algorithm [49], the Attention-FH [50] implements this strategy by resorting to a recurrent policy network for sequentially discovering attended patches and performing local enhancement in the reconstructed SR images. In this manner, the network can adaptively personalize an optimal searching path for each image according to its characteristics, and completely exploits the global intra-pixel dependence of images. Although these approaches demonstrate better performance to some extent, the recursive process needs a propagation path that increases the computational cost and challenges the training.

2.3.7 Pyramid Pooling

Zhao et al. [51] utilized the pyramid pooling architecture in their SR model which was inspired by the spatial pyramid pooling layer [52]. The pyramid pooling model employs global and local contextual information about an image. Figure 2.13 shows the architecture of the pyramid polling approach.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

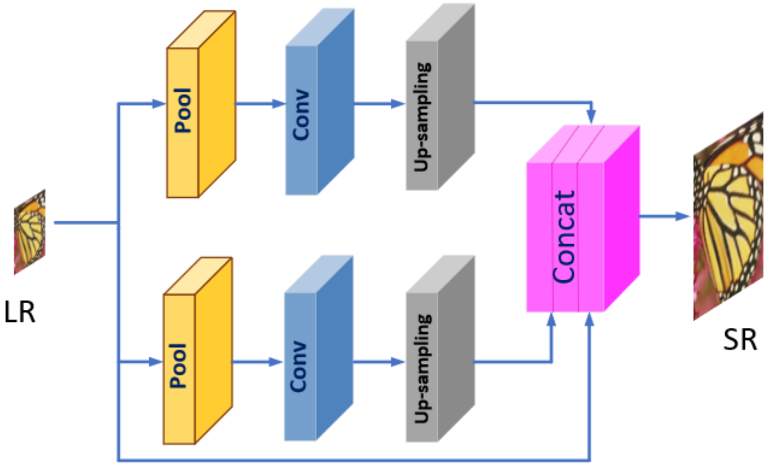


Figure 2.13: The pyramid polling learning architecture.

In this model, firstly each feature map is divided into $M \times M$ bins, then it goes through global average pooling and yield resulting in $M \times M \times C$ outputs. After that, a 1×1 convolution is applied on $M \times M \times C$ to compress the outputs into a single channel. Consequently, the bilinear interpolation up-sampling is applied on the single-channel LR feature map. In other words, utilizing different M helps integrate global and local contextual information effectively [53].

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

2.3.8 Generative Adversarial Network (GAN)

Generative Adversarial Network (GAN) is a branch of the machine learning framework. The GAN was inspired by the generative model presented by Goodfellow et al. 2014 [54]. Conceptually, the GAN structure operates as a supervised learning approach to do unsupervised learning by generating fake or synthetic data.

Due to the successful performance of the GAN structure, it has been presented in many applications in computer vision and graphics communities, such as image-to-image translation, image blending, face aging, object detection, 3D image synthesis, image super-resolution, human pose synthesis, image manipulation applications, language and speech synthesis, and video applications.

The GAN structure can be considered the training of two independent neural networks known as generator and discriminator simultaneously. Figure 2.14 demonstrates the structure of the GAN. *G* indicates the generator network, and the discriminator network is demonstrated by *D*. The generator is trained to generate more realistic and accurate data, In contrast, the discriminator is trained to distinguish between real or fake sample data generated by the generator network. In simple words, these networks play an adversarial game. The generator network aims to trick the discriminator while the discriminator attempts to prevent it. Both networks are skilled in this adversarial game by trainning the generator and discriminator networks. The generator produces more accurate data, and the discriminator improves distinguishing between fake and real generated samples.

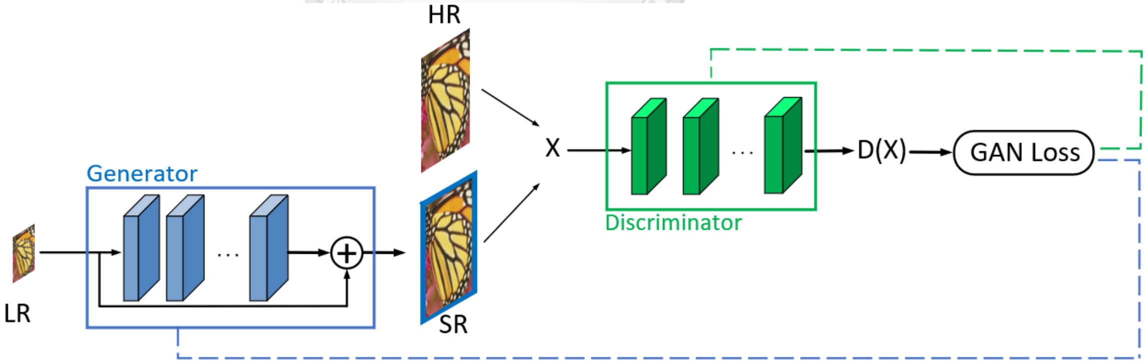


Figure 2.14: The GAN architecture.

From a mathematics perspective, the generator takes the input z from the probability distribution $p(z)$, then generates the fake data and feeds it into a discriminator network $D(x)$. The discriminator network takes two inputs from $G(z)$ and the real distribution data $p_{data}(x)$. This discriminator tries to solve a binary classification problem using the sigmoid function giving output in the range 0 to 1. In other words, these two networks D and G, play the following two-player min-max game with the value function $V(G,D)$ as formalized in Eq 2.1 and Eq 2.2.

$$\underset{G}{min}\ \underset{D}{max}\ V(D,G)\,, \tag{2.1}$$

$$V(D,G) = E_{x\sim P_{data(x)}}[\log D(x)] + E_{z\sim p_{z(z)}}\left[\log\left(1 - D\big(G(z)\big)\right)\right], \tag{2.2}$$

Where the first term is the entropy of real distribution data which passes through D and tries to maximize the first term to 1, the second term is the entropy of random noise data that passes through the generator to generate the fake sample. This fake sample then passes through the discriminator to maximize the second term to 0. There are two main procedures to train GAN. The first one is to freeze the generator by only training the discriminator with the following Eq 2.3 to update the discriminator.

$$\nabla_{\theta 1}= \frac{1}{m}\sum_{i=1}^{m} \log D(x^{i}) + \log(1 - D\left(G\big(z^{(i)}\big)\right))] \tag{2.3}$$

The second procedure is to freeze the discriminator and train only the generator with the following Eq 2.4 to update the generator.

$$\nabla_{\theta 2}= \frac{1}{m}\sum_{i=1}^{m} \log(1 - D\left(G\big(z^{(i)}\big)\right)) \tag{2.4}$$

Both the gradient-based updates use the standard gradient-based learning rule with the momentum parameters. The target of GAN is to generate fake data similar to real data.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Ledig et al. [9] proposed a single super-resolution model of SRGAN to reconstruct the SR image close to the manifold of natural images by utilizing a GAN model. SRGAN [9] model involves a three-loss formulation including L2 loss to encode pixel-wise similarity (MSE pre-trained), a perceptual loss to define high-level texture representation, and an adversarial loss (standard objective of GAN [54]) to maintain the min-max balances between generator and discriminator of this model. Based on the image reconstruction concept in GAN architecture, SRGAN [9] proposed favor results that are perceptually similar to the HR image. The limitation of the PSNR evaluation problem in this SR model leads to introducing a Mean Opinion Score (MOS) test to quantify the results. In the MOS test, people visually compare (bad/excellent quality) the reconstructed image quality.

An effective generator module with an appropriate loss is one of the most important factors in producing perceptually sharp and pleasing image textures. The SRGAN [9] model utilized EnhanceNet [55] concept in its architecture to have such an effective generator architecture. Sajjadi et al. [55] introduced the EnhanceNet model to focus on producing sharp textures. The non-compliance, with the perceptual quality of the reconstructed image due to utilizing a pixel-wise model, leads to producing the overly smoothed image that cannot reconstruct the sharp textures. This architecture is aimed to solve the overly smoothing problem by producing faithful texture details in the reconstructed SR images [55].

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Besides the regular pixel-level L2 (MSE) loss, the EnhanceNet [55] utilized two loss terms (combination of perceptual loss and texture matching loss). (a) The perceptual loss is defined by the intermediate features of a pre-trained network [56]. (b) The texture matching loss is defined to match the texture of the LR and HR images. Finally, the whole architecture is adversarially trained [54]. The generator architecture of the EnhanceNet is based on the Fully Convolutional Network [57] while utilizing the residual learning concept [58]. Although the EnhanceNet successfully increases PSNR and produces more realistic results and perceptually better SR images, it shows unpleasant artifacts in the highly textured regions of the image.

To solve the problem of unpleasant artifacts in the highly textured regions of the EnhanceNet, two GAN models of Single Image Super-resolution with Feature Discrimination [59] (SRFeat) and Cycle-in-cycle GAN (CinCGAN) [8] were proposed. The models mentioned above, utilize additional discriminator and generator networks and do not improve the objective functions of their models. The SRFeat [59] improves the realistic perception by utilizing an additional discriminator that forces the generator to produce high-frequency details.

CinCGAN [8] uses three generators and two discriminators. The SR image is generated in three steps. First, the LR image is denoised by a generator and a discriminator. Then the denoised LR image is used as the input for the next cycle of the model. Moreover, the intermediate image is upsampled by a pre-trained generator. Finally, the generator and discriminator are fine-tuned end-to-end and produce the SR image.

Although the SRFeat [59] and CinCGAN [8] were able to enhance the realistic SR image and somehow solved the problem of unpleasant artifacts in the highly textured regions, the complexity of these GAN models led to unstable training, increasing the network parameters, and are not considered efficient SR models.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Based on the SRGAN [9] architecture, Wang et al. [29] proposed Enhanced Super-resolution Generative Adversarial Networks (ESRGAN) to produce more realistic SR results compared to the SRGAN [9]. This model's discriminator architecture remains unchanged while the dense block architecture and long (global) residual connections are incorporated into the generator architecture. The ESRGAN [29] model also includes a three-loss formulation that is the a) L1 loss to encode pixel-wise similarity (MAE pre-trained) to utilize the VGG [60] model for content loss, b) a perceptual loss to define high-level texture representation, and c) an adversarial loss (standard objective of GAN [54]) to preserve the min-max balances between generator and discriminator of ESRGAN model. The visual result of this model (MOS test) not only is better than the previous GAN-based models but also shows improvement compared to non-GAN models such as the RCAN [38]. In terms of the quantitative measures (PSNR), the results of ESRGAN lag in comparison to the non-GAN models.

2.4 Loss Function

The loss function is used to evaluate reconstruction error and steer the model optimization during the training procedure of the SR field and has played a vital role in the learning phase of the deep learning model. The loss functions in the deep learning SR can be categorized into pixel-wise loss, content loss, and adversarial loss (GAN architecture).

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

The pixel-wise loss function measures the pixel-wise difference between the ground truth (GT) image and the reconstructed (SR) image. The L2 loss and L1 loss, known as the mean square error (MSE) and mean absolute error (MAE) belong to the pixel-wise loss function model. This section reviewed the literature on various loss functions in the deep learning SR models.

2.4.1 L2 Loss (MSE)

As pioneer research in the deep learning SR model, Dong et al. [17] utilized the L2 loss function in the SRCNN model. Attributable to the high correlation between pixel-wise loss and PSNR definition, the L2 loss functions (MSE) become the most common loss functions in SR models such as SRCNN [21], DRCN [24], FSRCNN [17], DRRN [23] SRResNet [9], MemNet [22], SRDenseNet [27], DBPN [33] and DSRN [28], given by:

$$L_{pixel-L2}\left(\hat{I},I\right)=\frac{1}{hwc}\sum_{i,j,k}\left(\hat{I}_{i,j,k}-I_{i,j,k}\right)^{2} \tag{2.5}$$

Where w, h and c are the width, height, and the number of channels of the assessed images, respectively. $\hat{I}$ and I denote the reconstructed image and GT image, respectively. The pixel loss constrains $\hat{I}$ to be close enough to I in terms of the pixel value.

According to the L2 loss equation, it penalizes a large error value that improves the model's capability to preserve the image's sharp edges while demonstrating more tolerance to its minor errors. Although the L2 loss shows robustness in reconstructing the edges details of the image, it suffers the independent Gaussian noise in the smooth regions. To solve this serious limitation, the L1 loss was introduced in the SR model.

2.4.2 L1 Loss (MAE)

The L1 loss is considered a pixel-wise loss type. The SR models such as EDSR [26], RDN [44], CARN [42], MSRN [36], RCAN [38], RNAN [61], SAN [46], and SRFBN [34], utilized the L1 loss (MAE) as objective function in the deep learning SR models. The L1 loss (MAE) is given by:

$$L_{pixel-L1}\left(\hat{I}, I\right) = \frac{1}{hwc}\sum_{i,j,k}\left|\hat{I}_{i,j,k} - I_{i,j,k}\right| \tag{2.6}$$

Where w, h and c are the width, height, and the number of channels of the assessed images, respectively. $\hat{I}$ and I represent the reconstructed image and GT image, respectively. The pixel loss constrains $\hat{I}$ to be close enough to I in terms of the pixel value.

In contrast to the MSE loss function, the MAE does not over-penalize the error. Therefore, it provides less independent noise and produces smoother results compared to MSE loss. Although the L1 objective function makes smoother results, it has a slower convergence speed in the training mode of the SR model. Ahn et al. [42] suggested utilizing the residual connection in the SR architecture to solve the slower convergence speed problem.

2.4.3 Charbonnier loss

Although the L1 loss objective function in the deep learning SR models outperforms visually pleasing results compared to the L2 loss, its output result is not optimum. To improve the L1 loss function, Lai et al. [30] utilized a variant of the L1 loss function known as the Charbonnier loss function, given by:

$$L_{pixel-Cha}\left(\hat{I}, I\right) = \frac{1}{hwc}\sum_{i,j,k}\sqrt{(\hat{I}_{i,j,k} - I_{i,j,k})^2 + \epsilon^2} \tag{2.7}$$

Where ϵ is a constant and set to 10^{-3} for numerical stability. $\hat{I}$ and I represent the reconstructed image and GT image, respectively. The pixel loss constrains $\hat{I}$ to be close enough to I in terms of the pixel value.

Because the definition of the pixel-wise loss function is highly correlated with PSNR calculation, thus minimizing pixel-wise loss directly maximizes the PSNR of the result. This capability leads to using the pixel-wise loss as the widest loss function in the SR area for getting a higher PSNR value while producing a natural and pleasing SR image that contains high-frequency details remains a challenge.

2.4.4 Content Loss

To solve the limitation of the pixel-loss function to evaluate the perceptual quality of images, the content loss model was introduced [56], [62].

The content loss measures the semantic differences between images using a pre-trained image network. This network demonstrates as φ and the extracted high-level representations on the l-th layer as $\varphi^{(l)}$. The content loss is calculated as the Euclidean distance between high-level representations of both the GT image and the reconstructed image, as follows:

$$l_{content}\left(\hat{I}, I; \varphi, l\right) = \frac{1}{h_1 \ w_1 \ c_1}\sqrt{\sum_{j,j,k}\left(\varphi^{(l)}_{i,j,k}(\hat{I}) - \varphi^{l}_{i,j,k}(I)\right)^2} \tag{2.8}$$

Where h_1, w_1 and c_1 are the height, width, and the number of channels of the representations on layer l, respectively. Fundamentally the content loss transfers the learned knowledge of hierarchical image features from the classification network φ to the SR network. The content loss encourages the $\hat{I}$ to be perceptually similar to the target image I , rather than match images pixel-wise. Therefore, the content loss

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

produces visually more perceptible results and is also widely utilized in the SR field [32], [34], [29], [55], [56], [63], [64], [65], where the VGG [60] and ResNet [35] are the most frequently used pre-trained CNNs.

2.4.5 Adversarial Loss

Due to attempts to improve the learning ability of the neural networks in recent years, the GAN [54] model attracted more attention. The GAN structure consists of two main parts, a generator and a discriminator, as discussed in Figure 2.14 in the GAN Network Architecture section. The generator is responsible for generating the image. The discriminator takes the generated result and the GT image to distinguish them as real sample or fake samples. During the training process of the GAN model, two strategies for operating the generator and the discriminator are alternately performed. The first is to fix the generator part and train the discriminator to discriminate better. The second is to fix the discriminator and train the generator to fool the discriminator. It means these two strategies play an adversarial game together. Through adequate iterative training, the generator can produce a result consistent with real data distribution. At the same time, the discriminator can distinguish between the real and generated data.

In the SR field, the GAN model implements adversarial learning with a generator as the SR model and the discriminator to judge whether the quality of the generated image is similar to the input image. Inspired by the GANs concept, Ledig et al. [8] proposed the first SRGAN model by using adversarial loss based on cross-entropy, as follows:

$$L_{gan-ce-g}\left(\hat{I}; D\right) = -\log D\left(\hat{I}\right) \tag{2.9}$$

$$L_{gan-ce-d}\left(\hat{I}, I_s; D\right) = -\log D(I_s) - \log\left(1 - D\left(\hat{I}\right)\right) \tag{2.10}$$

Where $L_{gan-ce-d}$ and $L_{gan-ce-g}$ represent the adversarial loss of the discriminator and the generator respectively. $\hat{I}$ and I_s denote the generated image and the GT image form dataset respectively.

Following this concept, Wang et al. [32] and Yuan et al. [8] utilized adversarial loss based on least square error for higher quality results and more stable training [66], given by:

$$L_{gan-ls-d}\left(\hat{I}; D\right) = \left(D\left(\hat{I}\right) - 1\right)^2 \tag{2.11}$$

$$L_{gan-ls-d}\left(\hat{I}, I_s; D\right) = \left(D\left(\hat{I}\right)\right)^2 + \left(D(I_s) - 1\right)^2 \tag{2.12}$$

Xu *et al.* [66] integrate a multi-class GAN consisting of a generator and multiple class-specific discriminators.

The ESRGAN [29] used relativistic GAN [67] to recover more detailed textures. In this algorithm rather than the probability to predict the real or fake images, it uses the probability that determines the real image is relatively more realistic than others. Although the GANs are considered difficult and unstable in the training process [68], [69], [70], the discriminator plays a vital role in extracting difficult-to-learn patterns of the real HR image. This important capability of the discriminator leads to forcing the generator to generate more realistic images [69], [70].

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

According to the adversarial loss and content loss concepts for reconstructing the SR image, the PSNR and SSIM evaluations show lower amounts compared to the pixel-loss-based models while the adversarial loss and content loss yield significant perceptual quality and realistic results compared to the pixel-loss-based results. To solve the PSNR and SSIM limitations and have a fair comparison, [9], [29] introduced the MOS tests.

2.5 Other Improvement

In addition to the previous strategies for improving the SR models, there are some other techniques such as Context-wise Network Fusion, Data Augmentation, Multi-task Learning, and Self-ensemble to improve the SR models.

2.5.1 Context-wise Network Fusion

The technique of context-wise network fusion (CNF) [40] refers to a stacking concept for fusing predictions from multiple SR networks. It means that the individual SR models with different architectures are trained separately, feed the prediction of each architecture into the individual convolutional layers, and conclusively the final prediction result is produced by summing up the outputs. The final structure of the CNF framework is constructed by three lightweight SRCNNs [21], and FSRCNN [17] and achieved comparable efficient performance with state-of-the-art models [40].

2.5.2 Data Augmentation

One of the common techniques for increasing the deep learning performance is data augmentation. It refers to increasing the data by adding slight modifications to the existing data. Some useful data augmentation items in the SR field include rotation,

Chulalongkorn University

cropping, scaling, flipping, color jittering [23], [26], [28], [30], [31], [71]. Moreover, Bei et al. [72] used the concept of augmentation by randomly shuffling the RGB channels, which augments data and mitigates color bias.

2.5.3 Multitask Learning

The multitask learning (MTL) [73] approach is used to improve the generalization capability of the deep learning network. It refers to solving multiple learning tasks by leveraging domain-specific information at the same time. Semantic segmentation and object detection [74], head pose estimation, and facial attribute inference [75] are the samples of multitask learning.

In the SR model, Wang et al. [65] integrate a semantic segmentation network to provide semantic knowledge and generate semantic-specific details. The spatial feature transformation is particularly employed to take semantic feature maps as input. Then the spatial-wise parameters of affine transformation are performed on the intermediate feature maps. The proposed algorithm [65] can generate more realistic and visually pleasing textures of reconstructed images.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

To solve the noise problem with this concept, DNSR [72] proposes his SR model with a denoising network and SR network which operate separately. The results of both networks are concatenated together and fine-tuned. Based on this idea, the cycle-in-cycle GAN (CinCGAN) [8] incorporates a cycle-in-cycle denoising framework with a cycle-in-cycle SR model to perform noise reduction and super-resolution tasks at the same time. Since different tasks produce different aspects of the data, thus a combination of the SR models with these tasks leads to improving the SR performance.

2.5.4 Network Interpolation

Network interpolation refers to the interpolation of two network results together. In the SR field, Wang et al., [76] propose a network by interpolating a PSNR-based and a GAN-based models to better balance the distortion and perception content. On the one hand, the PSNR-based algorithm produces images similar to the GT but has a blurring problem. On the other hand, the GAN-based model reconstructs better perceptual quality but has unpleasant artifacts. The PSNR-based and GAN-based models are trained separately. Then all the corresponding parameters of both networks are interpolated and produce meaningful results with fewer artifacts.

2.5.5 Self-ensemble

The self-ensemble method, enhanced prediction [71], is an inference technique usually utilized in the SR models. This method attempts to enhance the prediction by rotations with different angles and horizontal flipping of the LR images to produce a set of 8 images. Then the set of images is fed into the SR architecture and the corresponding inverse transformation is performed to the reconstructed HR images. The final prediction result is computed by the mean [26], [32], [37], [38], [44], or the median [25] of these outputs.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

2.6 Capsule Neural Network

Despite the success of Convolutional Neural Networks (CNN) in deep learning algorithms such as image classification, object detection, face recognition, and language translation, CNNs have some limitations. The CNN model is a translation invariance and cannot recognize the pose of objects (features). Furthermore, CNN has the polling operation layer that ignores many valuable data in the pooling layer. This problem leads to the requirement of numerous training data in our CNN model. To solve these two critical limitations of CNNs, Sabour et al. proposed the capsule network [77].

The Capsule Neural Network known as CapsNet is categorized as a field of machine learning and Artificial Neural Networks (ANN). The CapsNet can model the hierarchical relationships of data better than the CNN model. This approach operates based on a group of neurons that make the structure of a capsule. These capsules are added to a CNN and attempt to reuse the outputs of these capsules for more stable representations.

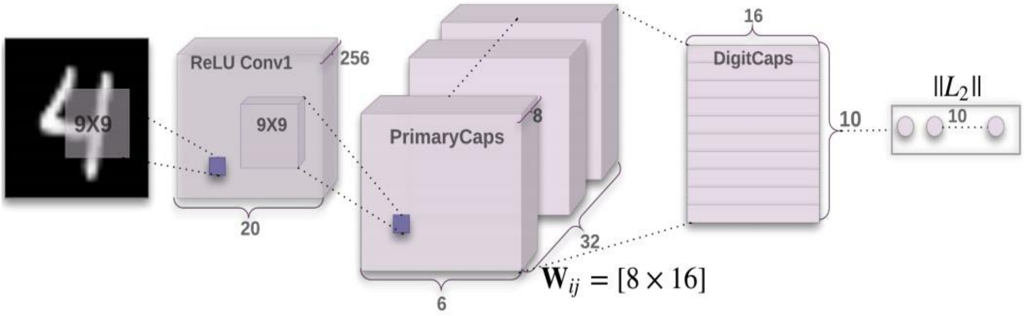


Figure 2.15: The CapsNet architecture.

The CapsNet architecture has a shallow structure. It contains two convolutional layers (the Primary Caps layer and the Digit Caps layer) and is followed by a fully connected layer, as shown in Figure 2.15. This Capsule architecture was first used for classificating handwritten digits of the MINIST datasets [78] and showed superior performance compared to the CNN architecture.

The first convolution layer has 256, 9 × 9 convolution kernels with a stride of 1, then continues with the ReLU activation function. The task of the first layer is to convert the pixel intensities to the activities of local feature detectors and the output has only one dimension (no orientation in its space). These local feature detectors are applied as the inputs to the Primary Capsules (first layer of the capsule structure).

Activation of the Primary Capsules corresponds to inverting the rendering procedure from an inverse graphics perspective. This Primary Capsule layer is a capsule convolutional layer with 32 channels of convolutional 8D capsules. That is to say; each, primary capsule consists of 8 units of convolutional task with a 9 × 9 kernel and a stride of 2. The Primary Capsules contain [32 × 6 × 6] capsule outputs. Each capsule output is a vector with 8 dimensions and each capsule in the [6 × 6] grid size distributes its weights to other capsules.

The DigitCaps is the final Layer of Capsule structure and has one 16 dimensions capsule per digit class (10-digit class). Each of these capsules in the Digit Layer receives input from all the capsules in the Primary Capsules layer. The dynamic routing algorithm is applied between the Primary Capsules and DigitCaps (the layers which contain the orientation). The whole routing logits are initialized to zero. Hence, a capsule output is initially sent to all parent capsules of $v0$ until $v9$ with equal probability. This implementation of handwritten digit classification was processed in TensorFlow [2] and the Adam optimizer [79] with its default parameters applied to it, to minimize the sum of the margin losses.

CHAPTER THREE
Proposed Network Architecture

3.1 Introduction

This section discusses the design methodology of our Caps-GAN method for single image SR. According to the GAN concept [54], two independent neural networks, generator and discriminator, are trained independently. The generator module is trained to generate a more accurate SR image, and the discriminator module is trained to distinguish precisely between real and fake images. In other words, the trained generator is skilled in fooling the discriminator while the trained discriminator attempts to prevent it, in an adversarial cycle [54]. The proposed Caps-GAN model consists of three main items: the generator module, the discriminator module, and the total losses. Figure 3.1 demonstrates the architecture of the proposed Caps-GAN model.

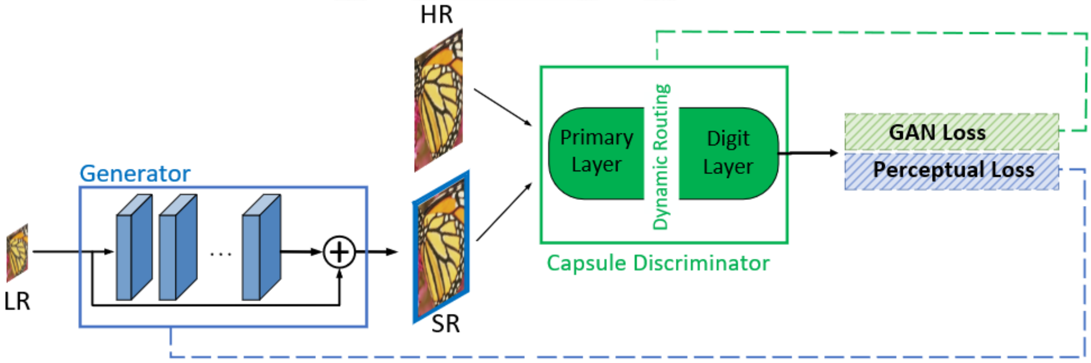


Figure 3.1: The proposed architecture of our Caps-GAN model.

To improve the GAN-based SR model, we are required to design all of these three items efficiently.

The generator is responsible for taking the LR image as the input and producing the SR (fake) image. An effective generator module should be able to generate the SR image as similar as possible to the GT image. Therefore, it should be capable of reconstructing the sophisticated tiny details and preserving the sharpness of the image without any degradation artifacts while considering the efficiency of the network complexity.

The discriminator module plays a crucial role in extracting difficult-to-learn patterns of the HR (real) image for comparing it with the SR (fake) image. Due to the ability of the capsule network to extract the hierarchical pattern relationships of data, our proposed capsule discriminator improves this binary classification (real or fake classification) procedure and force generator to produce a more realistic image.

Various types of losses are used in the proposed Caps-GAN model. The GAN loss is utilized for training the capsule discriminator while the perceptual loss (l^{SR}) is used for training our generator network. The proposed perceptual loss in our Caps-GAN consists of an adversarial loss (l^{SR}_{Gen}) to produce high-frequency details of the texture, and a content loss (l^{SR}_{Mix}) known as pre-trained to produce the overall appearance of the resulting image.

The rest of this section is categorized into the explanation of methodology, mathematical expression and hyperparameters of the generator network, the discriminator network, and the total loss of the proposed model.

3.2 Generator Architecture

In our Caps-GAN model, the generator module is responsible for taking the LR image and reconstructing the output SR image. The ultimate goal of a premier generator is to

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

have a robust module that can reconstruct the sophisticated image as similar to the HR image while preserving the tiny details such as lines, holes, quadrangular shapes, and lattice shapes in the SR image. Previous studies on the SR models were focused on the depth of the CNN network, which eventually led to improving the performance of the SR models. However, the deep SR architecture still faces some limitations as follows:

1) Increasing the depth of the SR network leads to an increase in the network parameters and consequently, it increases the complexity and computation cost of the network. The complexity and computation cost problems are important obstacles to utilizing the SR model in real-time applications.

2) Most existing deep learning SR models utilize the post-upsampling framework. Although this framework executes most of the computations in a low-dimensional space and helps reduce the complexity of computation, the post-upsampling raises the learning difficulties on the larger scale factors.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

To design a robust generator for our Caps-GAN network which can reconstruct the tiny details while solving these two mentioned limitations, it is necessary to design a lightweight progressively up-sampling architecture and utilize an effective pixel-wise objective function to produce the efficient model and increase the computational efficiency of our Caps-GAN architecture.

The feasible way to design a lightweight architecture for our generator is by employing a progressive up-sampling framework that performs fewer convolution layers in a low-dimensional space and progressively applies the predefined up-sampling operations in our network. Figure 3.2 illustrates our Progressive Multi

Residual Dense generator architecture consisting of the residual dense block (RDB) under a two-stage progressively up-sampling framework on a scale target of ×4.

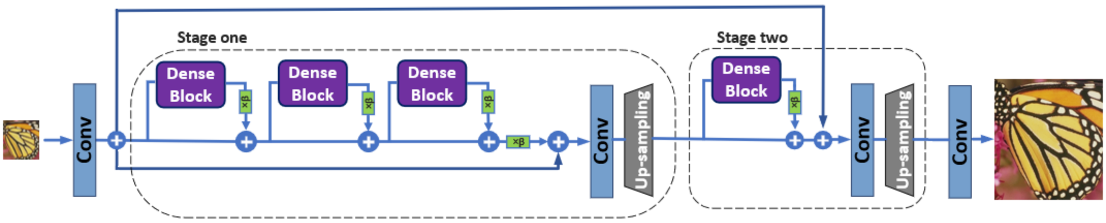


Figure 3.2: The proposed architecture of our generator using a Progressive Multi Residual Dense block structure

As shown in Figure 3.2, our proposed generator architecture includes the multi-level residual dense topology, called residual in residual dense blocks (RRDB), which will be used in the first stage of the progressively up-sampling framework and the second stage uses the RDB architecture.

Moreover, the depth-wise bottleneck projections are utilized to transfer the high-frequency details of the early layer to the up-sampling modules at the end of each up-sampling stage.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

The fusion objective function for training our SR model is made by combining of the L2 loss and Multiscale SSIM loss (MS-SSIM + L2).

The rest of this section is devoted to describing the in-depth implementation of the architecture of the generator network, the residual bottleneck projection, and the objective function to train the proposed generator of our GAN model.

3.2.1 The Architecture of Generator Network

To produce high-level features from low-level input, the residual network architecture shows excellent performance, especially in the SR models [22], [23], [29]. Due to the residual network's increased efficiency and computation costs, a combination of residual concept [26] and dense connections architecture [27] under the two-stage progressively up-sampling framework is employed as shown in Figure 3.2. The association of the multi-level residual network and the dense connections architecture (RRDB) [29], [80] is employed in the first stage of our progressive generator model. The residual dense block architecture (RDB) [44] is utilized at the second stage while the batch normalization (BN) layers are removed [29].

The BN layer has to normalize feature maps during the training and testing of the CNN model by using the mean and variance. In the training mode of any deep learning model, the BN layer performs based on the mean and variance of every batch. In the testing mode of any deep learning model the BN layer operates based on the mean and variance of the whole training dataset [26]. The differences between the mean and variance in the training and testing SR models lead to unpleasant visual artifacts, blurring effects, and inconsistent performance of the SR model [81].

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

To solve this unpleasant artifact using the BN layer in the SR model, the simplified architecture of the residual block is introduced as shown in Figure.3.3.

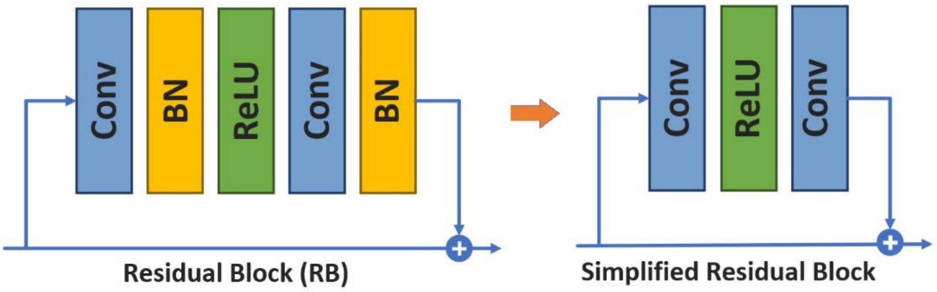


Figure 3.3: Simplified residual block by removing batch normalization layer.

The simplified structure of the residual block proved to increase the performance of computer vision tasks while dramatically decreasing the computational efficiency and memory usage [29], [26], [82].

The Dense Convolutional architecture (DenseNet) [43] aims to increase the information flow between layers in a feed-forward manner by connecting each layer of a network to every other layer. Due to the superior performance of this architecture (DenseNet) [43] in computer vision tasks, the successful SR models such as MemNet [22], CARN [42], RDN [44], RCAN [38], and ESRGAN [29] utilize this concept.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

According to the Dense Convolutional architecture, the feature maps of all previous layers are utilized as the inputs in every single layer. Subsequently, the yielded feature maps of each layer are used as inputs into all further layers. Based on the research evidence [33], [22], [82], employing more network layers and connections led to raising information flow between layers and consequently, it increases the SR model performance.

In our proposed generator model, we used the combination of the Dense block [43] and the simplified residual block [26] to create the Residual Dence Block (RDB) architecture. Figure 3.4 demonstrates the topology of RDB and the multi-level residual dense block network called Residual in Residual Dense Block (RRDB).

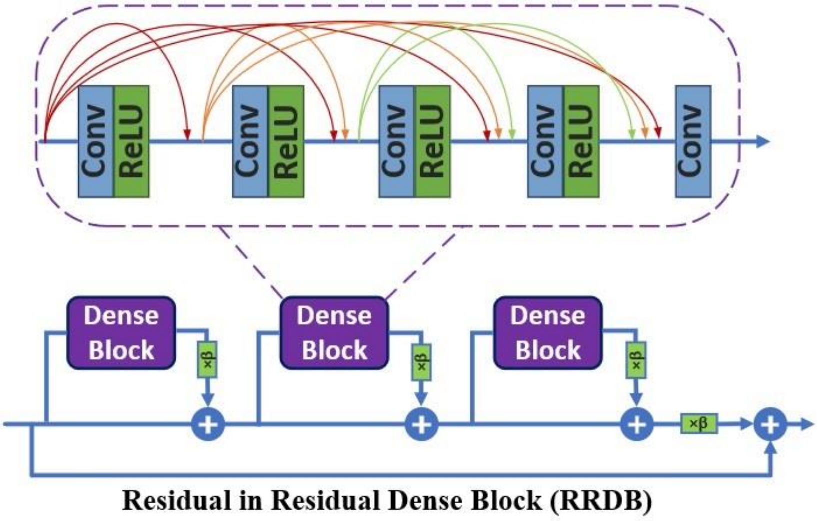


Figure 3.4: The dense block and the simplified residual in residual dense

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

β is the scaling parameter of the residual architecture from the range 0 to 1. The residual scaling parameter is multiplied by the output of the residual before adding to the main block as demonstrated in Figure 3.2 and Figure 3.4. According to the previous studies [29], [80], the optimum value for the residual scaling parameter is β = 0.2.

At the end of each stage of our progressive framework, the pixel shuffle [83] up-sampling model is used to enlarge the reconstructed features by a factor of 2. Among different up-sampling methods, the pixel shuffle [83] shows superior and efficient performances due to rearranging the feature map without losing information, without padding effect, and better PSNR result [84]. Sub-pixel convolution combines a single pixel on a multi-channel feature into an independent pixel on an image. Figure 3.5 illustrates the pixel shuffle operation concept for up-sampling on a scale of ×2.

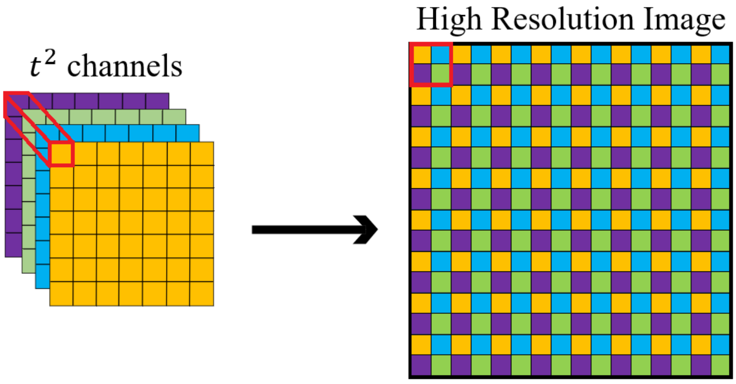


Figure 3.5: The pixel-shuffle module and transformation feature maps from the LR domain to the HR image on a scale of ×2.

Let’s assume that the feature map with dimensions H and W is up-sampling on a scale of t = 2. This model, groups the feature map into sets of $t^2 = 4$ channels C. Then rearrange each group into a 2 × 2 block of pixels. Finally, the output size is $(H \times r,$ $W \times r)$. In other words, the tensor shape of $(C \times t^2, H, W)$ is rearranged to the tensor shape of $(C, H \times r, W \times r)$ without losing information.

In summary, our generator structure is designed under a progressive up-sampling framework with two stages (scale ×4) of up-sampling to reduce the learning difficulties of our generator. The RDB contains five convolution layers followed by the ReLU activation function [92]. The mechanism of the proposed generator is that three RDB are residually connected to produce the RRDB architecture in the first stage. In contrast, in the second stage, it utilizes one RDB. At the end of each stage of our progressive model, the pixel shuffle [83] module is responsible for an upsample SR image on a factor of ×2.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.2.2 Residual Bottleneck Projection

The global residual connection was presented in many deep learning-based SR models [83], [85], [42], [36], [61]. The residual connection helps prevent gradient vanishing in the training of the CNN model and makes it feasible to design deeper network architecture. The residual projection is a variant of the residual connection that changes the dimension of the feature maps. In our proposed generator architecture, the features of the early layer at low dimension are upsampled by the bicubic interpolation technique and then fed into the higher-dimensional stages.

Multiple settings of the residual projection blocks were explored and represented in Figure 3.6 including the Residual Projection, Bottleneck Projection, and Depth-wise Bottleneck Projection.

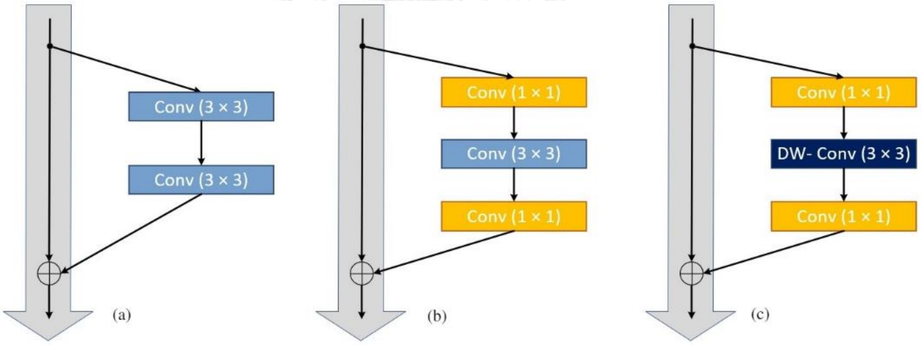


Figure 3.6: (a) The Residual Projection, (b) Bottleneck Projection, (c) Depth-wise (DW) Bottleneck Projection.

The residual projection architecture [35] shown in Figure 3.6.a consists of two convolutions with size 3 × 3, followed by a non-linear activation function.

The bottleneck projection stacks the 1 × 1, 3 × 3, and 1 × 1 convolution layers known as “bottleneck” building block [35], [39] is demonstrated in Figure 3.6.b. In this projection architecture, the first 1 × 1 convolution layer reduces the dimension of the feature map from 256-dimensional to 64-dimensional. The second convolution layer of 3 × 3 is used for computation, and via the final convolution layer, the feature dimension is changed to 256 by the last 1 × 1 convolution.

The proposed generator network utilizes an efficient bottleneck structure via depth-wise (DW) convolution known as a depth-wise bottleneck projection block. As shown in Figure 3.6.c, the first layer contains a 1 × 1 convolution to reduce the dimension of the feature map. The second layer includes a 3 × 3 depth-wise (DW) convolution operation. The DW convolution is a special form that aims to implement lightweight filtering via employing a single convolutional operation per every channel and then stacking them back [39]. The third layer of 1×1 convolution known as a point-wise convolution intends to construct new features by computing linear combinations of the input channels [33], [86], [87].

3.2.3 Objective Function

The objective function task is to measure the pixel-wise difference (error) between the reconstructed patch of the SR and the corresponding ground truth (GT) patch. To calculate an error function, the loss for a patch P can be mentioned as:

$$L^{\varepsilon}(P) = \frac{1}{N}\sum_{p \in P} \varepsilon(p) \tag{3.1}$$

Where P denotes the index of pixels and $\varepsilon(p)$ shows the values of the pixels in the error measurement.

Concerning a smoother SR result, the L1 loss function performs better than the L2 loss. However, both L1 and L2 losses correlate inadequately with image quality proved by human observation perception [88]. That is to say, using the loss function

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

that correlates independently with Human Vision System (HVS) is considered a feasible solution.

The sensitivity of HVS depends on the reconstructed image's local contrast, luminance, and structure [34], [88]. To enhance the network learning strategy based on the HVS that can reconstruct the SR image by attending to contrast, luminance, and structure qualities, the SSIM loss function is suggested [89].

Let's suppose $x(p)$and y(p) are two patches of the GT image and reconstructed image respectively. And let's assume μ_x and σ_x^2 are the mean and variance of x. The covariance of x and y is assumed to be σ_{xy}. Therefore, μ_x and σ_x^2 can be shown as the estimation of the luminance and contrast of x. Similarly, σ_{xy} measures the tendency of x and y to vary together and defines the structural similarity among x and y. The luminance, contrast, and structure evaluations are defined as follows:

$$l(x,y) = \frac{2\mu_x \mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1} \tag{3.2}$$

$$c(x,y) \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_x^2 + C_2} \tag{3.3}$$

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x + \sigma_y + C_3} \tag{3.4}$$

C_1, C_2 and C_3 are small constants defined by:

$$C_1 = (K_1 L)^2, \quad C_2 = (K_2 L)^2, \quad C_3 = {}^{C_2}\!/_{2} \tag{3.5}$$

Where L defines the dynamic range of pixel values (L = 255 for 8 bits/pixel images), and K_1 and K_2 define two scalar constants. The general form of the SSIM between GT patch and SR patch is explained as:

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$SSIM(x,y) = [l(x,y)]^{\alpha}.[c(x,y)]^{\beta}.[s(x,y)]^{\gamma} \tag{3.6}$$

Where α, β, and γ are parameters to define the relative importance of these components that are considered as $\alpha = \beta = \gamma = 1$. The SSIM index can be written as:

$$SSIM(p) = \frac{(2\mu_x \, \mu_y + C_1)}{(\mu_x^2 + \mu_y^2 + C_1)} \, \frac{(2\sigma_x \sigma_y + C_2)}{(\, \sigma_x^2 + \sigma_x^2 + \, C_2)} \tag{3.7}$$

$$SSIM(p) = l(p).cs(p) \tag{3.8}$$

Where the dependencies of means and standard deviations on pixel p are obtained. The Means and standard deviations are computed via a Gaussian filter with standard deviation σ_G, $G_{\sigma G}$. The SSIM loss function is presented as:

$$L^{SSIM}(P) = \frac{1}{N} \sum_{p \in P} 1 - SSIM(p) \tag{3.9}$$

In Eq 3.7, the calculation of $SSIM(p)$ requires exploring the neighborhood of pixel (p) as large as $G_{\sigma G}$ supports. The means computation of $L^{SSIM}(P)$ and its derivatives in some regions of the patch are not possible. The derivative computation at (p) for any other pixel (q) in the patch (P) can be defined as:

$$\frac{\partial L^{SSIM}}{\partial x(q)} = -\frac{\partial}{\partial x(q)} SSIM(p) = -\left(\frac{\partial l(p)}{\partial x(q)}.cs(p) + l(p).\frac{\partial cs(p)}{\partial x(q)} \right) \tag{3.10}$$

Where $cs(p)$ and $l(p)$ are the first and second terms of Eq 3.8 and their derivatives are:

$$\frac{\partial l(p)}{\partial x(q)} = 2.\, G_{\sigma G}(q-p)\,.\left(\frac{\mu_y - \mu_x \,.\, l(p)}{\mu_x^2 + \mu_y^2 + C_1} \right) \tag{3.11}$$

and

$$\frac{\partial l(p)}{\partial x(q)} = \frac{2}{\sigma_x^2 + \sigma_x^2 + C_2} \,.\, \mathrm{G}_{\sigma G}(q-p)\,.\left[\left(y(q)-\mu_y\right) - cs(q).\left(x(q)-\mu_x\right)\right] \tag{3.12}$$

Where $\mathrm{G}_{\sigma G}(q-p)$ is the Gaussian coefficient correlated with pixel q.

As mentioned earlier, the quality of the reconstructed image depends on σ_G. For example, the large value of σ_G tends to preserve noise in the edge while the small value of σ_G leads to unpleasant artifacts since it reduces our generator's ability to reconstruct the image's local structure. Using the multiscale structure of SSIM (MS-SSIM) designed according to a dyadic pyramid of M levels resolution, is a feasible solution for the SSIM limitation. Figure 3.6 demonstrates the MS-SSIM diagram, and it is defined as:

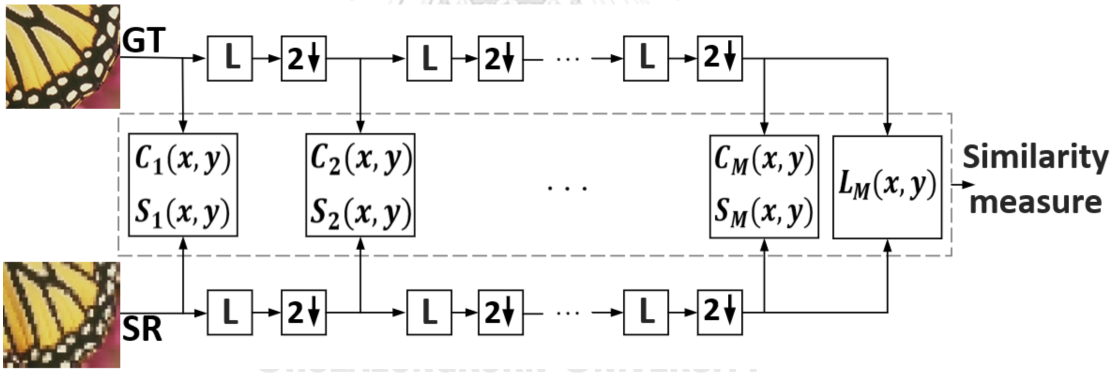


Figure 3.7: Block diagram of Multiscale SSIM (MS-SSIM).

$$MS - SSIM = l_M^{\alpha}(p).\prod_{j=1}^{M} cs_j^{\beta\gamma}(p) \tag{3.13}$$

Where $l_M(x,y)$, $c_j(x,y)$, and $s_j(x,y)$, demonstrate luminance, contrast, and similarity respectively. As shown in the structure of MS-SSIM in Fig 3.7, the GT patch and SR patch are taken as inputs. The low-pass filter and down-sample operation on a factor of 2 are applied iteratively on the inputs. The input patches are indexed as the first scale while the highest order scale is considered scale M which is obtained after $M-1$ iterations.

As demonstrated in the diagram and equation, the luminance comparison is calculated only on a scale of M and it is defined as $l_M(x,y)$. In contrast, the structure and contrast comparisons are computed at the j_{th} scale and are defined as $c_j(x,y)$, and $s_j(x,y)$ respectively. We set $\alpha = \beta\gamma = 1$. The final loss for the patch (P) with its center pixel (p) is defined as:

$$L^{MS-SSIM}(P) = 1 - MS\_SSIM(p) \qquad (3.14)$$

The derivative of the MS-SSIM loss function can be described as:

$$\left(\frac{\partial L^{MS-SSIM(P)}}{\partial x(q)}\right) = \left(\frac{\partial l_{M(p)}}{\partial x_{(q)}} + l_M(p).\sum_{i=0}^{M}\frac{1}{cs_j(p)}\frac{\partial cs_j(p)}{\partial x(q)}\right).\prod_{j=1}^{M} cs_j(p) \qquad (3.15)$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

However, The MS-SSIM loss makes a smoother SR result compared to the L2 loss and it preserves the image's contrast in high-frequency regions better than the L2 loss. On the other hand, L2 loss preserves the edges, and it is very sensitive to indicate the sharp intensity changes. To reconstruct the best result of our SR model, the combination of MS-SSIM loss and L2 loss function is proposed:

$$L^{Mix} = (1-\alpha)L^{MS-SSIM} + \alpha(G_{\sigma_G^M}.L^{L2}) \qquad (3.16)$$

We empirically set α = 0.8. We must note that we add a point-wise multiplication between $G_{\sigma_G^M}$ and L2.

3.3 Capsule Discriminator Architecture

Inspired by Goodfellow et al. [54] idea, the discriminator network $D_{\theta D}$ is designed to optimize the capsule network structure [77] and it operates in an alternating manner along with the generator to solve the adversarial min-max problem of our GAN structure as shown in Eq 3.17:

$$\begin{aligned} \underset{\theta G}{min}\ \underset{\theta D}{max}\ & E_{I^{HR}\sim P_{train}(I^{HR})}\left[\log D_{\theta D}\,(I^{HR})\right] \\ & + E_{I^{LR}\sim p_G(I^{LR})}\left[\log\left(1-D_{\theta D}\left(G_{\theta G}\left(I^{HR}\right)\right)\right)\right] \end{aligned} \tag{3.17}$$

This adversarial training of generator and discriminator simultaneously produces realistic perceptual (small details and texture) images in contrast to non-GAN models which only aim to minimize the pixel-wise error measurements. To make an efficient adversarial cycle (min-max problem) in our GAN, the robustness of the discriminator plays a critical role. Due to the capability of the capsule network to extract the hierarchical pattern relationships, and to learn the difficult-to-learn patterns compared to the CNN model, our proposed discriminator utilizes this approach. Figure 3.8 shows the proposed discriminator's architecture based on the capsule network.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

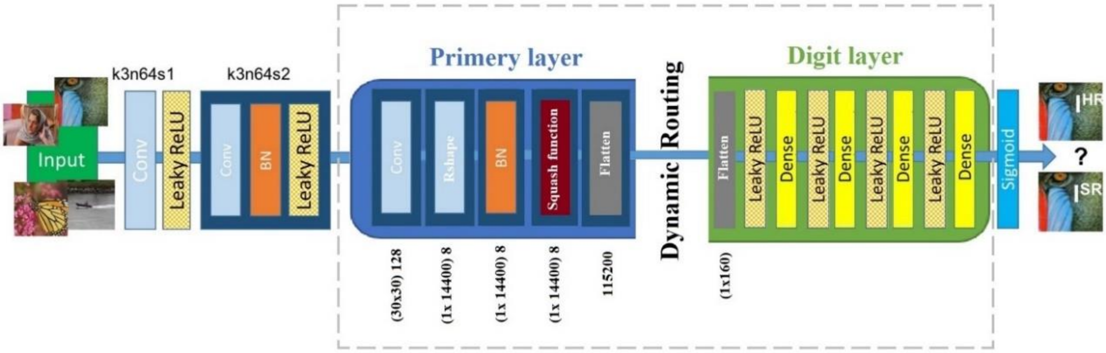


Figure 3.8: The architecture of Discriminator with Capsule layers in the Caps-GAN model.

As demonstrated in Figure 3.8, our discriminator prepares data by convolution layers and then feeds data to the primary layer of our capsule network. The capsule structure consists of two layers, including the primary layer and the digit layer, which dynamic routing operation is applied between these layers. The sigmoid function dose the final classification. In the following section, we will explain the details and pressures of the discrimination capsule network.

Our capsule discriminator architecture starts with a layer of convolution with 3×3 kernel size and 64 feature maps followed by the Leaku ReLU activation function. Then another layer of convolution with 3×3 kernel size and 64 feature maps and stride of two, followed by the batch-normalization layers [90] and ParametricReLU [91] as the activation function, prepares the input of the capsule layer.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

The capsule architecture layer consists of two sides. The primary side with blue color and the digit layer side is shown with green color in Figure 3.8. In the primary layer first, a convolution layer changes the input dimension to 128 feature maps. Then it is reshaped into 8 capsules with a dimension of 1×14480. After that, a layer of squash function is applied to vectorize data to limit their magnitude between 0 and 1, and then flatten them.

Dynamic Routing (Routing by agreement) is considered the first stage of the digit layer. It is a procedure of CapsNet that routs output vectors from layer l to layer l + 1. This procedure is utilized as a replacement for the Max Pooling layer of CNNs. The CapsNet learns the transformation matrices, or simply weights as the associations between capsule layers. In addition to these transformation matrices for routing information from layer to layer, each connection of CapsNet is also multiplied by a coupling coefficient demonstrated by C_{ij}. With the aim of decision making, the capsules adjust the coupling coefficient to multiply it by the output before sending it to the higher layer. The coefficient C is defined as a Softmax over b.

$$C_{ij} = \frac{\exp b_{ij}}{\sum_k exp\, b_{ik}} \tag{3.18}$$

Where b_{ij} is the similarity score that conders both probability and the feature properties in neurons. In addition, b_{ij} has remained low if the activation u_i of capsule i is low. Since the length of $\hat{u}_{j|i}$ is proportional to u_i therefore:

$$b_{ij} \leftarrow b_{ij} + \hat{u}_{j|i} \,.\, v_j \tag{3.19}$$

The C_{ij} are coupling coefficients calculated by the iterative dynamic routing process. Since $\sum C_{ij}$ are designed to sum up them to one, it measures how likely capsule ii may activate capsule jj.

$$s_j = \sum\nolimits_i C_{ij}\ \hat{u}_{j|i} \tag{3.20}$$

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Where s_j demonstrates by the output of the higher capsule after the routing procedure. However, the routing is not yet finished. The squashing function is applied to the output to keep the value between 0 and 1:

$$v_j = \frac{|s_j|^2}{1 + |s_j|^2} \cdot \frac{s_j}{|s_j|} \tag{3.21}$$

Where v_j is represented the output of a squashing function. This type of routing-by-agreement shows a more effective result than the Max Pooling layer of CNNs which ignores more image details [77].

After the dynamic routing process, the vectors flatten and the Leaku ReLU activation function is applied to them to create the dense layer. The Leaku ReLU and dense layer continued four times and finished by applying a sigmoid function to it. The sigmoid activation function [92] is responsible for binary classification that classifies the data as real or fake.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

3.4 Total Loss of Caps-GAN

According to the GAN concept, our proposed Caps-GAN model has to train both capsule discriminator and the generator. The generator is trained to generate a more realistic SR image (fake image). In contrast, the discriminator is trained to distinguish between the real (HR) and fake (SR) image generated by the proposed generator network [54]. In our Caps-GAN model, the capsule discriminator is trained with GAN loss [54], [29], and the generator is trained with a perceptual loss (l^{SR}) [9], [29]. Our designed perceptual loss (l^{SR}) consists of two types of losses including a content loss (l^{SR}_{Mix}), and an adversarial loss (l^{SR}_{Gen}). Figure 3.9 demonstrates the total loss functions utilized in our model.

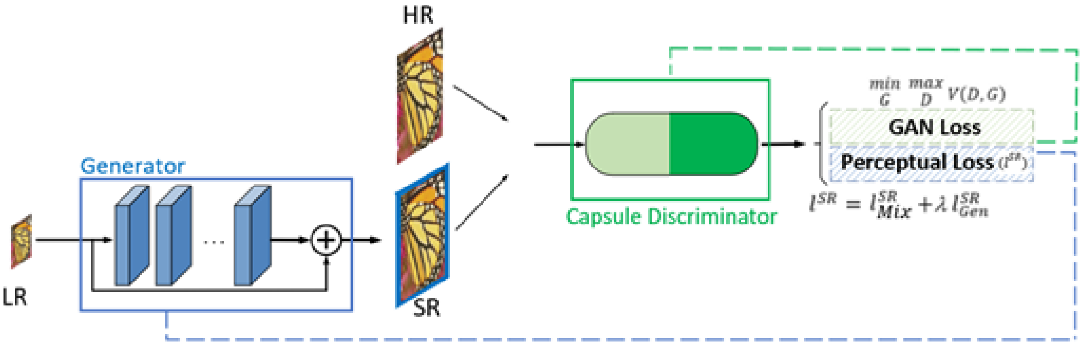

HR
Generator
LR
SR
Capsule Discriminator
$\min_G \max_D V(D,G)$
GAN Loss
Perceptual Loss (l^{SR})
$l^{SR} = l^{SR}_{Mix} + \lambda\, l^{SR}_{Gen}$

Figure 3.9: The GAN loss and perceptual loss of the Caps-GAN model.

In this section, we will explain the total loss functions including GAN loss, perceptual loss (Content loss and Adversarial loss), and their detailed combinations for our Caps-GAN model.

3.4.1 GAN Loss

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Inspired by the GAN structure [54], our capsule discriminator operates alternately along with the proposed generator to solve the adversarial min-max problem as demonstrated in Figure 3.9.

The capsule discriminator takes both generated image and the original HR image. A binary classification problem using the sigmoid function [92] gives the output from 0 to 1. This idea is known as the GAN loss which is formalized in Eq 3.22.

$$E_{I^{HR} \sim P_{train}(I^{HR})}\left[\log\, D_{\theta D}\,(I^{HR})\right] + E_{I^{LR} \sim p_G(I^{LR})}\left[\log\left(1 - D_{\theta D}\left(G_{\theta G}\left(I^{HR}\right)\right)\right)\right] \tag{3.22}$$

There are two phases for training our Caps-GAN model. The first phase is to freeze the proposed progressive generator and only train the capsule discriminator which tries to maximize the first term of formulation to 1.

The second phase freezes the capsule discriminator while our progressive generator produces the SR (fake) sample and passes through the capsule discriminator to minimize the second term of formulation to zero.

To be more precise, the capsule discriminator side of Eq 3.22 is used to train the capsule discriminator. In contrast, the generator side of Eq 3.22 is considered an adversarial loss utilized for training our generator.

3.4.2 Perceptual Loss

The perceptual loss function (l^{SR}) has a vital role in the performance of producing SR images with better perceptual quality. Since our model is a GAN-based model, the perceptual loss function only affects the generator network's training, as demonstrated in Figure 3.9.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Usually, the l^{SR} is modeled based on Pixel-wise type of loss function in most SR algorithms (non-GAN based models). However, in our Caps-GAN algorithm, another loss function approach (Adversarial loss) for the production of the SR image more naturally and realistically due to reconstructing high-frequency detail of the image texture. Eq 3.23 demonstrates the weighted sum of a content loss (l^{SR}_{Mix}) and an adversarial loss (l^{SR}_{Gen}) that applied as a perceptual loss (l^{SR}) function of our Caps-GAN model.

$$l^{SR} = l^{SR}_{Mix} + \lambda\, l^{SR}_{Gen} \tag{3.23}$$

Content Loss (l_{Mix}^{SR}): The content loss is responsible for producing the overall appearance of the SR image while it shows weakness in recovering the high-frequency detail. The concept of the content loss is similar to the objective function utilized in the non-GAN SR model and usually, the pixel-wise loss models such as L2 (MSE) or L1(MAE) are used. It is noticeable that this loss function in the GAN and non-GAN models have a direct effect on having higher PSNR and SSIM in the resulting image.

The content loss (l_{Mix}^{SR}) in the GAN models is used as a pre-trained model inspired by Bruna et al. [93], Gatys et al. [94], and Huang et al. [95].

In our Caps-GAN model, we introduced a novel objective function by combining MS-SSIM and L2 that was explained in section 3.2.3 and later defined in Eq 3.16. In our Caps-GAN model, we called the proposed MS-SSIM + L2 as Mix loss and defined it as Eq 3.24.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$$L^{Mix} = (1-\alpha)L^{MS-SSIM} + \alpha(L^{L2}) \tag{3.24}$$

To employ our proposed content loss (l_{Mix}^{SR}) in our Caps-GAN model, we initially train our proposed progressive generator with L^{Mix} loss. Since the overall appearance of the SR results depends on the content loss, it plays an important role in improving PSNR and SSIM evaluation in our Caps-GAN model. Eq 3.25 demonstrates our designed content loss (l_{Mix}^{SR}).

$$l_{Mix}^{SR} = \frac{1}{r^2WH}\sum_{x=1}^{rW}\sum_{y-1}^{rH}(I_{x,y}^{HR} - G_{\theta G}\left(I^{LR}\right)x,y) \tag{3.25}$$

Where W and H denote the dimension of the patch, $G_{\theta G}\left(I^{LR}\right)$ shows the generated image and $I_{x,y}^{HR}$ demonstrates the GT image.

Despite the benefit of getting higher PSNR and SSIM in the proposed content loss, the result often lacks of high-frequency detail which leads to perceptually unsatisfying results in representing textures details. The adversarial loss in our Caps-GAN model is responsible for recovering the high-frequency texture detail to produce a natural SR result.

Adversarial Loss (l_{Gen}^{SR}): The generative component loss function (adversarial loss) aims to compensate for the lack of high-frequency detail, especially in the texture regions of the SR image. Producing a realistic SR image containing high-frequency details requires adding adversarial loss to the content loss. Consequently, the weighted sum of these losses produces our perceptual loss as demonstrated in Eq 3.23. In our adversarial loss, the coefficient λ is set to 10^{-3}. It means that the adversarial loss encourages the generator to fool the capsule discriminator with the manifold of natural images.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

$l_{Gen}^{SR}$is determined based on the probabilities of the capsule discriminator network $D_{\theta D}(G_{\theta G}(I^{LR}))$ overall training samples and demonstrate as Eq 3.26.

$$l_{Gen}^{SR} = \sum_{n=1}^{N} -\log D_{\theta D}(G_{\theta G}\left(I^{LR}\right)) \tag{3.26}$$

Where $D_{\theta D}\left(G_{\theta G}(I^{LR})\right)$ denotes the probability which the reconstructed SR image $G_{\theta G}(I^{LR})$ is a natural HR images, and for better gradient behavior the probability of reconstruction is minimized as follows:

$$l_{Gen}^{SR} = \log\left[1 - D_{\theta D}\left(G_{\theta G}\left(I^{LR}\right)\right)\right] \tag{3.27}$$

Figure 3.10 visualizes the effect of content loss (overall appearance), adversarial loss (high-frequency texture detail) and our perceptual loss utilized in the proposed Caps-GAN model.

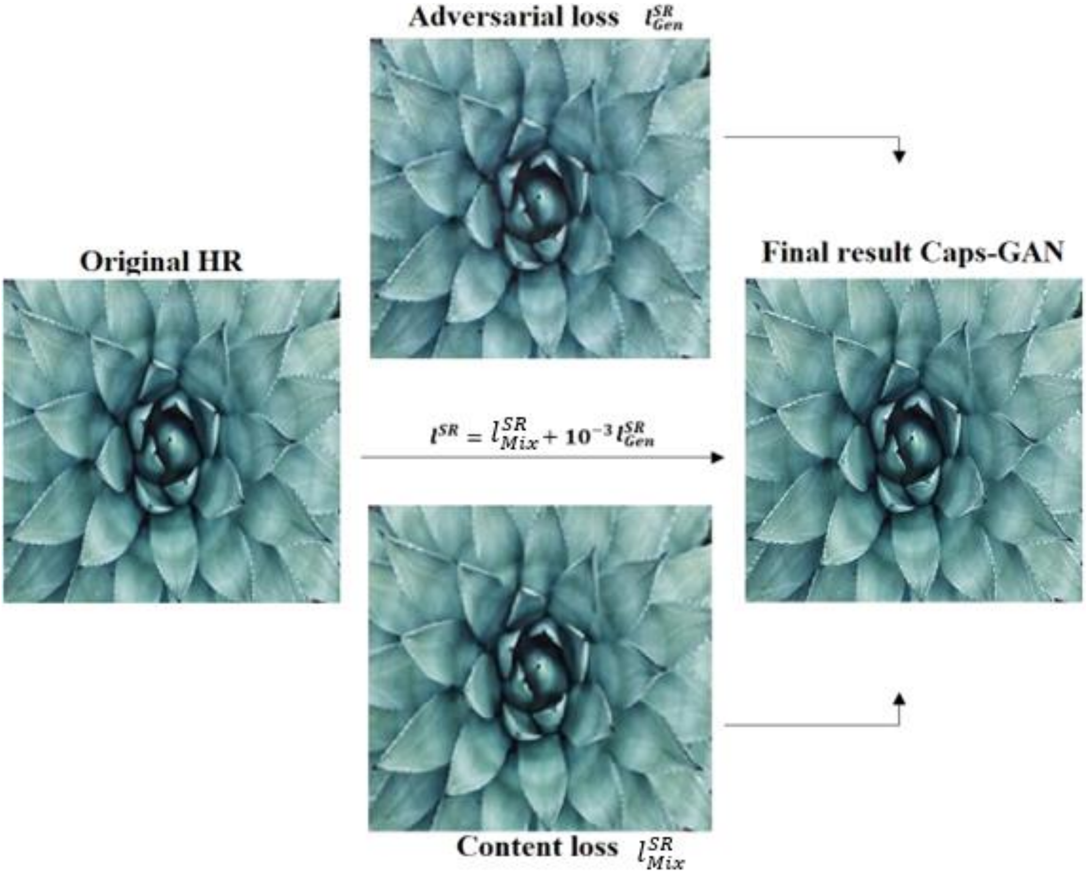

Adversarial loss l_{Gen}^{SR}
Original HR
Final result Caps-GAN
$l^{SR} = l_{Mix}^{SR} + 10^{-3} l_{Gen}^{SR}$
Content loss l_{Mix}^{SR}

Figure 3.10: The effect of content loss and adversarial loss to train our Caps-GAN model.

CHAPTER FOUR
Experimental Result and Discussion

4.1 Introduction

This chapter presents, the quantitative and qualitative experimental results of our Caps-GAN network. All experiments are conducted on a single image super-resolution and evaluated on scale factors of ×2, ×4, and ×8

The training and testing details of our Caps-GAN model are presented in sections 4.2 and 4.3 respectively.

In section 4.4, we will present the effectiveness of the proposed generator module in reconstructing the sophisticated structural images, and the effect of the proposed loss function (MS-SSIM + L2) in terms of quantitative and qualitative examinations which utilize pre-trained content loss (l_{MIx}^{SR}) for the final Caps-GAN model.

In section 4.5, the effect of the capsule discriminator on training our GAN model and, the visual comparisons for investigating the capsule architecture effects of the SR results are discussed.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Finally, the ultimate quantitative and qualitative experimental of our Caps-GAN model, as well as the complexity and execution time of the Caps-GAN model, is represented in section 4.6.

4.2 Training Details

The DIV2K dataset [96] is the most commonly used among various datasets for image super-resolution. The DIV2K dataset contains 800 high-quality (2K resolution) RGB images with a wide range of scene diversity used to train our model.

For each training batch of our model, the random LR patches in RGB mode with a size of 96×96 have been utilized as the inputs.

Adam optimizer [92] with a configuration of β1= 0.9, β2 = 0.999, and $\varepsilon = 10^{-8}$ is utilized for training the proposed model.

The Python 3.5.6 programming language under Keras 2.2.4 framework [97] with TensorFlow 1.5 as the back-end is used to implement our SR model.

Our model was trained on NVIDIA GeForce GTX 1080 Ti GPU with 24GB of Memory. The learning rate of our proposed model is set to 10^{-4}.

4.3 Testing Details

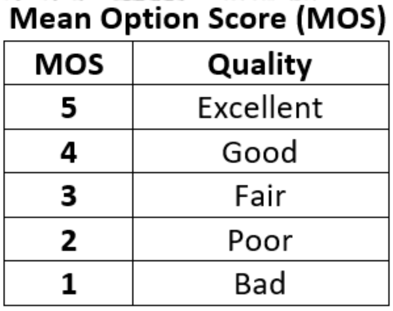
To test our model, we examined its performance on six benchmark datasets including Set5 [98], Set14 [99], BSD100 [100], Urban100 [95], Manga109 [101], and 100 images that belong to the testing purpose of the DIV2K [96]. The image representatives as mentioned above are demonstrated in Figure 4.1.

จุฬาลงกรณ์มหาวิทยาลัย

Figure 4.1: Testing Datasets used to evaluate the performance of the model.

For the quantitative evaluation of data, we used PSNR and SSIM [102], and MOS [9], [103] as image quality assessment metrics in which the higher values for the PSNR / SSIM / MOS indicate better performance of the resulting images. In the calculation of the PSNR and SSIM, we implemented the evaluation on the luminance (Y) channel of transformed YCbCr space. MATLAB 2018a is used to calculate the PSNR and SSIM of our proposed model.

The most straightforward method for obtaining the perceptual quality of an image is to acquire the MOS (Mean opinion score) [103] which is obtained from the perception of human observers [104]. The score of the MOS test is from 1 (bad quality) to 5 (excellent quality) as shown in Figure 4.2.



Mean Option Score (MOS)

MOS	Quality
5	Excellent
4	Good
3	Fair
2	Poor
1	Bad

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 4.2: The assessment criteria of the Mean Option Score (MOS)

In our MOS evaluation, 20 raters assigned their perception of our Caps-GAN results in the visual comparison section of this chapter on scales of ×4 and ×8 in 5 datasets including Set14 [99], BSD100 [100], Manga109 [101], Urban100 [95], and DIV2K [96].

4.4. Generator Module

As discussed earlier, our generator module is designed with the RDB architecture containing five convolution layers for each block. They are residually connected to make the RRDB architecture under the progressive up-sampling framework to reduce the learning difficulties of our generator. In addition, the depth-wise bottleneck projection approach is employed to convey the low-level information from the early convolution layer into each up-sampling module. The proposed content loss (L^{Mix}) that combining $L^{MS-SSIM}$ and L^{L2} is responsible for the reconstruction of the overall appearance of the SR image. Our generator with this content loss (L^{Mix}) is trained independently from our GAN model, then this pre-trained model is used as the pre-trained content loss of our Caps-GAN model.

The rest of this section demonstrates the effect of different projection approaches on our generator, compares different content losses in our model, and compares our pre-trained model with other non-GAN models.

4.4.1 Effect of Different Projections on Training

The effect of different projection approaches such as the Residual Projection, Bottleneck Projection, and Depth-wise Bottleneck Projection in the training stage of the proposed generator are compared in this section. Figure 4.3 and Figure 4.4 indicate the graphs for average training loss and average training PSNR (dB) on 800 training epochs of the DIV2K dataset.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

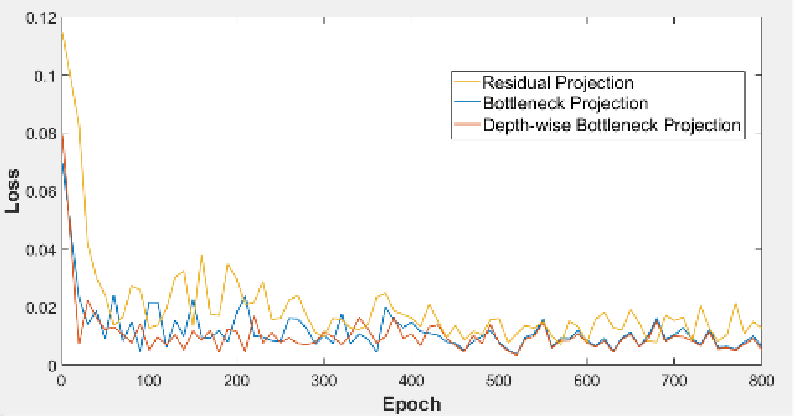
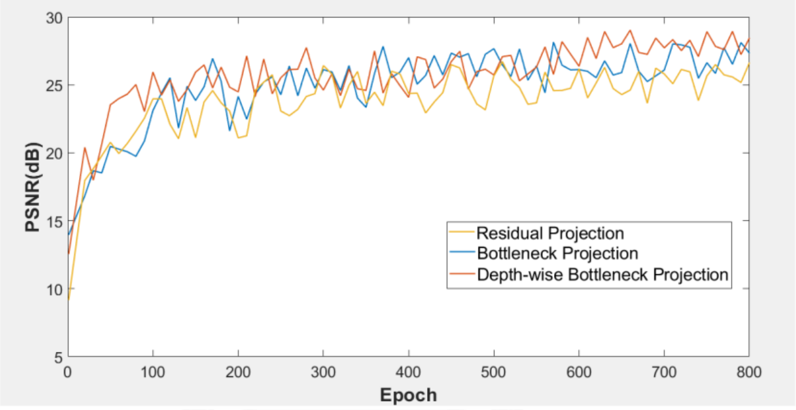


Figure 4.3: Average training loss per epoch for training our model with different projection approaches on the DIV2K dataset.



จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

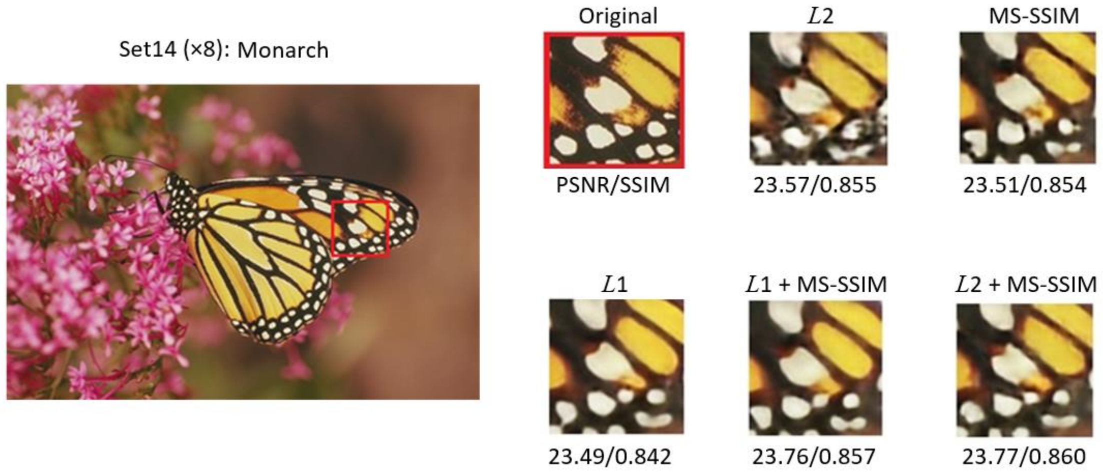
Figure 4.4: Average PSNR (dB) per epoch for training our model with different projection approach on the DIV2K dataset.

As shown in these graphs, the Bottleneck Projection (blue) and Depth-wise Bottleneck Projection (red) represent superior performance over the residual projection approach (yellow). Although the performance of Bottleneck Projection and Depth-wise Bottleneck Projection are almost the same, the Depth-wise Bottleneck approach shows smoother and better performances in both average loss and average PSNR (dB) in the training phase of the proposed generator.

4.4.2 Comparison of Different Content Losses

The effect of different objective functions (content losses) including L1 loss, L2 loss, MS-SSIM loss, L1+MS-SSIM loss, and L2+MS-SSIM loss are compared in this section.

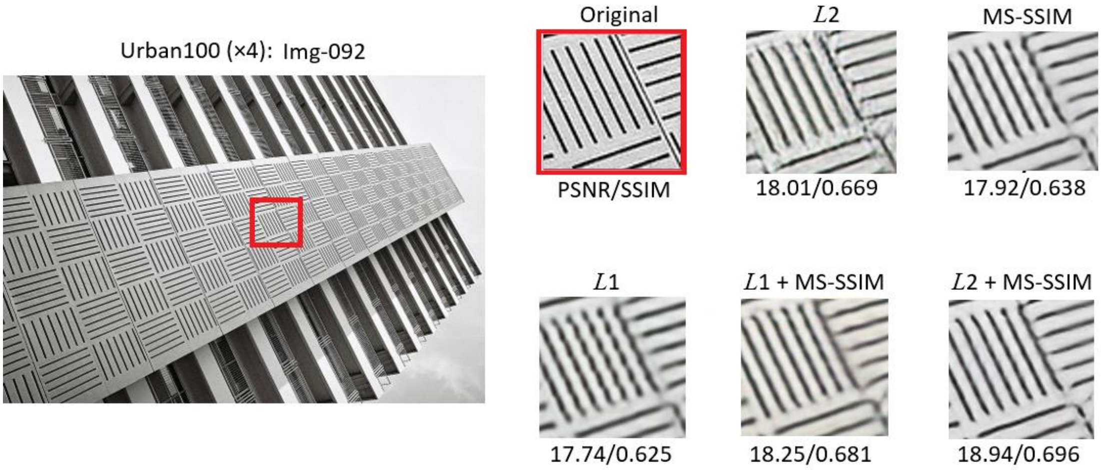
The visual compression of our pre-trained generator model with different content losses is shown in Figures 4.5 and 4.6. For a better visually distinguishing benchmark of the results of these content losses, we utilized the samples with different textural structures on different scale factors (scale ×4 and scale ×8).



CHULALONGKORN UNIVERSITY

Figure 4.5: Visual comparison of different content losses on the Monarch image of Set14 dataset on a scale of ×8

Figure 4.5 demonstrates the results of our generator model with the different content losses on the “Monarch” image of Set14 [99] dataset on a scale of ×8. Since the L1 loss function penalizes the smaller error compared to the L2 loss function, the result of L1 is much smoother and sharper than L2. At the same time, the high-frequency details and minor features in the regions connecting edges vanish. It means despite the smoothness of the L1 result, it shows weakness in reconstructing the minor details of the image similar to the original image. Although the result of MS-SSIM loss reconstructs a sharper image compared to L2 and represents more details than L1, it shows weakness in reconstructing the edges similar to the original image. The mix of L2 and MS-SSIM displays a more realistic result compared to the other content loss functions. It reconstructs a sharp result while more minor details are preserved around the edges. The PSNR (dB) and SSIM evaluations also show a superior performance of the proposed loss function.



จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Figure 4.6: Visual comparison of different content loss on Img-092 image of the Urban100 dataset on a scale of ×4.

Figure 4.6 demonstrates the results of the proposed generator model with different content losses on “Image-92” of the Urban100 [95] dataset on a scale of ×4. These results compare the effect of each content loss for reconstructing the vertical and horizontal lines over a constant surface. As observed in Figure 4.5, the L2 loss represents better performance for reconstructing the vertical and horizontal lines compared to other non-mixed content losses. However, lack of smoothness due to over noise amplification (greater penalizes error in L2) makes it a non-pleasing image. The mixed L2 and MS-SSIM loss function show the best performance to produce a sharp image while detecting all the lines similar to the original image. The PSNR (dB) and SSIM evaluations also indicate the best performances compared to the other content losses.

The quantitative performance comparison includes PSNR (dB) and SSIM on the benchmark datasets of Set5 [98], Set14 [99], BSD100 [100], Urban100 [95] and Manga109 [101] between these content losses on scales of ×2, ×4, and ×8 that are demonstrated in Table 4.1, Table 4.2, and Table 4.3. The red numbers indicate the best performance, and the blue numbers show the second-best performance.

Table 4.1: Quantitative comparison among different content losses on scale ×2

Content Loss	**Scale**	**Set 5 [98]**		**Set 14 [99]**		**BSD100 [100]**		**Urban100 [95]**		**Manga [101]**	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
L1	× 2	37.68	0.9598	33.30	0.9157	32.11	0.8964	31.30	0.9204	38.16	0.9758
L2	× 2	37.80	0.9613	33.39	0.9166	31.93	0.8971	31.15	0.9202	38.16	0.9767
MS-SSIM	× 2	37.61	0.9622	33.30	0.9159	32.12	0.8964	31.29	0.9210	38.18	0.9771
L1+MS-SSIM	× 2	37.79	0.9611	33.37	0.9162	32.17	0.8981	31.33	0.9213	38.19	0.9770
L2+MS-SSIM	× 2	37.81	0.9622	33.38	0.9168	32.16	0.8980	31.35	0.9214	38.18	0.9771

CHULALONGKORN UNIVERSITY

Table 4.2: Quantitative comparison among different content losses on scale ×4

Content Loss	**Scale**	**Set 5 [98]**		**Set 14 [99]**		**BSD100 [100]**		**Urban100 [95]**		**Manga [101]**	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
L1	× 4	32.65	0.9015	29.03	0.7890	28.10	0.7191	27.10	0.8112	31.94	0.9181
L2	× 4	32.70	0.9020	29.19	0.7911	28.18	0.7510	27.14	0.8131	32.09	0.9198
MS-SSIM	× 4	32.67	0.9015	29.11	0.7902	28.15	0.7501	27.13	0.8121	32.01	0.9190
L1+MS-SSIM	× 4	32.72	0.9025	29.23	0.7921	28.21	0.7512	27.18	0.8139	32.11	0.9201
L2+MS-SSIM	× 4	32.80	0.9032	29.30	0.7928	28.25	0.7532	27.22	0.8146	32.15	0.9210

Table 4.3: Quantitative comparison among different content losses on scale ×8

Content Loss	**Scale**	**Set 5 [98]**		**Set 14 [99]**		**BSD100 [100]**		**Urban100 [95]**		**Manga [101]**	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
L1	× 8	27.36	0.7980	25.27	0.6572	24.98	0.6067	23.14	0.6835	25.62	0.8131
L2	× 8	27.41	0.7987	25.31	0.6581	25.04	0.6072	23.25	0.6842	25.69	0.8141
MS-SSIM	× 8	27.38	0.7983	25.30	0.6579	25.00	0.6070	23.22	0.6839	25.66	0.8137
L1+MS-SSIM	× 8	27.51	0.7992	25.42	0.6590	25.09	0.6080	23.31	0.6849	25.80	0.8148
L2+MS-SSIM	× 8	27.54	0.7998	25.45	0.6598	25.17	0.6086	23.39	0.6865	25.86	0.8154

As the table shows, among all scales, the best performances belong to the mix of L2 and MS-SSIM objective functions utilized to pre-train our generator.

4.4.3 Comparison Results with Other Non-GAN models

In this section, we compare the effectiveness of our generator model with some state-of-the-art non-GAN SR models, including the visual comparisons and the quantitative comparisons on scales of ×2, ×4, and ×8. Since the generator performance directly effects on our final Caps-GAN result in terms of the PSNR, SSIM, and the overall perceptual quality, the robustness examination of our generator is very important.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

In a visual comparison, we compare the results of SRCNN [21], VDSR [58], LapSRN [30], MemNet [22], MSLapSRN [31], EDSR [26] and RCAN [38] models with our generator results on scales of ×4 and ×8 on the BSD100 [100], Manga109 [101] and Urban100 [95] datasets.

Figure 4.7 demonstrates the visual comparisons on a scale of ×4 for image “3096” of the BSD100 [100] dataset. As observed in Figure 4.7, other SR models show weakness in reconstructing the sharp image with small details and suffer blurring artifacts. In contrast, our generator model reduces the blurring effect and recovers minor geometric shapes better than other models.

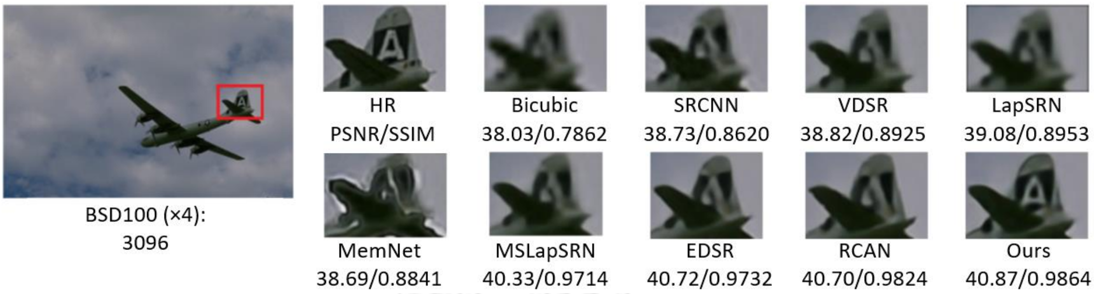


Figure 4.7: Visual comparison of the image "3096" on a scale of ×4 on the BSD100 dataset.

Figure 4.8, demonstrates visual comparisons on a scale of ×4 for the image “GakuenNoise” belonging to the Manga109 [101] dataset. The other models show weakness in representing the lattices' circular shapes. Some models suffer the blurring artifacts while in the RCAN [38] and EDSR [26], the reconstructed lattice shapes are not similar to the original HR image. In contrast, the proposed generator performs better to recover the lattice circular details.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

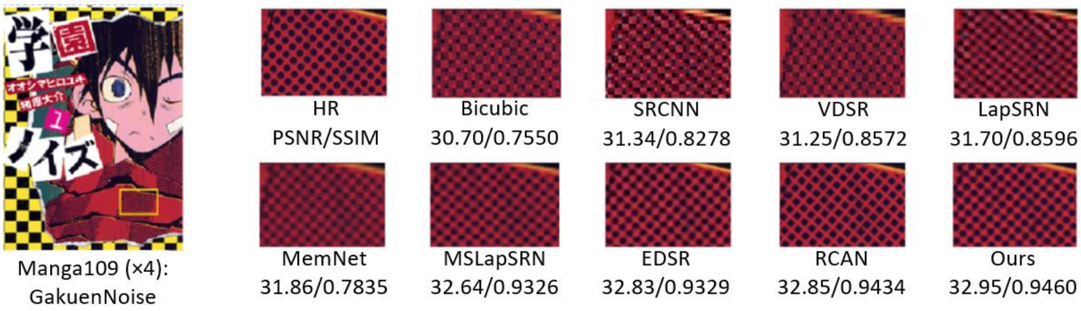


Figure 4.8: Visual comparison of the image "Gakuen Noise" on a scale of ×4 on the Manga109 dataset.

Figure 4.9 represents visual comparisons on a scale of ×4 for image “Img-012” belonging to the Urban100 [95] dataset. In contrast to the other SR models, our generator performs the parallel lines better while recovering the small details between these lines.

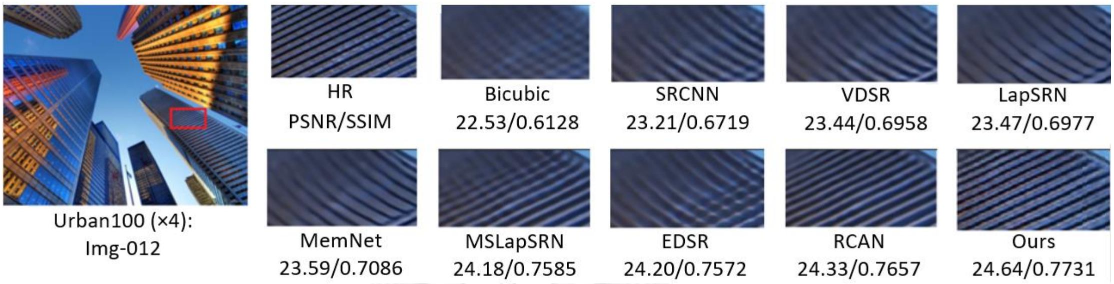


Figure 4.9: Visual comparison of the image "Img-012" on a scale of ×4 on the Urban100 dataset.

To further visual comparisons, we display the results on a scale of ×8. Figure 4.10 compares the results of other models of image “302008” belonging to the BSD100 [100] dataset. Due to the large-scale factor, the result of the Bicubic method’s result lost the tthe HR image's correct structure. Recovering the wrong structure because of a very large-scale factor also occurs in some other models such as SRCNN [21], VDSR [58], and LapSRN [30]. Our generator model demonstrates acceptable performance to recover the original structure of black lines compared to the other state-of-the-art models that suffer blurring artifacts, a lack of smoothness, and a weak capability to recover the tiny line connections.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

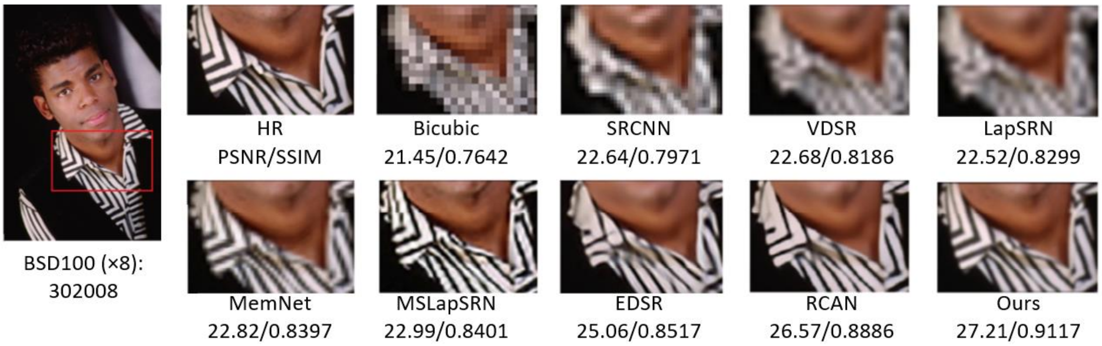


Figure 4.10: Visual comparison of the image "302008" on a scale of ×8 on the BSD100 dataset.

Figure 4.11 represents the image “KarappoHighschool” belonging to the Manga109 [101] dataset and the corresponding results of other SR models on a scale of ×8. Due to the large scale factor, the Bicubic, SRCNN [21], and VDSR [58] produce heavy blurring and wrong structure results. Although other SR models such as MemNet [22], EDSR [26], and RCAN [38] demonstrate better results compared to the Bicubic, SRCNN [20], and VDSR [58], they lose most high-frequency details that lead to the incorrect parallel line reconstruction.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

As demonstrated in Figure 4.11, both MSLapSRN [31] and our proposed generator model show robust performances to recover the most high-frequency details due to employing the progressively up-sampling framework with the projection approach. It indicates that the progressive up-sampling framework is more successful in recovering the most high-frequency information than the other SR technique such as the post-upsampling framework or the channel attention approach.

Besides the high-frequency recovering capability, the mixed L2 and MS-SSIM content loss help produce a sharper image, reducing the blurring artifacts and producing a better SR image compared to the other state-of-the-art models.

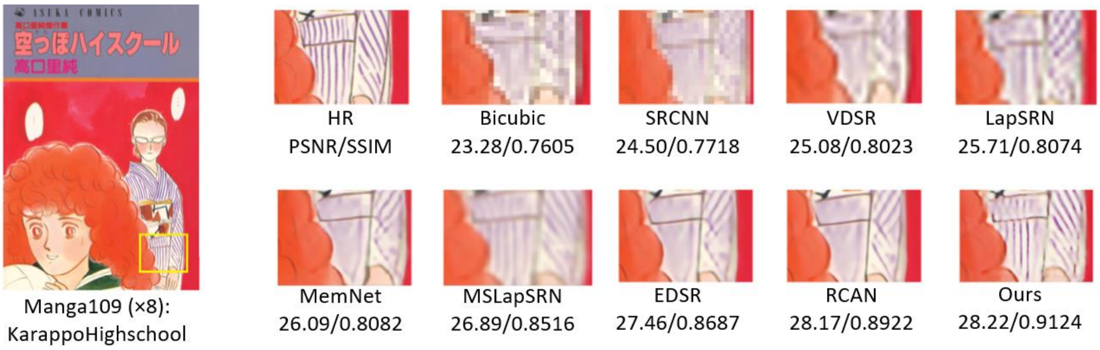


Figure 4.11: Visual comparison of the image "Karappo Highschool" on a scale of ×8 on the Manga109 dataset.

Figure 4.12 illustrates visual comparisons on a scale of ×8 for the image “Img-004” belonging to the Urban100 [95] dataset. Due to the sophisticated structure of this sample at a large-scale, most SR models have failed to recover the lattice structure of circular shapes. Although the RCAN [38] model successfully recovered the lattices texture, the original shape of the holes in the GT image is represented as a quadrangular shape. However, our generator module reconstructs circular lattice structure more faithfully than the other state-of-the-art methods.

จุฬาลงกรณ์มหาวิทยาลัย

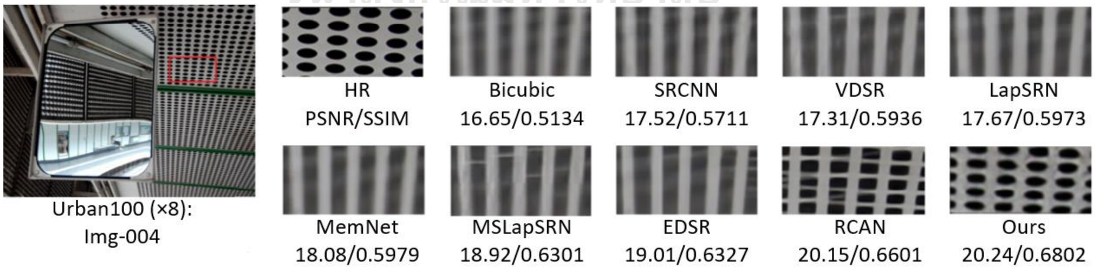


Figure 4.12: Visual comparison of the image "Img-004" on a scale of ×8 on the Urban100 dataset.

The quantitative comparison of the results by the PSNR (dB) and SSIM evaluations on scales of ×2, ×4, and ×8 on Set5 [98], Set14 [99], BSD100 [100], Urban100 [95] and Manga109 [101] datasets are presented in Table 4.4, Table 4.5, and Table 4.6. For the quantitative comparisons, we used eleven state-of-the-art models including Bicubic, SRCNN [21], FSRCNN [17], VDSR [58], LapSRN [30], MemNet [22], EDSR [26], SRMDNF [105], D-DBPN [33], RDN [44], DRLN [106] , RCAN [38] and LTE [107]. The results of other models are cited from their papers. The red numbers indicate the best performance, and the blue numbers demonstrate the second-best performance.

Table 4.4: Quantitative benchmark test results on a scale ×2.

Model	**Scale**	**Set 5**		**Set 14**		**BSD100**		**Urban100**		**Manga109**	
		PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Bicubic	× 2	33.66	0.9299	30.24	0.8688	29.56	0.8431	26.88	0.8403	30.80	0.9339
SRCNN [21]	× 2	36.66	0.9542	32.45	0.9067	31.36	0.8879	29.50	0.8946	35.60	0.9963
FSRCNN [17]	× 2	37.05	0.9560	32.66	0.9090	31.53	0.8920	29.88	0.9020	36.67	0.9710
VDSR [58]	× 2	37.53	0.9590	33.05	0.9130	31.90	0.8960	30.77	0.9140	37.22	0.9750
LapSRN [30]	× 2	37.52	0.9591	33.08	0.9130	31.08	0.8950	30.41	0.9101	37.27	0.9740
MemNet [22]	× 2	37.78	0.9697	33.28	0.9142	32.08	0.8978	31.31	0.9195	37.72	0.9740
EDSR [26]	× 2	38.11	0.9602	33.92	0.9195	32.32	0.9013	32.93	0.9351	39.10	0.9773
SRMDNF [105]	× 2	37.79	0.9601	33.32	0.9159	32.05	0.8985	31.33	0.9204	38.07	0.9761
D-DBPN [33]	× 2	38.09	0.9600	33.85	0.9190	32.27	0.9000	32.55	0.9324	38.89	0.9775
RDN [44]	× 2	38.24	0.9614	34.01	0.9212	32.34	0.9017	32.89	0.9353	39.18	0.9780
DRLN [106]	× 2	38.34	0.9619	34.43	0.9247	32.46	0.9032	33.54	0.9402	39.75	0.9792
RCAN [38]	× 2	38.33	0.9617	34.23	0.9225	32.46	0.9031	33.54	0.9399	39.61	0.9788
LTE [107]	× 2	38.33	--	34.25	--	32.44	--	33.50	--	--	--
Ours	× 2	37.81	0.9614	33.38	0.9168	32.16	0.8981	31.35	0.9214	38.18	0.9771

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Table 4.5: Quantitative benchmark test results on a scale ×4.

Model	**Scale**	**Set 5**		**Set 14**		**BSD100**		**Urban100**		**Manga109**	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	× 4	28.42	0.8104	26.00	0.7027	25.96	0.6675	23.14	0.6577	24.89	0.7866
SRCNN [21]	× 4	30.48	0.8628	27.50	0.7513	26.90	0.7101	24.52	0.7221	27.58	0.8555
FSRCNN [17]	× 4	30.72	0.8660	27.61	0.7550	26.98	0.7150	24.62	0.7280	27.90	0.8610
VDSR [58]	× 4	31.35	0.8830	28.02	0.7680	27.29	0.7260	25.18	0.7540	28.83	0.8870
LapSRN [30]	× 4	31.54	0.8850	28.19	0.7720	27.32	0.7270	25.21	0.7560	29.09	0.8900
MemNet [22]	× 4	31.74	0.8893	28.26	0.7723	27.40	0.7281	25.50	0.7630	29.42	0.8942
EDSR [26]	× 4	32.46	0.8968	28.80	0.7876	27.71	0.7420	26.64	0.8033	31.02	0.9148
SRMDNF [105]	× 4	31.96	0.8925	28.35	0.7787	27.49	0.7337	25.68	0.7731	30.09	0.9024
D-DBPN [33]	× 4	32.47	0.8980	28.82	0.7860	27.72	0.7400	26.38	0.7946	30.91	0.9137
RDN [44]	× 4	32.47	0.8990	28.81	0.7871	27.72	0.7419	26.61	0.8028	31.00	0.9151
DRLN [106]	× 4	32.74	0.9013	29.02	0.7914	27.86	0.7453	27.14	0.8149	31.78	0.9210
RCAN [38]	× 4	32.74	0.9013	28.98	0.7910	27.86	0.7455	27.10	0.8142	31.65	0.9208
LTE [107]	× 4	32.80	--	29.06	--	27.86	--	27.24	--	--	--
Ours	× 4	32.80	0.9032	29.30	0.7928	28.25	0.7532	27.22	0.8146	32.15	0.9210

Table 4.6: Quantitative benchmark test results on a scale of ×8.

Model	**Scale**	**Set 5**		**Set 14**		**BSD100**		**Urban100**		**Manga109**	
		PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM	PSNR	SSIM
Bicubic	× 8	24.40	0.6580	23.10	0.5660	23.67	0.5480	20.74	0.5160	21.47	0.6500
SRCNN [21]	× 8	25.33	0.6900	23.76	0.5910	24.13	0.5660	21.29	0.5440	22.46	0.6950
FSRCNN [17]	× 8	25.60	0.6970	24.00	0.5990	24.31	0.5720	21.45	0.5500	22.72	0.6920
VDSR [58]	× 8	25.93	0.7241	24.26	0.6148	24.49	0.5838	21.70	0.5710	23.16	0.7253
LapSRN [30]	× 8	26.15	0.7380	24.35	0.6200	24.54	0.5861	21.81	0.5810	23.39	0.7350
MemNet [22]	× 8	26.16	0.7410	24.38	0.6199	24.58	0.5840	21.89	0.5819	23.56	0.7380
EDSR [26]	× 8	26.97	0.7750	24.94	0.6399	24.80	0.5962	22.47	0.6220	24.56	0.7787
SRMDNF [105]	× 8	26.34	0.7558	24.57	0.6273	24.65	0.5895	22.06	0.5963	23.90	0.7564
D-DBPN [33]	× 8	26.96	0.7762	24.91	0.6420	24.81	0.5985	22.51	0.6221	24.69	0.7841
RDN [44]	× 8	27.21	0.7840	25.13	0.6480	24.88	0.6010	22.73	0.6312	25.14	0.7987
DRLN [106]	× 8	27.46	0.7916	25.40	0.6547	25.06	0.6070	23.24	0.6523	25.55	0.8087
RCAN [38]	× 8	27.47	0.7913	25.40	0.6553	25.06	0.6077	23.22	0.6524	25.58	0.8092
LTE [107]	× 8	27.35	--	25.42	--	25.03	--	23.17	--	--	--
Ours	× 8	27.54	0.7998	25.45	0.6598	25.17	0.6086	23.39	0.6865	25.86	0.8154

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

In contrast to scales ×4 and ×8, the results of our generator on a scale of ×2 are slightly less than RCAN [38] and RDN [44]. Since our generator has the progressive up-sampling framework, the generator on a scale of factor ×2 acts similar to a post-upsampling framework while it has less network depth compared to the other post-upsampling SR models at this scale.

Compared with other SR models on scales of ×4 and ×8, our generator indicates the best PSNR (dB) and SSIM results in all examined datasets. For example, on a scale of ×8 in the Urban100 [95] and Manga109 [101] datasets, our generator's PSNR gains 0.17 dB and 0.28 dB more than the second-best model (RCAN). In terms of SSIM evaluation in this example, our model also demonstrates improvement while the number of parameters in our SR model is less than the RCAN [38] model. These results show our progressive up-sampling framework with the proposed mixed perceptual loss (L2 + MS-SSIM) represents superior performance over the other SR models at larger scale factors (×4 and ×8).

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.4.4 Generator Network Size Analysis

The comparisons of our generator module size and the execution time are demonstrated in this section. For these comparisons, nine state-of-the-art models including SRCNN [21], FSRCNN [17], VDSR [58], LapSRN [30], MemNet [22], EDSR [26], D-DBPN [33], MDSR [26] and RCAN [38] are used. The implementations of these models have been done on GeForce GTX 1080 Ti GPU with 24GB of memory.

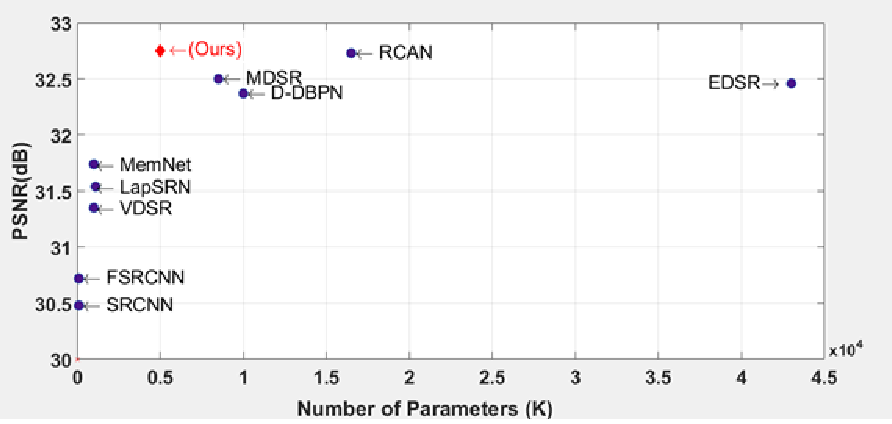


Figure 4.13: Performance and number of parameters evaluated on the Set5 dataset on a scale of ×4.

Figure 4.13 compares the performance and number of parameters on the Set5 [98] dataset on a scale of ×4. As observed in this graph, our generator model gains the highest PSNR (32.8 dB) while the number of parameters in our model (5 million) is less than RCAN [38] model as the second-best PSNR on this scale.

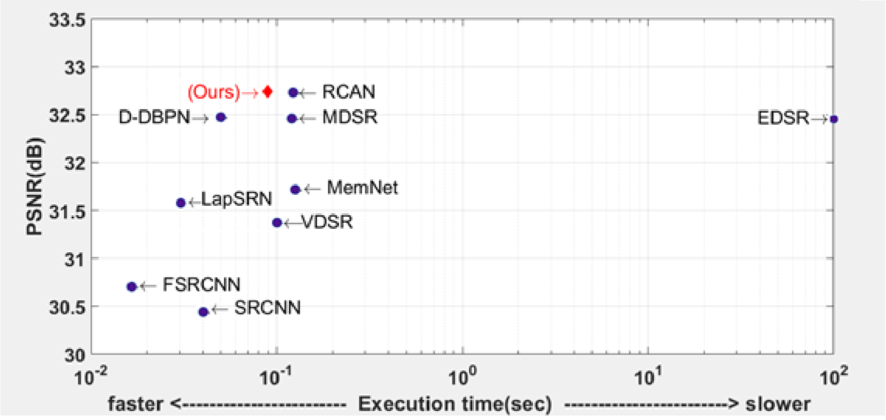


Figure 4.14: Performance and execution time evaluated on the Set5 dataset on a scale of ×4.

Figure 4.14 compares the performance and the execution time on the Set5 [98] dataset. According to this graph, our generator model gains the highest PSNR (32.8 dB) while its execution time is faster than EDSR [26], RCAN [38], MDSR [26], and MemNet [22].

4.5 Discriminator Module Performance

As discussed, our discriminator network is optimized with capsule structure [77] and operates alternately along with the generator to solve the adversarial min-max problem of our Caps-GAN model. Figure 4.15 demonstrates the discriminator loss of our model with capsule network and the discriminator with CNN-based models (CNN-based utilized in the SRGAN [9] and ESRGAN [29]) during training.

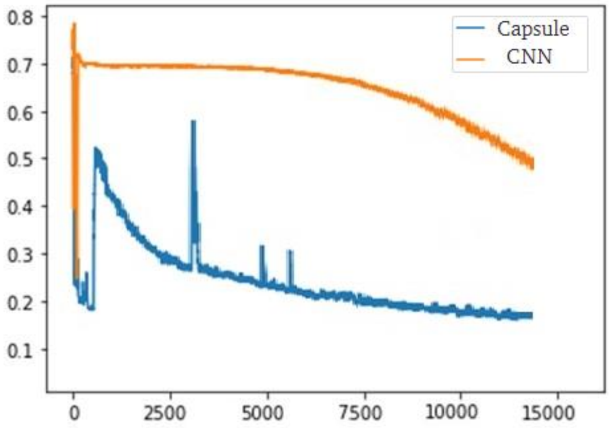


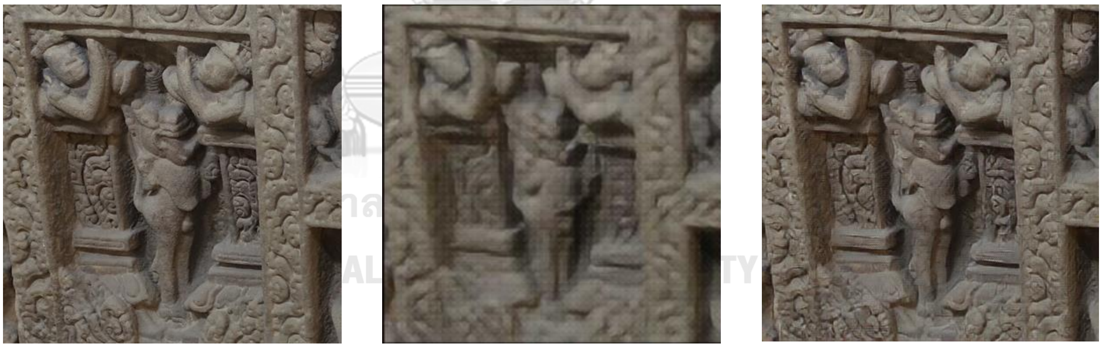
Figure 4.15: Discriminator loss during training model.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

The x-axis indicates the number of iterations (epochs), and the y-axis shows the loss. The orange graph is for the CNN-based discriminator while the blue color is for the proposed Caps-GAN. Due to the capability of capsule network architecture to extract the hierarchical feature relationships compared to the CNN concept, the capsule discriminator shows superior performance in extracting difficult-to-learn patterns in the training of the Caps-GAN model.

As demonstrated in Figure 4.15, the superior capability of our proposed capsule discriminator leads to training the discriminator module much faster in very less training iteration compared to the CNN-based discriminator. The blue graph of the proposed Caps-GAN indicates an anomaly at around 3,000 iterations and shows the sudden loss shoots up. This drastic fluctuation happened since the generator learned some trick in this range of iteration to fool the capsule discriminator. The capsule discriminator then starts to learn this trick and consequently the loss decreases. This sudden fluctuation indicates that our proposed GAN model plays the adversarial procedure very effectively in the training phase due to the applying a robust generator and robust discrimination in our Caps-GAN model.

Figure 4.16 and Figure 4.17 compare the results of our proposed model (same generator, objective functions, learning rate, training dataset, and training iterations) with the baseline CNN architecture of discriminator utilized in the SRGAN [9] and ESRGAN [29] on scales of ×8 and ×4.

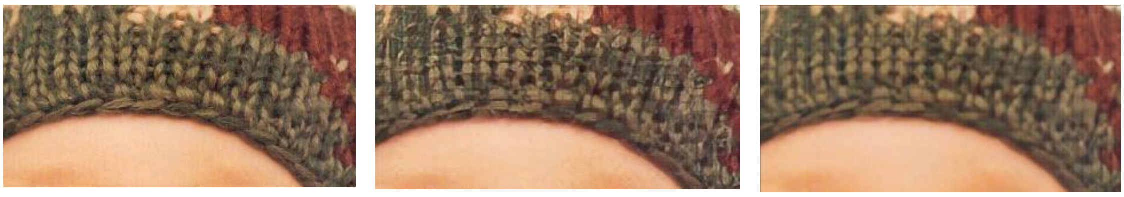


HR | CNN Discriminator | Capsule Discriminator

PSNR / SSIM | 16.51 / 0.5294 | 23.93 / 0.6992

Figure 4.16: Visual comparison of our GAN model with capsule and CNN discriminator on a scale of ×8.

HR | CNN Discriminator | Capsule Discriminator



PSNR / SSIM | 29.53 / 0.7906 | 29.79 / 0.8091

Figure 4.17: Visual comparison of our GAN model with capsule and CNN discriminator on a scale of ×4.

As Figure 4.16, the CNN discriminator cannot force our generator to produce acceptable SR results on a scale of ×8. In contrast, our proposed model with a Capsule discriminator network shows superior performance in training our GAN model and produces visually pleasing SR image at this scale.

Figure 4.17 also demonstrates the robustness of the capsule network discriminator over the CNN bades discriminator on a scale of ×4. As observed in Figure 4.17, the result of our proposed model with a capsule discriminator network demonstrates better performance in reconstructing sophisticated textures.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

4.6 Discussion of Caps-GAN Experimental Results

In this section, we compare the effectiveness of our Caps-GAN model with other state-of-the-art GAN models including the SRGAN [9] and ESRGAN [29].

First, the comparisons of the network complexity, execution time and the training iterations (training epochs) are demonstrated. Secondly, the quantitative comparisons on scales of ×2, ×4, and ×8 on the benchmark image datasets including Set5 [98], Set14 [99], BSD100 [100], Manga109 [101], Urban100 [95] and DIV2K [96] are displayed. Finally, the visual comparisons of our Caps-GAN model on scales of ×4, and ×8 on Set14 [99], BSD100 [100], Manga109 [101], Urban100 [95] and DIV2K [96] are shown.

4.6.1 Network Parameters, Execution Time, and Training Iterations

Table 4.7 demonstrates the network complexity, models' testing time, the training iterations of the SRGAN[9], ESRGAN [29] and our Caps-GAN model. The implementations of these models have been done on GeForce GTX 1080 Ti GPU with 24GB of memory.

Table 4.7: Comparison of execution time on the Set5 on a scale of ×4, network parameters, and training iterations of GAN-based super-resolution models.

Method	**Testing time (s)**	**Parameters**	**Training Epoch**
SRGAN [9]	2.63	1.790×10^7	50×10^3
ESRAGAN [29]	2.51	1.670×10^7	40×10^3
Caps-GAN (Ours)	2.23	4.983×10^6	10×10^3

According to Table 4.7, our Caps-GAN model demonstrates the best performance in all items. As discussed earlier, the progressive up-sampling framework helps to increase the deep learning network capability and to design our network with fewer parameters than the SRGAN [9] and ESRGAN [29] post up-sampling framework. Consequently, the execution time of our network with fewer parameters is decreased compared to the SRGAN [9] and ESRGAN [29] models. As discussed in the previous section, the capability of the Capsule network in our discriminator architecture has led to converging our Caps-GAN model very fast compared to the SRGAN [9] and ESRGAN [29] which utilized CNN-based discriminator. Therefore, the training iteration of our Caps-GAN model is decreased dramatically compared to the other GAN models.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 4.18 and Figure 4.19 demonstrate the comparison graphs that compare the number of parameters and the execution time of our Caps-GAN model via the performance (PSNR dB) on the Set5 dataset on a scale of ×4.

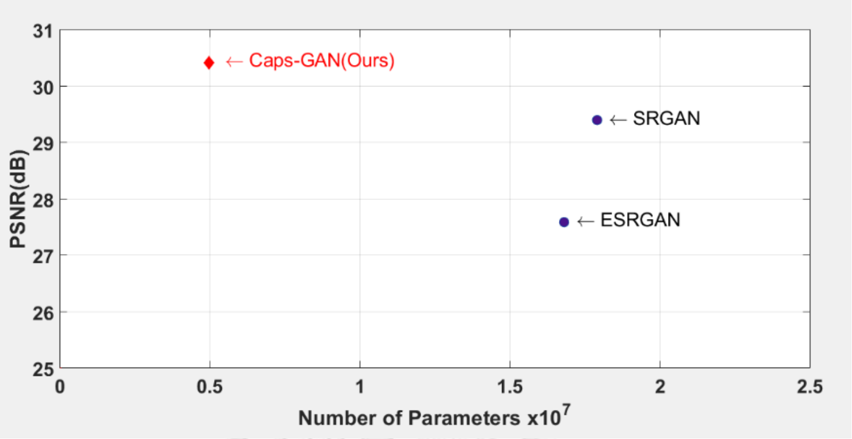


Figure 4.18: The GAN-based models' performances and parameters evaluated on the Set5 dataset on a scale of ×4.

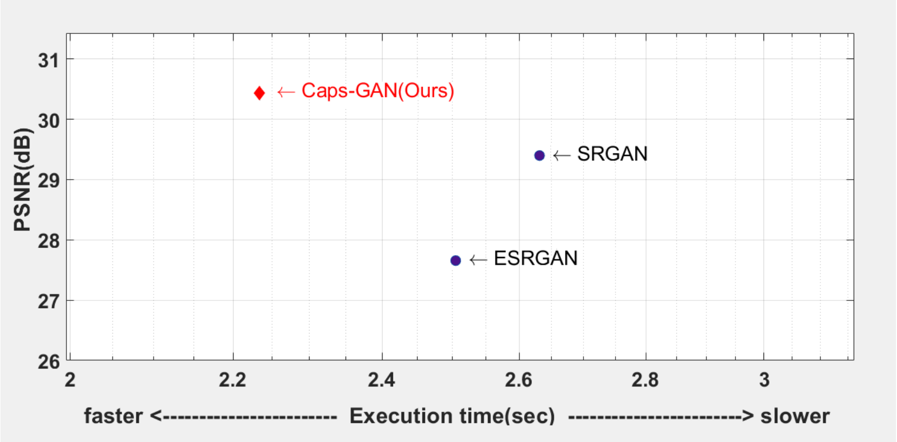


Figure 4.19: The GAN-based models' performances and execution time evaluated on the Set5 dataset on a scale of ×4.

Figure 4.18 and Figure 4.19 demonstrate our Caps-GAN model outperformance in terms of the PSNR and network complexity, as well as the PSNR and execution time compared to the other GAN models.

4.6.2 Quantitative Comparison of Caps-GAN model

The quantitative comparison of the proposed Caps-GAN by PSNR (dB) and SSIM evaluations on scales of ×2, ×4, and ×8 is demonstrated in Table 4.8, Table 4.9, and Table 4.10. For the quantitative comparisons, we used state-of-the-art GAN-based models, including the SRGAN [9] and ESRGAN [29]. The benchmark datasets including Set5 [98], Set14 [99], BSD100 [100], Manga109 [101], Urban100 [95] and DIV2K [96] are also utilized for comparison.

Table 4.8: Quantitative benchmark GAN results on a scale ×2.

Datasets	**Scale**	**SRGAN [9]**		**ESRGAN [29]**		**Caps-GAN(Ours)**	
		PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Set5 [98]	×2	33.27	0.8937	33.94	0.9145	**34.02**	**0.9152**
Set14 [99]	×2	32.14	0.8860	33.62	0.9150	**33.64**	**0.9155**
BSD100 [100]	×2	31.89	0.8760	31.98	0.8870	**31.99**	**0.8879**
Urban100 [95]	×2	27.66	0.8432	27.70	0.8441	**27.76**	**0.8450**
Manga109 [101]	×2	30.92	0.8956	30.98	0.8961	**30.98**	**0.8973**
DIV2K [96]	×2	25.08	0.7007	25.97	0.7801	**26.00**	**0.7809**

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

Table 4.9: Quantitative benchmark GAN results on a scale of ×4.

Datasets	**Scale**	**SRGAN [9]**		**ESRGAN [29]**		**Caps-GAN(Ours)**	
		PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Set5 [98]	×4	29.40	0.8472	27.59	0.7978	**30.41**	**0.8492**
Set14 [99]	×4	26.02	0.7379	30.50	0.7620	**31.18**	**0.7689**
BSD100 [100]	×4	25.16	0.6688	27.69	0.7120	**28.72**	**0.7197**
Urban100[95]	×4	27.16	0.8763	28.49	0.8851	**29.08**	**0.8890**
Manga109[101]	×4	27.39	0.8666	27.77	0.8709	**28.11**	**0.8754**
DIV2K [96]	×4	28.09	0.8210	28.68	0.8530	**28.98**	**0.8609**

Table 4.10:Quantitative benchmark GAN results on a scale of ×8.

Datasets	**Scale**	**SRGAN [9]**		**ESRGAN [29]**		**Caps-GAN(Ours)**	
		PSNR	**SSIM**	**PSNR**	**SSIM**	**PSNR**	**SSIM**
Set5 [98]	×8	---	---	---	---	**26.56**	**0.7696**
Set14 [99]	×8	---	---	---	---	**24.61**	**0.6388**
BSD100 [100]	×8	---	---	---	---	**24.75**	**0.5856**
Urban100[95]	×8	---	---	---	---	**22.36**	**0.6115**
Manga109[101]	×8	---	---	---	---	**24.78**	**0.7802**
DIV2K [96]	×8	---	---	---	---	**25.09**	**0.7839**

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

As observed in these tables, the PSNR and SSIM results of our Caps-GAN model are improved on scales of ×2 and ×4 compared to the other GAN-based models. This improvement in the PSNR and SSIM is related to our designed content loss (L^{Mix}) which combined MS-SSIM loss and L2 loss. The effectiveness of this combined loss function in terms of quantitative and qualitative performances was demonstrated and proven in the previous section. The results of the SRGAN and ESRGAN on a scale of ×2 are cited from research papers.

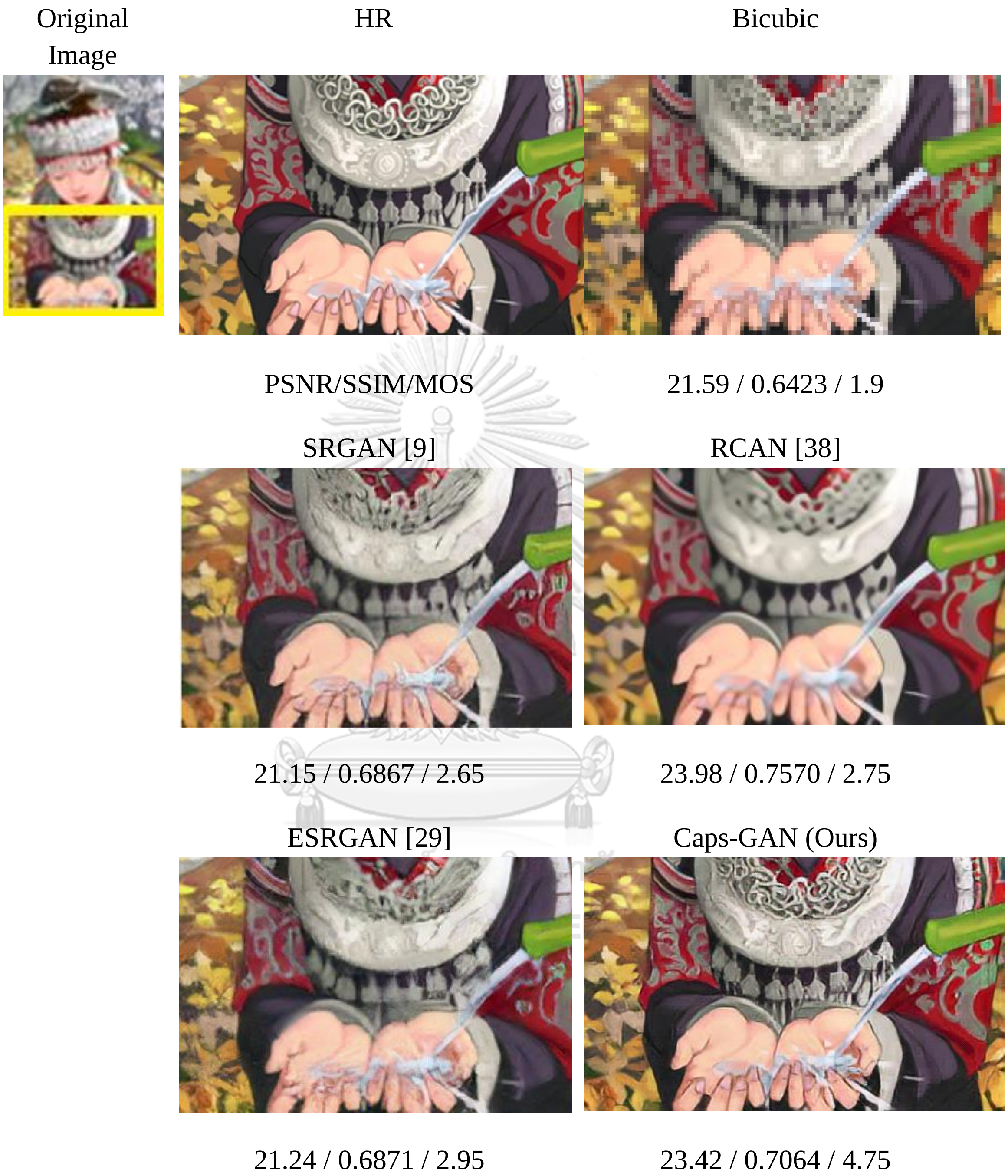
As shown in Table 4.9, the SRGAN [9] and ESRGAN [29] models cannot operates on a scale of ×8. It may happen because of the weakness of CNN-based discriminators in their GAN models to maintain effective training to reconstruct the SR images on a scale of ×8.

4.6.3 Visual Comparison of Caps-GAN model

In visual comparison, we compare the results of our Caps-GAN model on scales of ×4 and ×8 with other state-of-the-art models. The benchmark image datasets including Set14 [99], BSD100 [100], Manga109 [101], Urban100 [95] and DIV2K [96] are used.

Figure 4.20, Figure 4.21, Figure 4.22, Figure 4.23, and Figure 4.24 demonstrate the visual comparison of our Caps-GAN model with GAN-based models of the SRGAN [9] and ESRGAN [29] on a scale of ×4. Additionally, to visualize the differences between non-GAN and GAN models, the result of RCAN [38], as the best non-GAN model, is also displayed. Moreover, each sample image's PSNR, SSIM, and MOS evaluations are mentioned.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



Original Image

HR

Bicubic

PSNR/SSIM/MOS

21.59 / 0.6423 / 1.9

SRGAN [9]

RCAN [38]

21.15 / 0.6867 / 2.65

23.98 / 0.7570 / 2.75

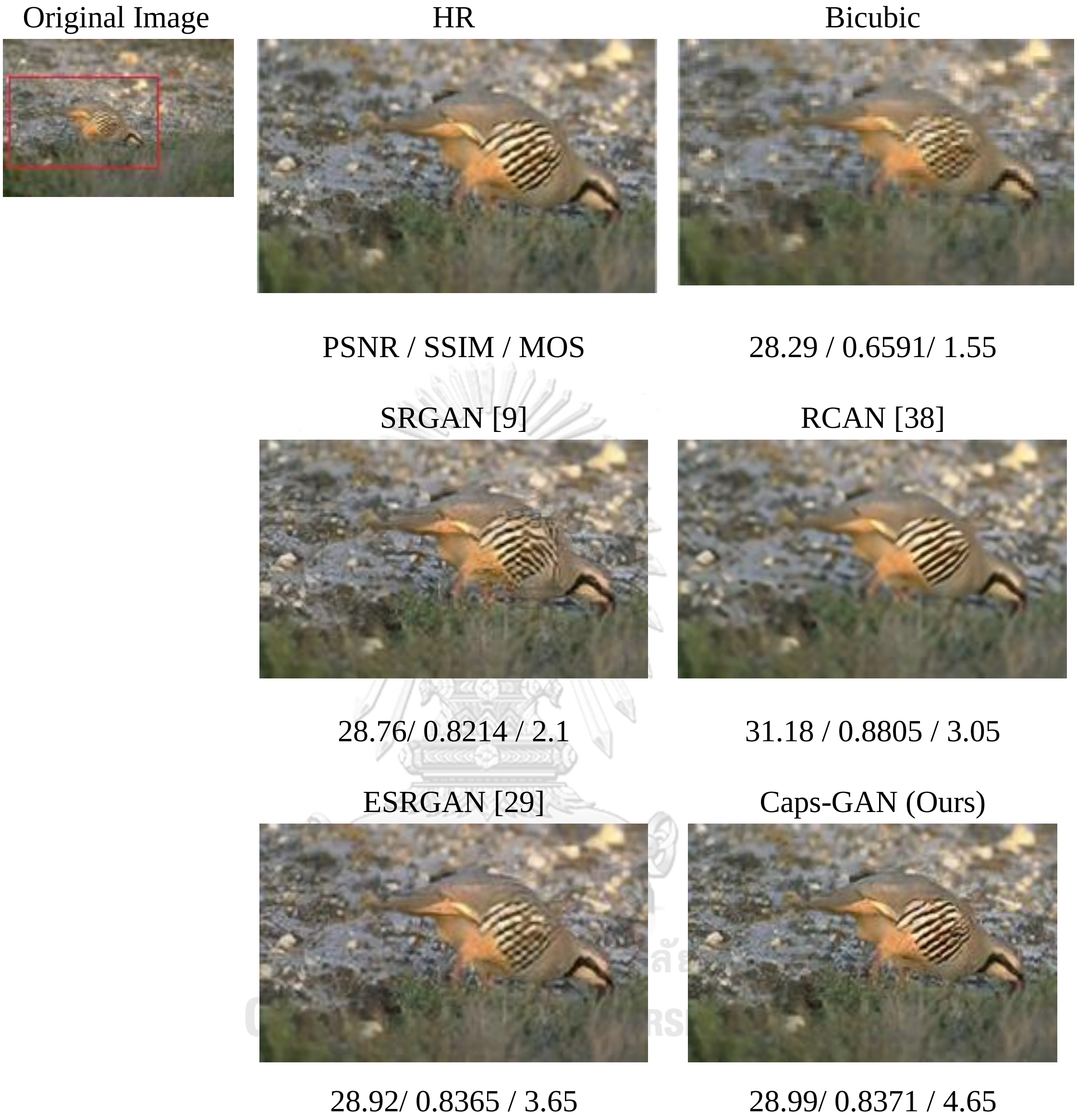
ESRGAN [29]

Caps-GAN (Ours)

21.24 / 0.6871 / 2.95

23.42 / 0.7064 / 4.75

Figure 4.20: Visual comparison of the image " Comic " on a scale of ×4 on the Set14 dataset.



Original Image

HR

Bicubic

PSNR / SSIM / MOS

28.29 / 0.6591/ 1.55

SRGAN [9]

RCAN [38]

28.76/ 0.8214 / 2.1

31.18 / 0.8805 / 3.05

ESRGAN [29]

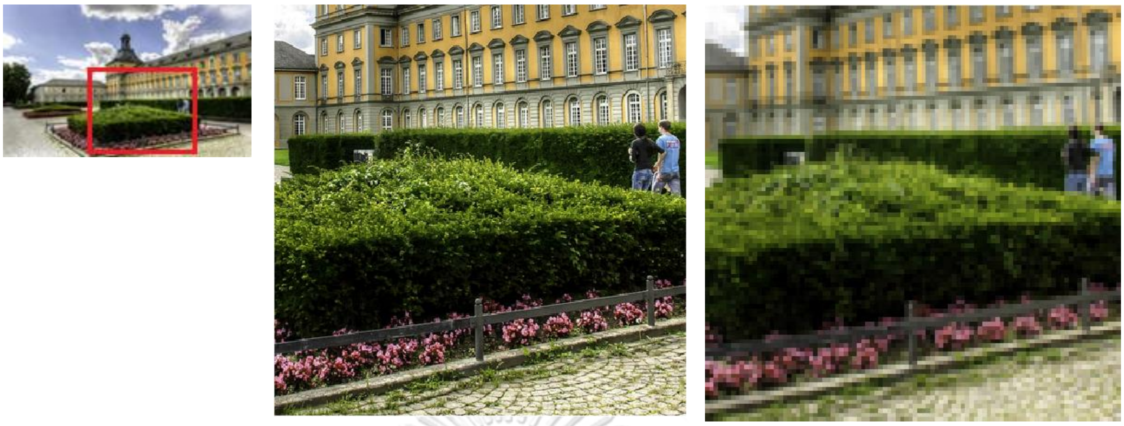
Caps-GAN (Ours)

28.92/ 0.8365 / 3.65

28.99/ 0.8371 / 4.65

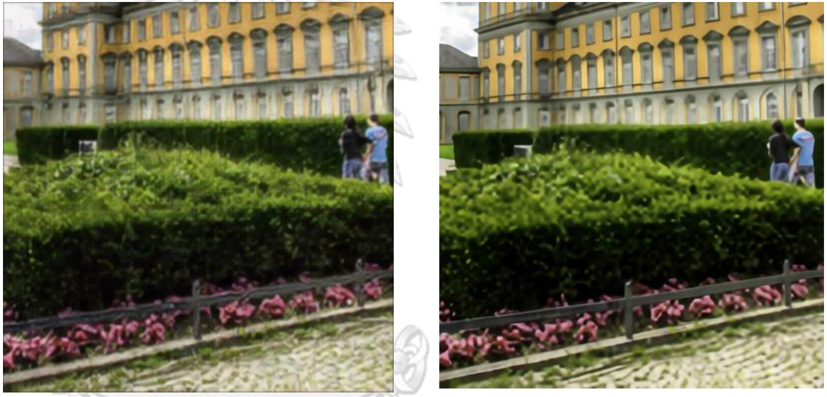
Figure 4.21: Visual comparison of the image "8023 " on a scale of ×4 on the BSD100 dataset.

Original Image HR Bicubic



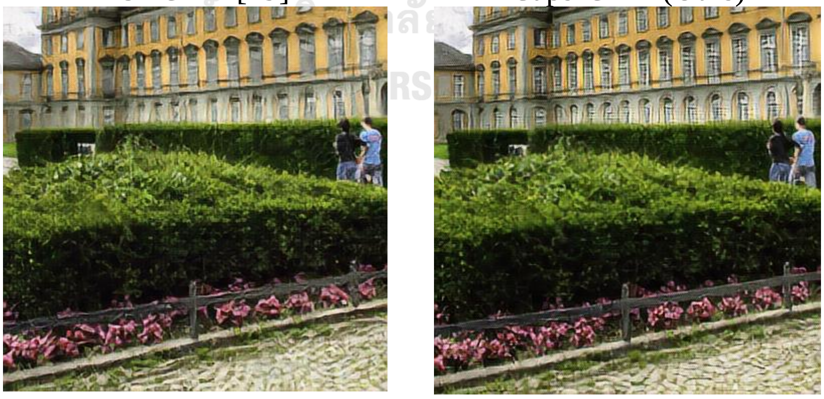
PSNR / SSIM / MOS 18.86 / 0.5885 / 1.25

SRGAN [9] RCAN [38]



19.09/ 0.6002 / 2.30 22.91 / 0.6924/ 3.65

ESRGAN [29] Caps-GAN (Ours)

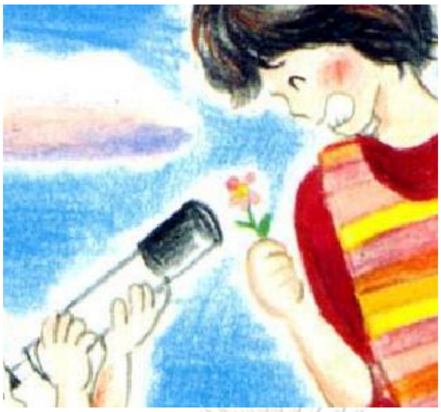


19.47 / 0.6390 / 3.25 19.90 / 0.6430 / 4.55

Figure 4.22: Visual comparison of the image "img_077" on a scale of ×4 on the Urban100 dataset.

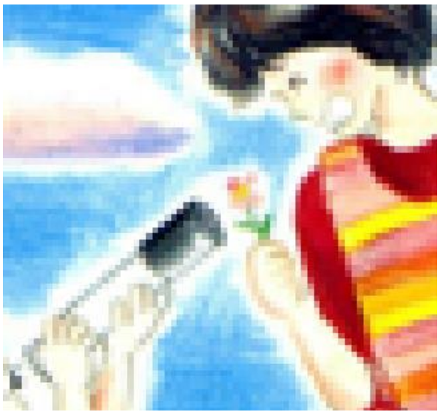
Original Image

HR



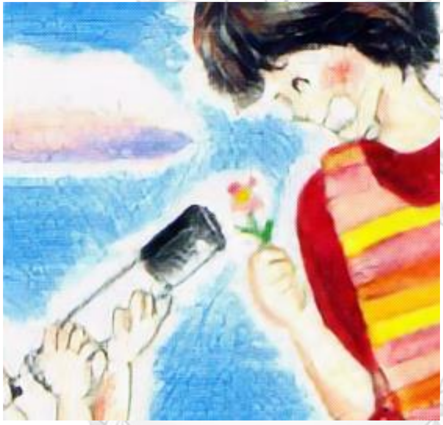
PSNR / SSIM / MOS

Bicubic



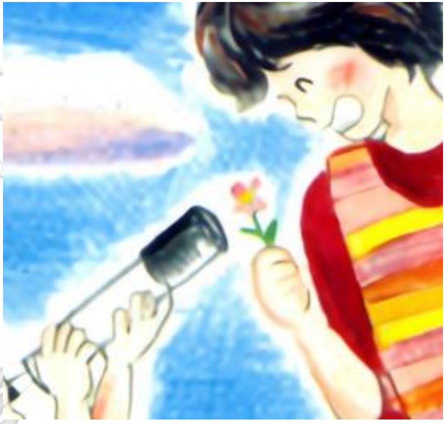
23.93 / 0.6951 / 1.05

SRGAN [8]



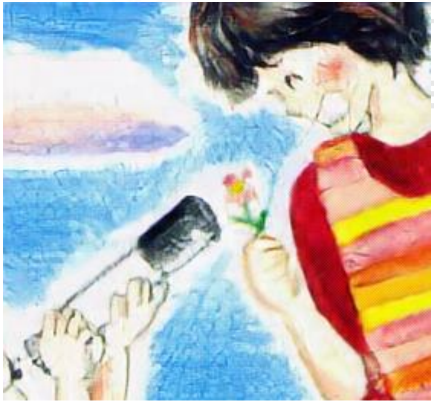
27.09 / 0.7206 / 2.9

RCAN [37]



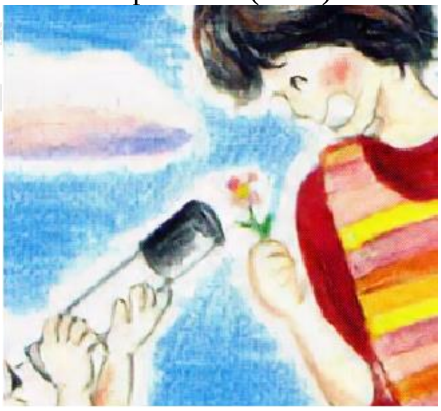
32.28/ 0.8212 / 2.55

ESRGAN [28]



27.12 / 0.7215 / 3.7

Caps-GAN (Ours)



29.60 / 0.7512 / 4.8

Figure 4.23: Visual comparison of the image "Aisazu NihaIrarenai " on a scale of ×4 on the Manga109 dataset.

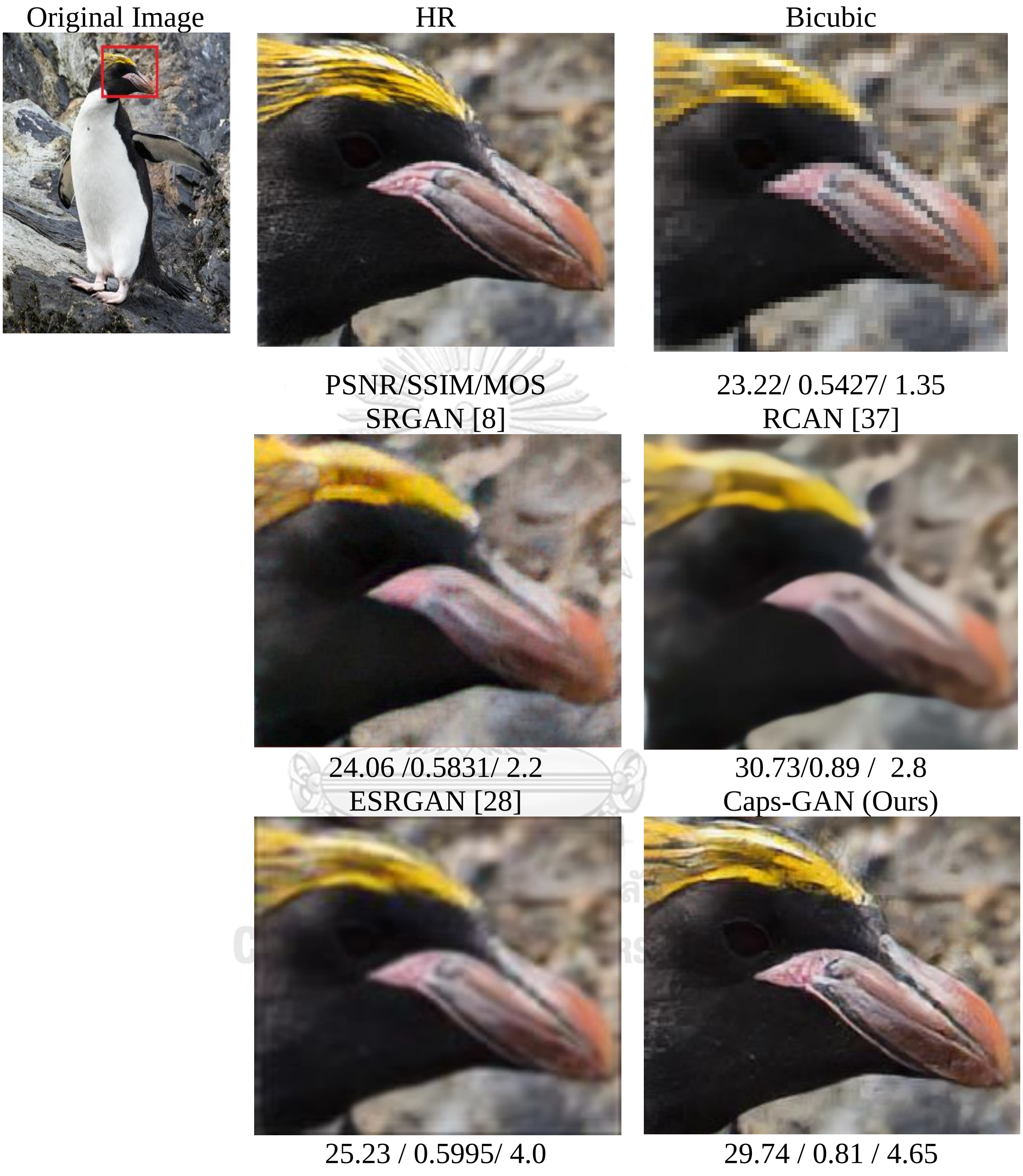


Figure 4.24: Visual comparison of the image "0801 " on a scale of ×4 on the DIV2K dataset.

Figure 4.20 represents visual comparisons on a scale of ×4 for the image “Comic” belonging to the Set14 [99] dataset. In contrast to the other models, our Caps-GAN shows superior performance in reconstructing the sophisticated details of the image. Although the PSNR and SSIM results of the GAN-based models are less than the non-GAN (RCAN), our Caps-GAN model improved the PSNR and SSIM and got the best MOS results among other models.

Figure 4.21 visually compared the image "8023" belonging to the BSD100 [100] dataset. As observed in Figure, our Caps-GAN successfully reconstructed the parallel line details of the bird compared to the SRGAN [9] and ESRGAN [29], while the background details were reconstructed naturally and very close to the HR image without any over-smoothing effects.

Figure 4.22 visually compares the image "img_077" belonging to the Urban100 [95] datasets. The results of other models show weakness in reconstructing the building details. Our Caps-GAN model successfully recovered the building details while preventing the blurring and over-smoothing effects in the entire parts of the image. Therefore, the highest MOS result belongs to our model.

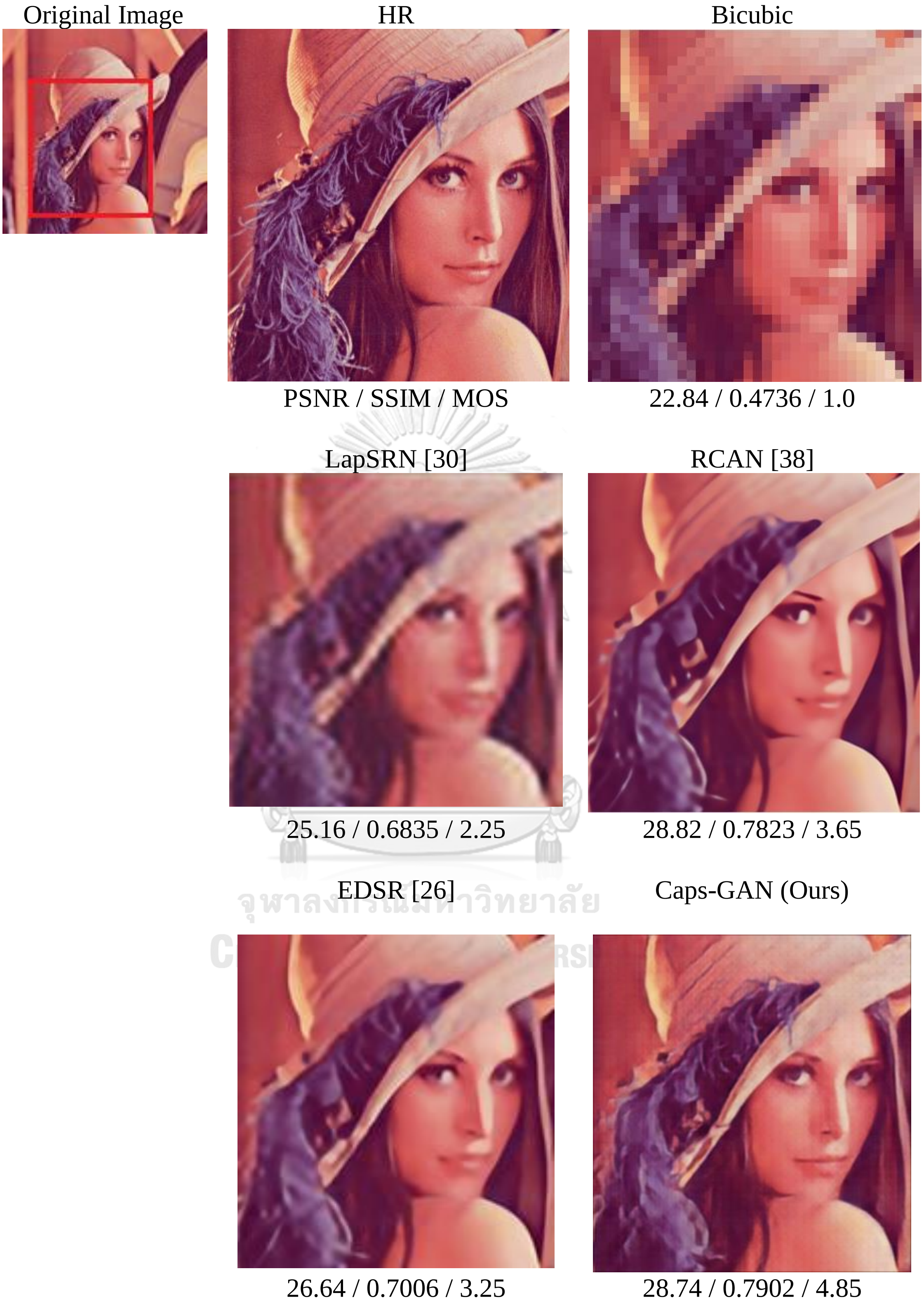
จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 4.23 indicates visual comparisons of the image "Aisazu NihaIrarenai" belonging to the Manga109 [101] dataset. Since this image is considered a simple structure to reconstruct on a scale of ×4, all produced results are acceptable in terms of the overall SR image structure. However, the most challenging part of this image is reconstructing the texture of the image. The effect of drawing on paper with a pencil made a special texture to this image. The RCAN model produces an over-smooth image that blurs the texture of the image. The adversarial loss of all GAN models aids in producing natural texture of the image. In contrast, the SRGAN and ESRGAN model are not successful in recover the correct texture, especially around the face region of the image. Our Caps-GAN got the best MOS result and produced the best texture compared to the other models.

Figure 4.24 represents visual comparisons of the image "0801" belonging to the DIV2K [96] dataset. As observed, the result of Caps-GAN outperforms in reconstructing the SR image very naturally and is similar to the GT image. It got 4.65 in the MOS test which is the highest among all models. The PSNR and SSIM also improved compared to other GAN-based models.

Figure 4.25, Figure 4.26, Figure 4.27, Figure 4.28, and Figure 4.29 show the visual comparison of our Caps-GAN model with other state-of-the-art models on a scale of ×8. Since the SRGAN [9] and ESRGAN do not have scale ×8 models, for visual comparison at this scale we demonstrate the EDSR [26], LapSRN [30], and RCAN [38] results on a scale of ×8. Due to outperforming the RCAN [38] model in terms of the PSNR and SSIM, the robustness of the SR image reconstruction capability of the EDSR [26] model, and the progressive up-sampling framework of the LapSRN [30] model, these models were selected for comparison. Moreover, each sample image’s PSNR, SSIM, and MOS evaluations are mentioned.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University



Original Image

HR

PSNR / SSIM / MOS

Bicubic

22.84 / 0.4736 / 1.0

LapSRN [30]

25.16 / 0.6835 / 2.25

RCAN [38]

28.82 / 0.7823 / 3.65

EDSR [26]

26.64 / 0.7006 / 3.25

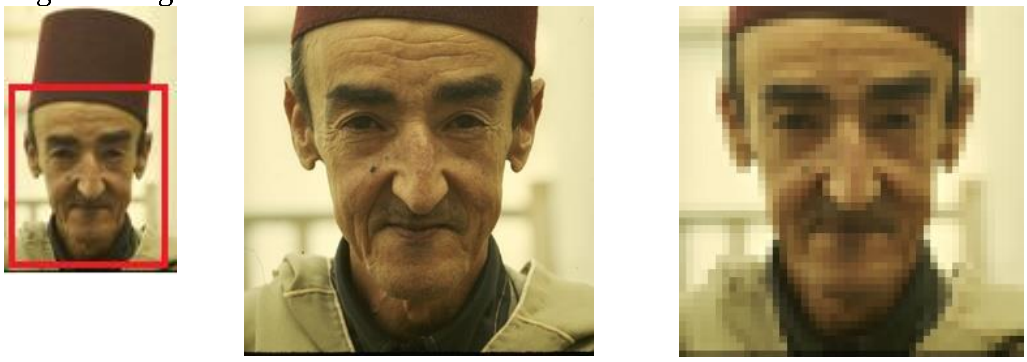
Caps-GAN (Ours)

28.74 / 0.7902 / 4.85

Figure 4.25: Visual comparison of the image "Lenna" on a scale of ×8 on the Set14 dataset.

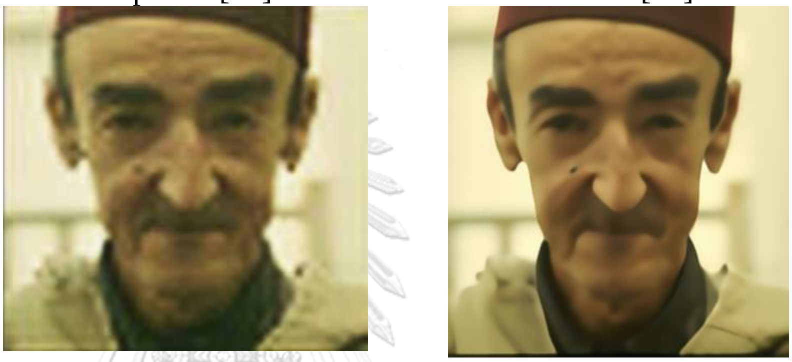
จุฬาลงกรณ์มหาวิทยาลัย

Original Image HR Bicubic



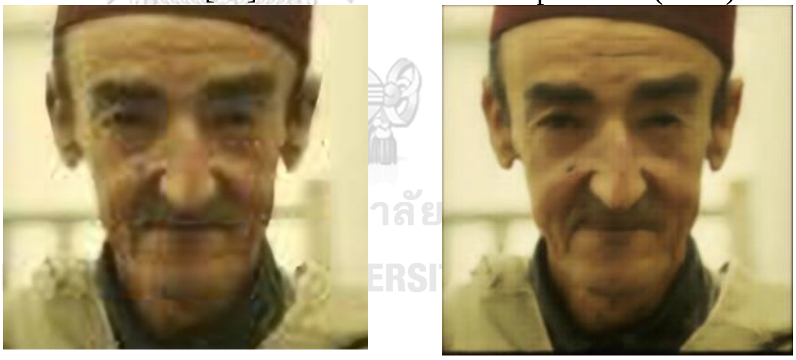
PSNR /SSIM / MOS 26.91 / 0.6901 / 1.4

LapSRN [30] RCAN [38]



28.76 / 0.7837 / 3.25 31.91 / 0.8379 / 3.95

EDSR [26] Caps-GAN (Ours)



27.92 / 0.7180 / 1.6 30.99 / 0.8276 / 4.8

Figure 4.26: Visual comparison of the image "189080 " on a scale of ×8 on the BSD100 dataset.

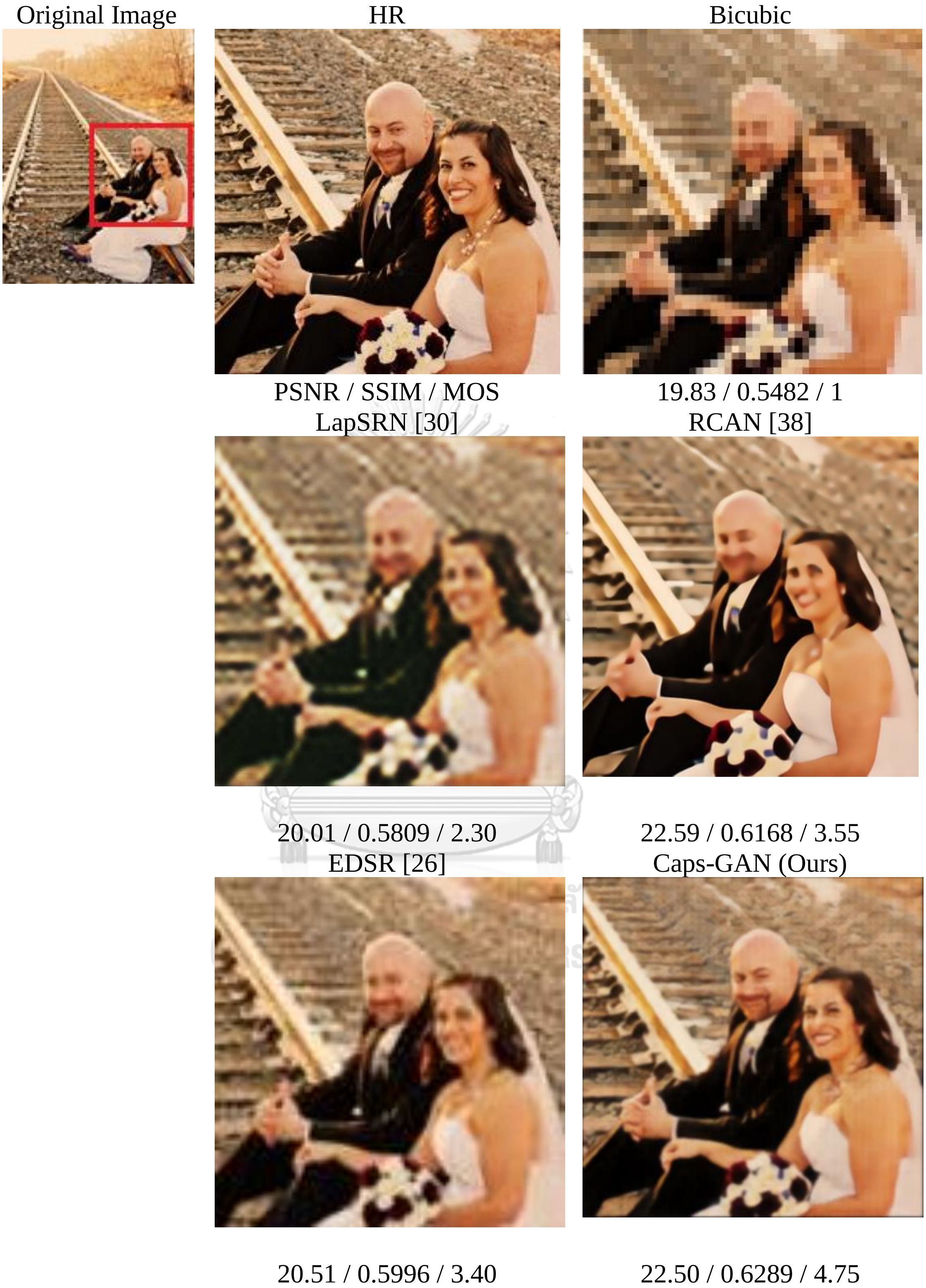


Figure 4.27: Visual comparison of the image "img_037 " on a scale of ×8 on the Urban100 dataset.

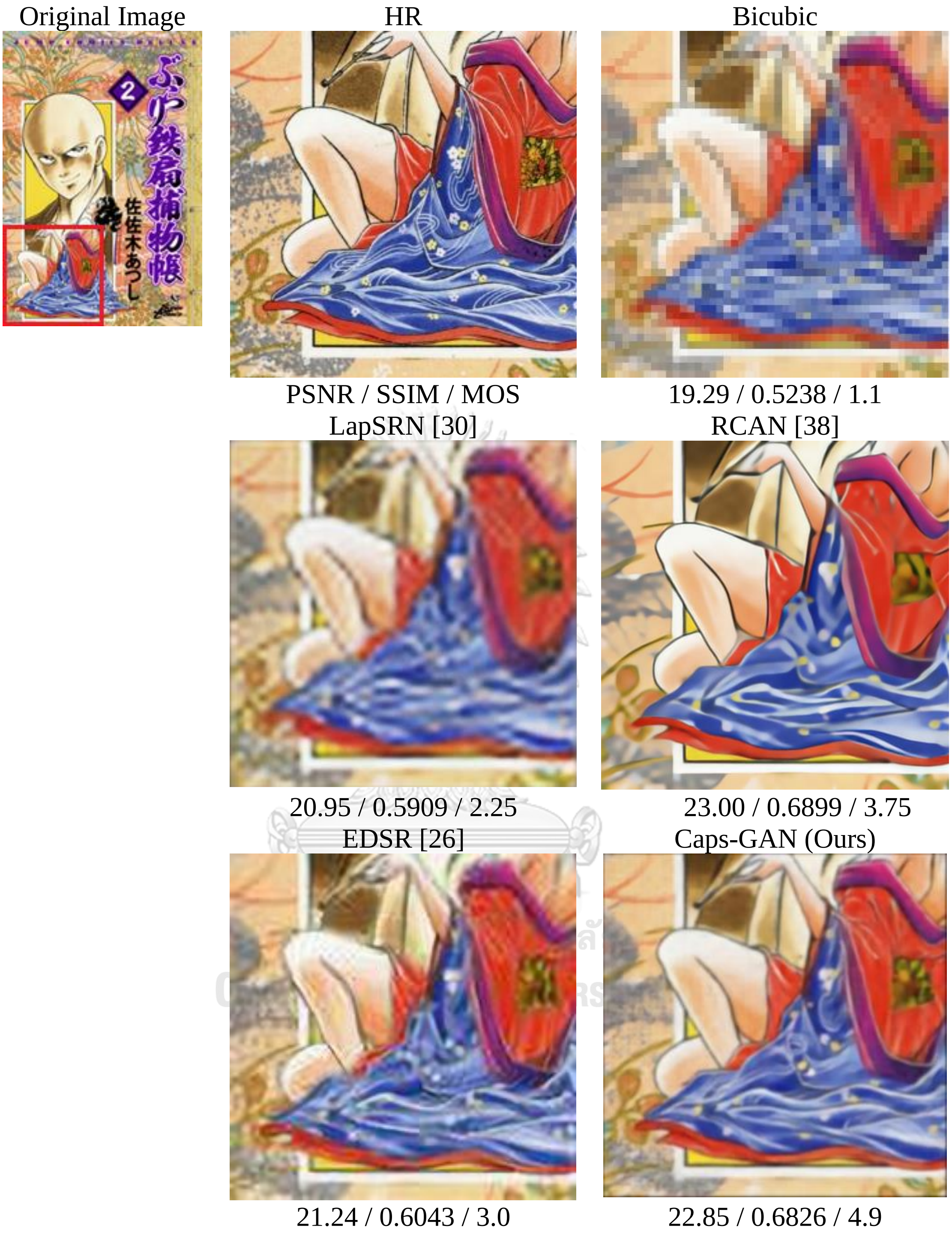
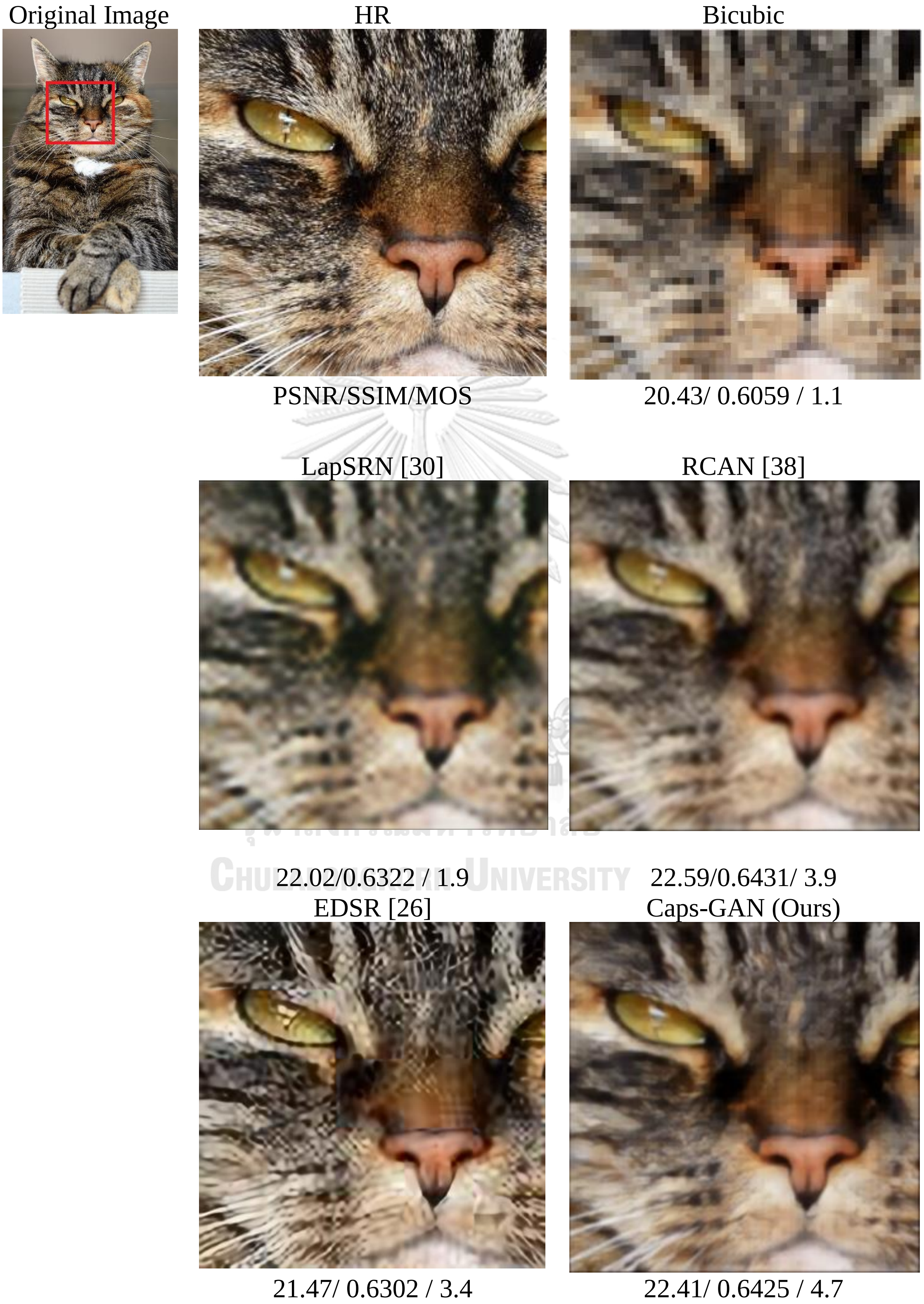


Figure 4.28: Visual comparison of the image "BurariTessen" on a scale of ×8 on the Manga109 dataset.



Original Image

HR

Bicubic

PSNR/SSIM/MOS

20.43/ 0.6059 / 1.1

LapSRN [30]

RCAN [38]

22.02/0.6322 / 1.9

22.59/0.6431/ 3.9

EDSR [26]

Caps-GAN (Ours)

21.47/ 0.6302 / 3.4

22.41/ 0.6425 / 4.7

Figure 4.29: Visual comparison of the image "0869" on a scale of ×8 on the DIV2K dataset.

Figure 4.25 represents visual comparisons on a scale of ×8 for the image “Lenna” belonging to the Set14 [99] dataset. The results of both RCAN [38] and EDSR [26] models suffer over-smoothing problems and show weakness in recovering the texture. The LapSRN [30] attempts to highlight the details due to the progressive up-sampling framework but the lack of high-frequency recovering capability leads to an unpleasant result. While our Caps-GAN model shows better results because of utilizing adversarial loss to recover the textural information without any blurring effects of face and background on this large scale.

Figure 4.26 visually compares the image "189080" belonging to the BSD100 [100] dataset. According to this Figure, our Caps-GAN model produced a smoother image while recovering the textual information of the face. Other models show weaknesses in their results due to over smoothing or noise amplification artifacts. The highest MOS rate also belongs to our results.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

Figure 4.27 shows the visual comparisons of the image "img_037" belonging to the Urban100 [95] datasets. Reconstructing the SR images at this scale is considered challenging due to the difficulty of recovering various textual information such as face and the background. Our Caps-GAN model demonstrates the high capability to reconstruct the SR image because of utilizing adversarial loss and an effective content loss in a progressive up-sampling framework. The highest MOS result belongs to our model.

Figure 4.28 visually compares the image "BurariTessen" belonging to the Manga109 [101] dataset. According to the SR results of these models, the Caps-GAN model generates a better SR image that preserves more small details without any noise and blurring artifacts.

Figure 4.29 represents visual comparisons of the image "0869" belonging to the DIV2K [96] dataset. The EDSR [26] model shows noise amplification artifacts in recovering the texture. The RCAN [38] and LapSRN [30] demonstrate blur results. Our Caps-GAN model produced a sharper image compared to the LapSRN [30] and RCAN models while preserving the perceptual quality of the SR image against noise amplification. This capability is yielded with a successful combination of adversarial and our mixed perceptual loss. The MOS test also indicates the outperformance of our model.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

CHAPTER FIVE
Conclusions and Future Works

5.1 Introduction

Recent decades have seen an increasing interest in utilizing videos and images in numerous applications. The image's resolution in any image-based application is considered the most important factor which directly affects the visual quality of an image. Due to the limitations of the increasing number of sensors in the image acquisition modules, the importance of increasing the spatial resolution of images via software algorithms in real-world applications is undeniable. Various super-resolution algorithms have been proposed to produce a visually pleasing upscale image. However, producing a high-resolution image that contains tiny details while executing in a short time has remained a challenge.

We proposed our Caps-GAN algorithm to improve the super-resolution algorithm and the reconstructed high-resolution image. The idea behind the Caps-GAN model is to upscale the LR image using the generative adversarial approach which consists of two main modules: generator and discriminator. These two independent modules are trained individually but in an adversarial cycle. Particularly, the generator module is trained to generate more realistic and accurate upsampled images. In contrast, the discriminator is trained for better distinguishment between real and fake sample data generated by the generator module.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

To improvement of both discriminator and generator modules is inevitable to enhance the GAN-based super-resolution model while having an efficient and fast algorithm. Based on these modification concepts, the capsule network approach is utilized in the discriminator module. Due to the ability of the capsule network to understand the hierarchical pattern relationships and learn the difficult-to-learn patterns compared to the CNN model, our proposed discriminator utilized this approach.

Since the generator module is responsible for generating the reconstructed HR images, it should be modified to produce a sophisticated image similar to the GT image while having less network complexity. Our generator module utilizes a lightweight architecture that applies a progressive up-sampling framework that performs fewer convolution layers in a low-dimensional space and progressively applies the predefined up-sampling operations. The residual in residual dense blocks (RRDB) is utilized for the first stage while the second stage utilizes the RDB architecture. To transfer the high-frequency details of the early network layer to the up-sampling modules and prevent gradient vanishing problems in the training of the proposed model, at the end of each up-sampling stage the depth-wise bottleneck projections approach is used.

In addition to the generator and discriminator modifications in our Caps-GAN model, perceptual loss plays a critical role in generating a better perceptual quality. The perceptual loss consists of content loss and adversarial loss. Due to the importance of the content loss in producing the overall appearance of the SR image and its contribution to improving quantitative results, the combination of the multiscale SSIM loss and L2 loss was utilized in our model. Additionally, the adversarial loss of the GAN model helps reconstruct the tiny high-frequency details of the texture and produce a realistic result similar to the GT image.

Chulalongkorn University

5.2 Contributions

The fundamental objective of our study is to investigate and propose a Caps-GAN (fusing) Single Image Super-Resolution Algorithm designed based on the GAN network while utilizing the Capsule Network Algorithm in the discriminator part of the GAN, to optimize the training procedure.

Besides, the study seeks to enhance the perceptual quality of the Caps-GAN results by introducing a combination of the MS-SSIM and L2 loss functions (content loss) and producing more perceptual satisfaction and realistic SR images, as well as improving the PSNR and SSIM of the previous GAN-based models.

The extensive experimental evaluations of our Caps-GAN model on six benchmark image datasets on 3 different scales proved that our SR model obtained a state-of-the-art accuracy in terms of the PSNR and SSIM as well as the perceptual satisfaction, and MOS.

5.3 Future Work

In future work, we will validate our Caps-GAN model on a single video frame and a video sequence due to the requirements of streaming services to provide high-quality videos from the low-resolution stored videos.

Furthermore, we will explore the capability of our Caps-GAN model in image deblurring, denoising, and inpainting.

In conjunction with our approach, we will finally examine our model's potential in the super-resolution phase of a face detection algorithm.

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

REFERENCES

[1] W. Jifara, F. Jiang, S. Rho, M. Cheng, and S. Liu, "Medical image denoising using convolutional neural network: a residual learning approach," *The Journal of Supercomputing,* vol. 75, no. 2, pp. 704-718, 2019.

[2] M. Abadi *et al.*, "Tensorflow: Large-scale machine learning on heterogeneous distributed systems," *arXiv preprint arXiv:1603.04467,* 2016.

[3] T. Tanprasert, C. Siripanpornchana, N. Surasvadi, and S. Thajchayapong, "Recognizing traffic black spots from street view images using environment-aware image processing and neural network," *IEEE Access,* vol. 8, pp. 121469-121478, 2020.

[4] T. Isobe *et al.*, "Video super-resolution with temporal group attention," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 8008-8017.

[5] S. Shakya, S. Kumar, and M. Goswami, "Deep learning algorithm for satellite imaging based cyclone detection," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 13, pp. 827-839, 2020.

[6] S. C. Park, M. K. Park, and M. G. Kang, "Super-resolution image reconstruction: a technical overview," *IEEE signal processing magazine,* vol. 20, no. 3, pp. 21-36, 2003.

[7] P. Milanfar, *Super-resolution imaging*. CRC press, 2017.

[8] 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*, 2018, pp. 701-710.

[9] C. Ledig *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*, 2017, pp. 4681-4690.

[10] M. Haris, G. Shakhnarovich, and N. Ukita, "Recurrent back-projection network for video super-resolution," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3897-3906.

[11] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv preprint arXiv:1511.05440,* 2015.

[12] E. L. Denton, S. Chintala, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," *Advances in neural information processing systems,* vol. 28, 2015.

[13] C. Li and M. Wand, "Combining markov random fields and convolutional neural networks for image synthesis," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2479-2486.

[14] R. Li *et al.*, "DeepUNet: A deep fully convolutional network for pixel-level sea-land segmentation," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing,* vol. 11, no. 11, pp. 3954-3962, 2018.

[15] M. Mostofa, S. N. Ferdous, B. S. Riggan, and N. M. Nasrabadi, "Joint-SRVDNet: Joint super resolution and vehicle detection network," *IEEE Access,* vol. 8, pp. 82306-82319, 2020.

[16] B. Na and G. C. Fox, "Object classifications by image super-resolution preprocessing for convolutional neural networks," *Advances in Science,*

CHULALONGKORN UNIVERSITY

Technology and Engineering Systems Journal (ASTESJ), vol. 5, no. 2, pp. 476-483, 2020.

[17] 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*, vol. 38, no. 2, pp. 295-307, 2015.

[18] R. Timofte, V. De Smet, and L. Van Gool, "Anchored neighborhood regression for fast example-based super-resolution," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 1920-1927.

[19] S. Dai, M. Han, W. Xu, Y. Wu, Y. Gong, and A. K. Katsaggelos, "Softcuts: a soft edge smoothness prior for color image super-resolution," *IEEE Transactions on Image Processing*, vol. 18, no. 5, pp. 969-981, 2009.

[20] G. Freedman and R. Fattal, "Image and video upscaling from local self-examples," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 2, pp. 1-11, 2011.

[21] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *European conference on computer vision*, 2014: Springer, pp. 184-199.

[22] 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*, 2017, pp. 4539-4547.

[23] 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*, 2017, pp. 3147-3155.

[24] 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*, 2016, pp. 1637-1645.

[25] 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*, 2018, pp. 3118-3126.

[26] 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*, 2017, pp. 136-144.

[27] 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*, 2017, pp. 4799-4807.

[28] W. Han, S. Chang, D. Liu, M. Yu, M. Witbrock, and T. S. Huang, "Image super-resolution via dual-state recurrent networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1654-1663.

[29] X. Wang *et al.*, "Esrgan: Enhanced super-resolution generative adversarial networks," in *Proceedings of the European conference on computer vision (ECCV) workshops*, 2018, pp. 0-0.

[30] 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*, 2017, pp. 624-632.

[31] 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, vol. 41, no. 11, pp. 2599-2613, 2018.

[32] 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*, 2018, pp. 864-873.

[33] 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*, 2018, pp. 1664-1673.

[34] 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*, 2019, pp. 3867-3876.

[35] 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*, 2016, pp. 770-778.

[36] 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)*, 2018, pp. 517-532.

[37] 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*, vol. 29, 2016.

[38] 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)*, 2018, pp. 286-301.

[39] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 1251-1258.

[40] 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*, 2017, pp. 54-61.

[41] C. Szegedy *et al.*, "Going deeper with convolutions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1-9.

[42] 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)*, 2018, pp. 252-268.

[43] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700-4708.

[44] 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*, 2018, pp. 2472-2481.

[45] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7132-7141.

[46] 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*, 2019, pp. 11065-11074.

[47] A. Van den Oord, N. Kalchbrenner, L. Espeholt, O. Vinyals, and A. Graves,

"Conditional image generation with pixelcnn decoders," *Advances in neural information processing systems,* vol. 29, 2016.

[48] R. Dahl, M. Norouzi, and J. Shlens, "Pixel recursive super resolution," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5439-5448.

[49] Q. Cao, L. Lin, Y. Shi, X. Liang, and G. Li, "Attention-aware face hallucination via deep reinforcement learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 690-698.

[50] Y. Shi, G. Li, Q. Cao, K. Wang, and L. Lin, "Face hallucination by attentive sequence optimization with reinforcement learning," *IEEE transactions on pattern analysis and machine intelligence,* vol. 42, no. 11, pp. 2809-2824, 2019.

[51] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2881-2890.

[52] K. He, X. Zhang, S. Ren, and J. Sun, "Spatial pyramid pooling in deep convolutional networks for visual recognition," *IEEE transactions on pattern analysis and machine intelligence,* vol. 37, no. 9, pp. 1904-1916, 2015.

[53] D. Park, K. Kim, and S. Young Chun, "Efficient module based single image super resolution for multiple problems," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 882-890.

[54] I. Goodfellow *et al.*, "Generative adversarial nets," *Advances in neural information processing systems,* vol. 27, 2014.

[55] 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*, 2017, pp. 4491-4500.

[56] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *European conference on computer vision*, 2016: Springer, pp. 694-711.

[57] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3431-3440.

[58] 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*, 2016, pp. 1646-1654.

[59] S.-J. Park, H. Son, S. Cho, K.-S. Hong, and S. Lee, "Srfeat: Single image super-resolution with feature discrimination," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 439-455.

[60] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[61] 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.

[62] A. Dosovitskiy and T. Brox, "Generating images with perceptual similarity metrics based on deep networks," *Advances in neural information processing systems,* vol. 29, 2016.

[63] X. Xu, D. Sun, J. Pan, Y. Zhang, H. Pfister, and M.-H. Yang, "Learning to super-resolve blurry face and text images," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 251-260.

[64] 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*, 2018, pp. 109-117.

[65] 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*, 2018, pp. 606-615.

[66] 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*, 2017, pp. 2794-2802.

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

[68] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *International conference on machine learning*, 2017: PMLR, pp. 214-223.

[69] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, "Improved training of wasserstein gans," *Advances in neural information processing systems*, vol. 30, 2017.

[70] T. Miyato, T. Kataoka, M. Koyama, and Y. Yoshida, "Spectral normalization for generative adversarial networks," *arXiv preprint arXiv:1802.05957*, 2018.

[71] 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*, 2016, pp. 1865-1873.

[72] 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*, 2018, pp. 874-881.

[73] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41-75, 1997.

[74] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961-2969.

[75] Z. Zhang, P. Luo, C. C. Loy, and X. Tang, "Facial landmark detection by deep multi-task learning," in *European conference on computer vision*, 2014: Springer, pp. 94-108.

[76] X. Wang, K. Yu, C. Dong, X. Tang, and C. C. Loy, "Deep network interpolation for continuous imagery effect transition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1692-1701.

[77] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," *Advances in neural information processing systems*, vol. 30, 2017.

[78] Y. LeCun, "The MNIST database of handwritten digits," *http://yann. lecun. com/exdb/mnist/*, 1998.

[79] K. Da, "A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[80] H. Wang, D. Su, C. Liu, L. Jin, X. Sun, and X. Peng, "Deformable non-local network for video super-resolution," *IEEE Access*, vol. 7, pp. 177734-177744, 2019.

[81] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural

network for dynamic scene deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 3883-3891.

[82] C. Duanmu and J. Zhu, "The image super-resolution algorithm based on the dense space attention network," *IEEE Access*, vol. 8, pp. 140599-140606, 2020.

[83] W. Shi *et al.*, "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*, 2016, pp. 1874-1883.

[84] M. Qin *et al.*, "Remote sensing single-image resolution improvement using a deep gradient-aware network with image-specific enhancement," *Remote Sensing*, vol. 12, no. 5, p. 758, 2020.

[85] 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.

[86] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510-4520.

[87] Z. Wang, J. Chen, and S. C. Hoi, "Deep learning for image super-resolution: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3365-3387, 2020.

[88] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47-57, 2016.

[89] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, 2003, vol. 2: Ieee, pp. 1398-1402.

[90] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015: PMLR, pp. 448-456.

[91] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026-1034.

[92] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[93] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," *arXiv preprint arXiv:1511.05666*, 2015.

[94] L. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis using convolutional neural networks," *Advances in neural information processing systems*, vol. 28, 2015.

[95] 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*, 2015, pp. 5197-5206.

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

[97] F. Chollet, "Keras: The python deep learning library," *Astrophysics source code library*, p. ascl: 1806.022, 2018.

[98] M. Bevilacqua, A. Roumy, C. Guillemot, and M. L. Alberi-Morel, "Low-

complexity single-image super-resolution based on nonnegative neighbor embedding," 2012.

[99] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *International conference on curves and surfaces*, 2010: Springer, pp. 711-730.

[100] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, 2001, vol. 2: IEEE, pp. 416-423.

[101] Y. Matsui *et al.*, "Sketch-based manga retrieval using manga109 dataset," *Multimedia Tools and Applications,* vol. 76, no. 20, pp. 21811-21838, 2017.

[102] 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,* vol. 13, no. 4, pp. 600-612, 2004.

[103] H. Talebi and P. Milanfar, "NIMA: Neural image assessment," *IEEE transactions on image processing,* vol. 27, no. 8, pp. 3998-4011, 2018.

[104] J. Kim, H. Zeng, D. Ghadiyaram, S. Lee, L. Zhang, and A. C. Bovik, "Deep convolutional neural models for picture-quality prediction: Challenges and solutions to data-driven image quality assessment," *IEEE Signal processing magazine,* vol. 34, no. 6, pp. 130-141, 2017.

[105] 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*, 2018, pp. 3262-3271.

[106] S. Anwar and N. Barnes, "Densely residual laplacian super-resolution," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2020.

[107] J. Lee and K. H. Jin, "Local Texture Estimator for Implicit Representation Function," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 1929-1938.

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY

จุฬาลงกรณ์มหาวิทยาลัย
Chulalongkorn University

VITA

NAME	Amir Hajian
DATE OF BIRTH	07 June 1985
PLACE OF BIRTH	Iran
INSTITUTIONS ATTENDED	Department of Electrical Engineering
HOME ADDRESS	1811/77 The Parkland Grand Asok, New Petchaburi Rd., Bangkapi, Huaykwang, Bangkok 10310
PUBLICATION	[1] Hajian, A., & Aramvith, S. (2022). Progressive Multi Residual Fusion Super-Resolution Network. IEEE Access (submitted). [2] Hajian, A., & Ramli, D. A. (2018). Sharpness enhancement of finger-vein image based on modified un-sharp mask with log-Gabor filter. Procedia Computer Science, 126, 431-440. [3] Hajian, A., & Ramli, D. A. (2017). Analysis of Finger Vein Feature Extraction Using Cross-Sectional Profile Approach. In 9th International Conference on Robotic, Vision, Signal Processing and Power Applications (pp. 609-616). Springer, Singapore. [4] Hajian, A., & Damavandinejadmonfared, S. (2014). Optimal feature extraction dimension in finger vein recognition using kernel principal component analysis. International Journal of Computer and Information Engineering, 8(9), 1637-1640.
AWARD RECEIVED	The 100th Anniversary Chulalongkorn University for Doctoral Scholarship

จุฬาลงกรณ์มหาวิทยาลัย
CHULALONGKORN UNIVERSITY