

# Performance Analysis of JPEG XR with Deep Learning-Based Image Super-Resolution



A Thesis Submitted in Partial Fulfillment of the Requirements  
for the Degree of Master of Engineering in Electrical Engineering  
Department of Electrical Engineering  
FACULTY OF ENGINEERING  
Chulalongkorn University  
Academic Year 2022  
Copyright of Chulalongkorn University

การวิเคราะห์ประสิทธิภาพของเจฟีกเอกซ์อาร์ด้วยการสร้างภาพความละเอียดสูงด้วยขดขึงภายใต้การ  
เรียนรู้เชิงลึก



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต  
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้า  
คณะวิศวกรรมศาสตร์ จุฬาลงกรณ์มหาวิทยาลัย  
ปีการศึกษา 2565  
ลิขสิทธิ์ของจุฬาลงกรณ์มหาวิทยาลัย

Thesis Title                      Performance Analysis of JPEG XR with Deep  
Learning-Based Image Super-Resolution  
By                                      Miss Taingliv Min  
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 Master of  
Engineering

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

THESIS COMMITTEE

..... Chairman  
(Assistant Professor SUREE PUMRIN, Ph.D.)  
..... Thesis Advisor  
(Associate Professor SUPAVADEE  
ARAMVITH, Ph.D.)  
..... External Examiner  
(Titipat Achakulvisut, Ph.D.)

จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY

เที่ยงลิป มิน : การวิเคราะห์ประสิทธิภาพของเจเนอเรชันอัพสเคอร์ด้วยการสร้างภาพความละเอียดสูง  
 ขวดยังภายใต้การเรียนรู้เชิงลึก. ( Performance Analysis of JPEG XR with  
 Deep Learning-Based Image Super-Resolution) อ.ที่ปรึกษาหลัก : รศ.  
 ดร.สุภาวดี อร่ามวิทย์

ความต้องการของการบีบอัดโคเดกของวิดีโอที่ทันสมัยและภาพในระดัสูงอย่างมีประสิทธิภาพเพิ่มขึ้น  
 อย่างกว้างขวาง วิธีการบีบอัดรูปภาพทั่วไป เช่น เจเนอเรชันอัพสเคอร์ใช้การควอนไทเซชันพารามิเตอร์ (คิวพ) เพื่อ  
 สร้างไฟล์ที่มีการบีบอัดสูงสำหรับรูปภาพ อย่างไรก็ตามค่าคิวพที่สูงขึ้นทำให้เกิดผลผิดพลาด นำไปสู่คุณภาพ  
 เชิงการมองเห็นลดลง การแก้ไขปัญหาข้อจำกัดนี้ คือ การลดขนาดภาพความละเอียดสูงโดยการสุ่มตัวอย่าง  
 ก่อนเข้ารหัสด้วยเจเนอเรชันอัพสเคอร์ จากนั้นใช้ขั้นตอนวิธีความละเอียดสูงขวดยังไปใช้กับภาพที่มีความละเอียดต่ำ  
 เพื่อสร้างภาพความละเอียดสูงใหม่ ในงานวิจัยนี้ เราใช้การสุ่มตัวอย่างลดขนาดของภาพขาเข้าก่อนเข้ารหัส  
 แบบเจเนอเรชันอัพสเคอร์ จากนั้น เราทดสอบประสิทธิภาพของการฝึกฝนใหม่ภายใต้การเรียนรู้เชิงลึก ด้วยขั้นตอน  
 วิธีสร้างคืนภาพความละเอียดสูงขวดยังแบบเอฟเอสอาร์ซีเอ็นเอ็น จากนั้นใช้การเข้ารหัสแบบเจเนอเรชัน  
 อัพสเคอร์ มาช่วยในด้านของคุณภาพและขนาดไฟล์ที่บีบอัด จากผลการทดลอง วิธีการที่นำเสนอมีผลดีกว่าการบีบ  
 อัดแบบเจเนอเรชันอัพสเคอร์ โดยลดขนาดของการเข้ารหัสไฟล์โดยเฉลี่ยเท่ากับ 557 กิโลไบต์สำหรับอัตราขยาย  
 2 เท่าและ 756 กิโลไบต์ที่อัตราขยาย 4 เท่า การรวมการฝึกฝนโมเดลใหม่กับการบีบอัดแบบเจเนอเรชันอัพสเคอร์  
 ได้ประสิทธิภาพที่ดีกว่าการบีบอัดแบบเจเนอเรชันอัพสเคอร์อย่างเดี่ยว ในการบีบอัดไฟล์ประมาณ 66  
 เฟอร์เซ็นต์สำหรับอัตราขยาย 2 เท่าและ 89 เฟอร์เซ็นต์สำหรับอัตราขยาย 4 เท่า โดยวิธีการที่นำเสนอยัง  
 สร้างขนาดไฟล์บีบอัดที่มีขนาดเล็ก สำหรับการบีบอัดแบบสูงและได้คุณภาพของภาพที่ดีกว่า ทั้งการบีบอัด  
 แบบเจเนอเรชันอัพสเคอร์ การบีบอัดแบบเจเนอเรชันอัพสเคอร์ด้วยวิธีไบคิวบิก และการบีบอัดแบบเจเนอเรชันอัพสเคอร์ด้วย  
 เอฟเอสอาร์ซีเอ็นเอ็น

จุฬาลงกรณ์มหาวิทยาลัย  
 CHULALONGKORN UNIVERSITY

สาขาวิชา วิศวกรรมไฟฟ้า

ลายมือชื่อนิติ

ปีการศึกษา 2565

ลายมือชื่อ อ.ที่ปรึกษาหลัก

# # 6372049121 : MAJOR ELECTRICAL ENGINEERING

KEYWORD Image super-resolution, JPEG XR compression, Bicubic  
RD: interpolation, Quantization parameter

Taingliv Min : Performance Analysis of JPEG XR with Deep Learning-Based Image Super-Resolution. Advisor: Assoc. Prof. SUPAVADEE ARAMVITH, Ph.D.

The demand for efficient high-level image and video codec compression has widely increased. Conventional image compression methods such as JPEG XR use a high quantization parameter (QP) to produce a highly compressed file for any given image. However, higher QP has displeasing artifacts that lead to perceptual quality degradation. A feasible solution to tackle this limitation is to reduce the high-resolution image size by downsampling it before encoding it with JPEG XR. Then, the super-resolution algorithm is applied to the resultant low-resolution image to reconstruct the high-resolution result. In this research, we downsample the input image before JPEG XR. Then, we investigate the performance of integrating a newly retrained deep learning-based FSRCNN super-resolution (SR) with JPEG XR in terms of quality and compressed file size. According to the experimental results, the experimental results show that the proposed method outperforms JPEG XR compression by shrinking the size of the encoded file by an average of 557 kB for scale two and 756 kB for scale four. The fusion of the newly trained model with JPEG XR compression can achieve higher performance than JPEG XR compression in compressing the file, around 66% for scale two and 89% for scale four. The proposed method also produces a small, compressed file size for high compression and achieves better visual quality than JPEG XR compression, JPEG XR with the bicubic method, and JPEG XR with FSRCNN.

Field of Study:	Electrical Engineering	Student's Signature
Academic Year:	2022	.....
		Advisor's Signature
		.....

## ACKNOWLEDGEMENTS

First, I would like to express my warmest gratitude toward my advisor, Assoc. Prof. SUPAVADEE ARAMVITH, Ph.D., allowed me to pursue higher study. Her guidance and advice helped me through all my work. When I found myself stuck with the difficulty of understanding the work, she always advised me and cheered me up.

I would also like to thank Asst. Prof. SUREE PUMRIN, Ph.D. for being the chairperson during my proposal exam and thesis defense. I am thankful to Dr. Titipat Achakulvisut for being on my committee. They gave me many valuable and fruitful suggestions and comments.

I gratefully acknowledge the financial support from Chulalongkorn University's Graduate Scholarship Programme for ASEAN or Non-ASEAN Countries.

I want to thank my parents, grandmother, and brother, who supported and motivated me to complete this work. They would listen to my worries and encourage me to finish my research.

Finally, I am so grateful to my seniors, VTRG lab-mates and friends for helping and giving me continuous support throughout my research.

Taingliv Min

# TABLE OF CONTENTS

	<b>Page</b>
ABSTRACT (THAI) .....	iii
ABSTRACT (ENGLISH).....	iv
ACKNOWLEDGEMENTS.....	v
TABLE OF CONTENTS.....	vi
LIST OF TABLES.....	viii
LIST OF FIGURES .....	ix
LIST OF ABBREVIATIONS.....	1
CHAPTER ONE INTRODUCTION.....	2
1. 1 Motivation and Research Problem .....	2
1. 2 Objectives .....	3
1. 3 Scope of Work.....	4
1. 4 Expected Output .....	4
1. 5 Thesis outline.....	4
CHAPTER TWO BACKGROUND.....	5
2.1 Pre-processing: Downsampling by the interpolation method .....	5
2.1.1 Nearest neighbor interpolation .....	5
2.1.2 Bilinear interpolation.....	6
2.1.3 Bicubic interpolation .....	6
2.2 Image Compression methods .....	6
2.2.1 JPEG compression.....	7
2.2.2 JPEG 2000 compression.....	8
2.2.3 JPEG XR compression .....	9
2.3 Post-processing: Image super-resolution methods .....	10
2.3.1 Traditional super-resolution models.....	11
2.3.2 Learning-based models.....	11

a. Super-Resolution Convolutional Neural Network (SRCNN) .....	11
b. Fast Super-Resolution Convolutional Neural Networks (FSRCNN).....	12
CHAPTER THREE METHODOLOGY .....	13
3.1 Overall Framework.....	13
3.2 Downsampling.....	14
3.3 JPEG XR compression .....	14
3.4 Super-resolution model .....	15
3.5 Loss function .....	17
CHAPTER FOUR EXPERIMENTAL RESULTS AND DISCUSSIONS .....	18
4.1 Experimental setup .....	18
4.2 Training dataset .....	20
4.3 Testing dataset .....	21
4.4 Quality metrics .....	21
4.4.1 Peak Signal-to-Noise Ratio (PSNR).....	21
4.4.2 Structural Similarity Index Measure (SSIM) .....	22
4.5 Results and discussions .....	23
CHAPTER FIVE CONCLUSION.....	37
REFERENCES .....	38
VITA.....	43



## LIST OF TABLES

	<b>Page</b>
Table 1: The Experiment setup details.....	19
Table 2: The results of PSNR (dB) and Encoded image file size (kB) on The New Test Image dataset.....	25
Table 3: The results of SSIM on The New Test Image dataset .....	26
Table 4: The average PSNR (dB) and Encoded file size (kB).....	27



## LIST OF FIGURES

	<b>Page</b>
Figure 1: Original HR image .....	2
Figure 2: The compressed image .....	3
Figure 3: General diagram of the motivation.....	3
Figure 4: General diagram of an image codec .....	7
Figure 5: General diagram of JPEG compression.....	7
Figure 6: General diagram of the JPEG 2000.....	8
Figure 7: JPEG XR compression process .....	10
Figure 8: Network structure of SRCNN .....	11
Figure 9: Network structure of FSRCNN .....	12
Figure 10: Overall framework .....	13
Figure 11: Diagram of the super-resolution module of the proposed model.....	15
Figure 12: Some images from the DIV2k dataset.....	20
Figure 13: Some images from The New Test Image dataset .....	21
Figure 14: The visual comparison of the ‘artificial’ image .....	29
Figure 15: The visual comparison of the ‘cathedral’ image .....	30
Figure 16: The visual comparison of the ‘fireworks’ image.....	31
Figure 17: The visual comparison of the ‘flower_foveon’ image .....	32
Figure 18: The visual comparison of the ‘hdr’ image.....	33
Figure 19: The visual comparison of the ‘leaves’ image.....	34
Figure 20: The visual comparison of the ‘nightshot’ image.....	35
Figure 21: The visual comparison of the ‘spider_web’ image .....	36

## LIST OF ABBREVIATIONS

SR	Super-Resolution
HR	High-Resolution
LR	Low-Resolution
SRCNN	Super-Resolution Convolutional Neural Networks
FSRCNN	Fast Super-Resolution Convolutional Neural Networks
JPEG	Joint Photographic Experts Group
JPEG XR	JPEG Extended Range
QP	Quantization Parameter
POT	Photo Overlap Transform
PCT	Photo Core Transform
LSB	Least Significant Bit
LBT	Lapped Biorthogonal Transform
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
PSNR	Peak Signal-to-Noise Ratio
SSIM	Structural Similarity Index Measure

# CHAPTER ONE

## INTRODUCTION

### 1.1 Motivation and Research Problem

Nowadays, multimedia data such as audio, animations, images, videos, and texts have been increasing due to the increasing popularity of the internet. The file sizes become large and consume lots of hard disk space. Moving them from place to place over the internet is also time-consuming.

Due to the limited bandwidth and memory space, sending, sharing, and storing large amounts of data have become a challenging problem. Data compression is required to solve this problem. It plays an essential role in high-volume data transmission and storage. Compression converts a large file into a smaller file. Compressing the data is a helpful solution when the file is quite large. Compressing data can save storage capacity, speed up file transfer, and decrease storage hardware and network bandwidth costs.

Image compression is a technique that shrinks the amount of image data while maintaining image quality. In image compression, the bits number in each pixel is reduced. The time it takes for an image to be uploaded or downloaded can also be reduced after compression. The original high-resolution image can be seen in Figure 1. The highly compressed image, the output from an image compression standard at high compression, is shown in Figure 2. The compressed image contains blocking artifacts that make the picture not pleasant to the eyes of the viewers.



Figure 1: Original HR image



Figure 2: The compressed image

That is why in this research, there is a need to have pre- and post-processing for the image compression process. The general diagram for the motivation of this research can be found in Figure 3. For the pre-processing step, downsampling is done to reduce the resolution of the input HR image. An image encoder then encodes the downsampled image. The compressed image can be transmitted over the network or stored in storage. The image decoder is used to decompress the image. Since the decompressed image contains blocking artifacts from the compression, the upsampling process is used as a post-processing step to produce an HR output image with better quality.

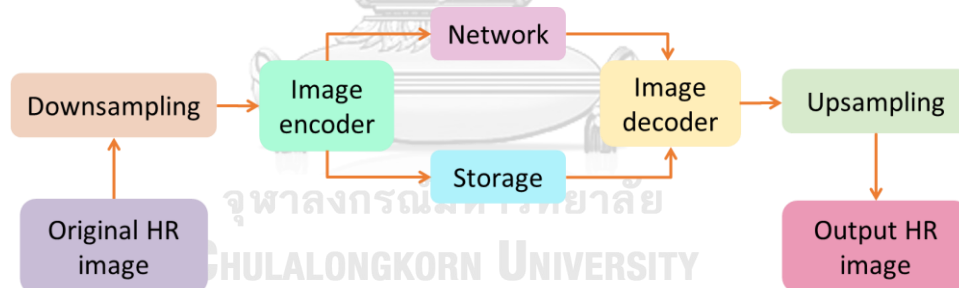


Figure 3: General diagram of the motivation

## 1.2 Objectives

- a. Investigate image compression with deep learning-based super-resolution.
- b. Analyze the super-resolution in different scale factors and quantization parameter values.
- c. Evaluate the performance of the proposed method with the image compression standard and conventional compression super-resolution method.
- d. For each scale factor, analyze the suitable QP values empirically after conducting several simulations.

### 1.3 Scope of Work

- a. Introduce a single image super-resolution along with JPEG XR image compression.
- b. Analyze the super-resolution of at least two scale factors and three quantization parameter values.
- c. Evaluate the performance of the proposed algorithm with JPEG XR compression and JPEG XR compression with conventional super-resolution method using PSNR.

### 1.4 Expected Output

- a. Produce output images with a smaller file size than the other two methods.
- b. Produce output images with better visual quality for high compression.

### 1.5 Thesis outline

The remainder of this work is organized as follows. Chapter II contains the background. The proposed method is described in Chapter III. Chapter IV consists of Experimental results and discussions. In Chapter V, the conclusion will be described.

## CHAPTER TWO

### BACKGROUND

Multimedia data compression can help solve the problem of limited bandwidth during the transmission and storage of large data. Since compression can reduce the size of the data, the compressed data also loses some information. In image compression, lossless compression can reduce the file size and restore the original file without any loss of information. On the other hand, lossy compression compresses the input data and loses much of the data compared to lossless compression. The loss of information can lead to quality degradation in the output image. Super-resolution is a method that produces a high-resolution image from a low-resolution input image. The quality of the output image from lossy compression can be accelerated by adding the super-resolution method to get a high-quality reconstructed image.

In this section, there are three main parts to be shown. The pre-processing part which is downsampling is first summarized. Then, image compression methods are also reviewed, and finally, the post-processing part for upsampling is briefed.

#### **2.1 Pre-processing: Downsampling by the interpolation method**

Interpolation is the process used for producing a continuing magnitude from disjointed image data. Image interpolation is commonly used for various purposes, such as resizing and scaling images [1].

There are three commonly known interpolation methods: Nearest neighbor interpolation, Bilinear interpolation, and Bicubic interpolation.

##### **2.1.1 Nearest neighbor interpolation**

Of all the commonly known interpolation methods, this technique is the most common and needs the least amount of time to process. By rounding the values of the target interpolation point, the nearest neighbor chooses the value of the closest pixel. With this procedure, every pixel in the final image is compared to its closest pixel in the original image. New pixels are formed the same way as others nearby. As the image expands, the pixels are replicated to generate new pixels. It produces pixilation,

jagged edges, or edges that divide curves into parts. The effects of this type of interpolation are typically undesirable for image enlargement and reduction [2].

### **2.1.2 Bilinear interpolation**

Bilinear interpolation takes four nearby pixels to calculate the average value and uses it as the last interpolated pixel value. This method interpolates pixels horizontally and vertically. The output image seems finer than the original image. Bilinear interpolation performs better than nearest neighbor interpolation and consumes less time to compute than the bicubic method [3].

### **2.1.3 Bicubic interpolation**

Bicubic interpolation takes the nearest 4x4 nearby pixels into account for a total of sixteen pixels. Nearest pixels are given a larger weighting in the computation since they are closer to the unknown pixel than farther away ones are. Bicubic creates sharper pictures than the other two techniques, which may be the best compromise between time consumed and output quality [3].

## **2.2 Image Compression methods**

The goal of compression is to reduce the number of bits needed to store or transmit an image with the least amount of information loss. Compression can be categorized into two types: lossless and lossy compression. Lossless compression compresses the images without information loss when it is uncompressed. It removes empty, needless, or duplicated information from the original file. The compressed file from lossless compression produces a smaller file with the same quality as the original. Lossless compression is typically used with text and spreadsheet files, where the loss of words or numbers would change the information. Lossy compression removes the original file's redundant, excessive, or unimportant bits. Lossy compression is useful with audio, video, and images, where removing some data bits has little or no noticeable effect on the representation of the content. It produces a higher compression rate than lossless compression, with the quality of the output image trade-off. The higher compression rate of lossy compression has quality degradation.





Figure 4: General diagram of an image codec

Many image compression formats exist today, such as JPEG, JPEG 2000, JPEG XR, JPEG-LS, and JPEG XT. The performance of image codecs is analyzed in [4]. The comparison was made using the same test conditions for each image codec, like JPEG, JPEG 2000, and JPEG XR. The workflow of an image codec can be described in Figure 4.

### 2.2.1 JPEG compression

JPEG (Joint Photographic Experts Group) was established in the mid-1980s as an international standard for grayscale and color still image compression [5]. It has become a popularly used image format. Most of the techniques in this category modify the LSB (least significant bit) of the JPEG coefficients, which are the outcomes of block-based two-dimensional (2-D) Discrete Cosine Transform (DCT) followed by quantization using a JPEG quantization table [6].

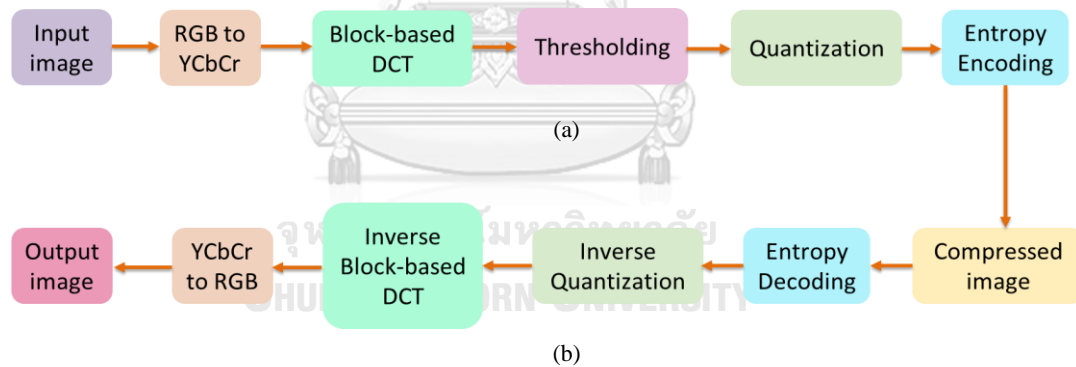


Figure 5: General diagram of JPEG compression

(a) Encoding part and (b) Decoding part

The general diagram of JPEG compression can be seen in Figure 5. In [7], JPEG image compression can be divided into two parts: (a) encoding part and (b) decoding part. There are five stages needed to perform the encoding part. Firstly, a color image in RGB format needs to be converted to a YCbCr color space. After the color conversion, the three color components are divided into many blocks and applied

block-based DCT. This operation converts an image from the spatial domain into the frequency domain.

The thresholding process changes the range of the element in the image block from  $[0, 255]$  to  $[-128, 127]$  [8]. The next stage is the quantization process. Quantization aspires to reduce the less important DCT coefficients to zero. The compression rate is higher when there are more zeros. Quantization is a method of optimally reducing a large number scale into a smaller one. The final procedure in the encoding part is entropy encoding. Entropy encoding further quantized the DCT coefficients so that more compression could be obtained. The decoding part works by reversing the process in the encoding part. Firstly, Entropy decoding is done, followed by inverse quantization. Then, the inverse block-based DCT is performed. Finally, the YCbCr color space is converted to RGB, and the output image is produced.

### 2.2.2 JPEG 2000 compression

In 1997, JPEG 2000 was launched as a new image codec for many still images. This standard can be operated at a low bit rate with higher rate distortion and image quality than the previously existing standards [5]. This new image coding system has been an advanced image compression technology since the efficiency and scalability of this standard are optimized.

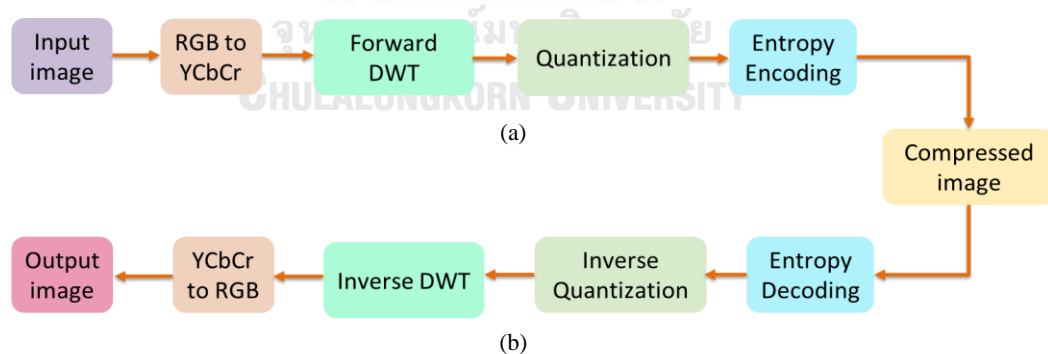


Figure 6: General diagram of the JPEG 2000

(a) Encoding part and (b) Decoding part

The process of JPEG 2000 can be described in Figure 6. JPEG 2000 relies on Discrete Wavelet Transform (DWT) to reduce the amount of information contained in a picture. The coefficients are obtained to make the compressed image blurry and contain ringing artifacts [9]. The encoding process works by applying the forward

DWT to the input image. The DWT transforms a signal into frequency and amplitude over time and is, therefore, more efficient. Then, the quantization and entropy encoding are done. The decoding part can be achieved by reversing the encoding part. The JPEG 2000 can achieve a higher compression rate than JPEG for similar image quality [10]. JPEG artifacts are much more disturbing, with the picture visibly broken down into small blocks at high compression ratios. JPEG 2000 requires codecs that are complex and computationally demanding. Another difference, in comparison with JPEG, is in terms of visual artifacts: JPEG 2000 only produces ringing artifacts, manifested as blur and rings near edges in the image, while JPEG produces both ringing artifacts and 'blocking' artifacts. JPEG 2000 is not commonly used since it is complex to be implemented [11].

### **2.2.3 JPEG XR compression**

JPEG XR is a new coding standard invented in 2007 and submitted to the Joint Photographic Experts Group (JPEG) for standardization. JPEG XR standard can be used to compress images and videos. The JPEG-XR encoding process utilizes two transform operations: Photo Overlap Transform (POT) and Photo Core Transform (PCT). During the encoding process, color conversion is performed as the first step. Then, the image is divided into non-overlapped blocks. The core transform and overlapped filter operations are applied to each block. In this step, core transform is performed twice. Finally, QP is used to perform quantization and then performs the entropy encoding [12]. During the encoding process, the quantization parameter greater than one makes the model lossy compression, and the output image contains degradation artifacts. These artifacts are presented by the nonlinearity of transforms used by JPEG XR [13].

The experimental research [14] has proven that JPEG XR and JPEG2000 demonstrate higher visual quality than JPEG compression under the high compression ratio. The JPEG2000 compression standard has a more complicated algorithm than the JPEG standard [15]. Compared to the JPEG2000 compression standard, JPEG XR has lower complexity and produces similar PSNR quality at the same bit rate. Due to the capability of JPEG XR, this compression standard maintains better image quality at a low bit rate and higher compression ratio. JPEG XR is composed of blocks that

compress continuous tone images. These blocks include Lapped Biorthogonal Transform (LBT), tiling, filters, and adaptive best suited for compression and decompression. JPEG XR standard has high compression efficiency, low complexity, and low computational cost [10]. Additionally, the ability to compress both lossy and lossless compressions makes it become a technique that can support the ideal results obtained [4]. The JPEG XR compression process is shown in Figure 7.

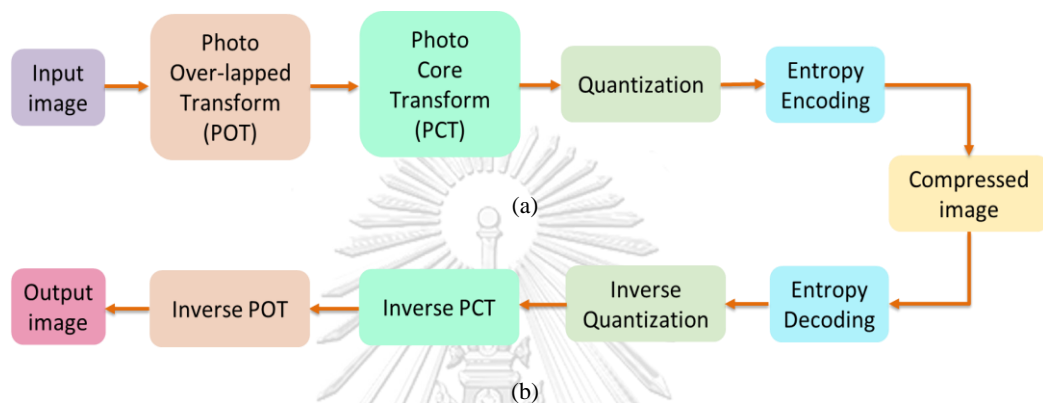


Figure 7: JPEG XR compression process

(a) Encoding process, (b) Decoding process

### 2.3 Post-processing: Image super-resolution methods

Image super-resolution is a technique that reconstructs low-resolution (LR) images into high-resolution (HR) images. There are three types of super-resolution methods: traditional interpolation-based, reconstruction-based, and learning-based. Image super-resolution is a technique that produces a high-resolution image from the information and details from the original input image [16]. The more resolution an image has, the more information it contains. The image stores information such as color and brightness. It is one of the important techniques in computer vision and image processing, such as medical imaging, security, surveillance, astronomical imaging, and satellite imaging. The high-resolution output image can be obtained by applying a super-resolution method to an LR image. Throughout the past decade, various super-resolution methods have been proposed and applied, such as interpolation, reconstruction, and learning-based designs [17].

### 2.3.1 Traditional super-resolution models

The interpolation super-resolution methods are the Nearest neighbor [18], Bilinear [19], Bicubic [20], directional Bicubic interpolation method [21], and so on. Directional Bicubic interpolation has low complexity and produces good results, but the edges in the output image from this method are not well reconstructed horizontally and vertically. Interpolation-based method interpolated the HR image from the LR input image using a smooth kernel function [22]. It upscales the LR image by estimating the pixels in the HR grids.

The reconstruction-based method can be classified into two types: deterministic approach and stochastic approach. A deterministic approach can produce smoothness for the output image. However, smoothness is not always the best option since it can blur the image. The stochastic approach can be easily integrated with other typical image processing tasks, such as denoising, de-convolution, enhancement, etc.

### 2.3.2 Learning-based models

Learning-based super-resolution shows the potential for reconstructing LR images with superior performance due to its ability to upsample them with better quality. The deep learning-based super-resolution method demonstrates outstanding performance to conventional techniques due to its ability to reconstruct the output image with a better objective and subjective quality. The deep learning-based super-resolution has attracted attention due to its robust reconstruction ability.

#### a. Super-Resolution Convolutional Neural Network (SRCNN)

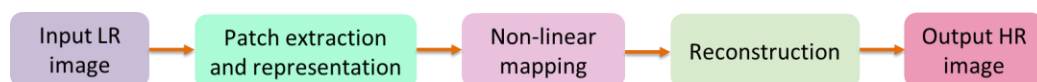


Figure 8: Network structure of SRCNN

Other research on learning-based super-resolution is suggested in the references [23-30]. Dong et al. [31] proposed the first CNN-based model, Super-Resolution Convolutional Neural Network (SRCNN). This model uses a very simple structure consisting of only three layers. The network of SRCNN is shown in Figure 8. An end-to-end learning-based approach reconstructs an LR input image into an HR image. Before the features are extracted, bicubic interpolation is used for upsampling LR

images as a pre-processing step. SRCNN only uses convolutional layers; it makes this model suitable for any image size but not patch-based [32]. The network consists of three main steps: patch extraction and representation, non-linear mapping, and reconstruction.

### b. Fast Super-Resolution Convolutional Neural Networks (FSRCNN)

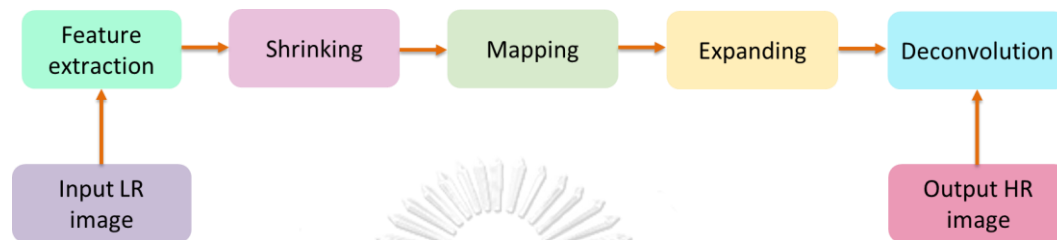


Figure 9: Network structure of FSRCNN

Later, Dong et al. proposed an updated model that was called Fast Super-Resolution Convolutional Neural Networks (FSRCNN). FSRCNN is different from SRCNN in three aspects [33]. The network structure of FSRCNN is shown in Figure 9. First, FSRCNN uses the original LR image as input without bicubic interpolation. Second, there are three steps in FSRCNN replacing the non-linear mapping in SRCNN: the shrinking, mapping, and expanding steps. Lastly, the filter sizes in the updated method are smaller, and the model has a deeper network structure. The FSRCNN model is lightweight and can train and test faster than SRCNN in different scaling factors.

## CHAPTER THREE

### METHODOLOGY

In this section, first, we explain the overall framework. Then, the downsampling process and JPEG XR compression are described. Finally, detailed information about the super-resolution model is presented.

#### 3.1 Overall Framework

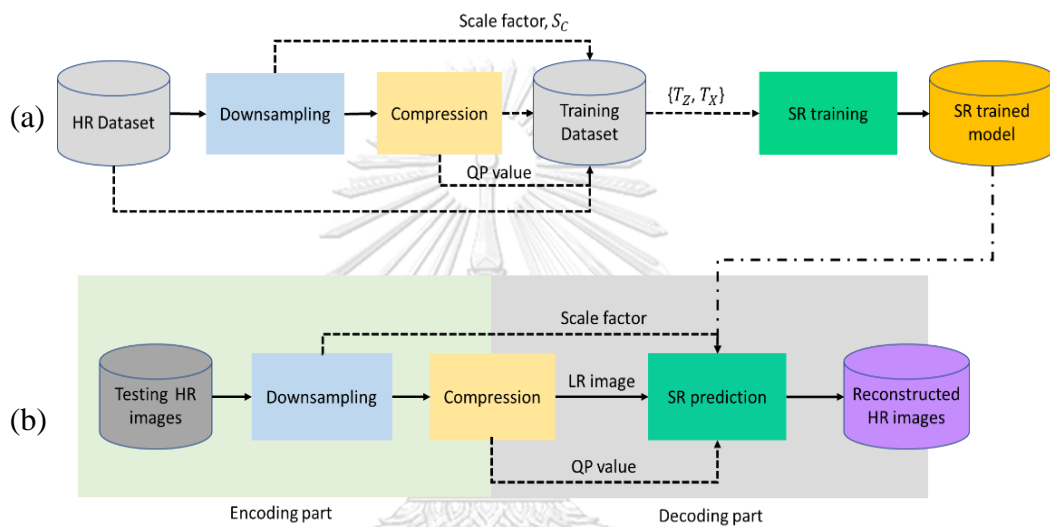


Figure 10: Overall framework

(a) Training phase, (b) Testing phase

The overall framework contains downsampling, compression, and super-resolution modules; The detail of the overall framework is shown in Figure 10. The training process (a) starts by taking the training image as input and downscaling with two scales (2 and 4). Then, the downsampled images are encoded and decoded by the image compression method with four QP values (20, 40, 60, and 80). After that, the scale factors (2 and 4), QP values (20, 40, 60, and 80), the training set of  $n$  decoded LR images ( $T_X = \{X_1, X_2, \dots, X_n\}$ ), and the training set of  $n$  original HR images ( $T_Z = \{Z_1, Z_2, \dots, Z_n\}$ ) are used as training input parameters in the SR model. Finally, we obtain the SR-trained model that can produce the best image after several training iterations.

The proposed method can be formulated as:

$$\hat{X} = F_{SR}(Z) \quad (3.1)$$

Where  $\hat{X}$  is the reconstructed HR image,  $F_{SR}$  is the super-resolution function of the SR model.  $Z$  is the LR input image.

The testing process (b) is conducted by firstly downscaling HR images. The compression procedure is done after getting the downsampled images. The upscaling process uses scale factor, QP value, and decoded image as input. SR prediction uses SR trained model to upscale the image and get the reconstructed image. The following sections describe downsampling, compression, and super-resolution modules.

### 3.2 Downsampling

The downsampling process is conducted to reduce the spatial resolution of input image data for the later stages. In the downsampling process, bicubic interpolation is used. It is the most common interpolation method. The scaling factors of the downsampling process are used later in the training process. The downsampling process is defined by Eq. 3.2.

$$Y_{S_c} = f_d(X, S_c) \quad (3.2)$$

where  $Y_{S_c}$  is the output image of the downsampling process.  $f_d(X, S_c)$  is the bicubic interpolation that uses an HR image  $X$  and scaling factor  $S_c$  as inputs.

### 3.3 JPEG XR compression

This work uses a JPEG XR encoder and decoder to compress and decompress images. JPEG XR has QP values from 1 to 255 to define compression from lossless to high lossy compression rate [34]. In the image compression process, the encoder of JPEG XR is used with different QP values to compress the image. The higher the QP, the worse the output image's quality can be. After that, the JPEG XR decoder decodes the compressed image. The QP values and the decoded images are also used as input for the training process of the super-resolution model.



The equation of JPEG XR compression can be determined as:

$$Z = JXR(Y_{S_c}, QP) = D(E(Y_{S_c}, QP)) \quad (3.3)$$

where  $Z$  is the output of JPEG XR compression.  $D$  and  $E$  represent the decoder and encoder, respectively.  $JXR$  is the decoding and encoding process of JPEG XR compression.

### 3.4 Super-resolution model

The block diagram of our deep learning super-resolution module (FSRCNN) is shown in Figure 11. The result of the decoding process is used as the input of the SR module, along with the QP value and scale factor.

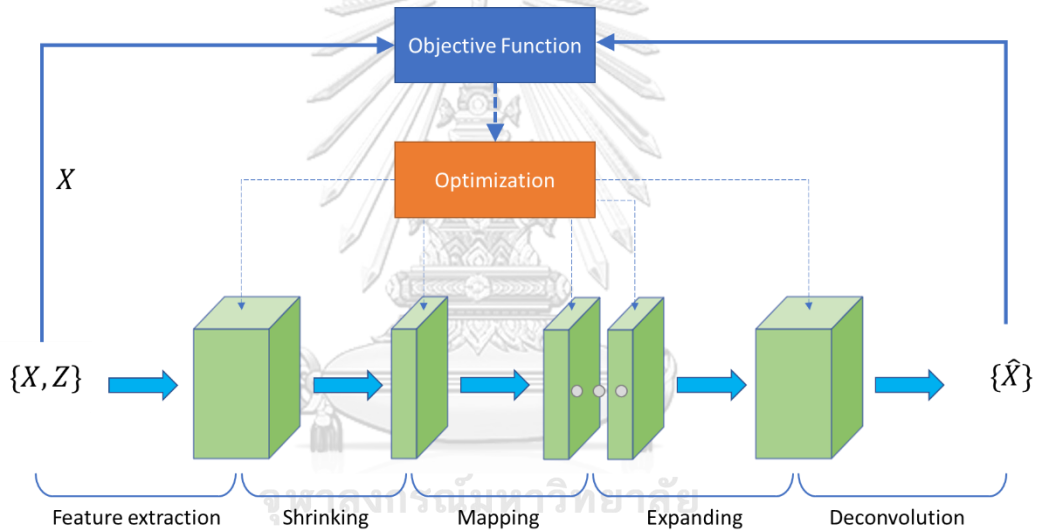


Figure 11: Diagram of the super-resolution module of the proposed model

Convolutional neural networks (CNN) for super-resolution formation can be described as follows: Let  $Z$  be the decoded LR image and  $F(Z)$  be the reconstructed HR image. Our super-resolution model contains three main stages: feature extraction and representation, Non-linear mapping, and reconstruction.

- 1) **Feature extraction:** features are extracted from  $Z$  image and represented by a high-dimensional vector. The vectors are made of a set of feature maps whose number is equivalent to the dimensionality of the vectors.

The first layer of the network  $F_1$  is defined as:

$$F_1(Z) = \max(0, W_1 * Z + B_1) \quad (3.4)$$

where  $W_1$  represents the filter parameters and  $B_1$  is the biases. *ReLU* is the rectified linear unit activation function.

- 2) **Shrinking:** the number of feature maps is reduced by  $1 \times 1$  convolution. After reducing the number of parameters, the network works faster because it needs fewer operations. This is one of the reasons that FSRCNN works faster than SRCNN.
- 3) **Non-linear mapping:** in this operation, a high-dimensional vector is a nonlinearity mapped onto another high-dimensional vector. Each mapped vector denotes a high-resolution patch. These vectors contain another set of feature maps. The equation of the second layer  $F_2$  is computed as:

$$F_2(Z) = \max(0, W_2 * F_1(Z) + B_2) \quad (3.5)$$

where  $W_2$  and  $B_2$  are the weight and biases parameters of the second layer, respectively.

- 4) **Expanding:** in this step,  $1 \times 1$  convolution is performed to increase the number of feature maps before going to the final step.
- 5) **Reconstruction:** This operation obtains the high-resolution output image by accumulating the high-resolution patch-wise representations. The output image  $F$  is determined by:

$$\hat{X} = F_{SR}(Z) = W_3 * F_2(Z) + B_3 \quad (3.6)$$

where  $W_3$  and  $B_3$  are model parameters of the up-sampled layer.

### 3.5 Loss function

We need to minimize the loss between the reconstructed output image and the original input image to train the model for predicting the output image.

Mean Square Error (MSE) is used as the loss function:

$$L = \frac{1}{n} \sum_{i=1}^n \|\hat{X}_i - X_i\|^2 \quad (3.7)$$

where  $n$  is the training sample numbers.  $i$  are the indices of layers and iterations.  $X$  is the original HR image, and  $\hat{X}$  is the reconstructed HR image corresponding to  $X$ .



## CHAPTER FOUR

### EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, the experimental results and discussions are described in detail to show the performance of the proposed method compared to the JPEG XR conventional image compression method and the JPEG XR with a traditional super-resolution method.

#### 4.1 Experimental setup

The proposed method takes a raw image input and then downsampling it. After the downsampling process, the image passes through the JPEG XR compression that encodes and decodes the downsampled image. The final step is reconstructing the image using the deep learning method. The output image from the reconstruction process is a high-resolution image.

We use bicubic as the downsampling method. The downscale factors are two and four. After the input images are downsampled, the output images from the bicubic interpolation pass through the JPEG XR compression. The encoder and the decoder of JPEG XR are utilized to compress and decompress the images. The output images from JPEG XR are in low resolution. Input image in RGB color space is converted to YCbCr formats. In this step, all the channels are normalized. YCbCr separates one luminance component (Y) representing brightness and two chroma components (Cb and Cr) representing color. Luminance components can be kept at high resolution and transmitted at high bandwidth. In contrast, chroma components can be kept at lower bandwidth than luminance and can be subsampled, coded, and managed individually to increase efficiency. Then, the Y color data is upscaled using a deep learning-based super-resolution method. The bicubic upscales the other two channels. Finally, the reconstructed image is converted back to RGB. We convert the RGB color space to YCbCr because human vision is more sensitive to brightness (Y) than color variation (Cb, Cr) [35]. By using only Y channel as training input, it can save us time and have better performance as well.

The input images are the first downsample using Bicubic interpolation at two scales (2 and 4). The downsampling procedure is done in MATLAB using the bicubic

method. After downsampling, the images proceed to JPEG XR compression conducted in Ubuntu. Four quantization parameters (20, 40, 60, and 80) are used during the encoding process to determine different levels of encoded image quality. Then, we decode the images using the JPEG XR decoder to obtain the decoded images. Once the images have been decoded, the training and testing process is done on NVIDIA GeForce GTX Titan 1080 Ti. The Python programming language version 3.8.13 uses PyTorch version 1.7.1 in the proposed method. We use Adam as an optimizer in the training process, learning rate = 0.001, batch size = 16, patch size = 96. The training is set to run for 100 epochs. We train eight models according to both scale factors and quantization parameters.

Table 1: The Experiment setup details

Method Model	Pre-processing	Image Compression	Post-processing
JPEG XR compression	-		-
JPEG XR with Bicubic	Bicubic Downsampling (scale 2 and 4)	JPEG XR Compression (QP 20, 40, 60, and 80)	Bicubic interpolation
JPEG XR with FSRCNN			Pre-trained FSRCNN
JPEG XR with Our trained model			Newly-retrained FSRCNN

The details of the experiments conducted for comparison are shown in Table 1. Three other experiments are carried out to compare with the proposed method. The first experiment, briefed in the first row, involves compressing and decompressing the images with four QP values using a JPEG XR encoder and decoder. The other three experiments use the Bicubic method to downsample the image in the pre-processing step and JPEG XR to compress and decompress the downsampled image. The second experiment, shown in the second row, uses a conventional super-resolution technique, the bicubic interpolation, in the post-processing to upsample the decoded image. The third experiment uses the pre-trained model of FSRCNN to reconstruct the output image in the final step. The last experiment, which is our proposed method, retrains the FSRCNN with the training datasets and uses the trained model to reconstruct the output image.

## 4.2 Training dataset

The images used for training are taken from the DIV2k dataset. The dataset consists of 800 images in '.png' format. Some of the training images used in the experiments of this research are shown in Figure 12.

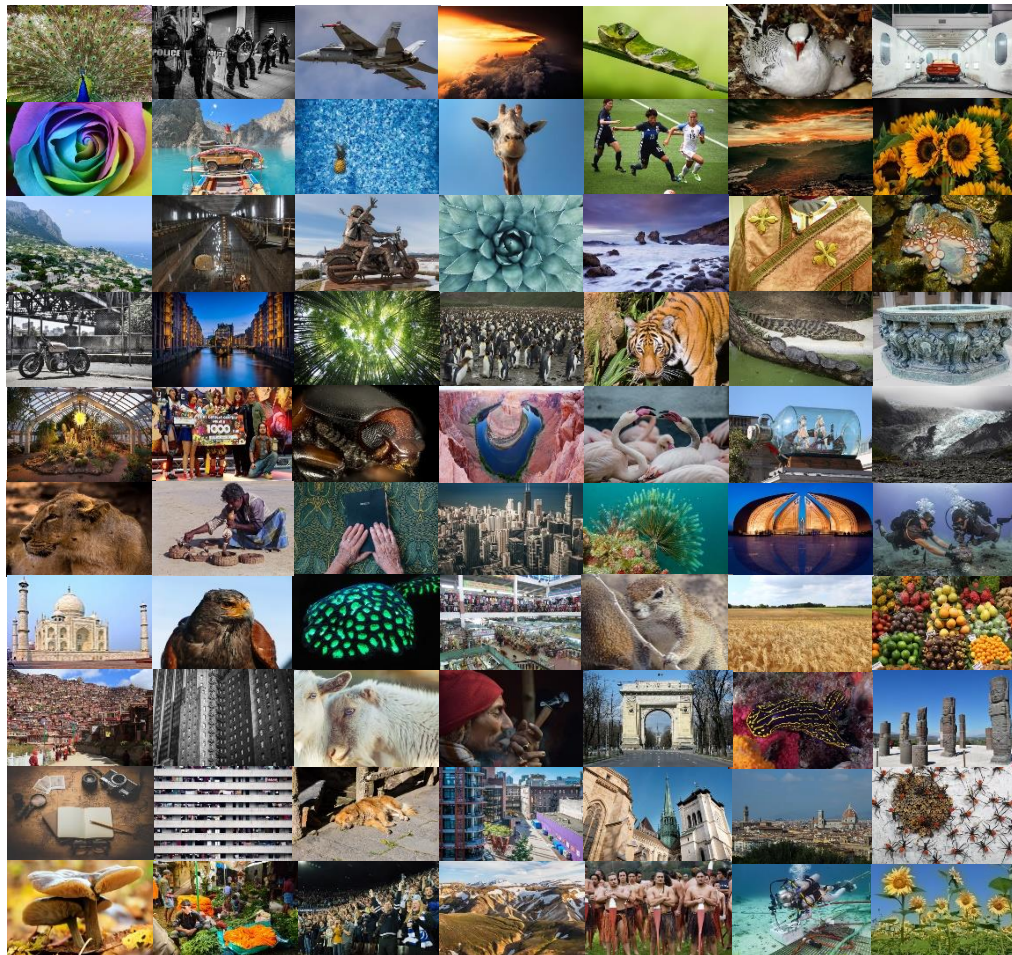


Figure 12: Some images from the DIV2k dataset

The images in the dataset are converted into '.ppm' format to be feasible with JPEG XR compression. The images are input for the downsampling process with two scales (2 and 4). And, then the 1600 downsampled images are encoded and decoded by JPEG XR encoder and decoder with four different QP values (20, 40, 60, and 80). Finally, we have 6400 images that will be used for training. Before the training begins, the decoded images are converted to '.png' format to go through the SR model training. The model is trained according to the scale factor and QP value. The final number of trained models is eight models.



The idea of converting the ‘.png’ format to ‘.ppm’ in the first place is for two reasons. One of the reasons is that if the data is sent or stored in ‘.png’ format, we want to have a small, compressed file size with good visual quality after decompression. And the second reason is that JPEG XR compression can only work with raw images in ppm, dng, raw, pnm, and so on.

The conversion doesn’t affect the quality of the image. We tested the original and the converted image by selecting one patch from the whole image and calculating the PSNR value. The result shows that both patches have the same value of PSNR. Therefore, we can conclude that the conversion still preserves the quality of the image no matter what the extension is.

### 4.3 Testing dataset

Eight out of fourteen images were used from The New Test Images dataset, which provides ppm raw images. Some of the testing images used in the experiments of this research are shown in Figure 13. There are various images with file sizes ranging from 10 to 117 MB. The minimum resolution of the input image is 2268×1512 pixels. The maximum input image resolution is 7216×5412 pixels. All fourteen images are in RGB format.

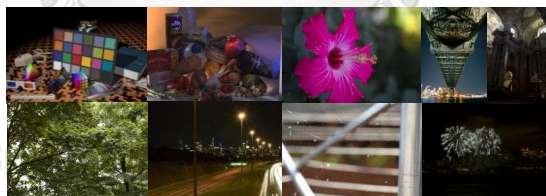


Figure 13: Some images from The New Test Image dataset

### 4.4 Quality metrics

#### 4.4.1 Peak Signal-to-Noise Ratio (PSNR)

The Y channel data from the original input image and the Y channel of the reconstructed image are used as parameters for calculating the PSNR value. Objective and subjective evaluations have been used to evaluate the performance of the proposed method. PSNR is used as an evaluation metric. PSNR is measured in decibels (dB), whose value ranges from 0 to infinity. A higher value of PSNR means a better quality of the reconstructed HR image.

The equation of PSNR is shown as follows:

$$PSNR = \frac{10 \times \log_{10}(2^b - 1)^2}{MSE} \quad (4.1)$$

$$MSE = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n (\hat{X}_{i,j} - X_{i,j})^2 \quad (4.2)$$

where  $m, n$  are the numbers of the row and column of the image.  $b$  is the bit-depth of the image.

#### 4.4.2 Structural Similarity Index Measure (SSIM)

The structural similarity index measure can be determined as follows:

$$SSIM(X, \hat{X}) = l(X, \hat{X})^\alpha \cdot c(X, \hat{X})^\beta \cdot s(X, \hat{X})^\gamma \quad (4.3)$$

Where  $l$  is the luminance,  $c$  is the contrast, and  $s$  is the structure.  $\alpha, \beta$ , and  $\gamma$  are constants with positive values. Luminance is the comparison between the brightness of two images. The contrast shows the difference between the brightest and darkest areas of the two images. The structure is used for comparing the pattern of the local luminance between two images so that the similarity and dissimilarity of the image can be found.

The expressions used to calculate luminance, contrast, and structure can be defined as:

$$l(X, \hat{X}) = \frac{2\mu_X\mu_{\hat{X}} + C_1}{\mu_X^2 + \mu_{\hat{X}}^2 + C_1} \quad (4.4)$$

$$c(X, \hat{X}) = \frac{2\sigma_X\sigma_{\hat{X}} + C_2}{\sigma_X^2 + \sigma_{\hat{X}}^2 + C_2} \quad (4.5)$$

$$s(X, \hat{X}) = \frac{\sigma_{X\hat{X}} + C_3}{\sigma_X\sigma_{\hat{X}} + C_3} \quad (4.6)$$

Where  $\mu_X$  and  $\mu_{\hat{X}}$  are the local means.  $\sigma_X$  and  $\sigma_{\hat{X}}$  are the standard deviation, and  $\sigma_{X\hat{X}}$  is the cross-covariance for image  $X$  and  $\hat{X}$ , respectively.  $C_1$  and  $C_2$  are used to avoid instability.  $C_3 = C_2/2$ .

$$\sigma_X = \left( \frac{1}{N-1} \sum_{i=1}^N (X_i - \mu_X)^2 \right)^{1/2} \quad (4.7)$$



$$\mu_X = \frac{1}{N} \sum_{i=1}^N X_i \quad (4.8)$$

$$\sigma_{X\hat{X}} = \frac{1}{N-1} \sum_{i=1}^N (X_i - \mu_X)(\hat{X}_i - \mu_{\hat{X}}) \quad (4.9)$$

The *SSIM* equation can be simplified as:

$$SSIM = \frac{(2\mu_X\mu_{\hat{X}}+C_1)(2\sigma_{X\hat{X}}+C_2)}{(\mu_X^2+\mu_{\hat{X}}^2+C_1)(\sigma_X^2+\sigma_{\hat{X}}^2+C_2)} \quad (4.10)$$

#### 4.5 Results and discussions

The PSNR values and encoded file sizes for eight images from The New Test Image dataset are displayed in Table 2. The PSNR value and encoded image size are evaluated from JPEG XR compression, JPEG XR fused with Bicubic, JPEG XR with FSRCNN, and JPEG XR with our trained model in scales 2 and 4 in different QP rates (20, 40, 60, and 80). The bicubic interpolation technique, FSRCNN super-resolution, and our trained model are compared for upsampling images. As observed in Table 5, the PSNR values from JPEG XR compression are higher than those from the other experiment in all QP values, while the Encoded file sizes are bigger. The file sizes of the encoded images for scale two are about three times smaller than those of JPEG XR compression. While for scale four, the encoded file sizes are around eight times smaller compared to the JPEG XR compression. Overall, we can see that almost at every QP with both scales 2 and 4, our trained model has the highest PSNR values among the three methods that use JPEG XR with SR. At high QP (QP = 80), the results from the JPEG XR experiment mostly have similar encoded file size, but lower PSNR values than the fusion of JPEG XR with SR scale two at QP = 60 and scale four at QP = 40.

For instance, the ‘flower\_foveon’ image in JPEG XR compression has all PSNR values higher than those from other experiments in scales two and four but bigger compressed file sizes. At QP = 80, JPEG XR compression has an encoded file size of 31 kB with PSNR = 39.65 dB, while JPEG XR with our trained model at scale two and QP = 60, has a similar compressed file size (32 kB), but higher PSNR value (41.46 dB). For scale four with QP = 40, JPEG XR with our trained model has PSNR = 41.51 dB, which is 1.86 dB higher than JPEG XR compression. At both scale two

and scale four, JPEG XR with our trained model achieves higher PSNR values in most cases of QP values than the ones from JPEG XR with Bicubic and JPEG XR with pre-trained FSRCNN.

The SSIM values from the experiments on The New Test Image dataset are shown in Table 3. The result indicates that JPEG XR compression has a high SSIM value compared to the other three models that use super-resolution combined with JPEG XR compression. The result shows that the percentage differences between JPEG XR compression and JPEG XR with Bicubic interpolation are 3.51% for scale two and 7.75% for scale four. The SSIM results from JPEG XR compression and using pre-trained FSRCNN with JPEG XR show the differences of 3.57% at scale two and 7.59% at scale four. For the last comparison, the average SSIM value from JPEG XR compression is higher than the SSIM value from our newly trained by 3.89% at scale two and 7.53% at scale four.

The proposed method outperforms JPEG XR compression by reducing the size of the encoded file by an average of 557 kB for scale two and 756 kB for scale four. The fusion of the newly trained model with JPEG XR compression can achieve higher performance than JPEG XR compression in compressing the file, around 66% for scale two and 89% for scale four.

Table 2: The results of PSNR (dB) and Encoded image file size (kB) on The New Test Image dataset

Image	QP	JPEG XR only		JPEG XR with SR scale 2				JPEG XR with SR scale 4			
		PSNR	Encoded file size	PSNR (Bicubic)	PSNR (FSRCNN)	PSNR (Ours)	Encoded file size	PSNR (Bicubic)	PSNR (FSRCNN)	PSNR (Ours)	Encoded file size
artificial	20	51.17	975	37.84	42.41	41.72	407	32.46	35.69	34.59	163
	40	45.55	539	36.98	39.65	39.78	232	32.04	34.46	33.82	95
	60	40.16	274	34.88	35.65	36.68	120	30.81	31.78	32.22	49
	80	35.26	124	31.60	31.63	32.90	53	28.42	28.50	29.38	22
cathedral	20	45.72	3,129	38.78	39.90	39.99	873	33.39	35.01	34.71	242
	40	40.85	1,155	36.93	37.37	37.54	344	32.61	33.64	33.55	97
	60	37.03	343	34.12	34.15	34.40	109	30.87	31.19	31.33	32
	80	33.41	95	30.92	30.86	31.16	32	28.54	28.54	28.75	10
fireworks	20	50.21	1,125	38.44	42.88	42.29	376	30.49	34.69	33.16	124
	40	46.28	435	37.68	40.68	40.98	169	30.28	33.91	33.01	62
	60	41.75	191	35.89	37.11	37.96	80	29.65	31.96	31.96	30
	80	36.98	90	32.83	32.93	34.07	36	28.21	28.87	29.37	13
Flower foveon	20	49.64	329	46.84	47.06	47.82	142	42.68	43.54	43.35	62
	40	47.07	149	44.53	44.51	43.89	69	41.02	41.35	41.51	31
	60	43.60	67	41.13	41.03	41.46	32	38.22	38.18	38.28	14
	80	39.65	31	37.21	37.14	37.39	14	34.61	34.50	34.67	6
hdr	20	48.96	937	44.66	45.34	45.93	316	39.98	41.26	41.37	110
	40	45.45	336	42.21	42.33	42.88	130	38.57	39.10	39.40	50
	60	41.72	124	38.96	38.86	39.34	51	36.05	36.04	36.45	21
	80	37.84	50	35.39	35.32	35.86	20	32.86	32.76	33.17	8
leaves	20	44.94	3,263	33.49	36.13	35.81	1,469	26.9	29.10	28.89	435
	40	39.06	1,514	32.39	33.84	33.90	760	26.55	28.32	28.25	250
	60	34.34	676	30.02	30.35	30.80	322	25.48	26.25	26.48	110
	80	30.07	274	26.69	26.60	27.12	118	23.28	23.29	23.62	40
Night shot	20	45.16	5,750	38.10	38.25	38.46	1,432	34.44	35.22	35.13	359
	40	39.64	2,507	36.30	36.29	36.39	620	33.77	34.34	34.35	158
	60	36.18	725	34.76	34.73	34.88	195	32.64	32.84	33.08	44
	80	34.29	169	32.84	32.79	33.11	42	30.72	30.68	31.06	13
Spider web	20	49.27	1,133	46.39	46.52	47.50	445	42.09	43.32	43.31	140
	40	46.53	345	44.31	44.31	44.86	132	40.63	41.19	41.39	46
	60	43.56	118	41.18	41.08	41.41	47	38.11	38.13	38.40	18
	80	39.52	57	37.32	37.25	37.77	20	34.93	34.84	35.12	7
Average		41.90	845	37.24	37.97	38.31	288	33.17	34.14	34.16	89

Table 3: The results of SSIM on The New Test Image dataset

Image	QP	JPEG XR	JPEG XR with SR scale 2			JPEG XR with SR scale 4		
			Bicubic	FSRCNN	Ours	Bicubic	FSRCNN	Ours
artificial	20	0.998	0.983	0.985	0.975	0.949	0.958	0.952
	40	0.993	0.974	0.973	0.961	0.937	0.942	0.902
	60	0.980	0.952	0.949	0.962	0.909	0.907	0.917
	80	0.953	0.912	0.908	0.873	0.862	0.856	0.863
cathedral	20	0.980	0.914	0.916	0.921	0.828	0.839	0.843
	40	0.936	0.868	0.868	0.871	0.792	0.797	0.795
	60	0.863	0.800	0.797	0.802	0.738	0.738	0.744
	80	0.780	0.727	0.726	0.727	0.680	0.679	0.687
fireworks	20	0.989	0.970	0.963	0.956	0.937	0.932	0.938
	40	0.976	0.954	0.943	0.923	0.919	0.910	0.839
	60	0.958	0.928	0.928	0.948	0.889	0.891	0.906
	80	0.921	0.891	0.892	0.807	0.849	0.850	0.856
flower foveon	20	0.994	0.992	0.991	0.993	0.990	0.989	0.990
	40	0.993	0.990	0.990	0.992	0.988	0.987	0.989
	60	0.990	0.987	0.986	0.988	0.982	0.981	0.984
	80	0.981	0.976	0.976	0.979	0.967	0.966	0.972
hdr	20	0.994	0.986	0.986	0.989	0.972	0.974	0.976
	40	0.989	0.979	0.978	0.981	0.964	0.965	0.967
	60	0.978	0.965	0.964	0.967	0.949	0.948	0.953
	80	0.959	0.943	0.942	0.947	0.924	0.923	0.927
leaves	20	0.995	0.953	0.961	0.963	0.864	0.889	0.892
	40	0.978	0.934	0.938	0.940	0.849	0.869	0.871
	60	0.941	0.891	0.891	0.895	0.809	0.818	0.827
	80	0.885	0.824	0.821	0.834	0.744	0.742	0.758
nightshot	20	0.988	0.890	0.893	0.902	0.789	0.789	0.797
	40	0.948	0.846	0.845	0.841	0.761	0.760	0.756
	60	0.857	0.776	0.775	0.766	0.713	0.712	0.714
	80	0.764	0.710	0.709	0.702	0.688	0.686	0.689
spider_web	20	0.994	0.988	0.988	0.989	0.983	0.982	0.984
	40	0.990	0.985	0.984	0.986	0.979	0.978	0.981
	60	0.984	0.979	0.978	0.968	0.973	0.971	0.975
	80	0.973	0.967	0.967	0.965	0.962	0.960	0.961
Average		0.953	0.920	0.919	0.916	0.879	0.881	0.881

Table 4: The average PSNR (dB) and Encoded file size (kB)

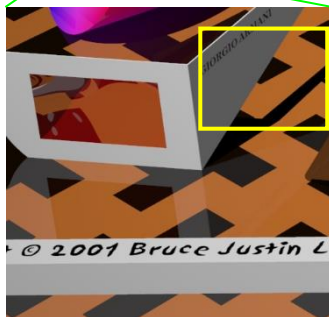
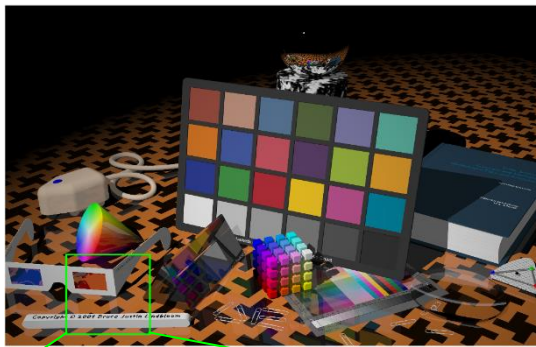
QP	JPEG XR only		JPEG XR with SR scale 2				JPEG XR with SR scale 4			
	PSNR	Encoded file size	PSNR (bicubic)	PSNR (FSRCNN)	PSNR (ours)	Encoded file size	PSNR (bicubic)	PSNR (FSRCNN)	PSNR (ours)	Encoded file size
20	48.13	2080	40.57	42.31	42.44	683	35.30	37.23	36.81	204
40	43.80	873	38.92	39.87	40.03	307	34.43	35.79	35.66	99
60	39.79	315	36.37	36.62	37.11	120	32.73	33.30	33.53	40
80	35.88	111	33.10	33.07	33.67	42	30.20	30.25	30.64	15

The average compressed file size and PSNR value of JPEG XR compression, JPEG XR with Bicubic, JPEG XR with FSRCNN, and JPEG XR with our trained model at scales 2 and 4 with four QPs are shown in Table 4. Compared to other methods, JPEG XR compression offers a greater PSNR value at each QP. However, for a greater QP value with a comparable compressed file size, the JPEG XR with our trained model has a much higher PSNR value than JPEG XR compression. For instance, at QP=80, the average file size of the JPEG XR experiment is 111 kB with a PSNR value of 35.88 dB. In JPEG XR with our method at scale two and QP = 60, the average compressed file size is 120 kB with a PSNR value of 37.11 dB. In JPEG XR with our method at scale four and QP = 40, the average encoded file size is 99 kB, and its PSNR value is 35.66 dB. For scale 4, the result is on par with the JPEG XR high QP compression if we compare the two results with similar encoded file sizes. The proposed method at scale two outperforms JPEG XR compression at a high QP value for similar encoded file sizes.

To compare the visual quality of the output images, we select one patch from the images to be analyzed. The output images shown from Figure 14 to Figure 21 are from the testing dataset. The patches selected have similar compressed file sizes from each experiment. We would like to compare the visual quality from the output images that have similar compressed file sizes from each experiment. On the patches selected, there is a rectangular box on the area that further shows the difference between the visual quality of the image from each experiment.

The output images from the JPEG XR compression at high compression rates that are shown in Figures 14 to 21 (d) contain blocking artifacts that can be easily seen, such as in Figure 18 (d) and Figure 21 (d). The output images from the experiment that uses JPEG XR with bicubic at scales two and four are shown in Figures 14 to 21 (b) and (c). The output images from using bicubic at scale four, i.e., Figure 14 (c), are very blurry and have low PSNR compared to JPEG XR. For scale two, i.e., Figure 14 (b), the visual quality is better even though the PSNR values are lower than those of JPEG XR compression. The output images are smoother than the output from JPEG XR. The output images from the experiment that uses JPEG XR with the pre-trained FSRCNN are shown in Figures 14 to 21 (e) and (f). The results indicate that at similar compressed file sizes, using JPEG XR with the pre-trained FSRCNN produces smoother output images than the ones from JPEG XR compression, even though the PSNR values are lower. Finally, our newly trained model's output images are displayed in Figures 14 to 21 (g) and (h) for scales two and four, respectively. The PSNR values from the patches selected from the experiment that uses our proposed method at scale two (Figure 14 to 21 (g)) are higher than the ones from JPEG XR compression at a similar compressed file size. Moreover, the output from both scales of our proposed method has a smoother and sharper texture than the image obtained from JPEG XR compression.

It is shown that the output image from JPEG XR with our trained model has the best visual quality of all other experiments. It can be concluded that the proposed method has better visual quality than JPEG XR compression and JPEG XR with the bicubic method for high compression. One of our objectives is to find a suitable QP that should be used with respecting scale according to the compromise of PSNR value, compressed file size, and visual quality. After discussing the result, we should select scale 2 with QP = 40, and scale four should be used with QP = 20.



(a) Original HR image

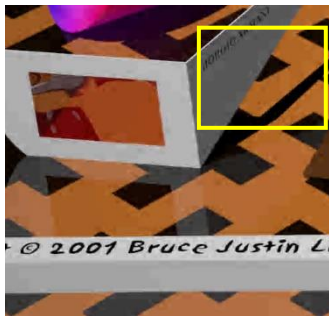
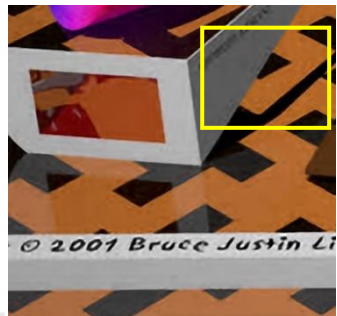
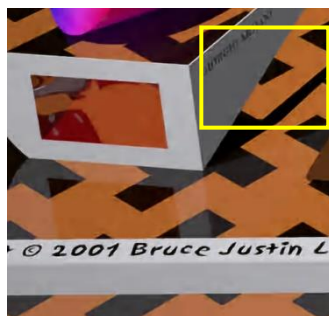
(b) Bicubic, scale 2,  
QP 60, PSNR 21.11 dB(c) Bicubic, scale 4,  
QP 40, PSNR 18.80 dB(d) JPEG XR only, QP 80,  
PSNR 32.68 dB(e) FSRCNN, scale 2,  
QP 60, PSNR 20.56 dB(f) FSRCNN, scale 4,  
QP 40, PSNR 17.95 dB(g) Our method, scale 2,  
QP 60, PSNR 34.01 dB(h) Our method, scale 4,  
QP 40, PSNR 29.44 dB

Figure 14: The visual comparison of the 'artificial' image





Figure 15: The visual comparison of the 'cathedral' image



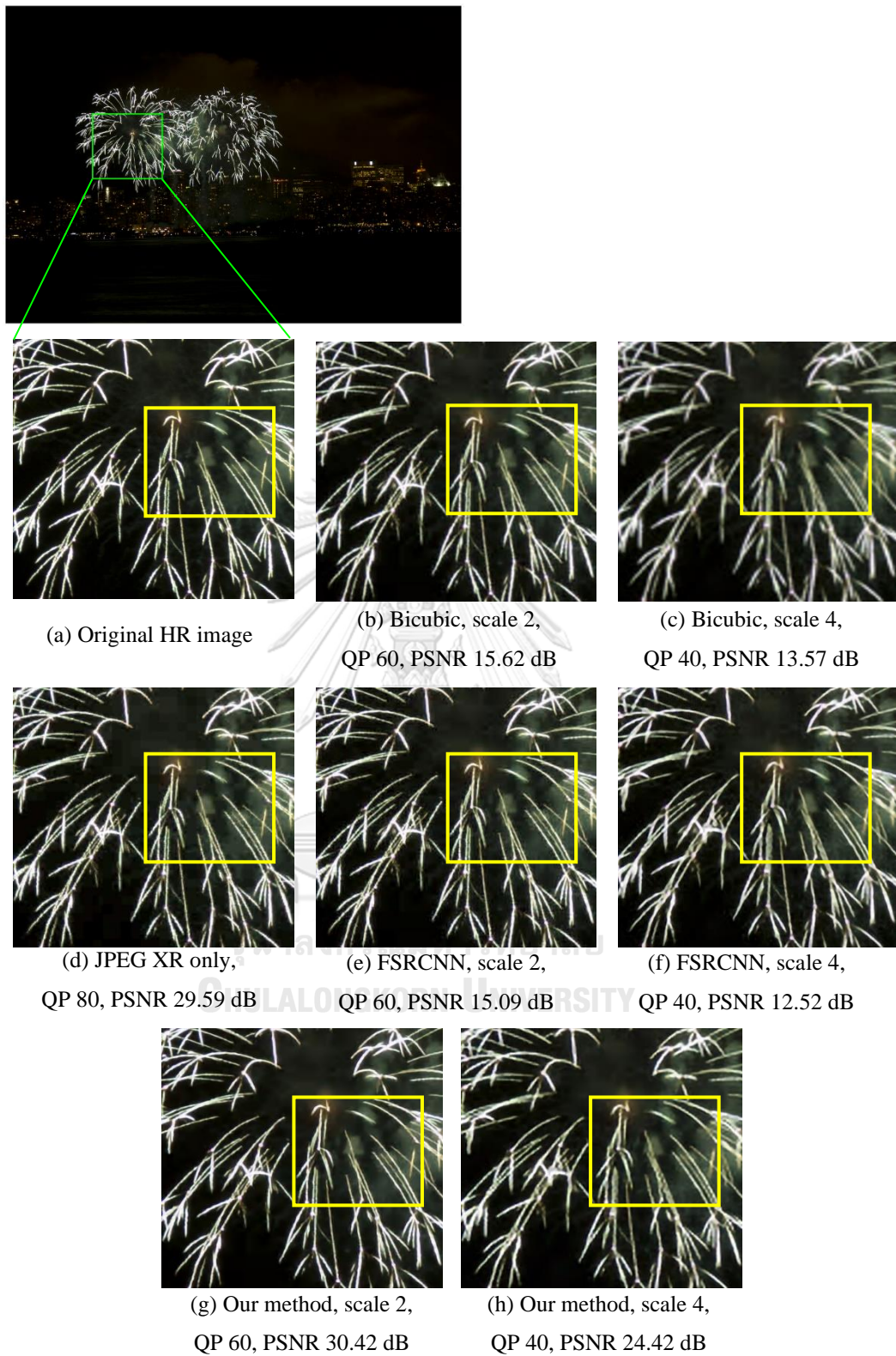


Figure 16: The visual comparison of the 'fireworks' image



Figure 17: The visual comparison of the 'flower\_foveon' image

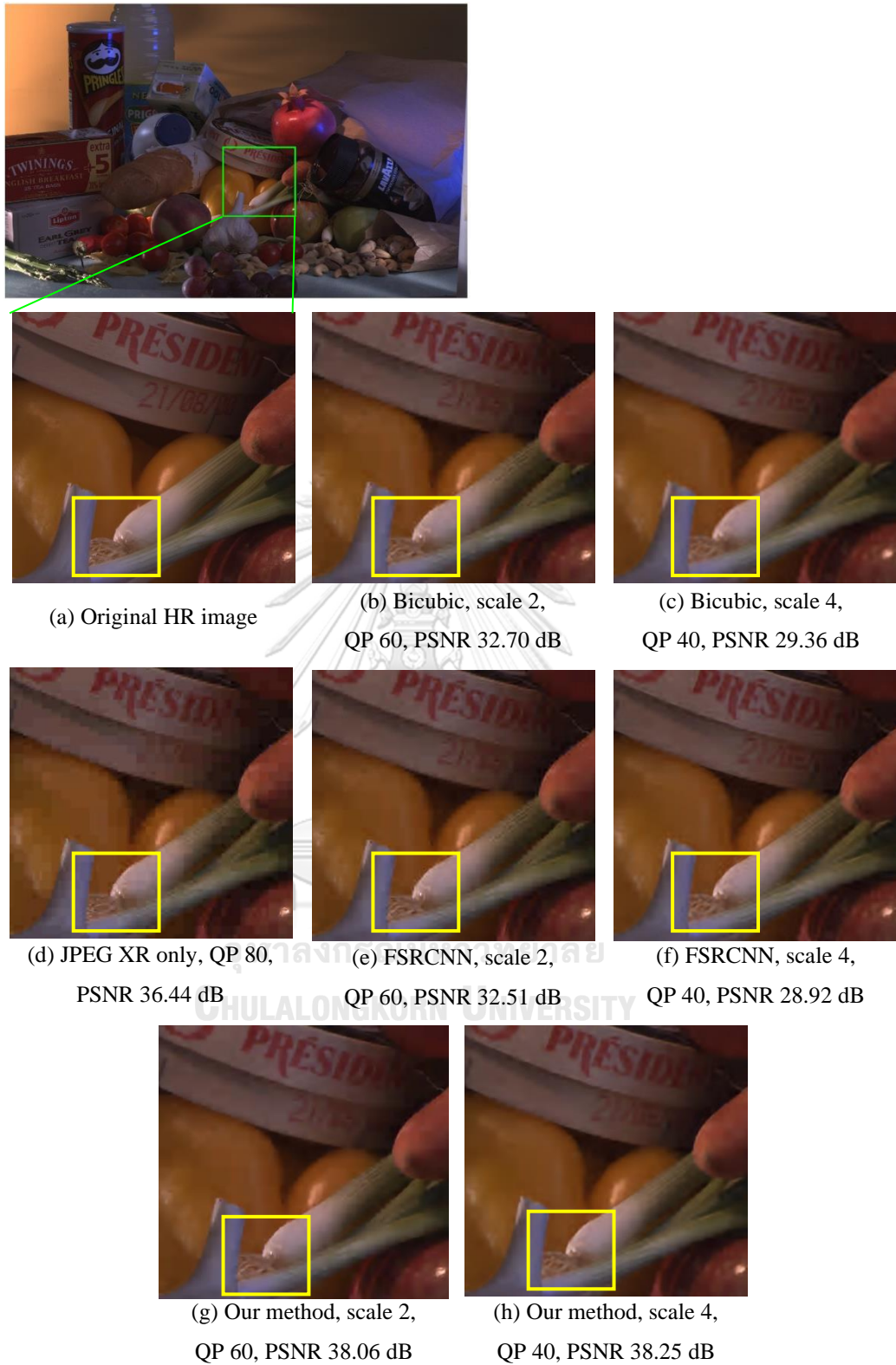


Figure 18: The visual comparison of the 'hdr' image



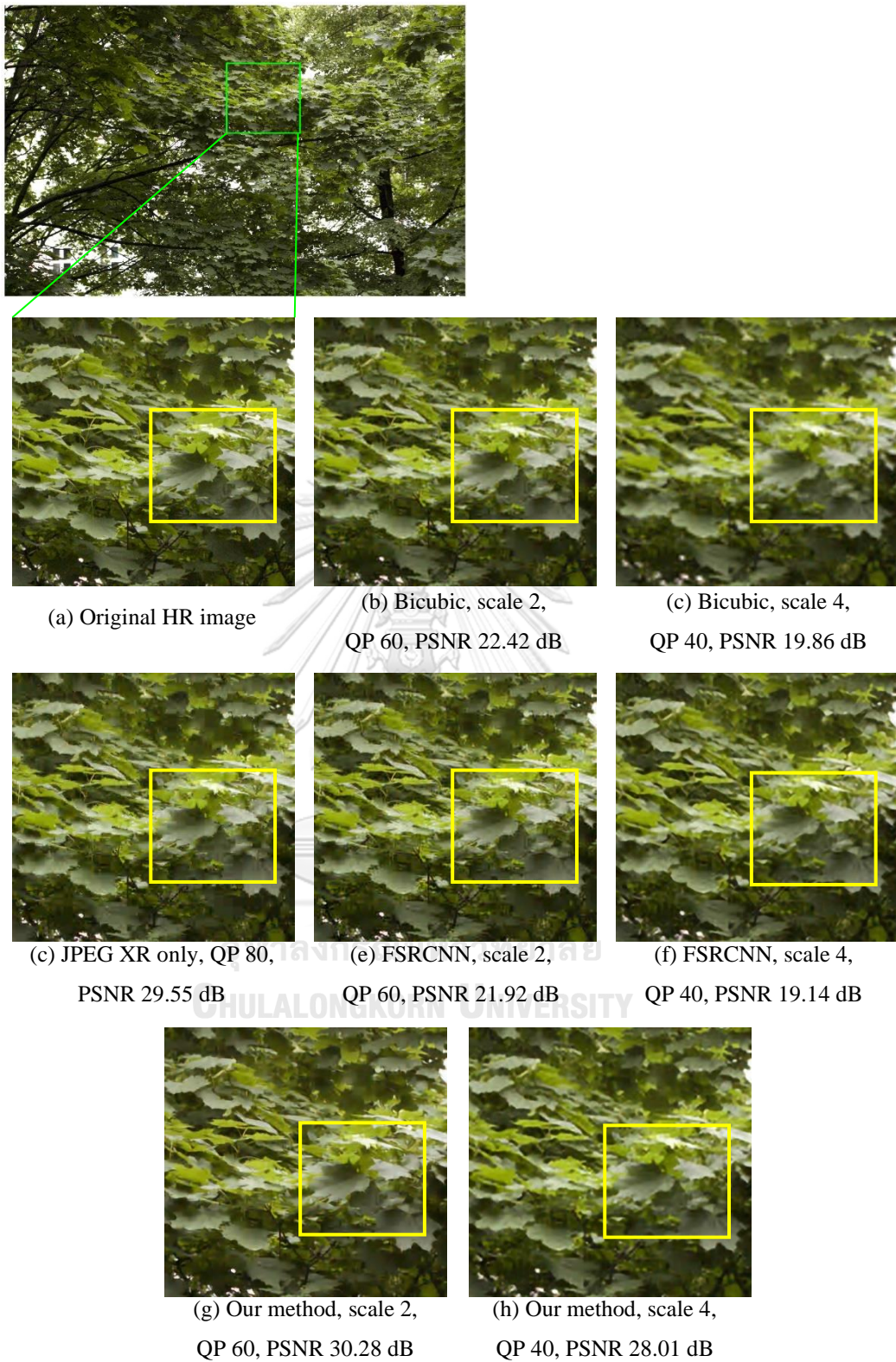


Figure 19: The visual comparison of the 'leaves' image



Figure 20: The visual comparison of the 'nightshot' image



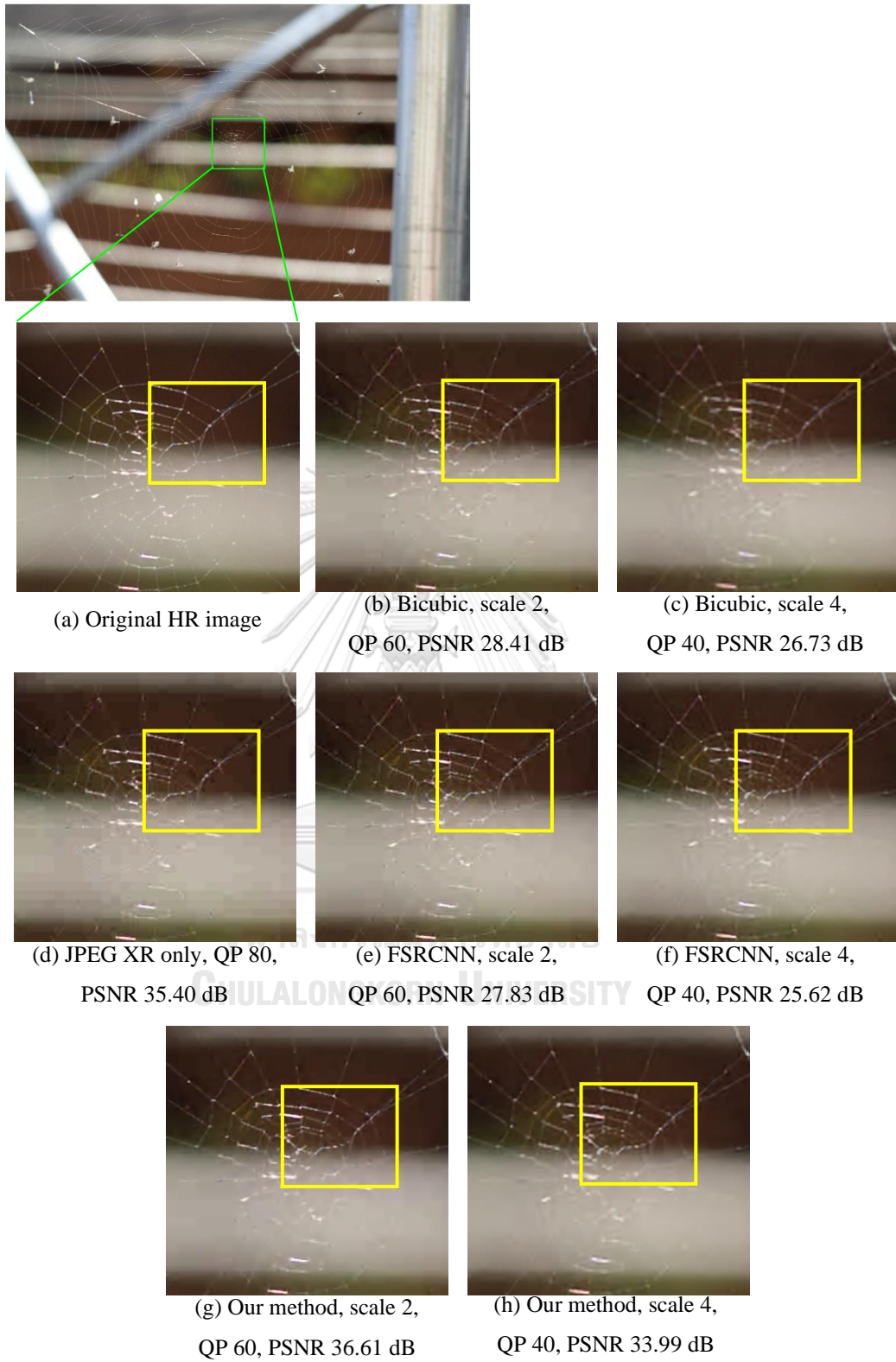


Figure 21: The visual comparison of the 'spider\_web' image

## CHAPTER FIVE

### CONCLUSION

This research analyzes the performance of JPEG XR with a deep learning-based image super-resolution method. Firstly, we downsample the input image. Then, the image is encoded and decoded by JPEG XR compression. The decoded image is up-sampled using a deep learning-based super-resolution method to reconstruct the high-resolution image. We use the FSRCNN model that uses decoded images, QP values from the compression, and scale factor in training a newly proposed method. The experimental results show that the proposed method outperforms JPEG XR compression by reducing the size of the encoded file by an average of 557 kB for scale two and 756 kB for scale four. The fusion of the newly trained model with JPEG XR compression can achieve higher performance than JPEG XR compression in compressing the file, around 66% for scale two and 89% for scale four. The proposed method also produces a small, compressed file size for high compression and achieves better visual quality than JPEG XR compression, JPEG XR with the bicubic method, and JPEG XR with FSRCNN. In the future, an end-to-end learning process can be considered in the model design to achieve a better result.

## REFERENCES



จุฬาลงกรณ์มหาวิทยาลัย  
**CHULALONGKORN UNIVERSITY**



- [1] B. K. Triwijoyoa and A. Adila, "Analysis of medical image resizing using bicubic interpolation algorithm," *J. Ilmu Komput*, vol. 14, no. 2, pp. 20-29, 2021.
- [2] P. Miklos, "Image interpolation techniques," in *2nd Siberian-Hungarian joint symposium on intelligent systems*, 2004.
- [3] P. Parsania and D. Virparia, "A review: Image interpolation techniques for image scaling," *International Journal of Innovative Research in Computer and Communication Engineering*, vol. 2, no. 12, pp. 7409-7414, 2014.
- [4] S. N. Kumar, M. V. Bharadwaj, and S. Subbarayappa, "Performance Comparison of Jpeg, Jpeg XT, Jpeg LS, Jpeg 2000, Jpeg XR, HEVC, EVC and VVC for Images," in *2021 6th International Conference for Convergence in Technology (I2CT)*, 2021: IEEE, pp. 1-8.
- [5] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal processing magazine*, vol. 18, no. 5, pp. 36-58, 2001.
- [6] Y. Q. Shi, C. Chen, and W. Chen, "A Markov process based approach to effective attacking JPEG steganography," in *International Workshop on Information Hiding*, 2006: Springer, pp. 249-264.
- [7] M. Sangeetha and P. Betty, "A Dynamic Image Compression Using Improved LZW Encoding Algorithm," 2017.
- [8] A. Raid, W. Khedr, M. A. El-Dosuky, and W. Ahmed, "Jpeg image compression using discrete cosine transform-A survey," *arXiv preprint arXiv:1405.6147*, 2014.
- [9] J. Zhang and T. M. Le, "A new no-reference quality metric for JPEG2000 images," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 743-750, 2010.
- [10] S. Merrouche, B. Bondžulić, M. Andrić, and D. Bujaković, "Accuracy analysis of lossless and lossy disparity map compression," *Ain Shams Engineering Journal*, vol. 13, no. 3, p. 101580, 2022.
- [11] R. Gonzalez, R. Woods, and S. Eddins, "Digital image processing using Matlab" Gatesmark Publishing," ed: LLC, 2009.

- [12] K. Iida and H. Kiya, "Fuzzy Commitment Scheme-Based Secure Identification for JPEG XR Images under Various Compression Ratios," *ITE Transactions on Media Technology and Applications*, vol. 5, no. 2, pp. 67-76, 2017.
- [13] K.-L. Hua *et al.*, "Reduction of Artefacts in JPEG-XR Compressed Images," *Sensors*, vol. 19, no. 5, p. 1214, 2019.
- [14] X. Chen, Y. Yuan, Y. Liu, and R. Cao, "High compression ratio static image coding technologies," *Acta Tech CSAV (ceskoslovensk Akademie Ved)*, vol. 62, no. 2, pp. 273-281, 2017.
- [15] S. S. Jadhav and S. K. Jadhav, "JPEG XR an image coding standard," *International Journal of Computer and Electrical Engineering*, vol. 4, no. 2, p. 137, 2012.
- [16] J. Yu, X. Gao, D. Tao, X. Li, and K. Zhang, "A unified learning framework for single image super-resolution," *IEEE Transactions on Neural networks and Learning systems*, vol. 25, no. 4, pp. 780-792, 2013.
- [17] 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.
- [18] R. Franke, "Scattered data interpolation: tests of some methods," *Mathematics of computation*, vol. 38, no. 157, pp. 181-200, 1982.
- [19] J. Allebach and P. W. Wong, "Edge-directed interpolation," in *Proceedings of 3rd IEEE International Conference on Image Processing*, 1996, vol. 3: IEEE, pp. 707-710.
- [20] F. N. Fritsch and R. E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp. 238-246, 1980.
- [21] J. Liu, Z. Gan, and X. Zhu, "Directional bicubic interpolation—A new method of image super-resolution," in *3rd International Conference on Multimedia Technology (ICMT-13)*, 2013: Atlantis Press, pp. 463-470.
- [22] A. Singh and J. Singh, "Survey on single image based super-resolution—implementation challenges and solutions," *Multimedia Tools and Applications*, vol. 79, no. 3, pp. 1641-1672, 2020.

- [23] K. Zhang, X. Gao, D. Tao, and X. Li, "Single image super-resolution with multiscale similarity learning," *IEEE transactions on neural networks and learning systems*, vol. 24, no. 10, pp. 1648-1659, 2013.
- [24] Y. Tian, F. Zhou, W. Yang, X. Shang, and Q. Liao, "Anchored neighborhood regression based single image super-resolution from self-examples," in *2016 IEEE International Conference on Image Processing (ICIP)*, 2016: IEEE, pp. 2827-2831.
- [25] Y. Tang and Y. Yuan, "Learning from errors in super-resolution," *IEEE Transactions on Cybernetics*, vol. 44, no. 11, pp. 2143-2154, 2014.
- [26] Y. Tang and L. Shao, "Pairwise operator learning for patch-based single-image super-resolution," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 994-1003, 2016.
- [27] K. I. Kim and Y. Kwon, "Example-based learning for single-image super-resolution and JPEG artifact removal," 2008.
- [28] S. Gou, S. Liu, Y. Wu, and L. Jiao, "Image super - resolution based on the pairwise dictionary selected learning and improved bilateral regularisation," *IET Image Processing*, vol. 10, no. 2, pp. 101-112, 2016.
- [29] P. P. Gajjar and M. V. Joshi, "New learning based super-resolution: use of DWT and IGMRF prior," *IEEE Transactions on Image Processing*, vol. 19, no. 5, pp. 1201-1213, 2010.
- [30] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *International journal of computer vision*, vol. 40, no. 1, pp. 25-47, 2000.
- [31] 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.
- [32] 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.
- [33] C. Dong, C. C. Loy, and X. Tang, "Accelerating the super-resolution convolutional neural network," in *European conference on computer vision*, 2016: Springer, pp. 391-407.

- [34] Y. Liu, "Quality and Rate Control of JPEG XR," 2016.
- [35] J. Zhang, M. Liu, X. Wang, and C. Cao, "Residual Net Use on FSRCNN for Image Super-Resolution," in *2021 40th Chinese Control Conference (CCC)*, 2021: IEEE, pp. 8077-8077.



## VITA

**NAME** Taingliv Min

**DATE OF BIRTH** 29 November 1998

**PLACE OF BIRTH** Phnom Penh, Cambodia

**INSTITUTIONS  
ATTENDED** Chulalongkorn University (2020-present)  
Institute of Technology of Cambodia (2015-2020)

**HOME ADDRESS** 488, 18 Phetchaburi road, Thanon Phetchaburi,  
Ratchathewi, Bangkok, 10400



จุฬาลงกรณ์มหาวิทยาลัย  
CHULALONGKORN UNIVERSITY