Learning-based Wavelet-like Transforms For Fully Scalable and Accessible Image Compression

Author:
Li, Xinyue

Publication Date:
2023

DOI:
https://doi.org/10.26190/unsworks/25308

Learning-based Wavelet-like Transforms For Fully Scalable and Accessible Image Compression

Xinyue Li

A thesis in fulfilment of the requirements for the degree of

Doctor of Philosophy

School of Electrical Engineering and Telecommunications

Faculty of Engineering

The University of New South Wales

Feb 2023

Declarations

ORIGINALITY STATEMENT

☑ I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.

COPYRIGHT STATEMENT

☑ I hereby grant the University of New South Wales or its agents a non-exclusive licence to archive and to make available (including to members of the public) my thesis or dissertation in whole or part in the University libraries in all forms of media, now or here after known. I acknowledge that I retain all intellectual property rights which subsist in my thesis or dissertation, such as copyright and patent rights, subject to applicable law. I also retain the right to use all or part of my thesis or dissertation in future works (such as articles or books).

For any substantial portions of copyright material used in this thesis, written permission for use has been obtained, or the copyright material is removed from the final public version of the thesis.

AUTHENTICITY STATEMENT

☑ I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.

Publications Statement

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in the candidate's thesis in lieu of a Chapter provided:

- The candidate contributed **greater than 50%** of the content in the publication and are the "primary author", i.e. they were responsible primarily for the planning, execution and preparation of the work for publication.
- The candidate has obtained approval to include the publication in their thesis in lieu of a Chapter from their Supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis.

☑ The candidate has declared that **some of the work described in their thesis has been published and has been documented in the relevant Chapters with acknowledgement**.

A short statement on where this work appears in the thesis and how this work is acknowledged within chapter/s:

> The work from "Neural Network Assisted Lifting Steps For Improved Fully Scalable and Accessible Lossy Image Compression in JPEG 2000" paper submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) covers the content from Chapter 7 to Chapter 11 of the thesis. The work from "Studies on Learning-based Lifting Schemes for Fully Scalable and Accessible Lossy Image Compression" paper submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI) covers the content from Chapter 12 to Chapter 13 of the thesis. These works have been cited properly within the relevant chapters, acknowledging the contribution of all authors (myself and my supervisors).

Candidate's Declaration

I declare that I have complied with the Thesis Examination Procedure.

Abstract

The goal of this thesis is to improve the existing wavelet transform with the aid of machine learning techniques, so as to enhance coding efficiency of wavelet-based image compression frameworks, such as JPEG 2000 [1].

In this thesis, we first propose to augment the conventional base wavelet transform with two additional learned lifting steps – a high-to-low step followed by a low-to-high step. The high-to-low step suppresses aliasing in the low-pass band by using the detail bands at the same resolution, while the low-to-high step aims to further remove redundancy from detail bands by using the corresponding low-pass band. These two additional steps reduce redundancy (notably aliasing information) amongst the wavelet subbands, and also improve the visual quality of reconstructed images at reduced resolutions.

To train these two networks in an end-to-end fashion, we develop a backward annealing approach to overcome the non-differentiability of the quantization and cost functions during back-propagation. Importantly, the two additional networks share a common architecture, named a proposal-opacity topology, which is inspired and guided by a specific theoretical argument related to geometric flow. This particular network topology is compact and with limited non-linearities, allowing a fully scalable system; one pair of trained network parameters are applied for all levels of decomposition and for all bit-rates of interest. By employing the additional lifting networks within the JPEG 2000 image coding standard, we can achieve up to 17.4% average BD bit-rate saving over a wide range of bit-rates, while retaining the quality and resolution scalability features of JPEG 2000.

Built upon the success of the high-to-low and low-to-high steps, we then study more broadly the extension of neural networks to all lifting steps that correspond to the base wavelet transform. The purpose of this comprehensive study is to understand what is the most effective way to develop learned wavelet-like transforms for highly scalable and accessible image compression. Specifically, we examine the impact of the number of learned lifting steps, the number of layers and the number of channels in each learned lifting network, and kernel support in each layer. To facilitate the study, we develop a generic training methodology that is simultaneously appropriate to all lifting structures considered. Experimental results ultimately suggest that to improve the existing wavelet transform, it is more profitable to augment a larger wavelet transform with more diverse high-to-low and low-to-high steps, rather than developing deep fully learned lifting structures.

Acknowledgement

First and foremost, I would like to express my sincere gratitude to my primary supervisor, Prof. David Taubman. I am very grateful to you for giving me this great and rare privilege to work under your supervision for my thesis work. You have always been supportive, encouraging and generous, from the beginning of my Ph.D. scholarship application to the last stage of writing this thesis. Your wisdom, your horizon and your immense knowledge have really shaped my mentality on research and my career. Your motivation, your patience and your invaluable feedbacks give me hope in many difficult times of my Ph.D. study and research. This is absolutely the best degree that I have ever had, and this beautiful and joyful journey would not be possible without you. I will cherish your mentorship and everything that you have taught me forever.

My sincere thanks also go to my joint supervisor, Dr. Aous Naman. You have always been patient, and your guidance helps me in all the time of my research and in writing of all publications, including this thesis. Your insightful comments and suggestions have always inspired me and widened my horizons from various perspectives. You have always been extremely supportive and responsive; every time that I encounter any issue in my study and research, I always know that you will be there to help. Without you, it would not be possible for me to complete my Ph.D. degree in a timely manner.

I am also grateful to my secondary supervisor, A/Prof. Elias Aboutanios and my colleague Dr. Reji Mathew, for their support and invaluable feedbacks. Although our interactions are limited due to the way my research evolves, I wish we could work closely in the future.

More importantly, I would like to express my deepest appreciation to my mother Zhongqin Guo, my grandma Zhixue Gui and my partner Chaoran Wang, for their unconditional love, their unrequited support and their long-term understanding throughout my Ph.D. research. Whenever I doubt myself and question my ability to complete this degree, you always believe, encourage and comfort me in many ways. You have always been the most powerful backing for me, and I would not have gone through this journey without you all.

In the end, I am also thankful to the University of New South Wales (UNSW) SYDNEY, who provides the Tuition Fee Scholarship and later on the Research Training Program for me. I would not be able to conduct this research without your financial assistance.

Publications and Patent

Journal Papers

- X. Li, A. Naman and D. Taubman. "Neural Network Assisted Lifting Steps For Improved Fully Scalable Lossy Image Compression in JPEG 2000," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, submitted for publication.
- X. Li, A. Naman and D. Taubman. "Studies on Learning-based Lifting Structures For Fully Scalable and Accessible Lossy Image Compression," *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, submitted for publication.

Conference Papers

- X. Li, A. Naman and D. Taubman, "Adaptive Secondary Transform For Improved Image Coding Efficiency In JPEG2000," *IEEE International Conference on Image Processing (ICIP)*, Abu Dhabi, United Arab Emirates, 2020, pp. 1216-1220, doi: 10.1109/ICIP40778.2020.9190654.
- X. Li, A. Naman and D. Taubman, "Machine-Learning Based Secondary Transform for Improved Image Compression in JPEG2000," *IEEE International Conference on Image Processing (ICIP)*, Anchorage, AK, USA, 2021, pp. 3752-3756, doi: 10.1109/ICIP42928.2021.9506122.
- X. Li, A. Naman and D. Taubman, "A Neural Network Lifting Based Secondary Transform for Improved Fully Scalable Image Compression in Jpeg 2000," *IEEE International Conference on Image Processing (ICIP)*, Bordeaux, France, 2022, pp. 1606-1610, doi: 10.1109/ICIP46576.2022.9897986.

Patent

- TAUBMAN, David Scott; NAMAN, Aous Thabit; LI, Xinyue, "Method, apparatus and computer readable medium for encoding an image," *Australian Provisional Patent Application* No. 2022902529, Filed on 2 September 2022.

Contents

List of Figures

List of Tables

List of Symbols

Δ	Quantization step size
Δ_β	Quantization step size for subband B_β
λ_1	The Lagrange multiplier which trades off distortion D and total coded length L
λ_2	The Lagrange multiplier which controls the emphasis on aliasing constraint
$\mathbf{A}$	The adjacency matrix of a graph $\mathcal{G}$
$\mathbf{c}$	The compressed bit-stream
$\mathbf{D}$	The diagonal degree matrix of a graph $\mathcal{G}$
$\mathbf{L}$	A combinatorial or unnormalized graph Laplacian
$\mathbf{s}_i$	The basis vectors of the one-dimensional DCT
$\mathbf{y}^{'}_{HH,d}$	The quantized HH band at resolution d
$\mathbf{y}^{'}_{HL,d}$	The quantized HL band at resolution d
$\mathbf{y}^{'}_{LH,d}$	The quantized LH band at resolution d
$\mathbf{y}^{'}_{LL,d}$	The quantized low-pass LL band at resolution d
$\mathbf{y}_{HH}$	The HH subband of the conventional wavelet transform
$\mathbf{y}_{HL}$	The HL subband of the conventional wavelet transform
$\mathbf{y}_{LH}$	The LH subband of the conventional wavelet transform
$\mathbf{y}_{LL}$	The LL subband of the conventional wavelet transform
$\mathcal{A}_H(x)$	The analysis of signal x into a high-pass band
$\mathcal{A}_L(x)$	The analysis of signal x into a low-pass band
$\mathcal{C}_0$	The non-empty intersection of closed convex sets

$\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_m$ The m well-defined closed convex sets

$\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ An underlying graph of a digital image with vertices $\mathcal{V}$ connected through edges $\mathcal{E}$ with weights $\mathbf{W}$

$\mathcal{L}^2(\mathbb{R})$ The Hilbert Space

$\mathcal{P}^H$ The horizontal-predict operator

$\mathcal{P}^V$ The vertical-predict operator

$\mathcal{P}_i$ The projection operator onto the convex set $\mathcal{C}_i$

$\mathcal{T}^A_{H2L}$ The adaptive high-to-low step

$\mathcal{T}^A_{L2H}$ The adaptive low-to-high step

$\mathcal{T}^W_{H2L}$ The linear high-to-low step

$\mathcal{T}^W_{L2H}$ The linear low-to-high step

$\mathcal{U}^H$ The horizontal-update operator

$\mathcal{U}^V$ The vertical-update operator

$\mathcal{V}^d$ The approximation space at resolution d of the Hilbert Space

$\mathcal{W}^d$ The detail space at resolution d of the Hilbert Space

$\overline{\mathbf{y}}_{HH,d}$ The less redundant HH band at level d

$\overline{\mathbf{y}}_{HL,d}$ The less redundant HL band at level d

$\overline{\mathbf{y}}_{LH,d}$ The less redundant LH band at level d

$\overline{\mathbf{y}}_{LL,d}$ The "cleaned" low-pass band at level d

$\overline{M^{-1}}$ The decompressor

$\overline{Q_\beta^{-1}}$ The dequantizer for subband B_β

$\overline{Q^{-1}}$ The dequantizer

$\overline{T^{-1}}$ The approximate inverse transform

$\widetilde{\mathbf{y}}^t_{LL,d}$ The aliasing model for the LL band at resolution d

$\widetilde{\mathbf{y}}_{HH,d}$ The redundant information within the HH band at level d

$\widetilde{\mathbf{y}}_{HL,d}$ The redundant information within the HL band at level d

$\widetilde{\mathbf{y}}_{LH,d}$ The redundant information within the LH band at level d

$\widetilde{\mathbf{y}}_{LL,d}$ The aliasing information within the low-pass band at level d

$\widetilde{Q}_\beta$	The continuous relaxation of the quantizer Q_β for subband B_β
A_X	The set of all possible outcomes of the random process X
b_{kj}	The bias between the k^{th} neuron at the previous layer and the j^{th} neuron at the current layer
$f_{s_2}(s_1)$	The horizontal cross-section of a continuous and consistently oriented signal $f(s_1, s_2)$ at the vertical position s_2
$G_{a,\beta}$	The extra analysis gain introduced by neural networks in addition to the base wavelet transform for subband B_β
$G_{s,\beta}$	The extra synthesis gain introduced by neural networks in addition to the base wavelet transform for subband B_β
$l_{i,\beta}$	The coded length resulting from the coding of quantization indices $q_{i,\beta}$ for all subbands B_β
M	The compressor
net_j	The inputs to the j^{th} neuron at a certain hidden layer
o_j	The output of the j^{th} neuron at a certain hidden layer
Q	The quantizer
Q_β	The quantizer for subband B_β
$q_{i,\beta}$	The quantization indices within subband B_β
T	The transform
T^{-1}	The inverse transform
T_β	The look-up table for mapping the quantization indices $q_{i,\beta}$ to the corresponding coded length $l_{i,\beta}$
w_{kj}	The weights between the k^{th} neuron at the previous layer and the j^{th} neuron at the current layer

Chapter 1

Introduction

1.1 Problem Statement

The wavelet transform has been successfully employed in a variety of codecs and open image compression standards; examples include JPEG 2000 [13] [1], the BBC's VC2 codec, and JPEG-XS [14]. The wavelet transform provides a balance between energy compaction and sparsity preservation, by analyzing the image with a hierarchical family of compact support operators, realized through successive filtering and down-sampling. Importantly, the wavelet transform naturally produces a multi-resolution representation of the image, which enables reconstructions at dyadically-spaced image resolutions, a feature known as *resolution scalability*.

Although the wavelet transform provides excellent energy compaction for horizontal and vertical edges, slanted features are poorly characterized by the separable wavelet filters, which leads to significant redundancy between all subbands and visually disturbing artifacts in the reconstructed images along diagonal edges. Solutions have been explored to improve directional sensitivity of the wavelet transform, which can be broadly categorized into traditional approaches and machine-learning based methods.

In the traditional approaches, oriented transforms [15–21] employing directional filter banks are proposed to capture geometric structures within an image. However, when such schemes are employed for image compression, the orientation information needs to be explicitly coded and communicated in order to correctly inverse the respective transform.

In the last decade, researchers experimented with machine-learning based approaches to improve coding efficiency in image compression applications, with very promising results. For wavelet-based image compression, which is the topic of this thesis, examples include [22–26]. These methods inherit the multi-scale representation from the wavelet transform, which provides resolution scalability; however, none of them explore quality scalability or region-of-interest accessibility. Additionally these works do not investigate ways to directly train neural networks for rate-distortion objectives; instead, alternative training objectives, such as energy compaction of the transformed coefficients or prediction residuals, are used as proxies for coding efficiency.

In contrast, some researchers experimented with image compression designs that employ neural networks only; these designs usually adopt an end-to-end optimization that explicitly targets rate-distortion objectives [8, 27–31]. Even though these end-to-end schemes achieve significantly better compression results, they lack resolution scalability, quality scalability and region-of-interest accessibility of wavelet-based compression frameworks. They also have significantly higher computational complexity and huge receptive field in the image domain. Additionally, network structures and trained parameters are mostly dependent on the target compression bit-rates.

As a result, in this thesis, we aim to develop low-complexity learning-based wavelet-like compression schemes, which inherit quality scalability, resolution scalability and random region-of-interest accessibility from the conventional wavelet transform. In these proposed schemes, our target is to achieve high coding efficiency and scalability by applying only one set of trained networks to all levels of decomposition and to all bit-rates of interest over a wide range. In addition, we also aim to take advantage of end-to-end optimization to jointly train all networks for optimized rate-distortion performance.

1.2 Our Contributions

The contributions of this thesis are carried out in two stages. In the first stage, we only consider augmenting the conventional base wavelet transform with two additional learned lifting steps. This first stage makes the following contributions:

- The disclosure of the role of geometric flow in untangling redundancy (notably aliasing information) between successive levels of the wavelet transform. We also demonstrate how this specific theoretical argument on geometric flow is connected with super resolution and optimized reconstruction problems.
- The development of three structures for reducing redundancy in the wavelet transform, named low-to-high, high-to-low and hybrid approaches.
- The development of a proposal-opacity network topology, which is inspired and guided by the underlying hypothesis about geometric flow. This structure involves a collection of purely linear filters, and so comes with fairly low computational complexity and a relatively small region of support in the image domain.
- The development of the end-to-end optimization framework with a backward annealing approach to manage discontinuities in quantization and cost functions during training, so as to jointly optimize all networks for rate-distortion performance.

Built upon the success of the first stage, we then study more broadly the incorporation of neural networks to all lifting steps that correspond to the base wavelet transform in the second stage. The contributions of this second stage are:

- A comprehensive study on what can be achieved by learning-based wavelet-like transforms, with respect to the depth of lifting structures, the diversity of lifting networks and region of support in the image domain.
- The development of a particular training schedule, which utilizes pre-defined oracle opacities to initialize and to progressively train all proposal-opacity lifting networks

in the end-to-end optimization framework.

- The recommendations on how to strategically deploy neural networks for improving wavelet-based image compression systems, especially for practical applications.

1.3 Outline of This Thesis

The rest of this thesis is arranged as follows. Chapter 2 to Chapter 6 are mostly literature survey, while Chapter 7 to Chapter 13 represent contributions of this thesis.

Chapter 2 introduces the basic concepts of image compression, especially the perspective that every image compression system can be understood as vector quantization.

Chapter 3 reviews the classic feedforward compression structure, which simplifies the compressor of vector quantization into three elements – transformation, quantization and source coding. Our emphasis is on transformation, especially on the discrete wavelet transform, which we intend to improve in this thesis.

Chapter 4 reviews predictive feedback compression structures, which present benefits as well as principle weaknesses over the classic feedforward compression structure.

Chapter 5 reviews post-processing and inverse transform optimization (or in other words optimized reconstruction) problems. This review provides another perspective to view the work in this thesis.

Chapter 6 reviews existing autoencoders with different neural network architectures; these autoencoders do not present desirable features, such as quality scalability, resolution scalability and region-of-interest accessibility, which otherwise exist in this thesis. We also review existing end-to-end training strategies, whose weaknesses motivate the proposed learning strategy in this thesis.

Chapter 7 reveals a specific theoretical argument related to the opportunity presented by geometric flow to remove redundancy between successive levels of the wavelet decomposition.

Chapter 8 introduces three general structures: low-to-high, high-to-low and hybrid structures to augment and enhance the conventional wavelet transform.

Chapter 9 provides insights on how the underlying theory about geometric flow drives the proposal-opacity network topology that we eventually select.

Chapter 10 introduces the proposed end-to-end learning strategy with a backward annealing approach to overcome the non-differentiability of the quantization and cost functions during back-propagation. This particular learning strategy has advantages over existing end-to-end training approaches.

Chapter 11 discusses the experimental results on augmenting the conventional wavelet transform with additional lifting networks.

Chapter 12 studies more broadly the extension of neural networks to all lifting steps that correspond to the base wavelet transform. The main contribution of this chapter is the development of an oracle-opacity training schedule.

Chapter 13 provides the experimental results for the performance of learning-based wavelet-like transforms, with respect to the depth of lifting structures, the diversity of lifting networks and region of support in image domain.

Chapter 14 concludes this thesis and highlights future directions.

Chapter 2

Review: Basic Concepts of Image Compression

2.1 Digital Images

A grayscale image is a two-dimensional collection of samples (or *pixels*), which can be represented as

$$\mathbf{x} \equiv x[n_1, n_2], n_1 \in [0, N_1), n_2 \in [0, N_2) \tag{2.1}$$

where n_1 and n_2 denote the horizontal and vertical coordinates/indices of each pixel, within finite extents N_1 and N_2 respectively. The pixel value $x[n_1, n_2]$ reflects the intensity, or can be understood as the luminance, of the image at the coordinates $[n_1, n_2]$. This value is often a B-bit signed or unsigned integer as

$$x[n_1, n_2] \in \left\{0, 1, \cdots, 2^B - 1\right\} \quad \text{for unsigned images, and} \tag{2.2}$$

$$x[n_1, n_2] \in \left\{-2^{B-1}, -2^{B-1} + 1, \cdots, 2^{B-1} - 1\right\} \quad \text{for signed images;} \tag{2.3}$$

therefore, the image $\mathbf{x}$ requires $N_1 N_2 B$ bits to represent all pixel values without any compression. For natural images, $B = 8$ is the most commonly encountered representation; larger bit-depth, e.g. 12-bit or 16-bit, is commonly adopted for high dynamic range (HDR) imagery, including professional and consumer imagery. In many cases, these B-bit integer values are often uniformly scaled into the range $[0,1]$ (unsigned) or $[-\frac{1}{2},\frac{1}{2}]$ (signed) to form a floating point representation of images.

A colour image, on the other hand, is a collection of M image components as

$$x_m[n_1,n_2], m=1,2,\cdots,M \tag{2.4}$$

The typical color images have three values at each pixel location, corresponding to red, green and blue primary color components; these images are known as RGB images. In this thesis, we restrict our attention to grayscale images having only one image component, as we can always develop and apply a compression system to each image component separately. Moreover, it helps to avoid confusing our proposed spatial transforms with color dependent questions, such as the optimal choice of color transforms, etc.

2.2 Image Compression As Vector Quantization

As we discussed in Section 2.1, the image $\mathbf{x}$ would require N_1N_2B bits to represent all pixel values if no compression was involved. Since these N_1N_2B bits are considered to be redundant, the primary purpose of image compression is to minimize the number of bits $\|\mathbf{c}\|$, i.e. the length of the compressed *bit-stream* $\mathbf{c}$, that are needed to code the image $\mathbf{x}$; in the context of compression, $\|\mathbf{c}\|$ should be smaller than N_1N_2B, i.e., $\|\mathbf{c}\| < N_1N_2B$.

In general, image compression can be categorized into *lossless compression* and *lossy compression*. Using Fig. 2.1 as guidance, lossless image compression requires the decompressor $\overline{M^{-1}}$ to be the perfect inverse of the compressor M; i.e., $\overline{M^{-1}} = M^{-1}$. Therefore, no *distortion* is introduced to the reconstruction; i.e., $\hat{\mathbf{x}} \equiv \mathbf{x}$. This perfect reconstruction is often demanded in scenarios where reconstruction errors cannot be tolerated, such as medical applications for diagnosis. However, it also imposes restrictions on the capability of the compression system to achieve higher compression performance for other applications with only limited storage space.

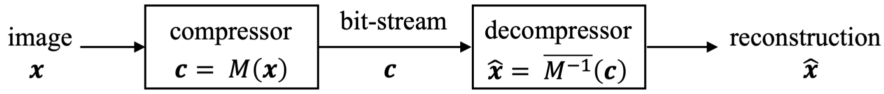


Figure 2.1: The global mapping operation of compressing the image $\mathbf{x}$.

For lossy image compression, which is the topic of this thesis, the compressor M is not invertible and the decompressor $\overline{M^{-1}}$ is only an approximate inverse of M; i.e., $\overline{M^{-1}} \approx M^{-1}$. Therefore, the distortion $D(\mathbf{x}, \hat{\mathbf{x}})$ must be present in the reconstruction. By allowing some level of distortion, we expect that the image $\mathbf{x}$ can be represented with even fewer bits compared with lossless compression. In fact, the amount of distortion introduced and the number of bits required for coding in lossy compression are closely related and are bounded by the *rate-distortion theory*; we shall see this more concretely in Section 3.4.

Essentially, the compressor M can be understood as an enormous look-up table with $2^{N_1 N_2 B}$ entries, which maps the image $\mathbf{x}$ into the bit-stream $\mathbf{c}$. If we assume the length of this compressed bit-stream $\|\mathbf{c}\|$ is fixed, then the decompressor $\overline{M^{-1}}$ can also be regarded as another enormous look-up table with $2^{\|\mathbf{c}\|}$ entries, mapping the bit-stream $\mathbf{c}$ into the reconstructed image $\hat{\mathbf{x}}$. The optimal bit-stream $\mathbf{c}$ can be constructed by selecting the bit-stream whose corresponding reconstructed image best approximates the original image $\mathbf{x}$ with minimum distortion; that is, to find

$$\arg\min_{\mathbf{c}} \; D(\mathbf{x}, \hat{\mathbf{x}}) = \arg\min_{\mathbf{c}} \; D\left(\mathbf{x}, \overline{M^{-1}}(\mathbf{c})\right) \tag{2.5}$$

Equation (2.5) is essentially the core concept of Vector Quantization (VQ); this conveys an important message that conceptually every compression system can be viewed as a vector quantizer. However, the size of the look-up table associated with the decompressor $\overline{M^{-1}}$ grows exponentially with the increase of the image size $N_1 N_2$; this makes vector quantization unrealistic for practical image compression applications.

Therefore, in practice, it is necessary to simplify the global mapping operations M and $\overline{M^{-1}}$ by decomposing them into elements that are interrelated. This motivates the classic feedforward image compression structure as described in Chapter 3.

Chapter 3

Review: Classic Feedforward Compression Structure

The classic feedforward structure for image compression is illustrated in Fig. 3.1. In this structure, the compressor M is composed of three elements: transform, quantization and coding; each of these elements has the ability to exploit statistical redundancy within the input image $\mathbf{x}$ or within the outputs of the previous stage, so as to minimize the length of the bit-stream $\|\mathbf{c}\|$ to achieve compression. Subsequently, the decompressor $\overline{M^{-1}}$ employs inverse transform, dequantization and decoding as the corresponding inverse elements to reconstruct the original image with some level of distortion. This classic structure is the most commonly encountered image compression structure, and is essentially the structure that we employ in this thesis.

3.1 Transform

The purpose of the transform is to *decorrelate* the statistical dependencies that exist in the original image $\mathbf{x}$; this is achieved by mapping the original image samples $\mathbf{x}$ into a new set of coefficients $\mathbf{y}$ using some operator T. The resulting transform coefficients $\mathbf{y}$

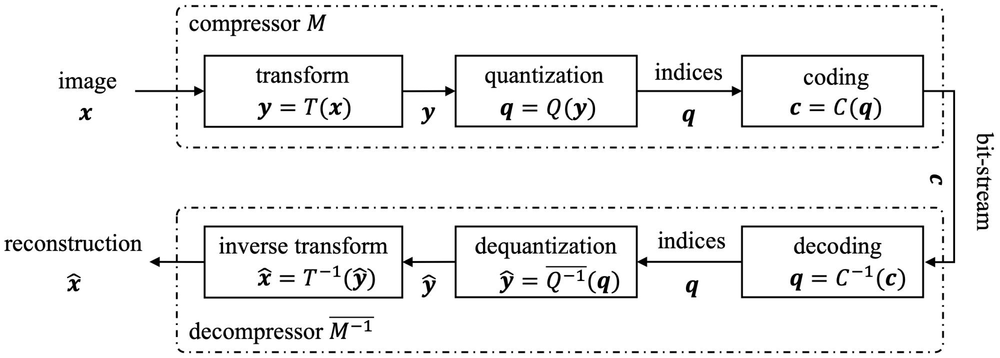


Figure 3.1: The classic feedforward image compression structure.

exhibit considerably less redundancies – in the ideal case, they could even be statistically independent – therefore can be represented and compressed more efficiently.

The transform T employed in this classic feedforward compression structure has two important properties. First, it is always invertible, therefore its inverse transform T^{-1} employed at the decompressor does not introduce any distortion; this is in contrast to the autoencoders in Chapter 6, in which the learning-based transforms are usually non-invertible and so do not have the ability to accurately represent the image $\mathbf{x}$ without any loss. Secondly, the transform T is usually linear; that means the synthesis vectors of the transform can be generally understood as a set of basis vectors. Since the number of non-zero transform coefficients should be limited for the purpose of image compression, this implies that the image $\mathbf{x}$ needs to be approximated well using only a small number of basis vectors. Additionally, this approximation should be well matched to the statistics of the image $\mathbf{x}$.

Apart from invertibility and linearity, there are many other considerations taken into account when designing the transform T within this classic compression structure. From the analysis (compressor) perspective, the transform design needs to consider *sparsity* and *energy compaction* of the basis representation; a good set of basis vectors has the ability to compact the energy of the image $\mathbf{x}$ into only a small set of significant coefficients, so that the other coefficients can be quantized to zero. The importance of this is revealed by recognizing that every non-zero quantized sample will consume significantly more than 1 coded bit with simple scalar quantization (Section 3.2), since the sign at least is usually uniformly distributed. From the synthesis (decompressor) aspect, the transform design also needs to consider the way that quantization errors expand through the inverse transform T^{-1} during reconstruction. These considerations essentially impose certain constraints on the design of the transform T or equivalently the inverse transform T^{-1} within this classic compression structure; this is in contrast to the autoencoders in Chapter 6, where the relationship between the transform T and the inverse transform T^{-1} is usually unconstrained, although still subject to a coding efficient objective that takes distortion into account.

Since this thesis focuses primarily on improving the existing wavelet transform, we find it necessary to review some transforms that have been widely adopted in the context of image compression, along with their respective weaknesses and benefits. Two representative transforms are explicitly addressed here: the Discrete Cosine Transform (DCT) [32] that is the basis of the JPEG image compression standard [2] and most video coding standards, and the Discrete Wavelet Transform (DWT) [33] that has been employed in the JPEG 2000 image compression standard [1].

3.1.1 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is a real-valued orthonormal transform, whose basis vectors are unit sampled cosine functions oscillating at different frequencies [32]. Specifically, the basis vectors $\mathbf{s}_i$ of the N-point one-dimensional DCT are defined as

$$s_{i,p} = k_i \cos\left(2\pi f_i \left(p + \frac{1}{2}\right)\right); \;\; f_i = \frac{i}{2N} \tag{3.1}$$

where k_i is the normalization scalar, defined as in (3.2), to ensure $\|\mathbf{s}_i\| = 1$ while f_i denotes the frequency of each cosine function $s_{i,p}$.

$$k_i = \begin{cases} \sqrt{\frac{1}{N}} & \text{if } i = 0 \\ \sqrt{\frac{2}{N}} & \text{if } i \neq 0 \end{cases} \tag{3.2}$$

The basis vectors of the two-dimensional DCT, therefore, can be defined as the separable extension of the one-dimensional DCT as

$$(s_{i_1,i_2})_{p_1,p_2} = k_{i_1} k_{i_2} \cos\left(2\pi f_{i_1} \left(p_1 + \frac{1}{2}\right)\right) \cos\left(2\pi f_{i_2} \left(p_2 + \frac{1}{2}\right)\right) \tag{3.3}$$

The DCT has many desirable features, particularly for image compression applications. A key attribute of the DCT is its ability to approximately diagonalize the covariance matrix of the source signal as N becomes large. This means that the DCT is a robust approximation to the optimal transform [34] – the Karhunen-Loeve transform (KLT) , which perfectly decorrelates the source signal [35]; decorrelation is a necessary condition for statistical independence, and so is usually taken as an objective for good linear transforms. The advantage of the DCT is that the DCT does not require adaptation to the statistics of the source signal, unlike the KLT; this is also the main reason why the KLT is not widely employed in practical image compression standards, despite being the optimal decorrelation transform.

Another important attribute of the DCT is its *energy compaction* property; it compacts most of the source signal energy into lower frequency coefficients, whereas higher frequency coefficients are small and can be quantized to nearly zero. This means that only a small number of quantized DCT coefficients needs to be coded compared to the original image samples, leading to higher coding efficiency. This feature is present in the Discrete Fourier Transform (DFT) as well, which also asymptotically converges to the KLT with increasing block size for a wide-sense stationary (WSS) random process. However, for small block sizes, the DCT converges faster and so presents better energy compaction than the DFT due to the lack of boundary transients.

Moreover, the DCT and the DFT have other things in common, as they both employ unit sampled sinusoids to map the source signal from the spatial domain to the frequency domain for decorrelation. The similarities lead to another desirable feature of the DCT; it can be efficiently computed with fast algorithms, e.g. the Fast Fourier Transform (FFT) algorithms that have been well developed for the DFT. In practice, these fast algorithms are usually block-based; the most commonly used block size is 8 x 8, which has been adopted by the JPEG image compression standard [2].

This block-based implementation, however, results in two principle weaknesses when the DCT is employed for image compression: 1) the loss of the opportunity to exploit statistical redundancy between different blocks, if they are coded independently; and 2) severe block

Figure 3.2: Severe block artifacts in the reconstructed image when the DCT is employed for image compression at low bit-rate (0.044 bpp).

artifacts in the reconstructed image, especially at low bit-rates, as seen in Fig. 3.2. The first issue is addressed to some extent in predictive coding methods; examples can be found in [36–38]. However, these methods cannot modify the visual structure of the quantization errors in the DCT; the errors are still expanded into the reconstructed image through the DCT basis functions, which are block-like and so cause block artifacts.

3.1.2 Discrete Wavelet Transform

3.1.2.1 Wavelets and Multi-resolution Analysis

The main limitation of the DCT is the block-based implementation, which means only samples within each independent block are decorrelated. However, there is no fundamental reason for us to believe that the neighbouring blocks are completely uncorrelated; in fact, neighbouring samples are highly correlated in natural images.

The wavelet transform, in contrast, does not suffer from this limitation; instead, it is implemented as a sliding window on the entire image, which effectively avoids blocking artifacts in the reconstructed image. More importantly, the wavelet transform serves as a powerful tool to analyse the source image with varying features over different scales; this is known as *multi-resolution analysis*. This multi-resolution analysis is of particular interest for image compression, as it enables reconstructions at dyadically-spaced image resolutions, a feature known as *resolution scalability*.

One way to understand the multi-resolution analysis property of wavelets is to suppose that we start with a continuous signal $x^0(t)$ at a certain resolution $\mathcal{V}^0$ of the Hilbert Space $\mathcal{L}^2(\mathbb{R})$, which can be characterized by the discrete sequence $y_0^0[n]$ as

$$x^0(t) = \sum_{n\in\mathbb{Z}} y_0^0[n]\varphi(t-n) = \sum_{n\in\mathbb{Z}} y_0^0[n]\varphi_n(t) \tag{3.4}$$

The symbol φ_n represents the basis function that spans the sub-space $\mathcal{V}^0$; the function $\varphi(t)$ is also known as the *scaling function*.

Suppose the space $\mathcal{V}^0$ can be further decomposed into another two orthogonally complementary sub-spaces, $\mathcal{V}^1$ (*the approximation space*) and $\mathcal{W}^1$ (*the detail space*), such that

$$\mathcal{W}^1 \perp \mathcal{V}^1 \quad \text{and} \quad \mathcal{W}^1 \oplus \mathcal{V}^1 = \mathcal{V}^0 \tag{3.5}$$

Then the signal $x^0(t)$ in (3.4) can be decomposed accordingly into low-pass sequence $y_0^1[n]$ and high-pass sequence $y_1^1[n]$ as

$$\begin{aligned} x^0(t) &= \sum_{n\in\mathbb{Z}} y_0^0[n]\varphi_n(t) \\ &= \sum_{n\in\mathbb{Z}} y_0^1[n]\varphi_n^1(t) + \sum_{n\in\mathbb{Z}} y_1^1[n]\psi_n^1(t) \end{aligned} \tag{3.6}$$

where φ_n^1 and ψ_n^1 are the basis functions of the sub-space $\mathcal{V}^1$ and $\mathcal{W}^1$, respectively.

If we continue recursively decomposing the low-pass sequence y_0^1, y_0^2 and so forth for D levels, eventually the signal $x_0(t)$ can be spanned using the basis functions $\left\{\psi_n^d\right\}_{d,n\in\mathbb{Z}}$ when D becomes infinitely large, as seen in (3.7).

$$\begin{aligned} x^0(t) &= \sum_{n\in\mathbb{Z}} y_0^D[n]\varphi_n^D + \sum_{d=1}^{D}\sum_{n\in\mathbb{Z}} y_1^d[n]\psi_n^d \\ &\approx \sum_{d=1}^{D}\sum_{n\in\mathbb{Z}} y_1^d[n]\psi_n^d, \text{ when } D \to \infty \end{aligned} \tag{3.7}$$

This reveals the core idea of the wavelet transform; that is to find the wavelet basis functions $\left\{\psi_n^d\right\}_{d,n\in\mathbb{Z}}$ that span all sub-spaces of $\mathcal{L}^2(\mathbb{R})$ as a multi-resolution hierarchy, and are all translated and dilated versions of a single mother wavelet $\psi(t)$. This mother wavelet $\psi(t)$ is also known as the *wavelet function*.

3.1.2.2 Wavelets from Subband Transform

In practice, the wavelet basis is usually constructed using subband transforms. To see the intimate connection between subband transforms and wavelet transforms, suppose the signal $x^0(t)$ in (3.4) is decomposed into the low-pass sequence $y_0^1[n]$ and the high-pass sequence $y_1^1[n]$ using a two-channel subband transform with analysis low-pass and high-pass filters, h_0 and h_1. Then $x^0(t)$ can be reconstructed using the synthesis low-pass and high-pass filters, g_0 and g_1, of the same subband transform as

$$\begin{aligned}
x^0(t) &= \sum_{n\in\mathbb{Z}} y_0^0[n]\varphi_n(t) \\
&= \sum_{n\in\mathbb{Z}} \left(\sum_{i\in\mathbb{Z}} y_0^1[i]g_0[n-2i] + \sum_{j\in\mathbb{Z}} y_1^1[j]g_1[n-2j] \right) \varphi_n(t) \\
&= \sum_{i\in\mathbb{Z}} y_0^1[i] \left(\sum_{n\in\mathbb{Z}} g_0[n]\varphi_n(t-2i) \right) + \sum_{j\in\mathbb{Z}} y_1^1[j] \left(\sum_{n\in\mathbb{Z}} g_1[n]\varphi_n(t-2j) \right) \\
&= \sum_{i\in\mathbb{Z}} y_0^1[i]\varphi_i^1(t) + \sum_{j\in\mathbb{Z}} y_1^1[j]\psi_j^1(t) \qquad (3.8)
\end{aligned}$$

We can see that the subband decomposition of $x^0(t)$ in (3.8) yields exactly the same equation as the wavelet decomposition in (3.6). This reveals the fact that subband transforms can serve as a vehicle for constructing the wavelet basis.

If we continue recursively decomposing the low-pass sequence y_0^1, y_0^2 and so forth for D levels using this subband analysis operation, a dyadic tree-structured subband transform results, as seen in Fig. 3.3. This tree-structured subband transform yields exactly the same decomposition of $x^0(t)$ as derived in (3.7); it essentially constructs the wavelet function $\psi(t)$ through infinite convolution of the subband filters $g_1[n]$ and $g_0[n]$, whose translates and dilates span $\mathcal{L}^2(\mathbb{R})$. For this reason, this subband transform is often known as the Discrete Wavelet Transform (DWT).

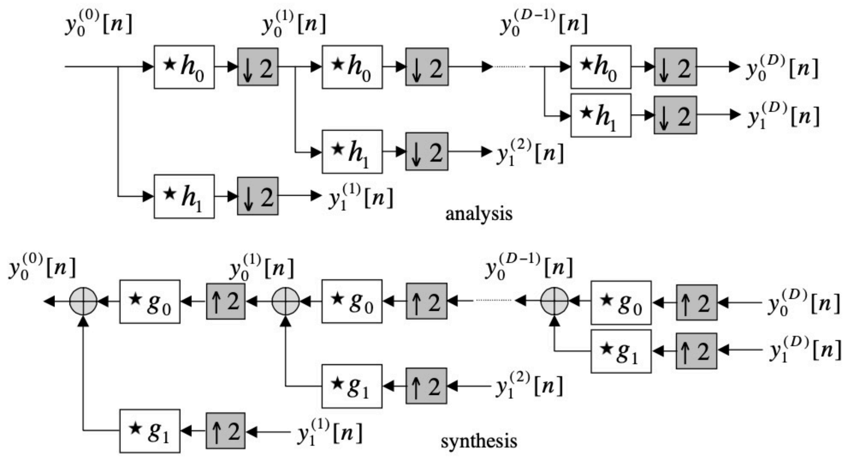


Figure 3.3: Dyadic tree-structured subband transform (i.e. the DWT) with analysis and synthesis operations for D levels.

Although subband transforms and wavelet transforms are often used interchangeably in the literature, it is important to highlight the fact that not every subband transform can be regarded as a wavelet transform. One important constraint is that infinite convolution of the subband filters must converge. This strict constraint on convergence reveals another important property of the wavelet transform, known as *self-similarity*. Of course there are other conditions for a subband transform to be a wavelet transform; interested readers can refer to [39–41] for more details.

3.1.2.3 Self-similar Property of the Wavelet Transform

To understand the self-similar property of the wavelet transform, we first restrict our attention to the continuous domain. As we have elaborated in Section 3.1.2.1, the underlying continuous basis functions of the wavelet transform are indeed scaled, and so are exactly self-similar in the hypothetical sub-spaces $\left\{\mathcal{V}^d\right\}_{d\in\mathbb{Z}}$ of the Hilbert Space $\mathcal{L}^2(\mathbb{R})$.

Subsequently, as we descend through multiple scales in the discrete domain in Section 3.1.2.2, the samples of the discrete basis functions of the DWT asymptotically converge to the samplings of the underlying continuous wavelet or scaling functions of the wavelet transform. This implies that the starting basis at the finest resolution becomes asymptotically irrelevant, because the resulting difference in the continuous wavelet basis functions only affects the high frequency components, which gradually disappear at lower resolutions. In other words, the hypothetical multi-scale model in the continuous domain as well as its self-similar property become valid in the discrete domain below the finest resolution.

This self-similar property is very useful, because in many cases natural images also exhibit a self-similar property across resolutions; this is known as a *fractal property* of natural images [42]. More importantly, this thesis is underpinned by the self-similar property of the wavelet transform, as it opens the opportunity for us to construct a fully scalable system, which only employs one set of networks (transforms) for all levels of the wavelet decomposition.

3.1.2.4 The Lifting Scheme

In contrast to the dyadic tree-structured implementation of the wavelet transform as described in Section 3.1.2.2, Wim Sweldens introduced an alternative implementation of the wavelet transform, called the *lifting scheme*. The key idea of the lifting scheme is to factorize the poly-phase matrix of the wavelet (or subband) filters into a sequence of elementary convolution steps, known as *lifting steps* [43–45]. This factorization effectively reduces the number of arithmetic operations by nearly a factor of two, leading to a fast implementation of the wavelet transform [46].

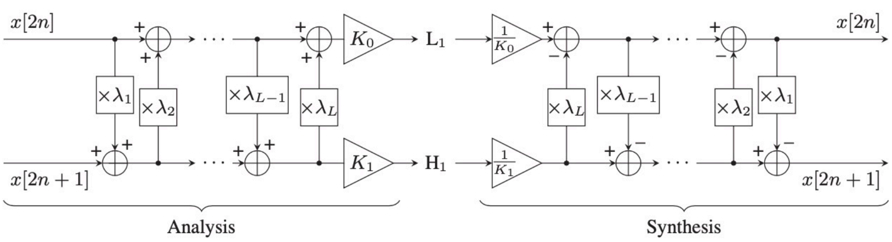


Figure 3.4: The structure of a typical lifting scheme of the wavelet transform.

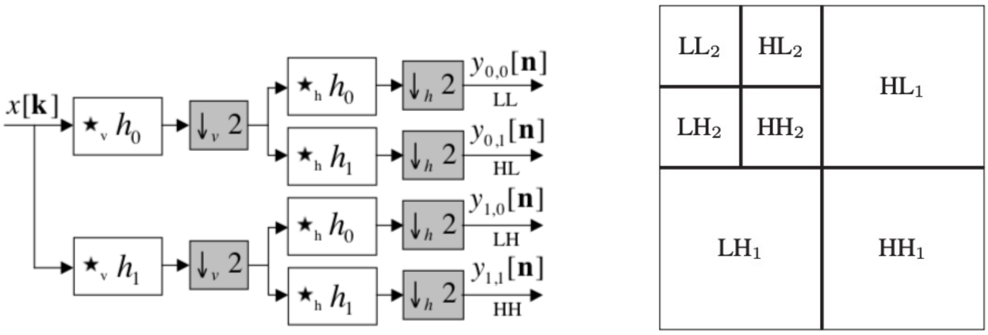
Fig. 3.4 illustrates the structure of a typical lifting scheme. The input signal $x[n]$ is first split into two sequences: even and odd samples, $x[2n]$ and $x[2n+1]$. The even samples $x[2n]$ are used to predict the odd samples $x[2n+1]$; this step is known as the *predict step*. Then the odd samples $x[2n+1]$, along with its prediction from $x[2n]$, are used to update the even samples $x[2n]$; this step is known as the *update step*. The predict and update steps alternate to produce the low-pass and high-pass subband, L_1 and H_1; the total number of lifting steps varies for different wavelet transforms. Of course, the same process can be recursively applied to the low-pass bands L_1, L_2 and so forth, to produce a multi-resolution wavelet representation of the input signal x in a cost-efficient way.

Apart from its low implementation cost, the lifting scheme also allows us to include (or employ) adaptive and non-linear elements in the wavelet transform, whose inverse transform can be easily found by inverting the lifting steps. This is of particular importance to this thesis, because we essentially utilize this property of the lifting scheme to design and implement our proposed lifting-based networks, as seen in Section 12.1. Interested readers can refer to [46] for more detailed explanations about other advantages of the lifting scheme.

3.1.2.5 The two-dimensional Discrete Wavelet Transform (2D-DWT)

Since the wavelet transform is separable, the 2D-DWT can be implemented by decomposing the input image $x[\mathbf{k}]$ first vertically and then horizontally, using the low-pass and the high-pass filters, h_0 and h_1, as illustrated in Fig 3.5(a). The subband $y_{0,0}[\mathbf{n}]$ is regarded as the *LL* subband as it captures low frequency along both the horizontal and the vertical directions. The subband $y_{0,1}[\mathbf{n}]$ is identified as the *HL* subband as it captures the horizontally-aligned high frequency components and the vertically-aligned low frequency components. Similarly, $y_{1,0}[\mathbf{n}]$ and $y_{1,1}[\mathbf{n}]$ are denoted as the *LH* and the *HH* subband, respectively. By further decomposing the LL bands, a two-dimensional multi-resolution wavelet analysis system results as illustrated in Fig. 3.5(b). To implement the 2D-DWT efficiently, the lifting scheme discussed in Section 3.1.2.4 is also applicable here; it is also the preferred implementation in modern wavelet-based image compression standards, such as JPEG 2000 [1].

Fig 3.6 illustrates an example of the 2D-DWT decomposition of a real image. We can observe that the LL band represents a lower resolution of the image. The HL bands mainly capture the vertical edges within the original image at different scales, while the LH and the HH bands capture the horizontal and the diagonal edges at each resolution respectively.



(a) (b)

Figure 3.5: The 2D-DWT analysis system.

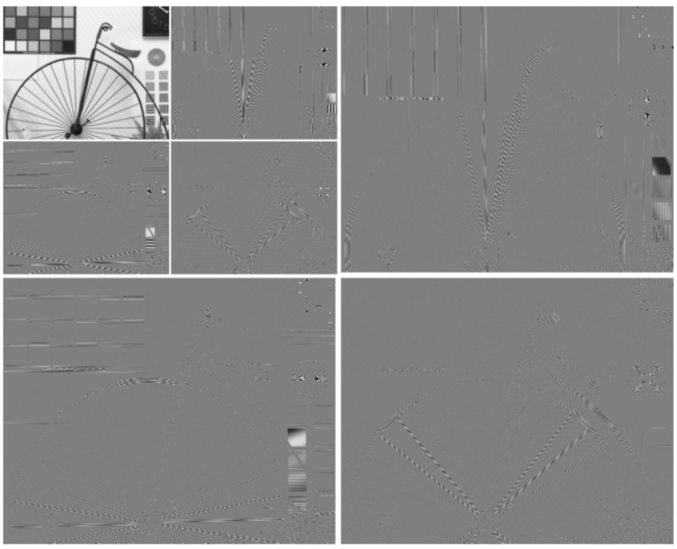
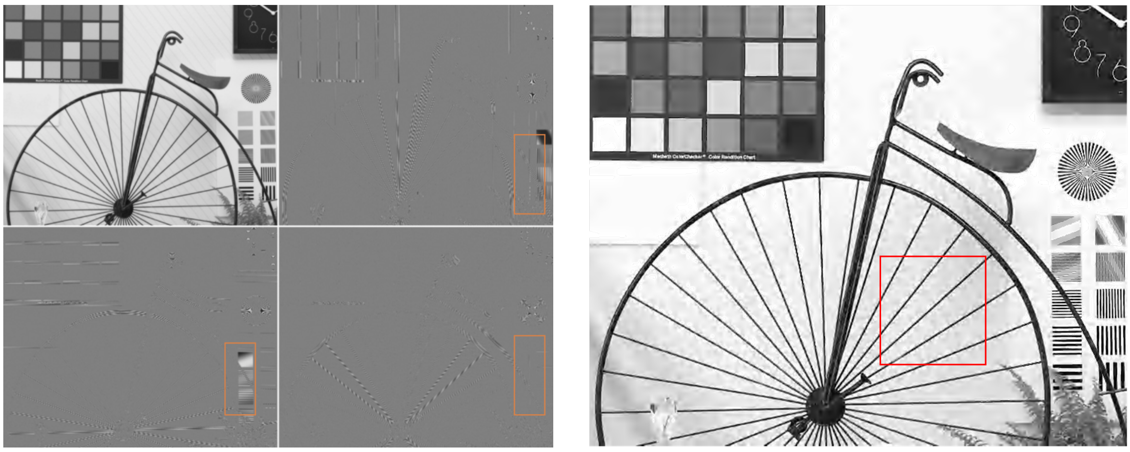


Figure 3.6: An example of the 2D-DWT decomposition of a real image.

3.1.2.6 Limitations of the Wavelet Transform

Although the wavelet transform provides excellent energy compaction for horizontal and vertical edges, slanted features are poorly characterized by the separable wavelet filters; that means these slanted edges are less sparse, spreading out and appearing in each subband, as exemplified in Fig. 3.7(a). More importantly, this poor directional sensitivity of the wavelet transform creates visually disturbing artifacts in the reconstructed images along directional edges, as illustrated in Fig. 3.7(b).



(a) (b)

Figure 3.7: Limits of the wavelet transform: (a) illustrates the slanted edges which are less sparse and appear in each subband; (b) illustrates the visually disturbing artifacts in the reconstructed image (at 0.1 bpp) along diagonal edges.

3.1.2.7 Existing Methods to Improve the Wavelet Transform

Since this thesis focuses primarily on improving the existing wavelet transform with the aid of neural networks, we find it useful to review some existing approaches that have been explored to overcome the limits of the wavelet transform; broadly, these methods can be categorized into traditional approaches and machine-learning based methods.

In the traditional approaches, oriented transforms are proposed to capture geometric structures within an image; examples include the Curvelet Transform [15, 16], the Ridgelet

Transform [17], the Contourlet Transform [18], the Bandelet Transform [19], directional wavelet transforms [20, 21], directional complex wavelet transform (CWT) [47–49] and dual-tree complex wavelet transform (DT-CWT) [50–52]. These methods involve constructing filters banks that are orientated at a variety of directions to capture smooth contours in images, so as to better characterize slanted edges. This is very important because heavily compressed images can only be represented by a relatively small number of non-zero coefficients; therefore, to capture oriented features accurately, the synthesis basis functions must also be able to exhibit similar orientations.

Furthermore, efforts are also invested to apply additional operations to the conventional wavelet transform, which effectively orient the wavelet basis functions to the direction of local geometric regularity. For example, Mehrseresht and Taubman [53] propose to employ a in-band shifting technique to each existing lifting step of the wavelet transform; this shifting operation essentially aligns the geometric features along the vertical and horizontal directions, so that each lifting step can be adapted to local image features. Similarly, Ding et al. [54] propose to incorporate directionally spatial prediction into the conventional lifting steps of the wavelet transform; each lifting step is applied at the direction which exhibits the strongest correlation, instead of only at horizontal and vertical directions, to improve directional sensitivity of the wavelet transform. Interested readers can refer to [55–58] for more related works.

However, when the aforementioned schemes are employed for image compression, they all encounter the same problem – orientation information needs to be explicitly coded and communicated in order to correctly inverse the respective transform.

In the last decade, researchers experimented with machine learning based approaches to improve wavelet-based image compression; examples include [22–26]. In [22], Ma et al. propose a neural network for context modeling in their JPEG2000-inspired arithmetic coder, which they identify as Pixel Convolutional Neural Network (PixelCNN); they also propose a post-processing step to enhance reconstructed image quality. In a later work [23], they propose an *iWave* transform; this transform replaces the predict step of the conventional wavelet transform with a CNN while keeping the update step as a simple averaging operation. The iWave transform improves energy compaction compared to the CDF 9/7-based wavelet transform of the JPEG2000 standard. In [24], Dardouri et al. propose to replace both the predict and update steps of the conventional wavelet transform with a Fully Connected Neural Network. This work is further extended in [25]; however, performance improvements over JPEG2000 could be obtained only for the SSIM metric and the uncommon PieAPP metric. In [26], Li et al. propose the reversible autoencoder (Rev-AE), which is a lifting based wavelet-like codec with theoretical guarantees on transform reversibility and robustness to reconstruction quantization errors; their proposed approach shows competitive results compared to JPEG2000.

These wavelet-like learning-based methods naturally inherit the multi-scale representation from the wavelet transform, which provides resolution scalability; however, none of them explore quality scalability or region-of-interest accessibility. Additionally, these works do not investigate ways to directly train the networks for rate-distortion objectives, as seen in Section 6.1; instead, alternative training objectives, such as energy compaction of the transformed coefficients or prediction residuals, are used as proxies for coding efficiency. These are in contrast to the autoencoders in Chapter 6, which directly target rate-distortion objectives but do not preserve important features such as multi-resolution analysis. These are also in contrast to this thesis, in which two proposed networks are end-to-end trained for the rate-distortion objective while still preserving resolution scalability, quality scalability and region-of-interest accessibility.

3.2 Quantization

Returning to the simplified model of the compression system in Fig. 2.1, as we have already mentioned, the transform is normally regarded as an invertible operator and so does not introduce any loss. The sole source of distortion in this classic compression structure is quantization. The purpose of quantization is to map the transform coefficients $\mathbf{y} = y[n_1, n_2]$ into a finite collection of symbols $\mathbf{q} = q[n_1, n_2]$, which can be coded more effectively; these symbols are known as *quantization indices*.

The simplest form of quantization is *scalar quantization*, which associates a quantization index $q[n_1, n_2]$ independently to each transform coefficient $y[n_1, n_2]$ according to

$$q[n_1, n_2] = i \quad \text{if} \quad y[n_1, n_2] \in \Delta_i \tag{3.9}$$

where Δ_i refers to a quantization interval and the collection of $\{\Delta_i\}_i$ covers the entire range of the transform coefficients $\mathbf{y}$, as exemplified in Fig. 3.8. If the quantization intervals are constant, i.e., $\Delta_i = \Delta$ for all i, Δ is then commonly known as the *quantization step size*. As we can see, quantization is essentially a many-to-one mapping and in general not invertible; the inverse quantization operator (the dequantizer $\overline{Q^{-1}}$) is only an approximation of Q^{-1}, therefore distortion is introduced.

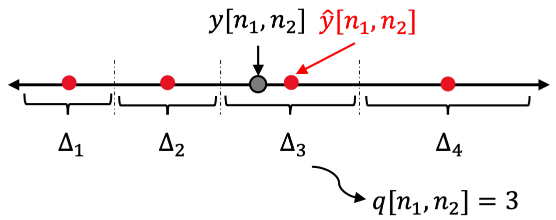


Figure 3.8: The simple scalar quantizer with four quantization indices.

There are many ways for the dequantizer $\overline{Q^{-1}}$ to reconstruct the transform coefficients $y[n_1, n_2]$ from the quantization indices $q[n_1, n_2]$. The simplest way is to utilize *midpoint reconstruction*; that is to select the reconstructed values $\hat{y}[n_1, n_2]$ as the midpoints of the quantization intervals Δ_i, as highlighted in Fig. 3.8.

The scalar quantizer and midpoint reconstruction are fairly straight-forward and are the most commonly used quantization and dequantization techniques in many modern image compression standards, such as the JPEG 2000 standard. Additionally, they also align with the idea of *simplifying* the global mapping operations M and $\overline{M^{-1}}$ in Chapter 2. Of course, there is a wealth of literature on developing more sophisticated quantization techniques; as our research is not devoted to improving forms of quantization, we will not review them any further here. Interested readers can refer to vector quantization [59–61], trellis coded quantization [62–64] and machine-learning based quantization [65–67] for more details.

3.3 Coding

Although the transform is designed to exploit statistical redundancies amongst the original image samples, in reality, it is hard to arrange for the transform coefficients to be statistically independent. More importantly, even if the coefficients are statistically independent, it is unlikely for them to be uniformly distributed. This is because the marginal distribution of the transform coefficients is expected to be highly skewed toward small amplitudes, as we have elaborated in Section 3.1. As a result, the outputs of quantization are also highly skewed toward the index whose quantization interval includes zero. Therefore, the purpose of source coding is to exploit this highly skewed distribution as well as the remaining statistical redundancy amongst the outputs of the quantization step, so as to minimize the average number of bits that are required to represent the quantization indices. Theoretically, this required average number of bits cannot be infinitely small; instead, it has a lower bound, which is associated with a quantity called *entropy*, according to Shannon's noiseless source coding theorem [68].

Suppose we have a random variable X, the entropy of which is defined as

$$\begin{aligned} H(X) &= -\sum_{x \in A_X} P_X(x) \log_2 P_X(x) \\ &= E\left[-\log_2 P_X(X)\right] \end{aligned} \tag{3.10}$$

where P_X denotes the probability mass function of the random variable X, while x is an outcome amongst the set of all possible outcomes A_X of the random process X. The essence of Shannon's noiseless source coding theorem is that the average number of bits required to code the outcomes of X must be no less than the entropy of X.

Three types of entropy are commonly encountered in the context of compression: joint entropy, marginal entropy and conditional entropy. To see the relationship of these three terms, assume we have a random vector $\mathbf{X}$ having m elements (random variables) $X_0, X_1, \cdots, X_{m-1}$. Following the concept in (3.10), the entropy of this random vector $\mathbf{X}$

can be defined as

$$
\begin{aligned}
H(\mathbf{X}) &= E\left[-\log_2 P_{\mathbf{X}}(\mathbf{X})\right] \\
&= E\left[-\log_2 \left(P_{X_0}(X_0)\cdots P_{X_{m-1}|X_{m-2},\cdots,X_0}(X_{m-1},\cdots,X_0)\right)\right] \\
&= H(X_0) + H(X_1|X_0) + \cdots + H(X_{m-1}|X_{m-2},\cdots,X_0) \\
&\leq \sum_{n=0}^{m-1} H(X_n)
\end{aligned} \tag{3.11}
$$

where $H(\mathbf{X}) = H(X_0,\cdots,X_{m-1})$ is often referred as the joint entropy of $X_0, X_1, \cdots, X_{m-1}$, while $H(X_n)$ is the marginal entropy of each random variable X_n. The entropy $H(X_{m-1} |X_{m-2},\cdots,X_0)$ is the conditional entropy of X_{m-1} given the other random variables $X_{m-2},\cdots,X_0$. As revealed in (3.11), conditional entropy is smaller than marginal entropy. Therefore, most of the source coding techniques employ conditional coding techniques that use a limited set of previously coded symbols to form a context, such that the coding of a symbol depends on its context. It is sufficient to note here that highly efficient coding techniques exist, such as adaptive arithmetic coding, that are able to nearly achieve the relevant entropy lower bound, by using appropriate contexts.

Since in this thesis, we are interested in learning additional transforms (or lifting steps) to the existing wavelet transform with the aid of machine learning, we need an entropy model for the bit-stream **c** during training. It is worthwhile to mention here that we utilize marginal entropy as the model during training, while the actual coding techniques employed for performance evaluation use conditional arithmetic coding. This means that statistical dependencies between quantized sample values are ignored during training, but they are utilised during the actual compression process. Although it is possible to take coding contexts into account also during training, this is a relatively minor refinement that can significantly increase the complexity of the already very expensive training process. Interested readers can refer to Chapter 6, in which some examples of coding contexts during training are briefly mentioned; they can also refer to [69–72] for more related works.

3.4 Rate-distortion Optimization

Although we explain the transform, quantization and coding as separate elements in the classic feedforward compression structure, in practice, they are closely related; this leads to the concept of *rate-distortion optimization* [73].

As we have explained before, in the classic compression structure, the transform is determined using some essential principles whereas quantization is solely responsible for introducing distortion. Therefore, there is only a small set of quantization parameters to be optimised – usually just one quantization step size for each type of subbands produced by the transform. These quantization step sizes essentially determine the level of distortion and the bit-rate required for the entropy coding of the corresponding quantization indices. In consequence, the purpose of rate-distortion optimization is to simultaneously adjust these quantization step sizes to minimize the distortion D subject to certain coded length L, or to minimize the coded length L subject to a certain level of distortion D. Specifically, this objective can be expressed using the Lagrange formulation; that is

$$J = D + \lambda L \tag{3.12}$$

where λ is the Lagrange multiplier that trades off distortion D and coded length L. In the literature, equation (3.12) is known as the standard rate-distortion objective; the optimization process that targets this objective is called rate-distortion optimization.

In contrast, in machine-learning based compression, which is the topic of this thesis, the transform is not fixed and so requires learning. This learning process involves a huge number of parameters, which cannot be tuned independently due to their complicated interdependencies with quantization and entropy coding, unlike that in the classic compression framework. As a result, we need to understand and anticipate the quantization process, the entropy coding process as well as the rate-distortion optimization process which must be included in any lossy compression framework, so as to robustly train the transform.

This means that machine-learning based approaches usually require an end-to-end optimization of the entire compression system, directly targeting rate-distortion objectives to achieve optimized coding efficiency. As we shall see in Chapter 6, this is essentially the case of the autoencoders. In this thesis, the proposed neural networks, which serve as secondary transforms in addition to the existing wavelet transform, are also optimized in an end-to-end framework, targeting the standard rate-distortion objective augmented with an extra aliasing constraint term; see Chapter 10 for more details.

3.5 Summary of The Classic Feedforward Compression Structure

We conclude this chapter by discussing the advantages and the weaknesses of the classic feedforward compression system. The purpose of this discussion is to motivate the introduction of predictive feedback compression structures in Chapter 4. More importantly, we would like to use this discussion to motivate the innovations in this thesis.

One major advantage of feedforward compression structures is that they can be used to construct highly scalable compressed representations of an image. In a scalable compression system, the bit-stream can be partially discarded to obtain reconstructions of the original image at different bit-rates and/or at different resolutions. Perhaps the best example of a highly scalable feedforward compression system is the JPEG 2000 image compression system [1], which generates a set of dyadically-spaced multi-resolution representations of the compressed image using the DWT (Section 3.1.2); each of these scales can be reconstructed from the compressed bit-stream, producing a smaller version of the original image, a feature known as *resolution scalability.* Moreover, at a given scale, the JPEG 2000 format also enables reconstructions at different distortion-rate operating points, a feature known as *quality scalability.* Apart from scalability, the JPEG 2000 image compression standard also provides *region-of-interest accessibility*, which means that any region of interest within the original image can be decompressed independently from the bit-stream, with no need for reconstructing the entire image.

The principle weakness of the classic feedforward system is that it is difficult to adapt the basis vectors of the transform to local geometric features, such as orientations of the input image, without communicating any side information to the decompressor, as we have mentioned in Section 3.1.2.7. This is because if a transform itself is dependent on features within the data that is being transformed, then it cannot be linear and its invertibility becomes very difficult to ensure.

One solution to avoid this difficulty is to use quantized values that have already been encoded and so are available at the decoder, to modify the way in which later sample values are transformed. This suggests some kind of feedback strategy, which is the topic of Chapter 4. However, such approaches tend to be incompatible with scalability, since in a scalable compression system the encoder cannot generally anticipate the quantized values that will be available to a decoder, which may be reconstructing a reduced quality and/or resolution version of the image from a subset of the original compressed content.

Another solution to overcome the difficulty of feedforward structures is to directly build a feedforward system which is capable of exploiting local geometric features without communicating any side information. As we shall see in Chapter 8, this is essentially the target of the hand-tuned algorithm proposed in our previous work [74]. Motivated by the limitations of this non-learning based solution in [74], we ultimately decide to use machine learning as a tool in this thesis to exploit local geometric features within the original image; strategic deployment of neural networks is the main content of Chapter 9.

Chapter 4

Review: Predictive Feedback Compression Structures

As we have explained in Section 3.5, the challenge for a classic feedforward system is to adapt the basis vectors of the transform to local geometric features, such as orientations. One possible solution to address this is to introduce some form of *predictive feedback*. The idea of predictive feedback is to exploit the statistical redundancy between the adjacent image samples by predicting a current image sample from previously reconstructed samples. Ultimately, only prediction errors (or *residuals*) need to be quantized. Since these prediction residuals are generally close to zero, the corresponding quantization indices should be highly skewed toward the index whose quantization interval contains zero, yielding a low coded bit-rate.

4.1 Differentiable Pulse Code Module (DPCM)

The basic predictive feedback structure is illustrated in Fig. 4.1. A scalar quantizer is employed in this structure to map its input array to quantization indices $q[\mathbf{n}] = q[n_1, n_2]$ in a raster scan order. Instead of taking the original image samples $x[\mathbf{n}] = x[n_1, n_2]$ as the input, this scalar quantizer quantizes prediction residuals $e[\mathbf{n}] = x[\mathbf{n}] - p[\mathbf{n}]$; the prediction $p[\mathbf{n}]$ is obtained by applying some function on some or even all previously reconstructed samples $\hat{x}[n_1, n_2 - i]$, for any $i > 0$. The simplest way of constructing $p[\mathbf{n}]$ is to utilize the most recently reconstructed sample $\hat{x}[n_1, n_2 - 1]$, i.e. $p[n_1, n_2] = \hat{x}[n_1, n_2 - 1]$; this kind of prediction is known as Differentiable Pulse Code Module (DPCM).

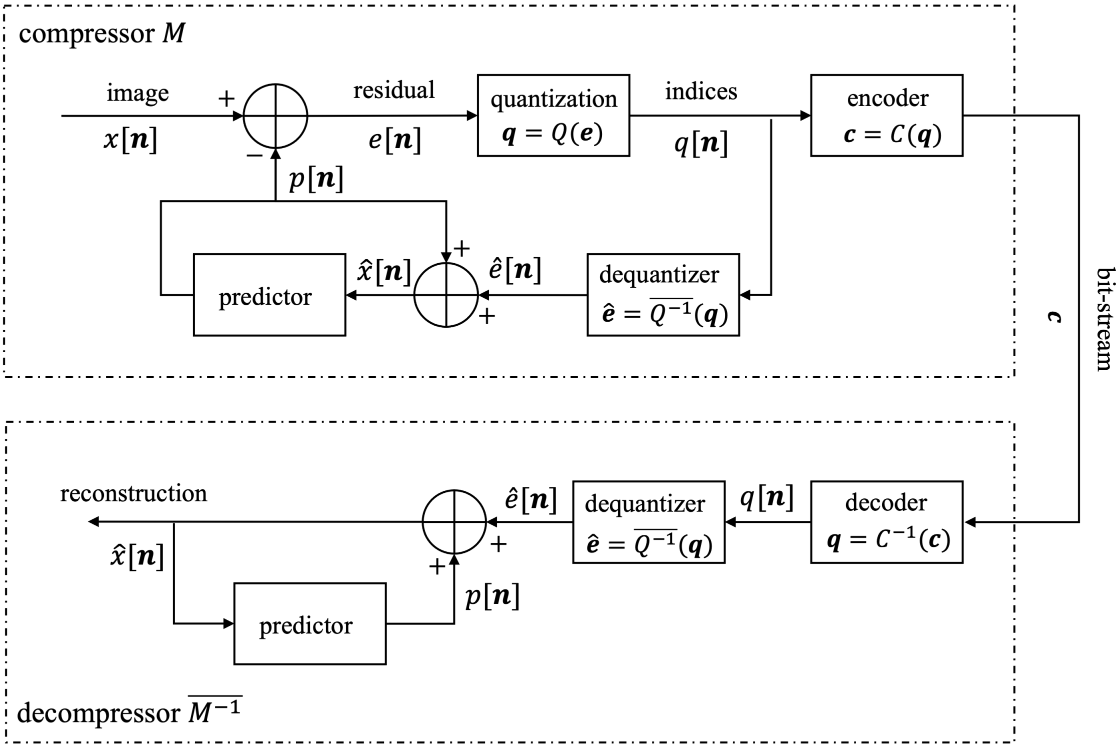
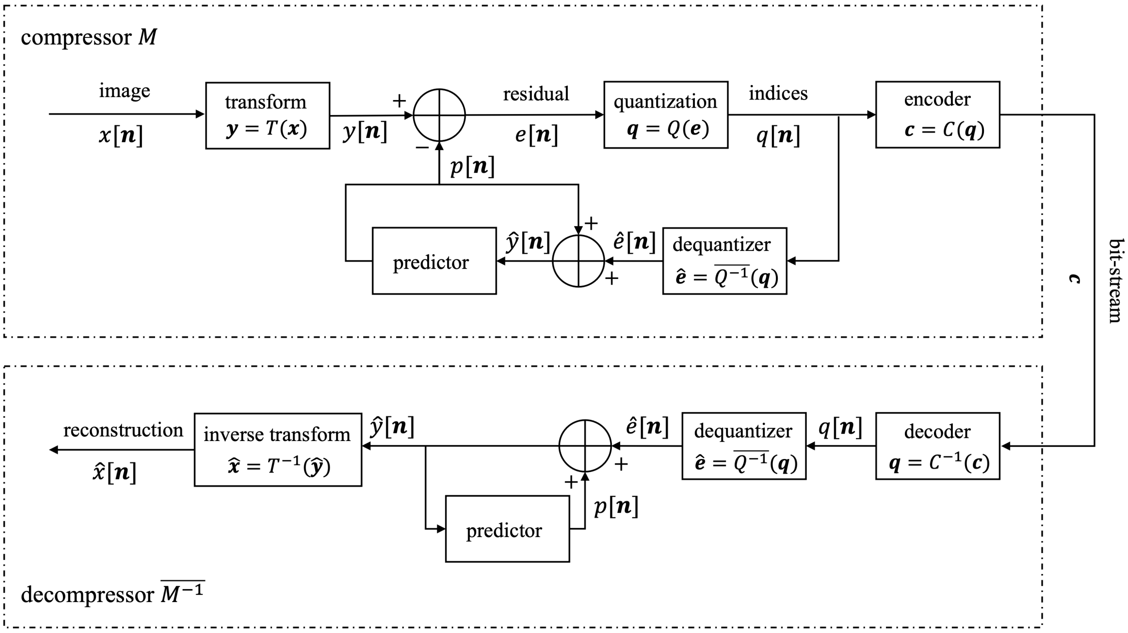


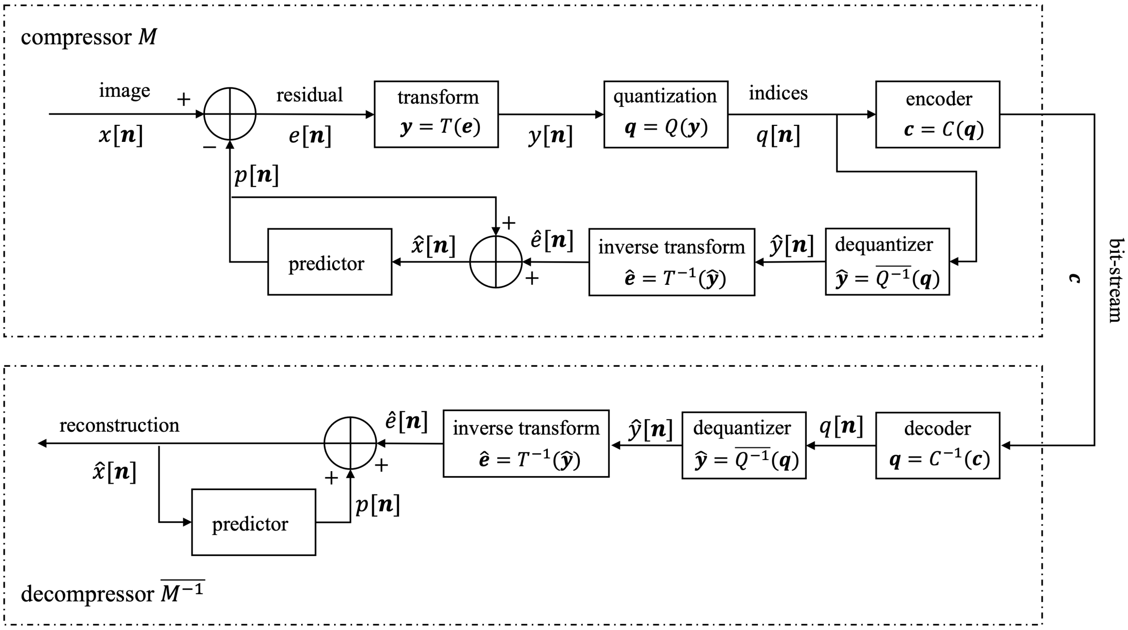
Figure 4.1: The Differential Pulse Code Modulation (DPCM) structure for image compression.

4.2 Hybrid Transform-Predictive Structure

Another popular form of predictive feedback structure is the *hybrid transform-predictive* structure, as depicted in Fig. 4.2(a). This structure essentially applies the predictive feedback loop on the transform coefficients $y[\mathbf{n}]$ instead of the original image samples; it has the benefit to further exploit redundancies that have not been captured by the transform T. Alternatively, the predictive feedback path can also include both the transform and the quantizer inside the loop, as shown in Fig. 4.2(b).



(a)



(b)

Figure 4.2: Two commonly used hybrid transform-predictive feedback structure for image compression.

These hybrid transform-predictive structures have been utilized in some modern image and video compression standards. For example, the lossless mode of the JPEG image compression standard [2] offers seven different predictors, which are weighted combinations of the reconstructed neighbours around the current transform coefficient, as seen in Fig. 4.3(a). Similarly, the latest video coding standard HEVC (High Efficiency Video Coding) [3, 75] utilizes neighbouring reconstructed samples to predict the current coding block with 33 angular intra prediction modes, the DC mode and the planar mode, as illustrated in Fig. 4.3(b). Other examples can be found in CCITT H.261/2/3 [76–78], H.264/AVC [79], AVS (Audio and Video Coding Standard in China) [80] as well as the ISO/IEC standards developed by the Motion Picture Experts Group (MPEG) [81, 82].

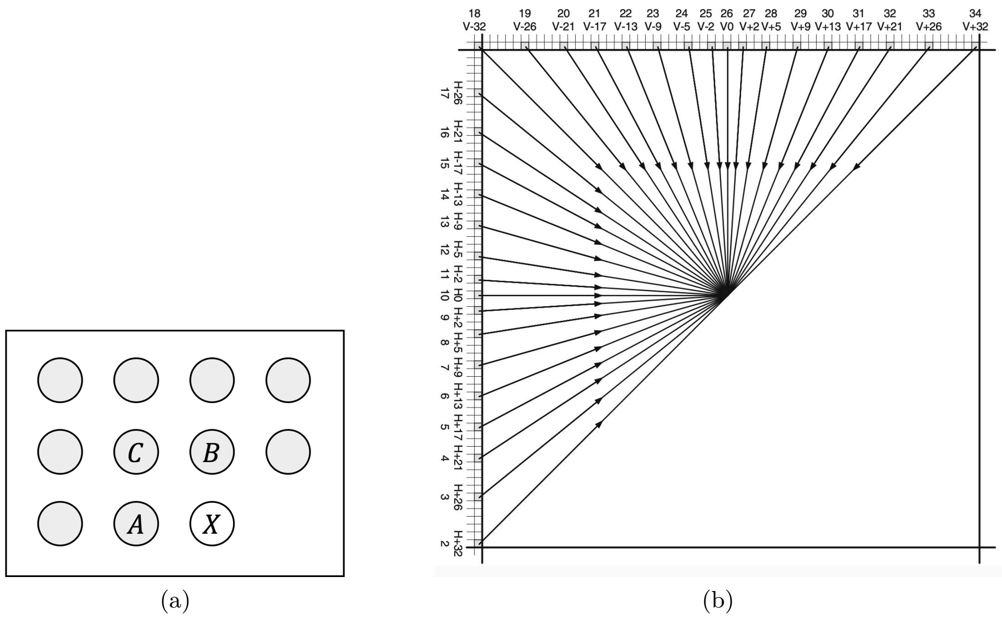


Figure 4.3: Examples of transform-predictive coding algorithms employed in modern image and video standards. (a) illustrates the lossless mode in the JPEG image compression standards [2], which supports the following predictors: $X = A$, $X = B$, $X = C$, $X = A + B - C$, $X = A + (C - B)/2$, $X = (A + C)/2$, $X = C + (A - B)/2$; A, B and C are previously reconstructed DCT coefficients. (b) shows the intra angular prediction employed in HEVC standard [3].

4.3 Relationship Between Predictive Feedback and Classic Feedforward Structures

As we have seen, the fundamental difference between the predictive feedback and the classic feedforward structure is the recursive prediction path from the reconstructed image. This feedback loop certainly breaks some of the constraints that otherwise exist in the classic feedforward structure, in return for some benefits. For example, there is no limit on the type of predictors that can be employed in predictive feedback structures. The predictors can be linear or non-linear, which leverage the ability of the system to explore complex features such as orientations in the original image.

However, there are also weaknesses associated with predictive feedback structures. The principle weakness is the difficulty for them to be employed in highly scalable systems. As we have seen in Fig. 4.1 and Fig. 4.2, predictive feedback structures require the compressor to replicate some or even all previously reconstructed samples at the decompressor. In other words, certain elements of the decompressor, e.g the dequantizer and/or the inverse transform, must be embedded inside the compressor. However, in highly scalable systems, the bit-stream can be decompressed in many ways that may not be the same as that in the compressor. This makes it rather difficult for the compressor in predictive feedback structures to replicate all the possible decompressions of the bit-stream.

Moreover, since the predictive feedback path is recursive, the support of the reconstruction system is notionally infinite. This means that the recursive nature of the reconstruction system leads to a dependence of later sample values on all previously coded samples, which naturally interferes with region-of-interest accessibility. Meanwhile, as quantization errors are incorporated inside the feedback loop and they are often *unstructured*, the performance of predictive feedback structures can be inferior to feedforward structures, under the condition that quantization errors are larger than prediction residuals.

Chapter 5

Review: Post-processing and Inverse Transform Optimization

As we have discussed in Chapter 3 and Chapter 4, both the classic feedforward and predictive feedback structures employ scalar quantization and midpoint reconstruction in the spirit of simplifying the compressor M and the decompressor $\overline{M^{-1}}$ in Fig. 2.1. This simple midpoint reconstruction, assigning a pre-determined fixed reconstruction value to each quantization index, is far from optimal. Therefore, it presents a source of significant potential for improvement, which essentially motivates the ideas of *post-processing* and *inverse transform optimization* (or *optimized reconstructions* in other words).

These optimized reconstructions can be generally understood as resolving the *inverse problem* [83]; that is, to use observed measurements to infer the information to an observation system. In the context of image processing, the observed measurements are degraded images due to additive noise, blurriness, down-sampling or distortion that is introduced by quantization, while the signal to be recovered is the original image.

The broad theme behind the inverse problem is to understand three things: 1) the relationship between the observations and the signal to be recovered, i.e. the *observation model*; 2) the characteristic of the errors (noise) in the observations, i.e. the *noise model*;

3) the prior statistical model of the signal that is to be recovered. As we shall see, these three questions are all explicitly or implicitly addressed in the optimized reconstruction algorithms discussed in this chapter.

There is a wealth of literature on developing optimized reconstruction algorithms. In this chapter, we choose to briefly review three representative works: projection onto convex sets (POCS), graph-based regularization and super resolution; as we shall see, POCS and graph-based regularization are conceptually connected to super resolution problems. The purpose of this brief overview is to provide another perspective to understand the work in this thesis. Furthermore, the idea of optimized reconstructions serves as a bridge for us to introduce the concepts of the autoencoders in Chapter 6.

5.1 Projection Onto Convex Sets (POCS)

Projection onto convex sets (POCS) is an iterative approach proposed by Youla and Webb to incorporate prior knowledge about the solution into the restoration process of the signal, given only partial data [84]. Specifically, suppose the original signal x is defined in the Hilbert Space $\mathcal{L}^2(\mathbb{R})$, and is known *a priori* to belong to the non-empty *intersection* $\mathcal{C}_0$ of m well-defined closed convex sets $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_m$, such that

$$x \in \mathcal{C}_0 \bigcap_{i=1}^{m} \mathcal{C}_i \tag{5.1}$$

These pre-defined convex sets $\mathcal{C}_i, i = 1, 2, \cdots, m$ can be understood as prior knowledge or constraints for reconstructing x. The intersection convex $\mathcal{C}_0$ is the space that all feasible solutions of restoration should be located to satisfy the prior constraints.

To find a solution of reconstruction inside the intersection $\mathcal{C}_0$, the POCS approach recursively projects any arbitrary signal x^n onto the closed convex sets $\mathcal{C}_1, \mathcal{C}_2, \cdots, \mathcal{C}_m$, so that

$$x^{(n+1)} = \mathcal{P}_m \mathcal{P}_{m-1} \cdots \mathcal{P}_2 \mathcal{P}_1 x^{(n)} \tag{5.2}$$

where $x^{(0)}$ can be an arbitrary starting point while n is the number of iterations. The symbol $\mathcal{P}_i$ denotes the projection operator onto the convex set $\mathcal{C}_i$ for $i = 1, 2, \cdots, m$. Although these iterative projections are not trivial tasks, it is still in general easier than finding only one projector $\mathcal{P}_0$ that projects $x^{(0)}$ onto $\mathcal{C}_0$ in one step [85].

It can be proved that the sequence $\left\{x^{(n)}\right\}_n$ must converge to a point within $\mathcal{C}_0$ in a limited number of iterations [84]; a simple example of this convergence is depicted in Fig. 5.1. It is worthwhile to point out that this convergence is not unique and is dependent on the starting point $x^{(0)}$; different starting points may lead to substantially different reconstructions. Therefore, in practice, a reasonable estimate of the original signal x, for instance Wiener estimate [86], is often in place to facilitate the rationale of the convergence.

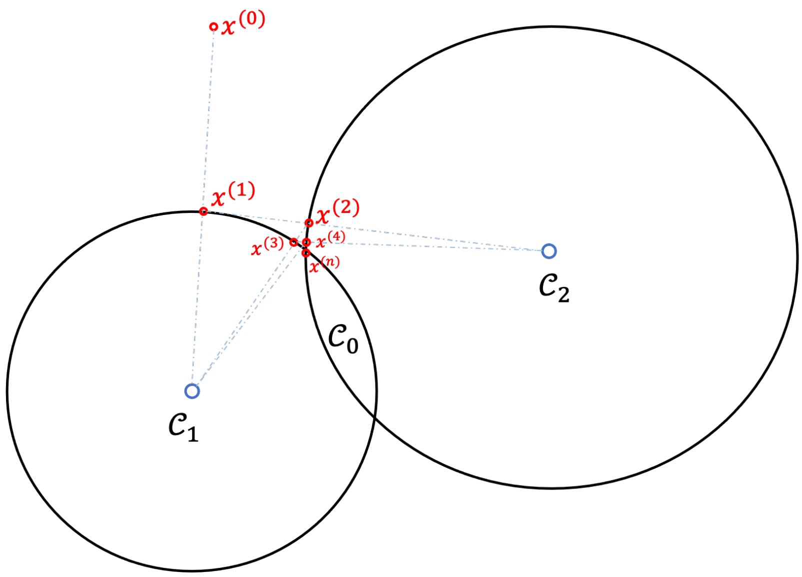


Figure 5.1: A simple example showing the convergence of the POCS method.

As we can see, when applying the POCS method to reconstruct/invert compressed images, the observation model of POCS is the transform that relates the original image and the observed quantized transform coefficients, such as the DCT. The noise model of POCS is characterized by the quantization intervals, where the original image samples should be. The simplest prior model that can be used for POCS is the *smoothness* of the original image; this indicates the reason why many algorithms employ *smoothness constraint sets* (SCS) as prior information for POCS-based reconstructions. For example, Zakhor et al. [87] employ SCS to iteratively restore a compressed image to its original artifact-free form. Subsequently, Yang et al. [88–90] propose to recover compressed images using SCS; they propose a new family of directional SCS derived from linear modelling of image edge structures, so as to reduce blocking and ringing reconstruction artifacts. Interested readers can refer to [91–95] for more sophisticated prior models developed for POCS-based reconstructions.

5.2 Graph-based Regularization

With the development of *graph signal processing* [96, 97] in the last decade, *graph-based regularization*, especially *graph Laplacian regularization*, has also shown its great potential in a wide range of image reconstruction applications, such as image denoising [98, 99], deblurring [100], bit-depth enhancement [101] and dequantization of JPEG images [102–106].

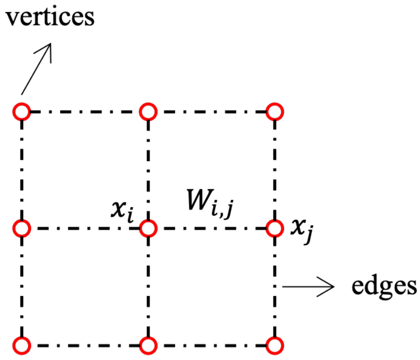


Figure 5.2: A digital image can be described by an underlying graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$.

Specifically, graph signal processing assumes that a digital image is highly structured and can be described by an underlying graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$, in which the pixels are viewed as *vertices* (or graph *nodes*) $\mathcal{V}$ connected through *edges* $\mathcal{E}$ with weights $\mathbf{W}$, as depicted in Fig. 5.2. The weight $W_{i,j}$ of the edge models the correlation, or can be understood as affinity, between adjacent (or neighbouring) pixels x_i and x_j. Given the edge weights $\mathbf{W}$, the *adjacency matrix* $\mathbf{A}$ and the diagonal *degree matrix* $\mathbf{D}$ of a graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ are defined as

$$A_{i,j} = W_{i,j} \tag{5.3}$$

$$D_{i,i} = \sum_j A_{i,j} \tag{5.4}$$

A *combinatorial* or *unnormalized graph Laplacian* $\mathbf{L}$ can then be defined in relation to $\mathbf{A}$ and $\mathbf{D}$ as

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{5.5}$$

Subsequently, the *graph Laplacian regularizer* is defined using $\mathbf{L}$ to describe the squared variance of the signal $\mathbf{x}$ with respect to the graph $\mathcal{G}$ as

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \sum_{i,j} (x_i - x_j)^2 W_{i,j} \tag{5.6}$$

From (5.6) we can see that the graph Laplacian regularizer essentially detects the level of smoothness for each connected pixel pair (x_i, x_j); if the image $\mathbf{x}$ is heavily distorted due to quantization or even loses some compressed blocks during transmission, the regularizer $\mathbf{x}^T \mathbf{L} \mathbf{x}$ becomes huge under a fixed pre-defined graph $\mathcal{G}$. If we incorporate this regularizer inside the cost function of the inverse problem as

$$\text{minimize} \quad g(\mathbf{x}) + \alpha^2 \mathbf{x}^T \mathbf{L} \mathbf{x} \tag{5.7}$$

where $g(\mathbf{x})$ is an application-dependent term while α specifies the amount of regularization, then the additional graph Laplacian regularizer is forced to be small, so as to minimize the global cost function. This essentially encourages the reconstructed images to be smooth. For this reason, the graph Laplacian regularization and its variants are of particular interest in dequantizing JPEG images to reduce block artifacts; interested readers can refer to [102–106] for more details.

As we can see, in these graph-based reconstructions, the prior model of the original image is determined by the graph $\mathcal{G}(\mathcal{V}, \mathcal{E}, \mathbf{W})$ with weights $\mathbf{W}$. The noise model can be understood as the external weights between the graph nodes and the observed compressed image samples. The observation model is still the transform that relates the original image and the quantized transform coefficients, similar to POCS-based reconstructions.

5.3 Super Resolution

As exemplified in Section 5.1 and Section 5.2, optimized reconstruction algorithms essentially attempt to restore transform coefficients that have been quantized to zero in the bit-stream. Theoretically, most of the coefficients that are quantized away are located at high-resolution levels; in other words, at low bit-rates, the information that we first lose due to compression is at high resolutions. This means that even if the observed compressed images are at full-resolution, the essence of any reconstruction technique is to try to restore high-resolution information using low-resolution observations. This is exactly the concept of super resolution – reconstructing a high-resolution image given a single or multiple low-resolution images.

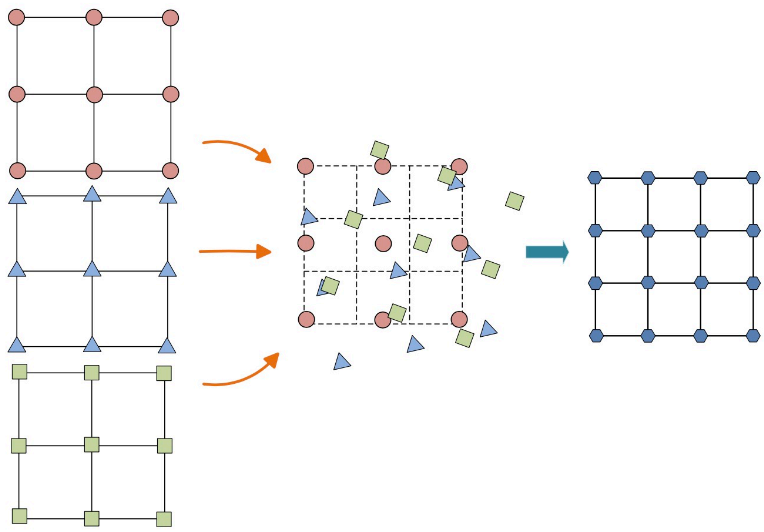


Figure 5.3: The core concept of Multi-frame Super Resolution.

The principle of the classic *Multi-frame Super Resolution* is as follows. If multiple low-resolution images with sub-pixel misalignment of the same scene are available, then the reconstruction of a high-resolution image can be achieved by fusing the complementary information observed in these low-resolution images with some algorithms, as illustrated in Fig. 5.3. The observation model for super resolution normally involves a disciplined down-sampling process to model the imaging system of a camera [107, 108]. The noise model employed for super resolution is usually the Gaussian imaging noise; in the context of compression, this becomes the quantization noise model. In the literature, there is a pile of prior models that have been developed for super resolution; examples include Gibbs prior [109], Markov random field prior [110], Gaussian prior [111] and Huber-Markov Gibbs prior [112]. This infers the reason why machine learning has become extremely popular for super resolution nowadays [4, 113–115], because the learning process essentially avoids incorporating any explicit prior statistical model of the high-resolution image; instead, the prior model is implicitly embedded inside the network weights.

Amongst all existing methods that have been developed for Multi-frame Super Resolution, there is one particular super-resolution algorithm intimately connected to this thesis; that is the frequency domain approach first proposed by Tsai and Huang [116]. This approach makes explicit usage of aliasing information that exists in each low-resolution image to reconstruct the high-resolution image. In [116], Tsai and Huang prove that given multiple *aliased* views of the same underlying continuous image, where each view is obtained with a different shift, the minimum mean squared error best estimate of the original scene can be found using Wiener filtering.

For the purpose of this thesis, we only have *one* aliased view, i.e. the low-pass LL band of the same underlying continuous image that is to be reconstructed; therefore, we lose most of the prior knowledge presented in [116] for reconstruction. However, as we shall see in Chapter 7, geometric regularity along oriented edges serves as another form of prior knowledge, which can be effectively used to identify multiple shifted copies of the same feature within the low-resolution image, so as to recover a higher resolution image. This reveals the intimate connection between this thesis and super resolution problems.

Moreover, it is worthwhile to mention that the existing *Single-image Super Resolution* algorithms may have potentially exploited this geometric regularity as prior knowledge for reconstruction in one way or another. As exemplified in Fig. 5.4, some first-layer filters trained in [4] are clearly exploiting local orientations.

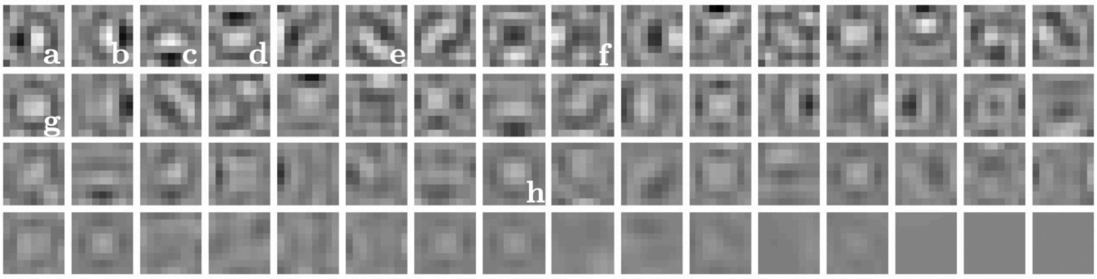


Figure 5.4: The first-layer filters trained in [4] with an upscaling factor of 3.

5.4 Connection of This Thesis with Optimized Reconstructions and Super Resolution

The learned neural network structures employed in this thesis can be understood as introducing an optimised reconstruction step, more specifically, a super resolution step, into the synthesis process of the wavelet transform.

Using Fig. 5.5 as guidance, even if all the encoded detail wavelet subband samples wind up being quantized to zero in this thesis, the proposed low-to-high network $\mathcal{T}_{L2H}$ still has the ability to partially restore the original detail samples using the coded low-pass residuals $\overline{\mathrm{LL}}'_d$. Subsequently, the proposed high-to-low network $\mathcal{T}_{H2L}$ is then able to at least partially reconstruct the original coded low-pass image LL'_d at resolution d of the wavelet transform, using the restored detail samples $\widetilde{\mathrm{HL}}'_d$, $\widetilde{\mathrm{LH}}'_d$ and $\widetilde{\mathrm{HH}}'_d$. Eventually, the restored low-pass and high-pass subbands are synthesized to reconstruct the coded low-pass image LL'_{d-1} at the next higher resolution $d-1$.

This means that the combination of the high-to-low and the low-to-high structures in this thesis essentially introduces an optimal reconstruction into the synthesis process of the wavelet transform. Moreover, in this extreme case, we essentially synthesize a high-

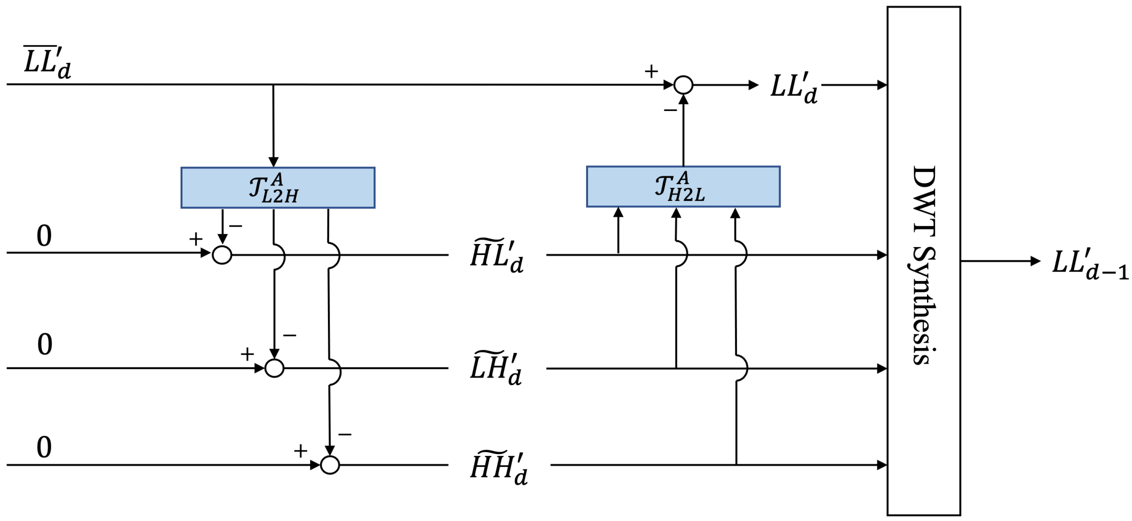


Figure 5.5: The learned neural network structures, $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ in this thesis, can be understood as introducing an optimised reconstruction step, or a super resolution step, into the synthesis process of the wavelet transform.

resolution image LL'_{d-1} using the observation of a low-resolution image $\overline{\mathrm{LL}}'_d$. Therefore, this combination of the high-to-low and the low-to-high structures can also be viewed as introducing a super resolution step into the synthesis process of the wavelet transform.

Chapter 6

Review: Autoencoders

Built upon the existence of optimized reconstruction algorithms in Chapter 5, it is natural to consider also optimizing the analysis system according to the optimized synthesis system; this is known as *analysis-by-synthesis* [117], which has been widely adopted in many speech-coding applications in the literature [118–120]. The *autoencoders* introduced in this chapter can be understood from this perspective; the synthesis system of an autoencoder produces essentially an optimized reconstruction for the original image using learnable parameters, while the analysis system is also learned simultaneously in accordance with the synthesis system to achieve high coding efficiency.

6.1 Basic Concepts

Fig. 6.1 illustrates the general compression architecture of an autocoder. It learns a latent representation $\mathbf{y}$ of the input image $\mathbf{x}$ through one or more hidden layers, which can also be understood as the analysis transform T of the image $\mathbf{x}$ with learnable parameters $\boldsymbol{\phi}_a$, i.e. $\mathbf{y} = T(\mathbf{x}; \boldsymbol{\phi}_a)$. The latent representation $\mathbf{y}$ usually has reduced dimensions compared with the input image $\mathbf{x}$, and so can be quantized effectively to produce the integer-valued latents $\mathbf{q} = Q(\mathbf{y}) = Q(T(\mathbf{x}; \boldsymbol{\phi}_a))$. The quantizer Q employed in an autoencoder is often the uniform scalar quantizer with quantization step size $\Delta = 1$; this is sufficient, because we can always modify the analysis transform to include a rescaling or incorporate non-linearities into the analysis transform to make the quantizer non-uniform. The integer-valued latents $\mathbf{q}$ are eventually coded and decoded for the synthesis $\overline{T^{-1}}$ of the reconstructed image $\hat{\mathbf{x}}$ with learnable parameters $\boldsymbol{\phi}_s$; that is, $\hat{\mathbf{x}} = \overline{T^{-1}}(\mathbf{q}; \boldsymbol{\phi}_s) = \overline{T^{-1}}(Q(T(\mathbf{x}; \boldsymbol{\phi}_a)); \boldsymbol{\phi}_s)$. To achieve optimized coding efficiency, the analysis and the synthesis parameters $\boldsymbol{\phi}_a$ and $\boldsymbol{\phi}_s$ are usually end-to-end optimized for the rate-distortion objective

$$\begin{aligned} J(\boldsymbol{\phi}_a, \boldsymbol{\phi}_s) &= E\left[\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2\right] + \lambda E\left[-\log_2 P_{\mathbf{V}}(\mathbf{q}; \boldsymbol{\phi}_a)\right] \\ &= E\left[\left\|\mathbf{x} - \overline{T^{-1}}(Q(T(\mathbf{x}; \boldsymbol{\phi}_a)); \boldsymbol{\phi}_s)\right\|_2^2\right] + \lambda E\left[-\log_2 P_{\mathbf{V}}\left(Q(T(\mathbf{x}; \boldsymbol{\phi}_a)); \boldsymbol{\phi}_a\right)\right] \end{aligned} \quad (6.1)$$

where the first term denotes the distortion between the original image $\mathbf{x}$ and the reconstructed image $\hat{\mathbf{x}} = \overline{T^{-1}}(Q(T(\mathbf{x}; \boldsymbol{\phi}_a)); \boldsymbol{\phi}_s)$, while the second term represents the average coded length for coding the integer-valued latents $\mathbf{q} = Q(T(\mathbf{x}; \boldsymbol{\phi}_a))$ that are drawn from the random variable $\mathbf{V}$ with probability mass function $P_{\mathbf{V}}(\boldsymbol{\phi}_a)$.

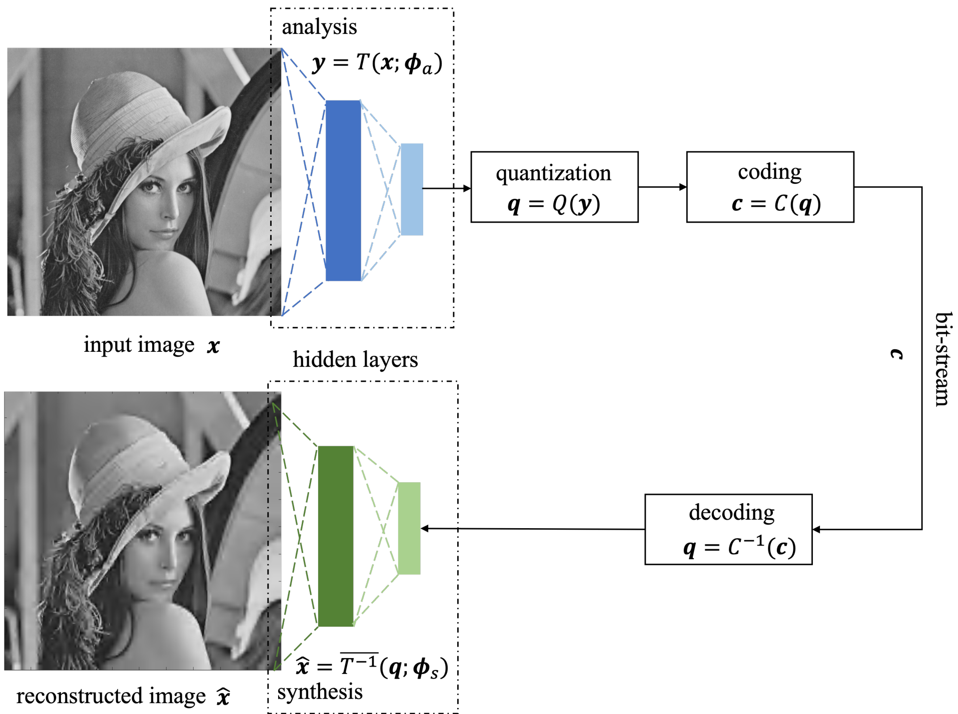


Figure 6.1: The general compression architecture of autoencoders.

As we have seen, autoencoders break the classic feedforward struture discussed in Chapter 3 in many aspects. The main difference is that the transforms (or the networks) in these autoencoders are not invertible. There are two reasons behind this: 1) these networks usually involve reduction of dimensions for the purpose of compression, which means that there must be some image sample sequences that cannot be represented exactly, even without any quantization errors; 2) the operators employed in these networks usually involve non-linearities, whose invertibility are hard to prove.

As a result, these autoencoders do not generally preserve some of the important features enjoyed by classic feedforward compression systems, as discussed in Section 3.5; they are simply *unstructured* machines that target rate-distortion objectives to achieve optimized coding efficiency only. This is in contrast to other wavelet-like machine-learning based methods discussed in Section 3.1.2.7, as well as the method presented in this thesis.

For the purpose of this thesis, since we primarily focus on improving the wavelet transform with the aid of neural networks, we first review the existing neural network architectures that have been successfully employed for image compression; these networks can be broadly categorized into Fully Connected Networks (Section 6.2.1), Convolutional Neural Networks (Section 6.2.2), Recurrent Neural Networks (Section 6.2.3) and Generative Adversarial Neural Networks (Section 6.2.4).

Moreover, since learned compression systems usually require end-to-end training, most of the machine-learning optimization techniques, e.g. gradient decent, rely on differentiability for back-propagation [121]. However, both the distortion and the average coded length in (6.1) depend on the quantizer Q, whose derivative is either zero or infinity everywhere. Therefore, we also review the existing techniques that have been developed to overcome the discontinuity imposed by the quantizer in the context of the autoencoders; this is the main purpose of Section 6.3.

As we shall see through Section 6.2.1 to Section 6.2.4, autoencoders were initially underperforming conventional non-learning based techniques, such as JPEG and JPEG 2000. Subsequently, with enormous explosion in computational complexity, the development of extremely elaborated learning-based coding schemes and sophisticated training methodology, the performance of autoencoders is now comparable with the existing state-of-the-art compression standards, such as HEVC and AVC/H.264. However, as we have mentioned, these autoencoders sacrifice most of the properties that are desirable for practical applications, as discussed in Section 3.5.

6.2 Different Neural Network Architectures

6.2.1 Fully Connected Neural Networks (FCNNs)

As inferred by the name, Fully Connected Neural Networks (FCNNs) consist of a series of fully connected layers that connect every *neuron* in one layer to every neuron in another adjacent layer, as seen in Fig. 6.2; for historical reasons, FCNNs are also known as Multi-layer Perceptrons (MLPs). Specifically, the output of the j^{th} neuron at a certain hidden layer can be represented as the weighted combination of all neurons o_k from the previous layer, subjected to an activation function $f(x)$ as

$$o_j = f\left(net_j\right) = f\left(\sum_k \omega_{kj} o_k + b_{kj}\right) \tag{6.2}$$

where net_j denotes the inputs to the j^{th} neuron. w_{kj} and b_{kj} are the *weights* and the *bias* between the k^{th} neuron at the previous layer and the j^{th} neuron at the current layer. The activation function $f(x)$ can often be understood as a type of switch, which decides whether or not a certain neuron should be activated [122]. This activation function needs to involve non-linearities, such as the sigmoid function, the Rectified Linear Unit (ReLU), the tanh function and etc.

Since early theoretical analysis had demonstrated the potential of FCNNs (or MLPs) to approximate any continuous function with an arbitrary precision [123], researchers started to experiment with FCNN-based image compression. These experiments are amongst the very first attempts of using neural networks as a tool for image compression. Although the explored FCNN structures were very naive and the performance was no where near

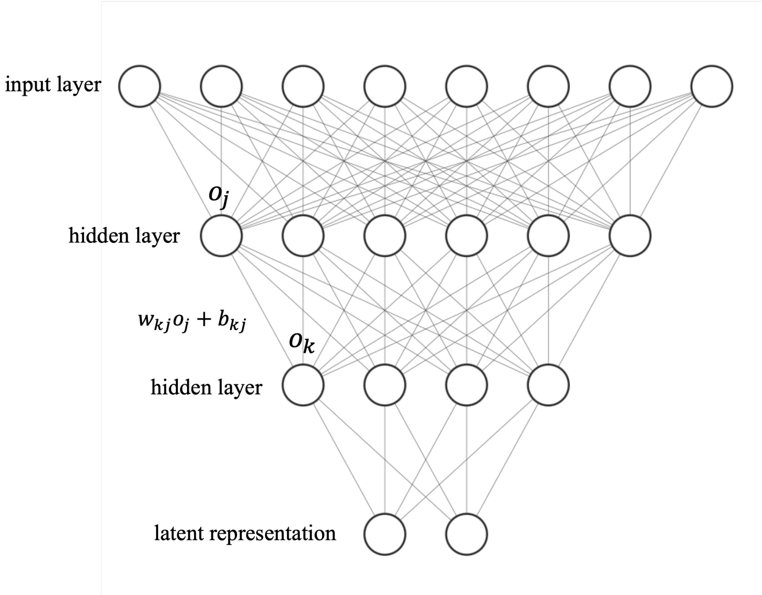


Figure 6.2: The general structure of fully connected neural networks.

any modern image compression standard, important properties of neural networks were revealed and studied at this stage, such as generalization-capacity trade-off (known as *overfitting* behaviour nowadays).

Specifically, Cottrell et al. [124] first proposed to train an FCNN with only one hidden layer of 16 neurons to compress each 8 x 8 non-overlapped block of the input image sequentially. Although this simple structure has the ability to compress and reconstruct the input image with seemingly low normalized mean squared error, it is hard to adapt to variable compression ratios as the number of neurons in the hidden layer are fixed.

Instead of processing each image block sequentially as in [124], Sonehara et al. proposed to use parallel sub-FCNNs to concurrently process a batch of overlapped image patches [5], as illustrated in Fig. 6.3. The authors, for the first time, discovered the *overfitting* issue encountered by learning-based image compression schemes. They found that each sub-FCNN in [5] was tuned to a specific image pattern, and could only work when the untrained image patches were substantially similar to patches found in the training set. This severe overfitting issue was reflected in the average Signal-to-Noise (SNR) of the *unlearned* compressed images, which was significantly lower (around 10 dB) than that of the *learned* images.

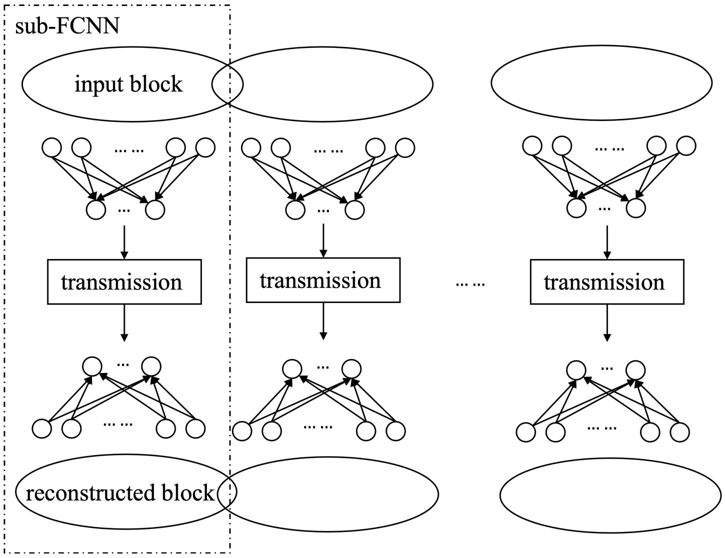


Figure 6.3: The sub-FCNNs employed in [5].

In contrast to the single-hidden-layer FCNN employed in [124] and [5], Namphol et al. proposed a multipatch hierarchical FCNN structure as illustrated in Fig. 6.4, which had the ability to further exploit the correlation between hidden layers and between different images using a nested training algorithm [6]. More importantly, the authors mathematically addressed the overfitting issue encountered in [5]. They proved that the *generalization ability* and the *capacity* of the network were inversely related, and a trade-off could be found by employing a proper size of the input image patch; generalization represented the ability of a network to achieve significant coding gain for unlearned images, while capacity described the ability of a network to capture global features and improve coding efficiency for training images.

Other variants of FCNNs are also explored for the purpose of image compression; examples include [125–129]. As we have mentioned before, these early-phase explorations are very primitive with only limited coding performance. Later on, the focus of the work shifted from improving FCNNs themselves into developing more sophisticated network structures, such as convolutional neural networks, recurrent neural networks and generative adversarial networks. FCNNs then are often used as an element within or in conjunction with these more advanced network structures.

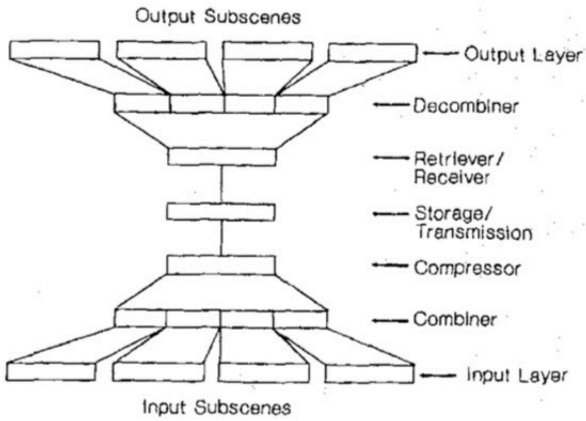


Figure 6.4: The multipatch hierarchical FCNN proposed in [6].

6.2.2 Convolutional Neural Networks (CNNs)

The introduction of convolutional neural networks (CNNs) led the development of learning-based image compression into a new era, with numerous advantages over the FCNN/MLP-based image compression.

First, if the source content is stationary, then shift-invariant processing is the only processing that makes sense. Since neural network layers are all just linear operators with non-linear point-wise activation functions, a shift-invariant neural network then necessarily involves a linear operator that is shift invariant – and that is by definition *convolution*. Therefore, CNNs are and have always been the sensible choice for processing stationary random processes rather than FCNNs/MLPs. Moreover, if the source content is not stationary, CNNs also require less trainable parameters due to the *weight-sharing* nature of convolutional operations, reducing the risk of overfitting compared with FCNNs/MLPs.

In addition, by cascading multiple convolutional operations CNNs can adapt better, exposing more higher-order features within the source images than FCNNs/MLPs. For example, [130–133] explicitly visualize the learned feature maps from convolutional layers, demonstrating the capability of CNNs to extract and analyse edges, curved shapes or even faces and objects, when the source image propagates toward deeper convolutional layers. This also explains why CNNs have been successfully employed in many high-level computer vision tasks, such as classification and object detection [134–139].

For the purpose of compression, CNN-based compression is comparable with many existing non-learning based image compression standards, such as the JPEG and the JPEG 2000 standards; however, FCNN/MLP-based compression that we mentioned in Section 6.2.1 can hardly achieve this performance.

In terms of the network structure, the so-called *convolutional layer* is at the heart of CNNs, which employs filtering to exploit the correlation between neighbouring image samples. Each layer consists of a bank of learned filters to produce a set of two-dimensional *feature maps* from the original image, or from the output feature maps from the previous layer; the outputs from each filter are subjected to a simple non-linearity, such as the RELU activiation function – a half rectifier that truncates negative values to 0. By cascading many convolutional layers as seen in Fig. 6.5, hierarchical statistical redundancy within the source image can be exploited [140].

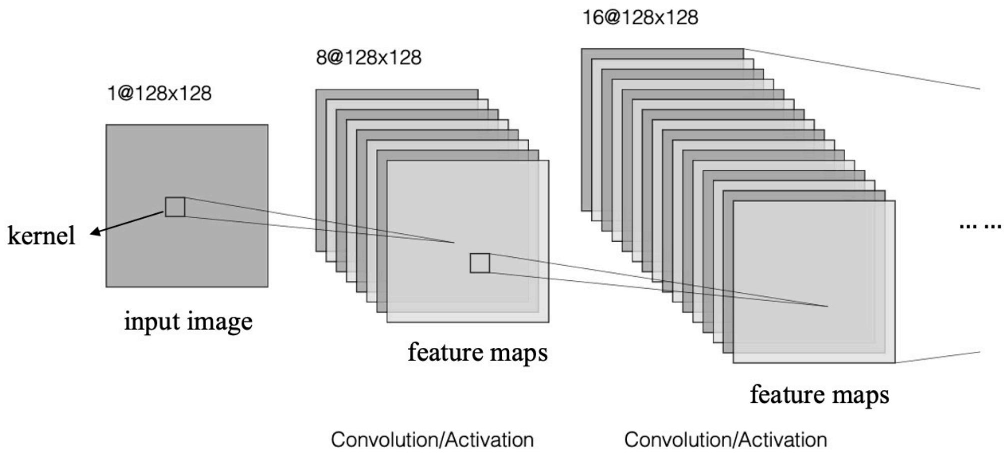


Figure 6.5: An example of cascading multiple (in this case two) convolutional layers as employed in typical CNNs, in which NxN denotes the size of the input image patch or each feature map. C@ represents the number of kernels employed in each convolutional layer.

To reduce the spatial size while retaining the dominant features that are rotational and positional invariant, a *pooling layer* is often placed between the adjacent convolutional layers for down-sampling as seen in Fig. 6.6. The two commonly used pooling/down-sampling approaches are *maximum pooling* and *average pooling*; maximum pooling selects the largest value within each *receptive field* (the area that the CNN kernel covers) for down-sampling, whereas average pooling returns the averaged value as output.

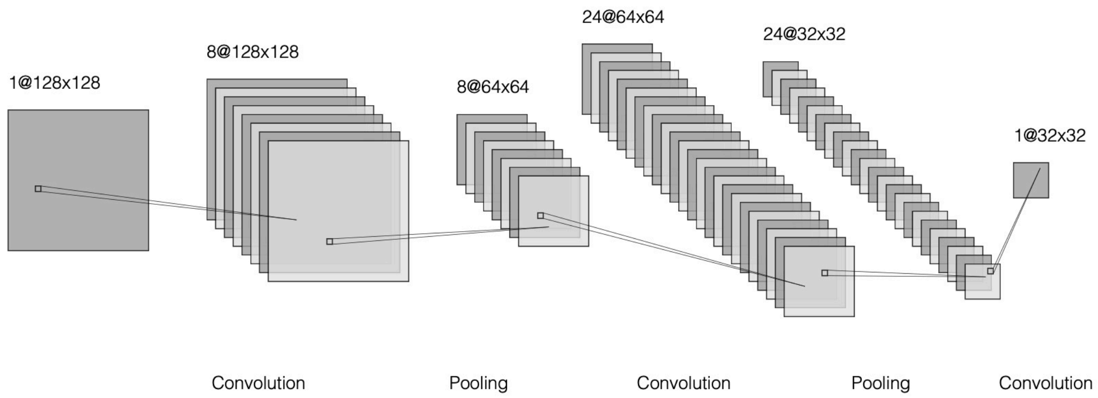


Figure 6.6: A typical CNN structure with convolutional layers and pooling layers.

To experiment with CNN-based lossy image compression schemes, Ballè et al. first introduced an end-to-end optimized CNN for image compression with uniform scalar quantization [11, 27], in which generalized divisive normalization (GDN) with optimized parameters [141] was employed to reduce mutual information between the transformed channels from pooling layers. The authors demonstrated that this CNN-based scheme outperformed the JPEG and the JPEG 2000 standards under both PSNR and MS-SSIM metrics, along with more perceptually appealing reconstructed images. Subsequently, they developed an additional scale hyperprior in [28] to further exploit spatial dependencies in the latent representation, which approximated the performance of HEVC in terms of PSNR. Similarly, Theis et al. [29] proposed to employ Gaussian scale mixtures to model the probability density function of quantization indices and to estimate their entropy; the entropy was then used to estimate bit-rates and drive the backpropagation-based training. Other works that focus on improving entropy estimation of CNN-based compression systems can be found in [142–149].

Many researchers managed to improve other aspects of CNN-based compression systems. For example, [150] proposed a hierarchical distortion loss function to protect both pixel-level fidelity for region of interest and improved structural similarity for the reconstructed image. Liu et al. [151] proposed to utilize the combination of a perception loss and an additional adversarial loss to improve the subjective reconstruction quality. Other modifications to the loss function can be found in [152, 153]. Moreover, other artifact removal techniques were also explored to improve reconstruction quality of CNN-based compression systems; examples include a U-Net-like deblocking network [154] and a multi-scale reshuffling network [155]. In addition, efforts were also invested to incorporate pre-processing or side information to improve the performance of CNNs; examples can be found in [156, 157], which proposed to utilize the wavelet decomposition as pre-steps in addition to deep CNNs, whereas Ayzik et al. [158] considered using side information containing synthetic images to obtain high-quality reconstructed images.

6.2.3 Recurrent Neural Networks (RNNs)

In contrast to the CNN architectures, another class of neural networks, named Recurrent Neural Networks (RNNs), are developed explicitly for the purpose of learning long-dependencies amongst source image samples [159, 160]. The typical RNN structure allows the outputs o_{t-1} of the hidden unit H_{t-1} to be connected/combined with the inputs x_t to the next hidden unit H_t, as illustrated in Fig. 6.7; the hidden units H_t can consist of one or multiple hidden layers (either fully connected layers or convolutional layers), each of which has its own weights, bias and activation functions. Thanks to the feedback loops adopted by the RNNs, the transformed information from previous executions can be propagated and affect the context of the inputs to the current unit, exploiting long-dependencies between input image samples or patches.

Although conceptually RNNs are capable of exposing long-dependencies, theoretical analysis has shown that this particular ability of RNNs actually fades away as time t increases [161] [162]. The reason for this is because the errors back-propagated through

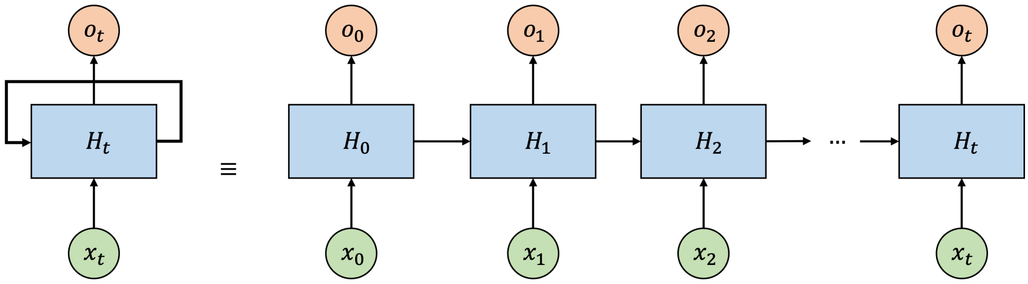


Figure 6.7: The typical RNN structure with hidden units H_t connected with themselves via a feedback loop.

RNNs often vanish or explode exponentially when the network structure goes deeper; see more mathematical explanations in [162].

To resolve the vanishing/exploding gradient issue, Long Short-Term Memory (LSTM) units are proposed, which can be used in hidden layers of a RNN, in order to maintain a constant error flow during back-propagation [7, 163]. Fig. 6.8 illustrates the typical diagram of a LSTM block; it is composed of an input gate, a forget gate, an output gate and a self-connected cell in the middle. The input gate controls the new information that the neural network is going to store in the cell state, while the forget gate determines the old information that the network is going to throw away from the cell state. The self-connected cell has two functions – updating the old cell state with the new cell state, while trapping the error signal to maintain a constant error flow for back-propagation [7]. In the end, the output gate decides which part of the cell state should be outputted.

There are other works developed to improve the performance of RNNs; representative examples include adding *peephole* connections [164] to LSTM blocks, as well as Gate Recurrent Unit (GRU) which simplifies the LSTM by combining the input and the forget gates into a single update gate [165].

Inspired by the successful deployment of RNNs in many high-level computer vision and speech tasks [166–168], researchers also started to experiment with image compression tasks using RNNs. Toderici et al. [30] first introduced an RNN architecture for compressing 32 x 32 thumbnail images in a variable rate compression scheme. This work was further extended in [8], in which a general RNN-based framework for compressing full resolution

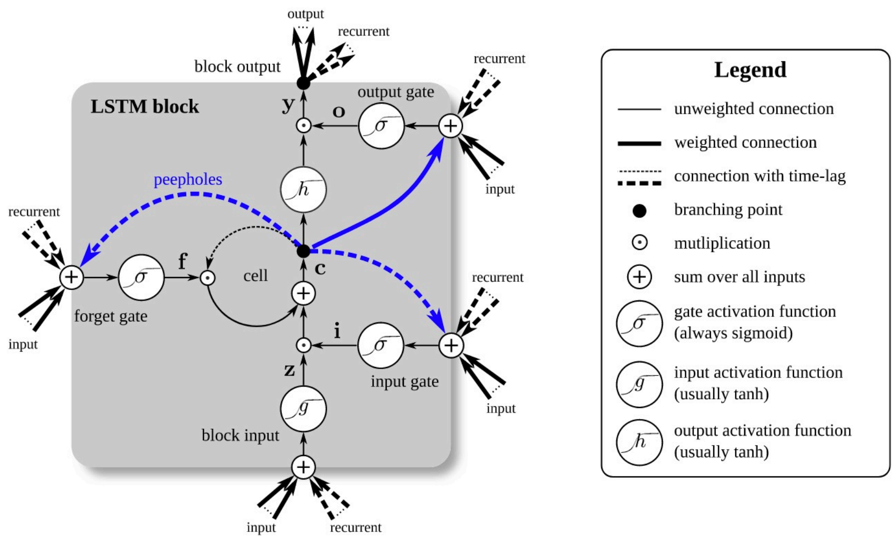


Figure 6.8: The diagram of the peephole LSTM, which is used in the hidden layers of a RNN [7].

images was proposed. Specifically, each iteration in [8] consisted of an encoding network, a binarizer and a decoding network as seen in Fig. 6.9; only the encoding and the decoding networks were RNNs. The residuals between the input and the reconstructed image from the decoding network were fed again as inputs to the encoding network at the next iteration for further compression. In this way, this scheme supported variable bit-rate compression in a progressive manner, with demonstrated improvement over the JPEG compression standard.

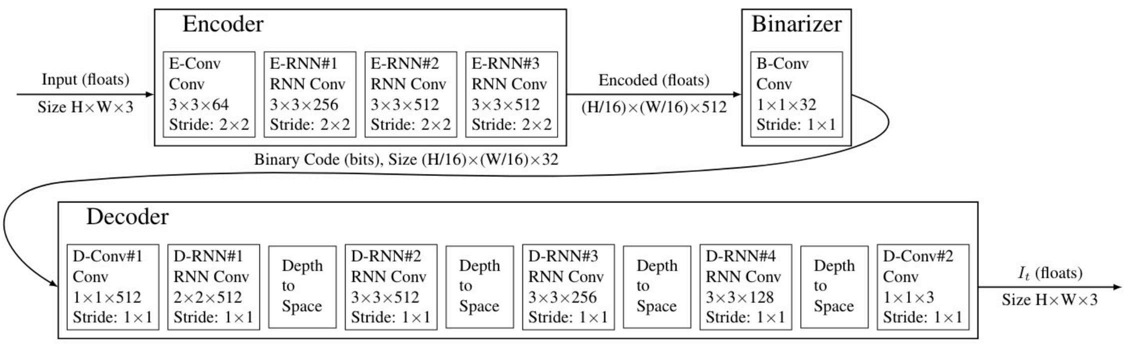
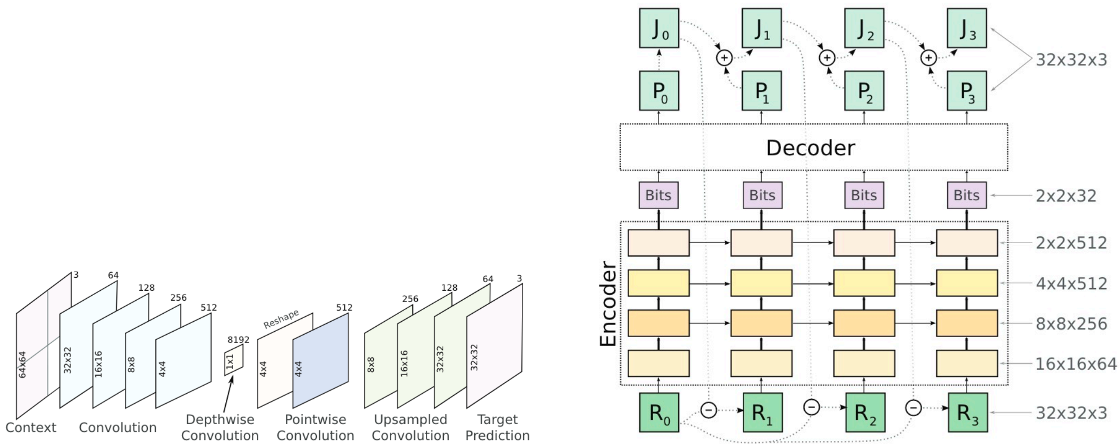


Figure 6.9: A single iteration of the proposed RNN structure in [8].

The work was further improved in [9], in which deep RNNs were combined with an additional tiled network to achieve quality-sensitive bit-rate adaptation, so as to improve both PSNR and qualitative performance. The proposed network first divided input images into tiles, each of which was preliminarily predicted using a CNN-based spatial context predictor as seen in Fig. 6.10(a). Prediction residuals were then encoded progressively using a deep RNN, so that the sum of iteration outputs P_i approximated the original image, as illustrated in Fig. 6.10(b). The authors demonstrated that by introducing this spatially adaptive tiled network, significantly better rate-distortion performance and reconstruction quality could be obtained in comparison to [8] and the JPEG standard. Similar work can also be found in [169], which proposed to embed convolutional and generative divisive normalization layers to achieve bit-rate adaptation of the RNN-base compression framework.



(a) (b)

Figure 6.10: The proposed spatially adaptive RNN-based compression scheme in [9]. (a) illustrates the context prediction network, which uses convolutional networks to extract features from the context tiles and generate a prediction for the target tile. (b) shows the residual encoder with deep RNN structure; R_i denotes the residuals from the context predictor as the input to each hidden RNN unit while P_i is the output of each iteration.

There are many other works focusing on improving other aspects of RNN-based compression. For example, Lin et al. [170] proposed to utilize two hyperprior networks to better estimate entropy parameters for improved compression performance; the authors also employed a block-based LSTM recurrent network to further exploit redundancy between adjacent image patches. Similarly, Johnston et al. [31] also developed a spatial contextual entropy model to improve entropy estimations in a RNN-based compression scheme. Other methods on improving entropy models within RNN-based compressions can be found in [171, 172]. Interested readers can also refer to [173–175] for more RNN-related works.

6.2.4 Generative Adversarial Neural Networks (GANs)

As we have discussed, the autoencoders in Section 6.2.1, Section 6.2.2 and Section 6.2.3 are usually trained with reference-based distortion metrics, such as mean squared error (MSE), which poorly reflects the perceptual fidelity of natural images. One way to address this issue is to utilize another learned machine that acts as a non-reference-based distortion measurement, whose goal is to discriminate images with good reconstruction quality from those with bad quality; this leads to the concept of Generative Adversarial Networks (GANs). As we shall see, GANs are traditionally utilized to generate realistic images from random noise. However, for the purpose of image compression, GANs are often used as a learnable replacement for simple objective distortion metrics such as MSE in autoencoders, so as to improve perceptual quality of the reconstructed images.

The concept of GAN was first introduced by Goodfellow et al. in [176], in which two network models, called *generator* and *discriminator*, are jointly optimized to minimize an adversarial loss. Specifically, the main role of the generator is to learn the function $G(\mathbf{z}, \boldsymbol{\theta}_G)$ with trainable parameters $\boldsymbol{\theta}_G$, which can transform unstructured noise input $\mathbf{z} \sim p_{\mathbf{z}}(\mathbf{z})$ into realistic samples to fool the discriminator. In contrast, the discriminator inspects its input $\mathbf{x}$ and returns an estimation $\mathcal{D}(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{D}})$ with trainable parameters $\boldsymbol{\theta}_{\mathcal{D}}$; this estimation $\mathcal{D}(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{D}})$ determines whether the input samples $\mathbf{x}$ are real images (following the distribution of real data p_{data}) or fake images produced by the generator. The generator G and the discriminator $\mathcal{D}$ are optimized simultaneously using back-propagation during

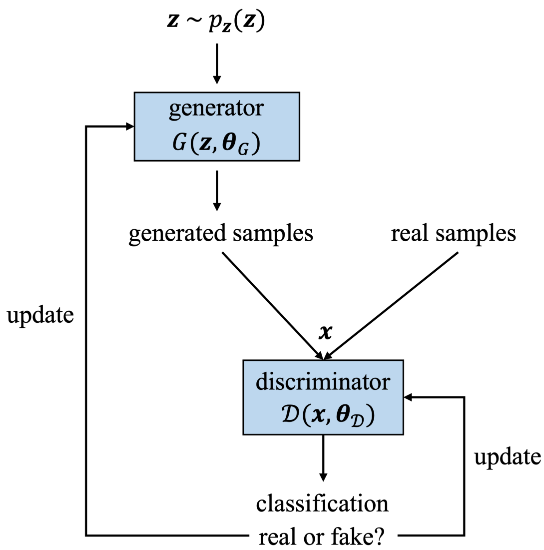


Figure 6.11: The typical GAN structure with the generator and the discriminator models.

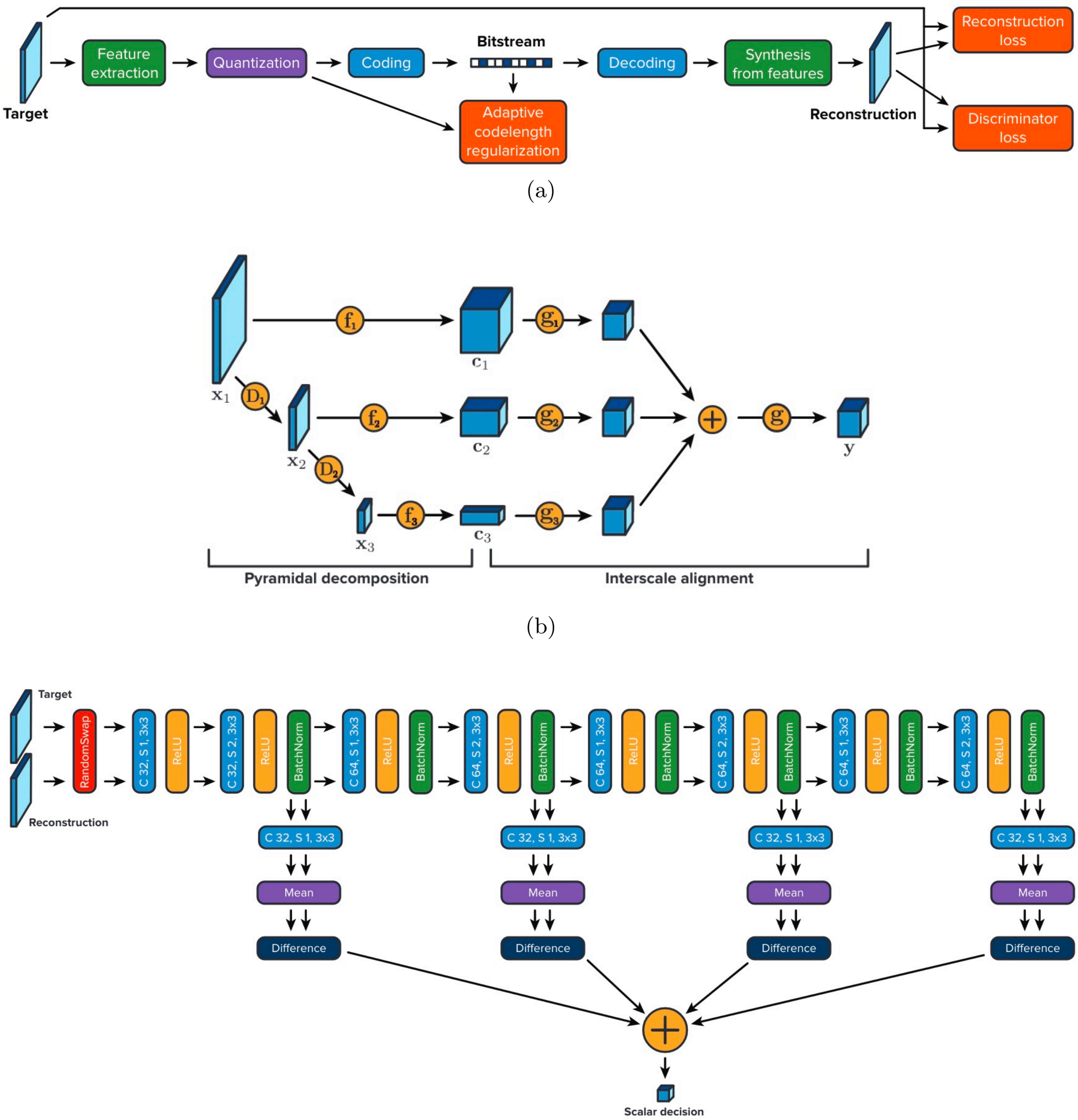
training for the adversarial loss J as

$$\min_{G} \max_{\mathcal{D}} J\left(\boldsymbol{\theta}_{\mathcal{D}}, \boldsymbol{\theta}_{G}\right) = E_{\mathbf{x}\sim p_{\mathrm{data}}(\mathbf{x})}\left[\log \mathcal{D}\left(\mathbf{x}, \boldsymbol{\theta}_{\mathcal{D}}\right)\right] + E_{\mathbf{z}\sim p_{\mathbf{z}}(\mathbf{z})}\left[\left(\log(1 - \mathcal{D}(G(\mathbf{z}, \boldsymbol{\theta}_{G}))\right)\right] \quad (6.3)$$

An the end of training, an equilibrium point should be achieved, i.e. the samples produced by the generator G are indistinguishable from the real data by the discriminator $\mathcal{D}$. An illustration of the typical GAN structure can be found in Fig. 6.11.

There are many variations of the GAN structure in the literature; examples include cGAN [177], InfoGAN [178], CoGAN [179], cycleGAN [180] and WGAN [181]. As these variants are barely used for compression purposes, we will not discuss them any further here.

In the context of image compression applications, GANs are often utilized to enhance visual quality of reconstructed images especially at very low bit-rate, which also improves coding efficiency. Rippel and Bourdev [10] were the first to introduce the adversarial loss within a pyramidal autoencoder for real-time image compression. Fig. 6.12 illustrates the proposed GAN-based compression scheme in [10]. The encoder-decoder pipeline is considered as the generator model in GAN, which compresses the input image via a pyramidal feature extraction module as seen in Fig. 6.12(b), and then reconstructs it to calculate the reconstruction loss. The discriminator, on the other hand, has a multiscale archi-



(c)

Figure 6.12: (a) illustrates the general GAN-based compression scheme proposed in [10]. (b) is the insight of the feature extraction module, which analyses the input using a pyramidal decomposition and then aligns feature maps across different scales to discover the joint structure. (c) represents the proposed discriminator structure used in adversarial training; scalar outputs at different scales are accumulated and averaged to attain the final value provided to the objective sigmoid function, in order to discriminate between the target and the reconstructed images.

tecture as seen in Fig. 6.12(c), which utilizes an averaged scalar decision to discriminate between the target/original and the reconstructed images. The authors demonstrate that this adversarial architecture significantly improves the perceptual quality of reconstructed images over the JPEG, the JPEG 2000 and the WebP standards, along with much smaller compression file sizes.

Similarly, Agustsson et al. also proposed a GAN-based scheme with a multiscale discriminator for extremely low bit-rate image compression [182]. The authors explicitly show that the proposed GAN structure is capable of synthesizing more textures for homogeneous regions in decoded images with the aid of a semantic label map, yielding more visually appealing reconstructions. On the other hand, the GAN-based view synthesis is also explored in the context of Light Field (LF) image compression [183]; by incorporating a unsupervised perceptual learning model with the proposed LF-GAN, the contents of an arbitrary positioned sub-aperture image can be reliably generated, along with the state-of-the-art compression performance. Similar work can be found in [184]. Interested readers can also refer to [185–187] for more GAN-related compression approaches.

6.3 Existing Training Strategies

As we have introduced in Section 6.1, autoencoders are often optimized in an end-to-end fashion for the rate-distortion objective (6.1) in the context of image compression. Most of the machine-learning optimization techniques, e.g. gradient decent, rely on differentiability for back-propagation [121]. However, both the distortion and the average coded length in (6.1) depend on the quantizer, whose derivative is either zero or infinity everywhere. In the literature, many approaches have been proposed to resolve this issue, which can be broadly categorized into additive noise approaches, straight-through-estimator (STE) and soft-to-hard annealing.

6.3.1 Additive Noise Approaches

Ballè et al. first proposed to replace the quantizer with an additive uniform noise source [11]; similar work can also be found in [188]. Specifically, a uniform scalar quantizer with quantization step size $\Delta = 1$ operating on the transform coefficient y_i can be denoted as

$$q_i = \mathrm{round}(y_i) \tag{6.4}$$

Then the probability mass of the n^{th} quantization bin can be calculated as

$$P_{q_i}(n) = \left(p_{y_i} * \mathrm{rect}\right)(n), \quad \text{for all} \quad n \in \mathbb{Z} \tag{6.5}$$

where $\mathrm{rect}(t)$ is a uniform distribution on the interval $[-\frac{1}{2}, \frac{1}{2}]$ and the symbol $*$ denotes the convolution operation.

Suppose we add independent uniformly distributed noise $\Delta y_i \sim \mathrm{rect}$ to the transform coefficient y_i, forming $\tilde{y}_i = y_i + \Delta y_i$. It can be proven that the probability density function of $\tilde{y}_i$, i.e.

$$p_{\tilde{y}_i} = p_{y_i} * \mathrm{rect} \tag{6.6}$$

is identical to P_{q_i} at all integer locations, yielding a continuous relaxation of the uniform scalar quantizer as seen in Fig. 6.13.

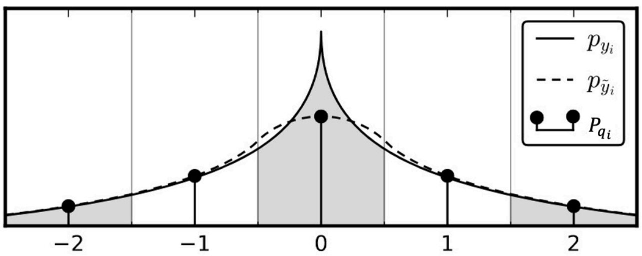


Figure 6.13: The additive noise approach proposed in [11]. We can see that $p_{\tilde{y}_i}$ is a continuous relaxation of P_{q_i} across all quantization bins.

However, these additive noise models have the *forward-backward discrepancy*, because the forward-pass uses realizations of the noise model while the backward-pass employs the probability distribution (statistical model) of the noise. Moreover, these approaches also suffer from *train-test discrepancy* that the forward-pass with additive noise is different from the actual quantization that takes place in the real testing phase.

6.3.2 Straight-through-estimator (STE)

Another approach is to use a straight-through estimator (STE) [189, 190] to overcome the discontinuity of the quantization operation. The idea of STE is to include the discontinuous quantization operation directly in the forward path of the network, but replace it with the identify during back-propagation. With enough diversity of content, this approach, which essentially draws a straight line through the quantization step function for the purpose of back propagation, should lead to more accurate modelling than additive noise approaches.

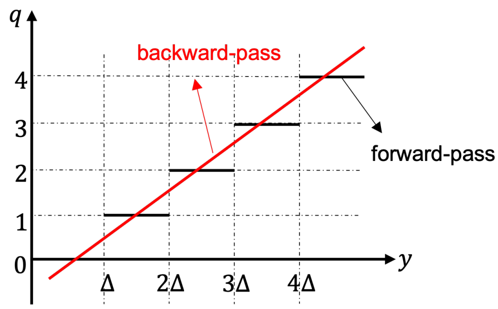


Figure 6.14: The core concept of the STE, which essentially draws a straight line through the quantization step function for the purpose of back propagation. The forward pass still adopts the non-differentiable quantization step function.

As illustrated in Fig. 6.14, the STE approach essentially avoids the train-test discrepancy, as the forward-pass of the STE retains the same non-differentiable quantization behaviour as in the testing phase; therefore, it has been proven to have a major benefit in performance over the additive noise approaches. However, the STE approach still suffers from the forward-backward mismatch, which has a fundamental impact on the convergence of the learning system.

6.3.3 Soft-to-hard Annealing

In contrast, soft-to-hard annealing approaches are proposed to develop a continuous relaxation of quantization for both the forward- and the backward-pass during training [191, 192]. As illustrated in Fig. 6.15, the derivative of the continuous relaxation function is well defined everywhere, and so suitable for the purpose of back-propagation. Since the forward- and the backward-pass share the same behaviour during training, there is no forward-backward mismatch in soft-to-hard annealing approaches. In addition, soft-to-hard annealing approaches utilize some *cooling parameters* to gradually anneal this continuous relaxation toward the actual non-differentiable quantization step function throughout training to reduce the train-test discrepancy. However, the neural networks in these annealing methods do not have early access to the real quantized data that is actually used during testing, which might present adverse impact on the performance of the learning system.

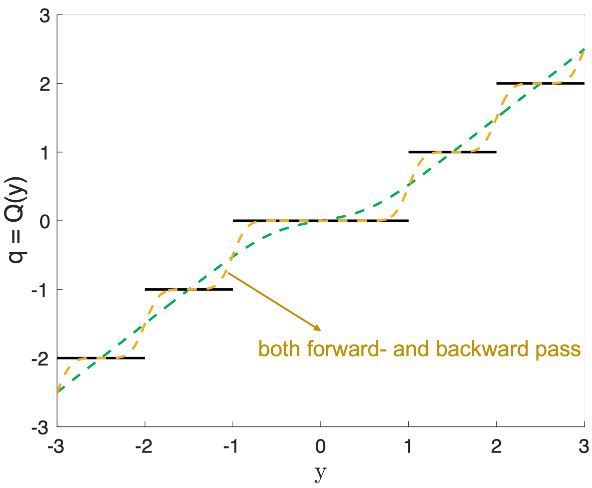


Figure 6.15: The illustration of soft-to-hard annealing approaches, in which both the forward- and the backward-pass of back-propagation employ a continuous relaxation function (green dashed curves). This continuous relaxation gets gradually annealed toward the actual quantization function throughout training, as seen in the yellow dashed curves, to reduce the train-test discrepancy.

Chapter 7

Contribution: The Significance of Geometric Flow

This chapter highlights the generic problems and opportunities associated with reducing residual redundancy in the wavelet transform, especially with the aid of geometric flow in the two-dimensional (2D) scenario; the significance of geometric flow underpins this thesis.

Although deterministic redundancy, i.e. oversampling, is avoided in the wavelet representation, statistical redundancy, especially the aliasing-related residual redundancy, is still inevitably presented amongst the wavelet subbands. This is because the wavelet transform imposes strong conditions on the critically sampled filter banks, which prevents the redundancy from being eliminated between different subbands. Specifically, the analysis and synthesis filters h_0 and g_0 of a two-channel critically sampled filter bank must satisfy the following constraint in the Fourier domain:

$$\hat{h}_0(\omega)\hat{g}_0(\omega) + \hat{h}_0(\pi - \omega)\hat{g}_0(\pi - \omega) = 1 \tag{7.1}$$

which means in particular that $\hat{h}_0(\pi - \omega)\hat{g}_0(\pi - \omega) = \frac{1}{2}$ at $\omega = \frac{\pi}{2}$. Since finite support filters must have continuous transfer functions, the low-pass analysis filter h_0 must have a significant response to frequencies $\omega > \frac{\pi}{2}$, which corresponds to aliasing in the low-pass subband. This aliasing content is both visually disturbing and a form of redundancy.

Similarly the high-pass analysis filter h_1, which is in mirror symmetry with g_0, necessarily has a significant response to frequencies $\omega < \frac{\pi}{2}$. This pollution of the high-pass subband with low frequency content is another form of information redundancy between the two subbands.

The goal of this thesis is to reduce this redundant content between the low- and high-pass subbands in the wavelet transform. There are many possible ways to achieve this. One can attempt to improve the subband filters themselves, but no amount of improvement in the filters can escape from the implications of (7.1), which indicates that h_0 cannot be designed to avoid aliasing. If h_0 does manage to roll off close to 0 by $\frac{\pi}{2}$, then g_0 would require a huge gain around the half-band frequency $\frac{\pi}{2}$, greatly amplifying quantization errors. The same is true for the high-pass analysis and synthesis filters h_1 and g_1. Although designs involving longer filters can have smaller transition bands around $\frac{\pi}{2}$, this comes at the cost of a loss of sparsity – innovative features such as edges in the space domain produce more non-zero subband samples, which adversely impact coding efficiency.

More generally, additional operators could be introduced to untangle the redundant information amongst the wavelet coefficients, which is the main focus of this thesis. Let $\mathcal{A}_L(x)$ and $\mathcal{A}_H(x)$ denote the analysis of signal x into the low-pass band $y_L = \mathcal{A}_L(x)$ and the high-pass (detail) subband $y_H = \mathcal{A}_H(x)$ respectively, within one level of a Discrete Wavelet Transform (DWT). For a 2D DWT following the Mallat decomposition structure, y_H stands for the collection of all three detail subbands, denoted as HL, LH and HH, but most of the material in this chapter is most easily presented in 1D in the first instance.

In particular, suppose an operator $\mathcal{T}^A_{H2L}$ can be found to estimate the aliased component $\tilde{y}_L$ of y_L using y_H, written as $\tilde{y}_L = \mathcal{T}^A_{H2L}(y_H)$, then the non-aliased component $\bar{y}_L$ can be separated from y_L as $\bar{y}_L = y_L - \tilde{y}_L$. Since $\bar{y}_L$ is at least approximately free from aliasing, now all of the aliasing information $\tilde{y}_H$ inside y_H arises from the content in $\bar{y}_L$. This means that $\bar{y}_L$ can then be used to discover the aliasing contribution within y_H, written as $\tilde{y}_H = \mathcal{T}^W_{L2H}(\bar{y}_L)$. In fact, the operator $\mathcal{T}^W_{L2H}$ can simply be an LSI filter, because $\tilde{y}_H$ should ideally be equal to $\mathcal{A}_H(\mathcal{I}(\bar{y}_L))$, where $\mathcal{I}$ stands for the ideal interpolator. We have chosen to use the superscript ^{W} for this second operator, to emphasize the fact that it could potentially be obtained as a conventional Wiener filter.

In this scenario, the development of the operator $\mathcal{T}^A_{H2L}$ is the one that presents the greatest challenge; it cannot simply be an LSI filter. If it were, then the transformation steps $\bar{y}_L = y_L - \mathcal{T}^A_{H2L}(y_H)$, followed by $\bar{y}_H = y_H - \mathcal{T}^W_{L2H}(\bar{y}_L)$ could be seen as augmenting the original wavelet transform with two additional LSI lifting steps, so that the complete transform is equivalent to choosing a different set of subband analysis and synthesis filters, which we have already ruled out as a viable solution. As we shall see, the operator $\mathcal{T}^A_{H2L}$ should at least be adaptive to local geometric structure, and we use the superscript ^{A} here to highlight both its role in untangling aliasing and the need for local adaptivity.

From a different perspective, suppose an operator $\mathcal{T}^A_{L2H}$ can be developed to discover the aliased part $\tilde{y}_H$ of y_H using y_L, written as $\tilde{y}_H = \mathcal{T}^A_{L2H}(y_L)$. Then the "cleaned" high-pass band $\bar{y}_H = y_H - \tilde{y}_H$ can be used to untangle the aliasing information $\tilde{y}_L$ in y_L using an operator $\mathcal{T}^W_{H2L}$. In this converse scenario, the operator $\mathcal{T}^W_{H2L}$ becomes conceptually simple, potentially being a Wiener filter, whereas $\mathcal{T}^A_{L2H}$ cannot be LSI.

The two perspectives demonstrate that at least one of these two operators $\mathcal{T}^A_{L2H}$ and $\mathcal{T}^A_{H2L}$ is difficult to develop, depending on which we choose to perform first. Although these two approaches may seem equally plausible considering only one level of DWT decomposition, the difference appears with multiple levels of decomposition. This will be further discussed in Chapter 8.

For the moment, considering only one level of decomposition, we first examine the fundamental difficulties and opportunities to untangle the aliasing in the high-pass band using the low-pass band, i.e. the operator $\mathcal{T}^A_{L2H}$. Although there is no general deterministic way to construct $\mathcal{T}^A_{L2H}$ to untangle the aliasing, prior statistical signal models can be used to derive a posterior distribution for the aliasing component, from which an estimate can be formed. This is essentially the basis of super resolution algorithms, for which the key challenge is to estimate original high frequency components that appear as aliasing in a low-resolution source image, as discussed previously in Section 5.3.

Estimating the aliasing component of a signal is much easier to do in the image domain than in one dimension, since geometric flow in images provides a strong form of prior knowledge. Specifically, we expect that edges in the underlying spatially continuous image are smooth along their contours, so that the innovative aspects of an edge, namely its profile, change only slowly along the edge, i.e. along the geometric flow. This geometric regularity provides an opportunity to untangle aliasing in the 2D-DWT.

We can see this more concretely by considering a continuous and consistently oriented signal $f(s_1, s_2)$, such that the edge profile is exactly the same along an orientation with slope α as shown in Fig. 7.1(a). This 2D continuous signal $f(s_1, s_2)$ can be understood as an ensemble of multiple shifted copies of the prototype 1D signal $f(s_1)$. That is $f(s_1, s_2) \equiv f_{s_2}(s_1) = f(s_1 - \alpha \cdot s_2)$, where f_{s_2} represents the horizontal cross-section of f at the vertical position s_2 as highlighted in Fig. 7.1(a). In this thesis, the 2D underlying continuous signal f is a Nyquist band-limited image, whose samples correspond to the discrete image x. To model the discrete wavelet transformation of x, f is then subjected to the continuous analogues of the wavelet analysis low-pass filter h_L and high-pass filter h_H, producing the low- and high-pass images f_L and f_H respectively.

The cross-section f_{L,s_2} of f_L and its discrete counterpart x_{L,n_2} can be written in the horizontal Fourier domain as

$$\begin{aligned} \hat{f}_{L,s_2}(\omega) &= \hat{h}_L(\omega)\hat{f}(\omega)e^{-j\alpha s_2 \omega} \\ &= \hat{h}_L(\omega)\hat{x}(\omega)e^{-j\alpha n_2 \omega} = \hat{x}_{L,n_2}(\omega) \end{aligned} \tag{7.2}$$

The discrete wavelet low-pass subband y_{L,n_2} is just a sub-sampled version of x_{L,n_2}; considering only one level of decomposition, y_{L,n_2} can be written as

$$\begin{aligned} \hat{y}_{L,n_2}(\omega) &= \tfrac{1}{2}\hat{x}_{L,n_2}(\omega/2) + \tfrac{1}{2}\hat{x}_{L,n_2}(\omega/2 - \pi) \\ &= \tfrac{1}{2}\hat{h}_L(\omega/2)\hat{x}(\omega/2)e^{-j\alpha n_2 \omega/2} \\ &+ \tfrac{1}{2}\hat{h}_L(\omega/2 - \pi)\hat{x}(\omega/2 - \pi)e^{-j\alpha n_2(\omega/2-\pi)} \end{aligned} \tag{7.3}$$

which reveals its aliased and non-aliased components; an illustration of this can be found in Fig. 7.1(b).

Averaging the inverse shifted signals over a vertical neighborhood $\mathcal{N}_2$ yields

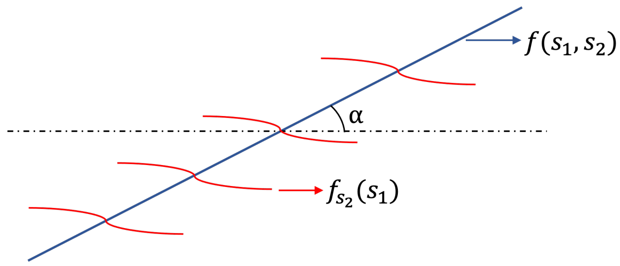
$$\begin{aligned}\bar{y}_L(\omega) &= \frac{2}{\|\mathcal{N}_2\|}\sum_{\mathcal{N}_2}\hat{y}_{L,n_2}(\omega)e^{j\alpha n_2\omega/2} \\ &= \hat{h}_L(\omega/2)\hat{x}(\omega/2) + \frac{2}{\|\mathcal{N}_2\|}\sum_{\mathcal{N}_2}\hat{h}_L(\omega/2-\pi)\hat{x}(\omega/2-\pi)e^{j\alpha n_2\pi}\end{aligned} \tag{7.4}$$

The last term above averages aliasing components and can be expected to be small, so long as α is not an integer and the averaging neighbourhood is sufficiently large. As a result, $\bar{y}_L(\omega) \approx \hat{h}_L(\omega/2)\hat{x}(\omega/2)$. Once aliasing components are effectively untangled, $\bar{y}_L$ can then be employed to estimate the aliasing contribution $\tilde{y}_H$ within the wavelet high-pass subband y_H, using an LSI filter as we have elaborated before. This entire process, starting from y_L to untangle $\tilde{y}_H$, provides a viable solution for constructing $\mathcal{T}^A_{L2H}$.

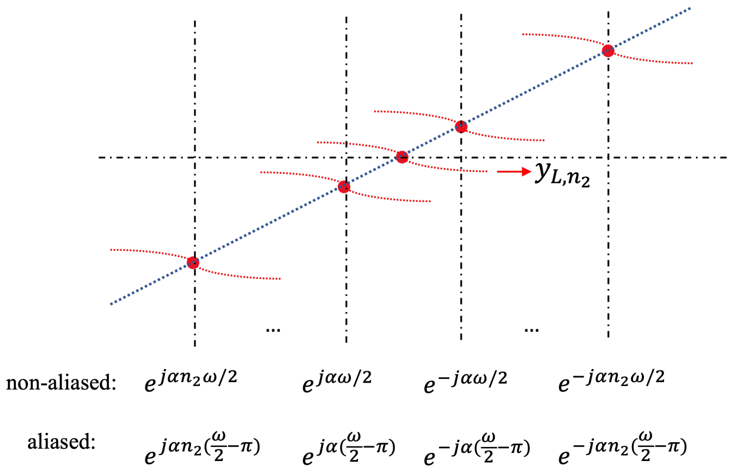
Moreover, $\bar{y}_L$ can be combined with y_L to recover an estimate of the original image f_L. This demonstrates the connection between untangling aliasing from a low-pass subband and the well studied problem of super resolution. More generally, the simple averaging process suggested above can be replaced by a Wiener filter. As we have explicitly addressed in Section 5.3, [116] proves that given multiple aliased views of the same underlying continuous image, where each view is obtained with a different shift, the minimum mean squared error best estimate of the original scene can indeed be found using Wiener filtering.

That is, the problem of untangling aliasing can in fact be solved using a filter-based strategy, so long as we can identify multiple copies of the same underlying feature, with known shifts between each copy – i.e. known geometric flow. Since geometric flow is a local property within an image, the untangling of aliasing requires either an adaptive filtering solution or a bank of filters with an adaptive strategy for combining their responses, so the overall operator $\mathcal{T}^A_{L2H}$ cannot be LSI and will generally need to be non-linear. As we shall see in Chapter 9, this is essentially the structure that we have found to work best. Although this discussion has been limited to the case in which we start from the low-pass subband[1], the dual problem, in which the first step uses the high-pass subband to discover and clean the redundant aliasing information, has exactly the same properties.

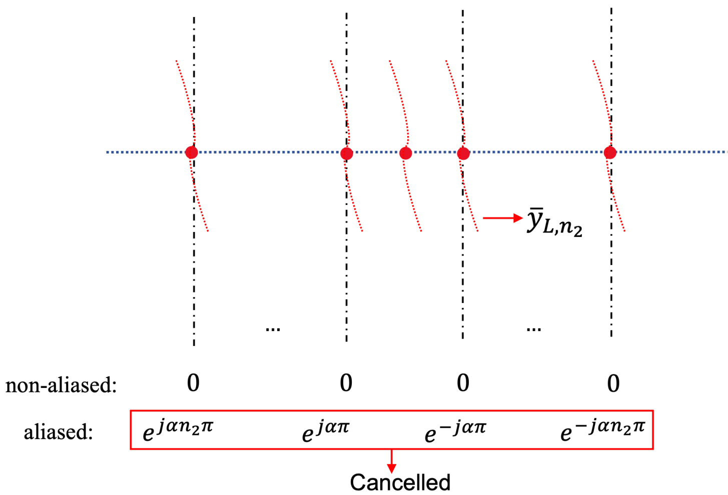
[1] We have done this to help clarify the connection with super resolution, which is always understood as starting from a low-resolution image.



(a)



(b)



(c)

Figure 7.1: (a) shows the orientated image feature along with its geometric flow as highlighted in red, while (b) illustrates different phases of the non-aliased and aliased components after DWT filtering and down-sampling by a factor of 2. (c) demonstrates different phases of aliased components after compensation (inverse shift), which are eventually cancelled out over an averaging neighbourhood.

Chapter 8

Contribution: Structures to Augment The Existing Wavelet Transform For Reducing Redundancy

This chapter summarizes three generic architectures which can exploit geometric flow and untangle aliasing content within the wavelet subbands; we refer to these as low-to-high, high-to-low and hybrid approaches.

8.1 Low-to-high Approach

The low-to-high approach aims to suppress redundant information within the detail bands HL, LH and HH with the aid of the low-pass (LL) band from the same decomposition level, as illustrated in Fig. 8.1. We do this using an operator $\mathcal{T}^{A}_{L2H}$, which can be understood as forming a prediction of HL, LH and HH from the LL band. More specifically, we expect this operator to be able to exploit local geometric flow to predict the aliased components within HL, LH and HH, as explained before. Conceptually, if the operator $\mathcal{T}^{A}_{L2H}$ completely removes redundancy within the detail bands, then further cleaning aliasing $\tilde{\mathbf{y}}_{LL}$ in the LL band can be achieved simply using a linear operator $\mathcal{T}^{W}_{H2L}$ as explained in Chapter 7.

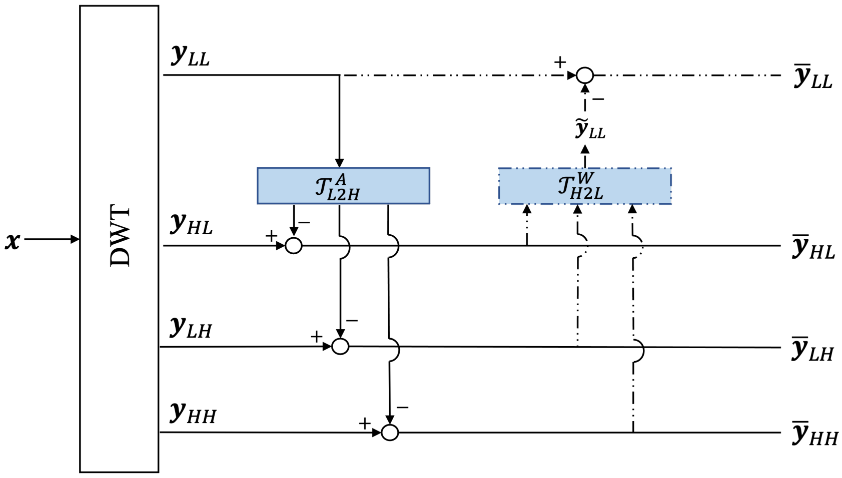


Figure 8.1: The architecture of the low-to-high approach, which can be viewed as additional lifting steps to the wavelet transform. The symbols $\mathbf{y}_{LL}$, $\mathbf{y}_{HL}$, $\mathbf{y}_{LH}$ and $\mathbf{y}_{HH}$ represent the LL, HL, LH and HH bands of the wavelet transform. The symbols $\bar{\mathbf{y}}_{LL}$, $\bar{\mathbf{y}}_{HL}$, $\bar{\mathbf{y}}_{LH}$ and $\bar{\mathbf{y}}_{HH}$ denote the less redundant ("cleaned") LL, HL, LH and HH bands. The dashed lines indicate that the operator $\mathcal{T}^{W}_{H2L}$ is only optional.

In our previous work [74], we proposed a simple yet effective, hand-tuned solution for the operator $\mathcal{T}^{A}_{L2H}$, which explicitly targets the discovery of local geometric flow in the low-pass band to untangle aliasing within the detail bands. This is certainly not the only way to design $\mathcal{T}^{A}_{L2H}$, and redundant information might be exploited in a more general way. The purpose of [74] is to demonstrate that an algorithm designed exclusively to exploit geometric flow, without statistical modeling or learning, is capable of untangling redundant information within the detail subbands. Moreover, since [74] successfully exploits local orientations in a critically sampled subband transform without any need to communicate side information, it provides an example method for addressing the key challenge of designing a feedforward system that is adaptive to local geometric features, as outlined in Section 3.5.

Unfortunately, this approach does not extend well to coarser levels in the wavelet decomposition. The reason for this can be understood with the aid of Fig. 8.2. We see the

LL band at the first level of decomposition ($\mathbf{y}_{LL,1}$) cannot be regarded as samples of a continuous Nyquist band-limited image, as it contains the aliasing component $\tilde{\mathbf{y}}_{LL,1}$ due to down-sampling. This aliasing component then accumulates through the DWT hierarchy, and forms part of the LL band at the next level of decomposition ($\mathbf{y}_{LL,2}$). Given the increasing amount of aliasing presented in $\mathbf{y}_{LL,2}$, it becomes harder to discover local properties such as geometric flow, reducing the effectiveness with which redundancy can be suppressed within the detail bands $\mathbf{y}_{HL,2}$, $\mathbf{y}_{LH,2}$ and $\mathbf{y}_{HH,2}$.

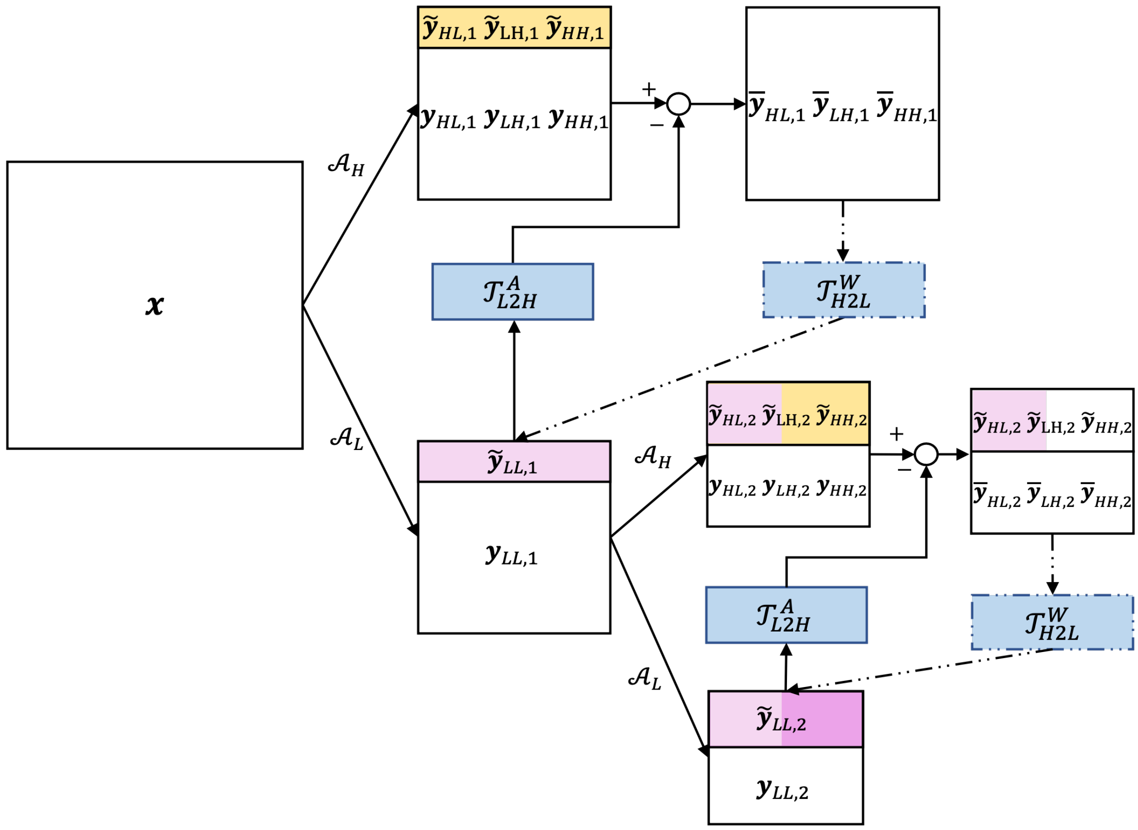


Figure 8.2: The illustration of extending the low-to-high approach to coarser levels, where $\mathbf{y}_{LL,d}$, $\mathbf{y}_{HL,d}$, $\mathbf{y}_{LH,d}$ and $\mathbf{y}_{HH,d}$ represent the low- and high-pass bands at the d^{th} level of decomposition. The symbols $\tilde{\mathbf{y}}_{LL,d}$, $\tilde{\mathbf{y}}_{HL,d}$, $\tilde{\mathbf{y}}_{LH,d}$ and $\tilde{\mathbf{y}}_{HH,d}$ denote the redundant (aliasing) information within the low- and high-pass bands at level d. The symbols $\bar{\mathbf{y}}_{HL,d}$, $\bar{\mathbf{y}}_{LH,d}$ and $\bar{\mathbf{y}}_{HH,d}$ stand for the less redundant detail bands after applying the operator $\mathcal{T}^A_{L2H}$. The dashed lines indicate that the operator $\mathcal{T}^W_{H2L}$ is only optional.

8.2 High-to-low Approach

In the light of this fundamental difficulty, we choose not to pursue the development of more sophisticated low-to-high approaches. Instead, we propose to adopt a high-to-low approach, which uses the high-pass subbands $\mathbf{y}_{HL,d}$, $\mathbf{y}_{LH,d}$ and $\mathbf{y}_{HH,d}$ to remove redundant aliasing $\tilde{\mathbf{y}}_{LL,d}$ from the LL band $\mathbf{y}_{LL,d}$ at each level d, before proceeding to the next level in the decomposition. We do this using an operator $\mathcal{T}^A_{H2L}$ as seen in Fig. 8.3. Similar to $\mathcal{T}^A_{L2H}$, we expect the operator $\mathcal{T}^A_{H2L}$ to also be capable of adaptively exploiting local geometric features from the detail bands to predict aliasing within the LL band. Conceptually, if the operator $\mathcal{T}^A_{H2L}$ successfully targets aliasing untangling within the LL band, then further reducing redundancy within the detail bands could be achieved simply using a linear operator $\mathcal{T}^W_{L2H}$ as explained in Chapter 7.

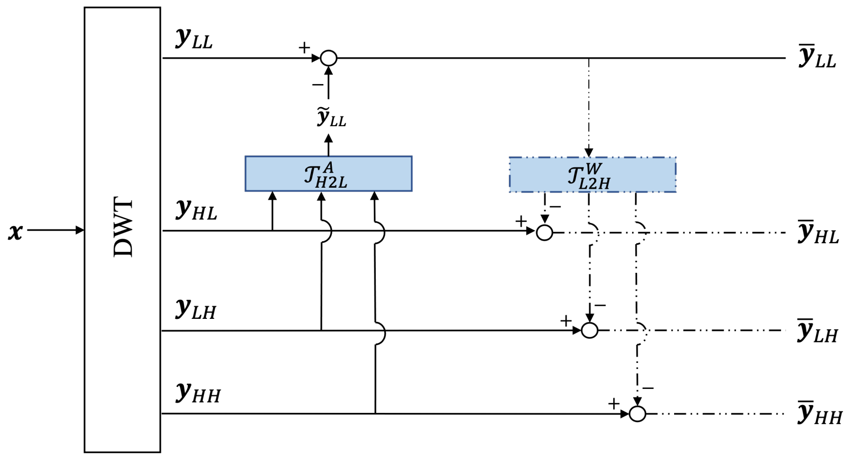


Figure 8.3: The architecture of the high-to-low approach, which can be viewed as additional lifting steps to the wavelet transform. The symbols $\mathbf{y}_{LL}$, $\mathbf{y}_{HL}$, $\mathbf{y}_{LH}$ and $\mathbf{y}_{HH}$ represent the LL, HL, LH and HH bands of the wavelet transform. The symbols $\bar{\mathbf{y}}_{LL}$, $\bar{\mathbf{y}}_{HL}$, $\bar{\mathbf{y}}_{LH}$ and $\bar{\mathbf{y}}_{HH}$ denote the less redundant ("cleaned") LL, HL, LH and HH bands. The dashed lines indicate that the operator $\mathcal{T}^W_{L2H}$ is only optional.

Contrary to the low-to-high approach, the high-to-low approach is expected to be more successful at untangling redundancy within the LL band; the accumulation of aliasing is then effectively avoided through the DWT hierarchy, which makes the method applicable to multiple levels of decomposition as seen in Fig. 8.4. Moreover, by effectively cleaning aliasing within the LL band at each level, reconstructed images at different scales indeed turn out to have significantly higher visual quality than the original LL bands obtained from the wavelet transform.

To develop the operator $\mathcal{T}^{A}_{H2L}$, preliminary experiments have been conducted for the high-to-low method using the hand-tuned solution presented in [74], which was not very successful. This is because it is more difficult to discover local geometric flow from the detail bands than from the low-pass band, at least without the aid of strong prior statistical models. For this reason, it seems appropriate to adopt machine learning as a tool for the methods presented in this section. More details concerning the proposed neural network structures themselves are presented in Chapter 9, but here we focus on architectural aspects.

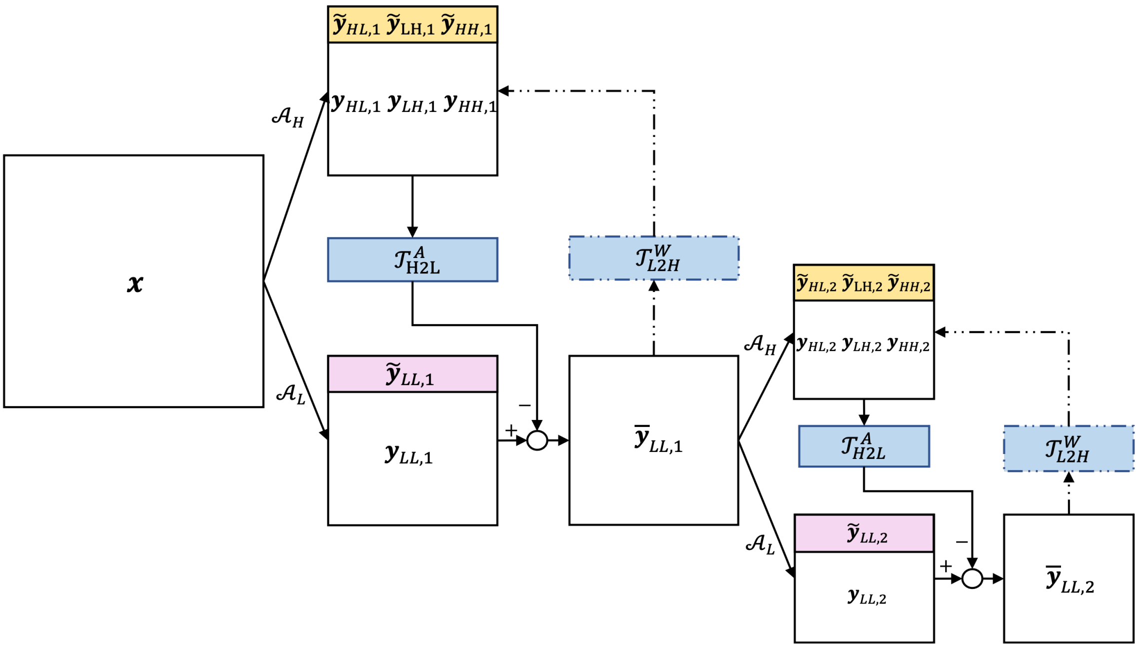


Figure 8.4: The illustration of extending the high-to-low approach to coarser levels, where $\mathbf{y}_{LL,d}$, $\mathbf{y}_{HL,d}$, $\mathbf{y}_{LH,d}$ and $\mathbf{y}_{HH,d}$ represent the low- and high-pass bands at the d^{th} level of decomposition. The symbols $\tilde{\mathbf{y}}_{LL,d}$, $\tilde{\mathbf{y}}_{HL,d}$, $\tilde{\mathbf{y}}_{LH,d}$ and $\tilde{\mathbf{y}}_{HH,d}$ denote the redundant (aliasing) information within the low- and high-pass bands at level d. The symbol $\bar{\mathbf{y}}_{LL,d}$ stands for the less redundant ("cleaned") LL band at level d after applying the operator $\mathcal{T}^{A}_{H2L}$. The dashed lines indicate that the operator $\mathcal{T}^{W}_{L2H}$ is only optional.

8.3 Hybrid Approach

Building on the high-to-low approach, we introduce a third *hybrid* architecture to further improve coding efficiency [193]. Rather than employing a linear low-to-high operator $\mathcal{T}^{W}_{L2H}$ as described in the high-to-low approach, the hybrid architecture adopts an adaptive low-to-high operator $\mathcal{T}^{A}_{L2H}$ after implementing $\mathcal{T}^{A}_{H2L}$ as seen in Fig. 8.5. Although conceptually $\mathcal{T}^{W}_{L2H}$ is sufficient to suppress redundancy within the detail bands, it is strictly true only if the first operator $\mathcal{T}^{A}_{H2L}$ pre-cleans all aliasing from the LL band. By introducing an adaptive low-to-high operator, the hybrid approach can maintain the benefits of coding efficiency even if $\mathcal{T}^{A}_{H2L}$ fails to clean aliasing from the low-pass band in the first place.

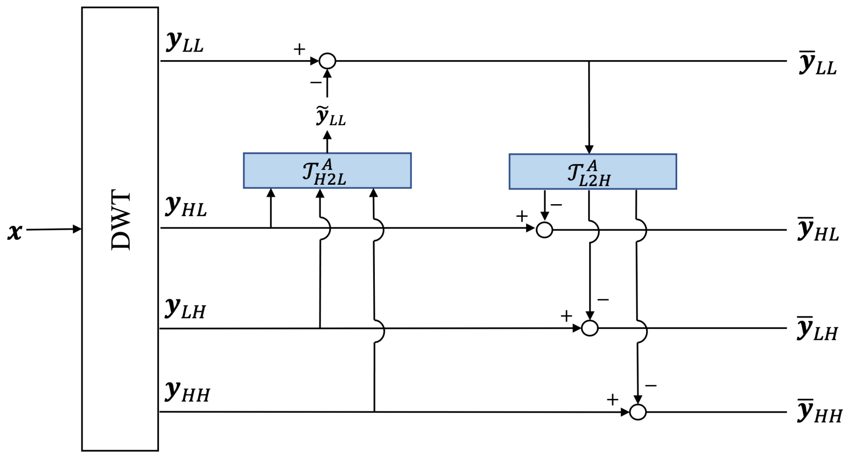


Figure 8.5: The architecture of the hybrid method, which can be viewed as additional lifting steps (predict and update steps) to the wavelet transform. The symbols $\mathbf{y}_{LL}$, $\mathbf{y}_{HL}$, $\mathbf{y}_{LH}$ and $\mathbf{y}_{HH}$ represent the LL, HL, LH and HH bands of the wavelet transform. The symbols $\bar{\mathbf{y}}_{LL}$, $\bar{\mathbf{y}}_{HL}$, $\bar{\mathbf{y}}_{LH}$ and $\bar{\mathbf{y}}_{HH}$ denote the less redundant ("cleaned") LL, HL, LH and HH bands.

8.4 Encoding Systems

In terms of the encoding system, it can be implemented in either *open-loop* or *closed-loop* fashion. The difference between the two approaches rests in how quantization errors are treated and propagated in the synthesis step. The details of each encoding approach are given below.

8.4.1 Closed-loop Encoding System

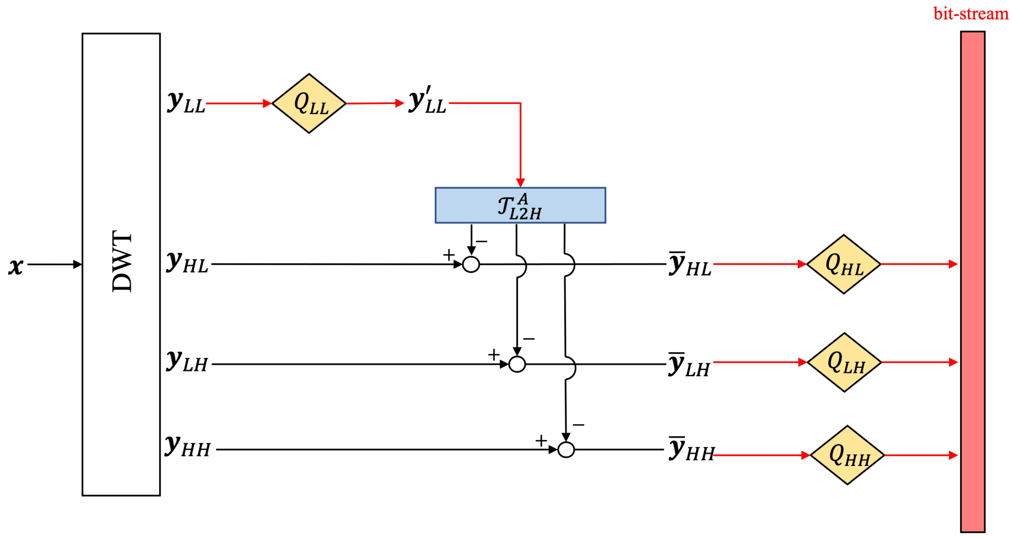
The closed-loop encoding approach is conceptually appealing in the context of non-linear operators; it avoids the propagation of quantization errors, which otherwise are expanded in an uncontrollable way through non-linearities in the networks. To achieve this, the closed-loop encoding system essentially embeds the decoder inside the encoder, so that the transform is designed at the decoder with quantized data.

In our scenario, the low-to-high and the high-to-low architectures can be developed respectively in the closed-loop encoding framework as seen in Fig. 8.6. In both cases, adding additional Wiener filters $\mathcal{T}^{W}_{L2H}$ and $\mathcal{T}^{W}_{H2L}$ is infeasible, as it creates cyclic dependencies between the adaptive operators $\mathcal{T}^{A}_{L2H}$ and $\mathcal{T}^{A}_{H2L}$; this prevents us from finding a deterministic process for determining the quantized subband samples. For this same reason, the closed-loop encoding system is incompatible with the hybrid architecture. Considering these fundamental difficulties, we choose to focus on developing the open-loop encoding system as presented in the following section.

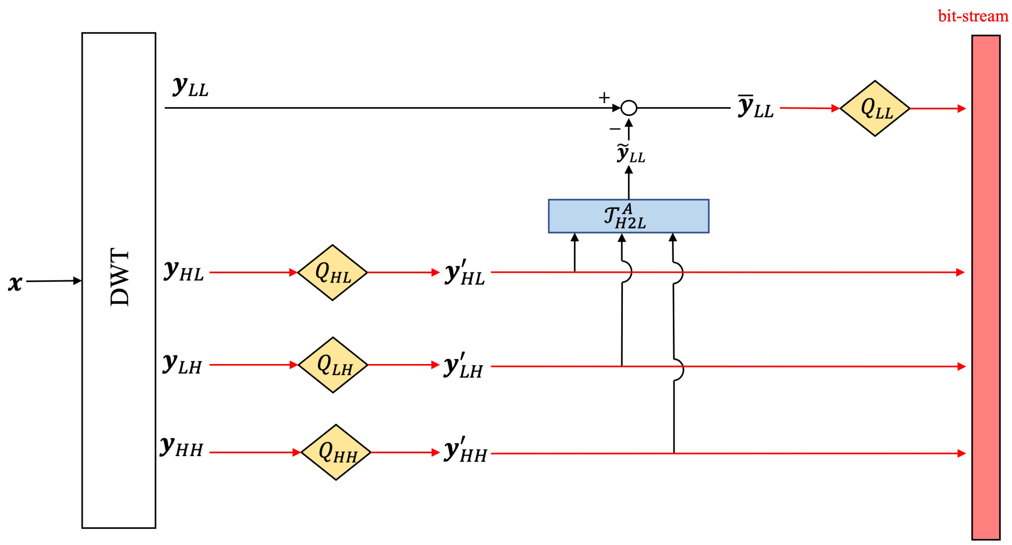
8.4.2 Open-loop Encoding System

In the so-called "open-loop" approach, the transform is designed at the encoder without any quantization, whereas the decoder receives quantized samples to invert the operation. In this scenario, the hybrid architecture is feasible, as illustrated in Fig. 8.7, which is of particular interest due to its ability to adaptively remove redundancy within both the $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$ steps.

The main challenge for open-loop encoding is that quantization errors propagate through multiple adaptive operators that necessarily entail non-linear elements, in addition to the linear wavelet synthesis operators themselves. This introduces the potential for quantization errors to be amplified, in ways that are strongly data dependent and so harder to bound. Ultimately, this will require careful modelling during the training of our neural network based operators. Nonetheless, it turns out that it is possible to develop open-loop hybrid architectures that achieve significant gains in coding efficiency across a wide range of bit-rates, in a completely scalable setting.



(a)



(b)

Figure 8.6: The proposed closed-loop encoding framework for the low-to-high and high-to-low systems, respectively. The symbols Q_{LL}, Q_{HL}, Q_{LH} and Q_{HH} represent the quantizer for the LL, HL, LH and HH bands respectively. The symbols $\mathbf{y}'_{LL}$, $\mathbf{y}'_{HL}$, $\mathbf{y}'_{LH}$ and $\mathbf{y}'_{HH}$ denote the quantized low- and high-pass bands.

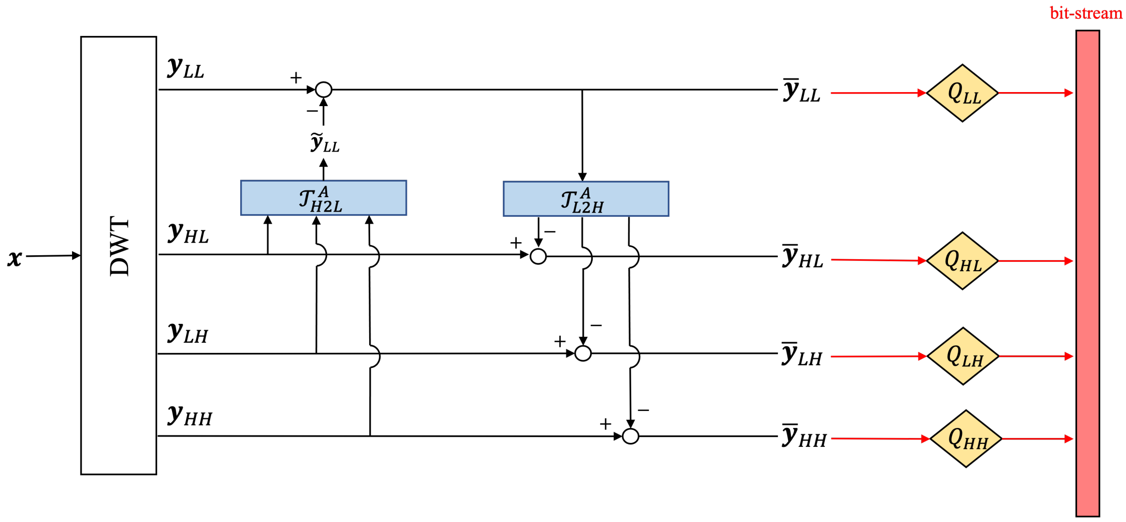


Figure 8.7: The proposed open-loop encoding system for the hybrid architecture. The symbols Q_{LL}, Q_{HL}, Q_{LH} and Q_{HH} represent the quantizer for the LL, HL, LH and HH bands respectively. The symbols $\mathbf{y}_{LL}$, $\mathbf{y}_{HL}$, $\mathbf{y}_{LH}$ and $\mathbf{y}_{HH}$ represent the LL, HL, LH and HH bands of the wavelet transform. The symbols $\overline{\mathbf{y}}_{LL}$, $\overline{\mathbf{y}}_{HL}$, $\overline{\mathbf{y}}_{LH}$ and $\overline{\mathbf{y}}_{HH}$ denote the less redundant ("cleaned") LL, HL, LH and HH bands.

Chapter 9

Contribution: Neural Network Architectures

In Chapter 7, we have mathematically elaborated the opportunity that exists to exploit the residual redundancy from the existing wavelet transform, i.e. a simple linear solution is sufficient to untangle the redundant (notably aliasing) information within regions with consistent geometric flow. The purpose of this chapter is to give insight on how this underlying hypothesis drives the structure of the neural networks that we select. Eventually, we find that the best solution does indeed involve banks of optimized linear filters controlled dynamically by an opacity network. This confirms our underlying hypothesis that the solution to our problem (redundancy exploitation) can be a linear filter if the local orientation is known a priori.

In a preliminary exploration phase, we explore the merits of different structures. This exploration phase does not involve end-to-end training for the full rate-distortion optimization problem. Instead, we measure the energy compaction potential of different structures, and we explore robustness to quantization error propagation by considering just one level of decomposition in isolation. Later, after identifying the most suitable structures, we develop a comprehensive end-to-end training strategy that is capable of modelling the complex interactions between quantization and adaptive processing steps across the decomposition hierarchy; this is the subject of Chapter 10.

9.1 Benefits of Proposal-opacity Structures

Our exploration phase starts with a focus only on the adaptive high-to-low operator $\mathcal{T}^{A}_{H2L}$. This is because $\mathcal{T}^{A}_{H2L}$ is the most critical element to avoid propagation of aliasing through the DWT hierarchy, and opens the opportunity for the transform architecture to be extended to multiple levels of the wavelet decomposition. This is surely not the only way to approach the problem, but the initial explorations involving only $\mathcal{T}^{A}_{H2L}$ turn out to be very insightful.

We begin by considering a fairly straightforward high-to-low network structure in [12]. This structure is composed from three sub-networks involving conventional convolution and *Leaky ReLU* operators, as seen in Fig. 9.1. Variations on this structure were also explored, involving concatenation of the HL, LH and HH source channels ahead of the first convolutional layer.

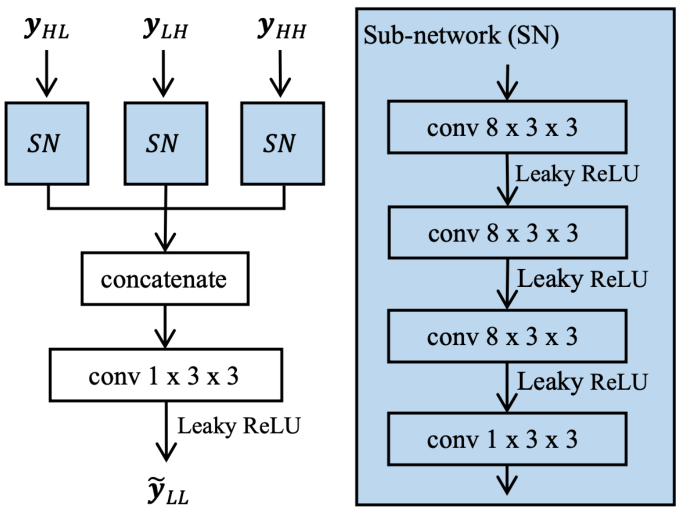


Figure 9.1: The initial high-to-low network structure proposed in [12], where N x K x K denotes N filters (or channels) with kernel support K x K.

To evaluate the potential of these high-to-low network structures without building a complete end-to-end optimization system, our primary training objective is aliasing suppression within the LL bands. This objective is chosen for two reasons: 1) removal of aliasing is necessary to ensure that the approach can be effectively applied also at lower levels in the DWT hierarchy, as elaborated in Fig. 8.4; and 2) aliasing suppression will also help to reduce redundant information from the subbands that are derived from the "cleaned" LL band. This naturally leads to higher energy compaction, which can be employed as an evaluation criterion when assessing the performance of these high-to-low networks.

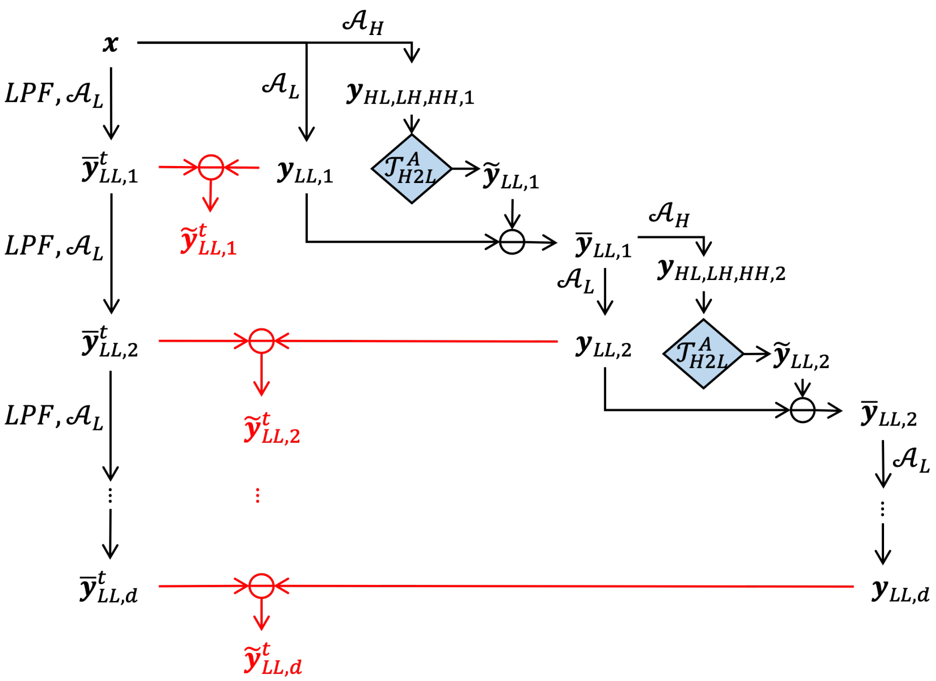


Figure 9.2: The proposed structure to construct the aliasing model $\tilde{\mathbf{y}}^t_{LL,d}$ for the LL band at each level d.

To be more specific, the idea of developing our training objective is to construct a model $\tilde{\mathbf{y}}^t_{LL,d}$ for the aliasing in the LL band at each level of decomposition d, by subtracting $\mathbf{y}_{LL,d}$ from $\bar{\mathbf{y}}^t_{LL,d}$ as seen in Fig. 9.2. The accent $^-$ is used to indicate the subband free of aliasing, while the superscript ^{t} denotes the training target. The subband $\bar{\mathbf{y}}^t_{LL,d}$ is obtained by low-pass filtering $\bar{\mathbf{y}}^t_{LL,d-1}$ and then subjecting it to the low-pass wavelet analysis operator $\mathcal{A}_L$, while $\mathbf{y}_{LL,d}$ is derived from the "cleaned" LL band $\bar{\mathbf{y}}_{LL,d-1}$ after applying the high-to-low operator $\mathcal{T}^{A}_{H2L}$. The low-pass filter (LPF) employed here has a windowed sinc impulse response with bandwidth 0.7π.

The objective can be either the l_2-norm $\left\|\widetilde{\mathbf{y}}_{LL,d} - \widetilde{\mathbf{y}}^{t}_{LL,d}\right\|_2^2$ or the l_1-norm $\left\|\widetilde{\mathbf{y}}_{LL,d} - \widetilde{\mathbf{y}}^{t}_{LL,d}\right\|_1$, where $\widetilde{\mathbf{y}}_{LL,d}$ is the aliasing predicted by the high-to-low operator (network) $\mathcal{T}^{A}_{H2L}$. The difference between these two objective metrics will not be explicitly addressed, as we have empirically verified that their impacts on the performance of different high-to-low networks are neglectable.

For the experimental results, Adam algorithm [194] with 75 image batches comprising 16 patches of size 256 x 256 from DIV2K image dataset[1] are employed, while other images in DIV2K dataset that are not included in the training are used for testing. To evaluate the performance of different high-to-low network structures, we consider two objective measurements: 1) energy compaction, that is the ratio of the energy of the original detail bands obtained through LeGall 5/3 wavelet transform to the detail bands $\mathbf{y}_{HL,d}$, $\mathbf{y}_{LH,d}$ and $\mathbf{y}_{HH,d}$ decomposed from the "cleaned" LL band $\overline{\mathbf{y}}_{LL,d-1}$; and 2) visual enhancement of the "cleaned" LL band ($\overline{\mathbf{y}}_{LL,d}$) at different resolutions d.

Table 9.1(a) provides numerical results to illustrate the averaged energy compaction of the initial high-to-low network structure shown in Fig. 9.1 across all images in the testing set. In the experiment, we employ 5 levels of the LeGall 5/3 bi-orthogonal DWT, applying the proposed neural network prediction strategy for all the levels. As we see, the energy compaction of the detail subbands at all the levels affected by the operator $\mathcal{T}^{A}_{H2L}$ can be reduced considerably; those levels not affected by $\mathcal{T}^{A}_{H2L}$ are identified by a "–" in Table 9.1. The visual enhancement of the "cleaned" LL band obtained from this simple structure can be found in Fig. 9.4(b). Variations on this network structure are not explicitly shown here, as they empirically show similar potential to the initial one in Fig. 9.1.

Although this initial network structure appears to work, the underlying theory presented in Chapter 7 suggests that it should be possible to develop a linear solution to untangle redundant (aliasing) information within regions where local geometric flow is consistent.

[1] https://data.vision.ee.ethz.ch/cvl/DIV2K/

Table 9.1: Energy Compaction

(a) Energy compaction of the initial network structure proposed in [12] as seen in Fig. 9.1.

	LL	HL	LH	HH
level 1	99.7%	–	–	–
level 2	99.9%	91.2%	88.9%	76.7%
level 3	99.9%	96.4%	93.5%	83.4%
level 4	100.5%	97.5%	94.8%	85.1%
level 5	100.8%	102.4%	96.6%	93.3%

(b) Energy compaction of the proposal-opacity network structure with linear proposals as seen in Fig. 9.3.

	LL	HL	LH	HH
level 1	99.5%	–	–	–
level 2	99.6%	88.4%	85.8%	68.2%
level 3	99.0%	94.8%	91.2%	74.7%
level 4	98.4%	90.4%	86.4%	69.8%
level 5	99.1%	86.9%	87.9%	65.1%

(c) Energy compaction of the proposal-opacity network structure with non-linear proposals as seen in Fig. 9.5.

	LL	HL	LH	HH
level 1	99.4%	–	–	–
level 2	99.3%	85.3%	83.7%	64.4%
level 3	99.0%	90.4%	87.1%	67.2%
level 4	98.7%	90.6%	85.2%	67.8%
level 5	99.5%	89.0%	88.0%	65.9%

This reasoning suggests that we would do well to decompose the high-to-low network in two aspects: a bank of learned linear filters, each capable of responding to different geometric features; and a separate feature detector network, which is necessarily non-linear.

Specifically, we explore a proposal-opacity structure, as shown in Fig. 9.3, where the non-linear opacity network ($N = 8$) is understood as analysing local scene geometry to produce opacities (or likelihoods) in the range 0 to 1 that are used to blend linearly generated proposals for the aliasing prediction term. The structure of the opacity network is inspired by [195], employing residual blocks that have been demonstrated to be useful in feature detection, while the proposals are chosen to have the same region of support as the opacity network. Since the proposals are completely linear, if our training objective is the l_2-norm $\left\|\tilde{\mathbf{y}}_{LL,d} - \tilde{\mathbf{y}}^t_{LL,d}\right\|_2^2$, the proposal system amounts to a linear least mean-squared error (LLMSE) best estimator conditioned on the opacities, so it is effectively a bank of Wiener filters.

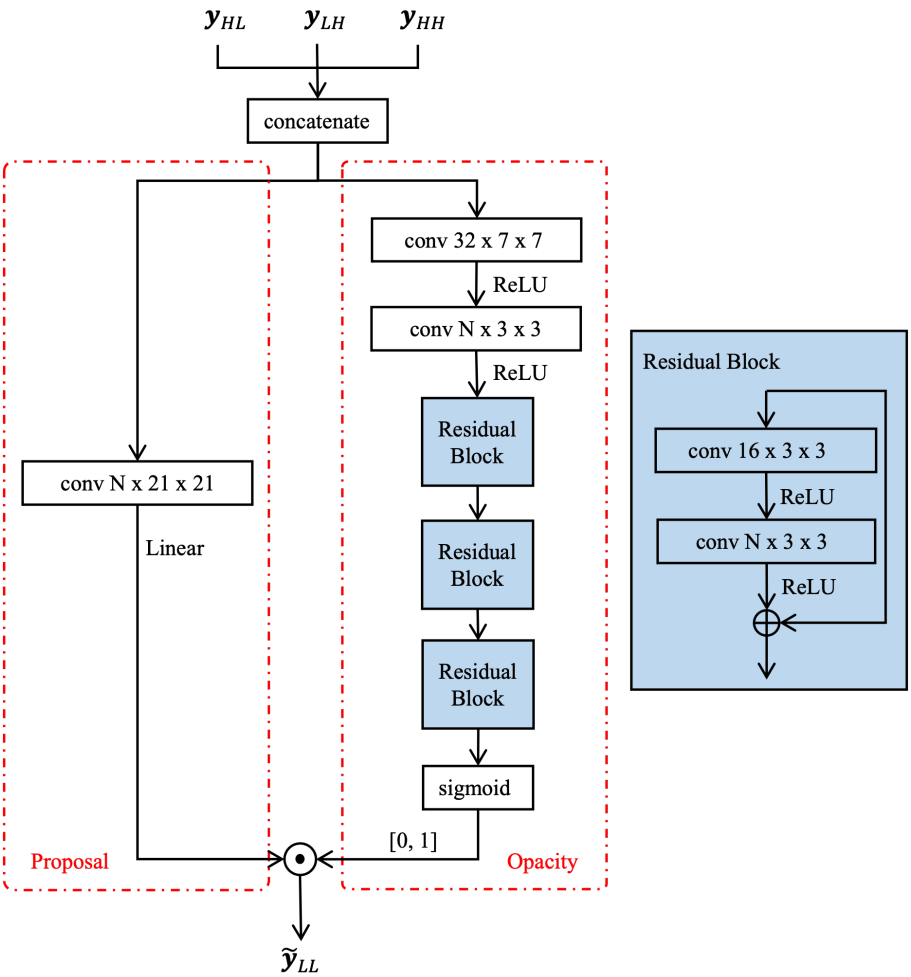


Figure 9.3: The proposed proposal-opacity structure for the high-to-low network $\mathcal{T}^A_{H2L}$ with linear proposals; N x K x K denotes N filters (or channels) with kernel support K x K.

By comparing the energy compaction in Table 9.1(a) and (b), it can be seen that the proposal-opacity network structure indeed achieves considerably higher energy compaction for all the relevant detail bands across all levels. Moreover, this proposal-opacity structure does produce more visually meaningful LL bands at different resolutions with less "staircases" around edges, compared with that of the LeGall 5/3 wavelet transform and the initial structure in Fig. 9.1; see examples in Fig. 9.4(a)(b)(c).

(a) the original LL band, with lots of aliasing ("staircases") along edges.

(b) the initial structure proposed in [12], with much less aliasing

(c) proposal-opacity structure with linear proposals and sigmoid activation

(d) proposal-opacity structure with non-linear proposal network and sigmoid activation

(e) proposal-opacity structure with linear proposals and log-like activatiom

Figure 9.4: Visual quality of the "cleaned" LL bands at the third finest resolution from different network structures. We are specifically looking for aliasing suppression – less staircase-like artifacts around edges.

9.2 Sufficiency of Linear Proposal Structures

It is worth considering whether the proposal-opacity structure can be improved by introducing non-linearities into the proposal network as well. Specifically, we choose the proposal network ($N = 8$) to be substantially similar to the opacity network, as depicted in Fig. 9.5, whereas *ReLU* and linear activation functions alternate to ensure zero-mean outputs. By comparing the prediction effectiveness in Table 9.1(b) and (c) and the visual quality of the LL bands in Fig. 9.4(c) and (d), the linear proposal structure seems to have comparable performance to the non-linear one for the high-to-low operator $\mathcal{T}^{A}_{H2L}$.

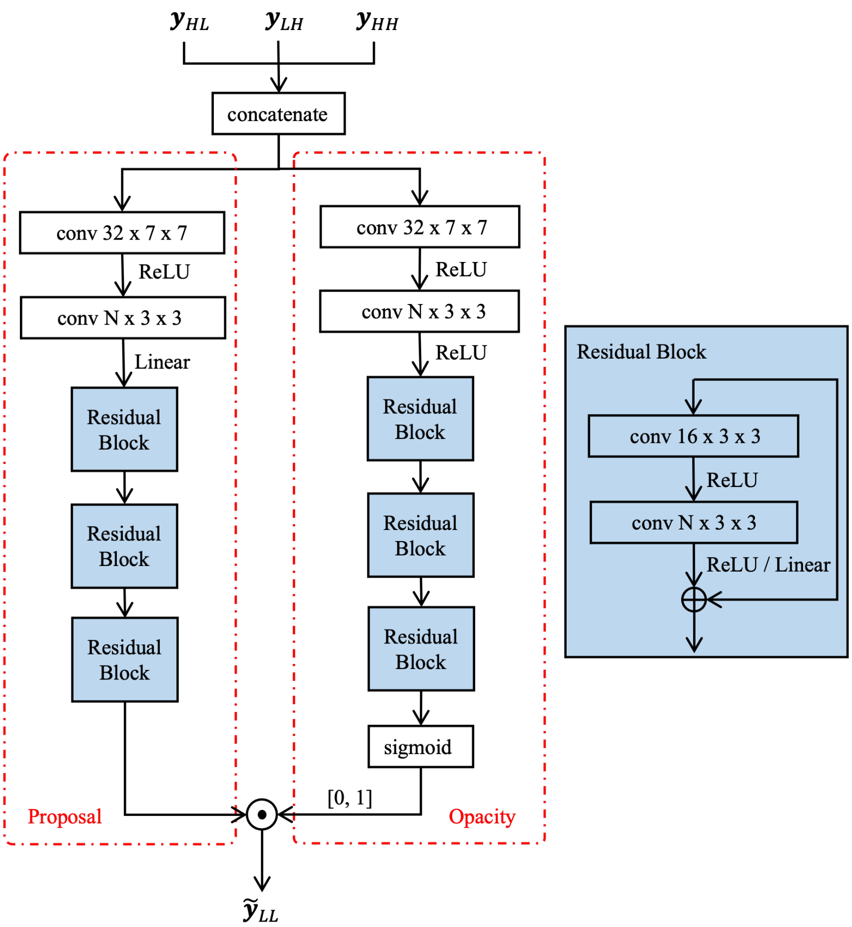


Figure 9.5: The proposed proposal-opacity structure for the high-to-low network $\mathcal{T}^{A}_{H2L}$ with non-linear proposals, where N x K x K denotes N filters (or channels) with kernel support K x K.

To gain further insight into the benefits of the linear versus the non-linear proposal structures, we construct a complete hybrid architecture by extending the proposal-opacity concept to the low-to-high network $\mathcal{T}^A_{L2H}$ with linear or non-linear proposals ($N = 8$), as seen in Fig. 9.7(a) and Fig. 9.7(b). As a result, the open-loop coding efficiency can now be explored instead of using energy compaction as a proxy, to understand the potential of different network structures.

In this open-loop setting, both $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ are trained with full-quality data, i.e. without incorporating any quantization errors during training. $\mathcal{T}^A_{H2L}$ explicitly targets the aliasing model $\tilde{\mathbf{y}}^t_{LL,d}$ during training, as described in Section 9.1. $\mathcal{T}^A_{L2H}$ is trained to minimize the prediction residuals of the detail bands at each level d; that is either l_1-norm $\|\mathbf{y}_{HL,d} - \tilde{\mathbf{y}}_{HL,d}\|_1$ or l_2-norm $\|\mathbf{y}_{HL,d} - \tilde{\mathbf{y}}_{HL,d}\|_2^2$, as exemplified in Fig. 9.6. Although the objective metric used to train $\mathcal{T}^A_{L2H}$ can be either l_1-norm or l_2-norm, we have empirically verified that l_1-norm training results in higher open-loop coding efficiency. For simplicity, $\mathcal{T}^A_{H2L}$ is trained first, after which $\mathcal{T}^A_{L2H}$ is trained while keeping $\mathcal{T}^A_{H2L}$ fixed.

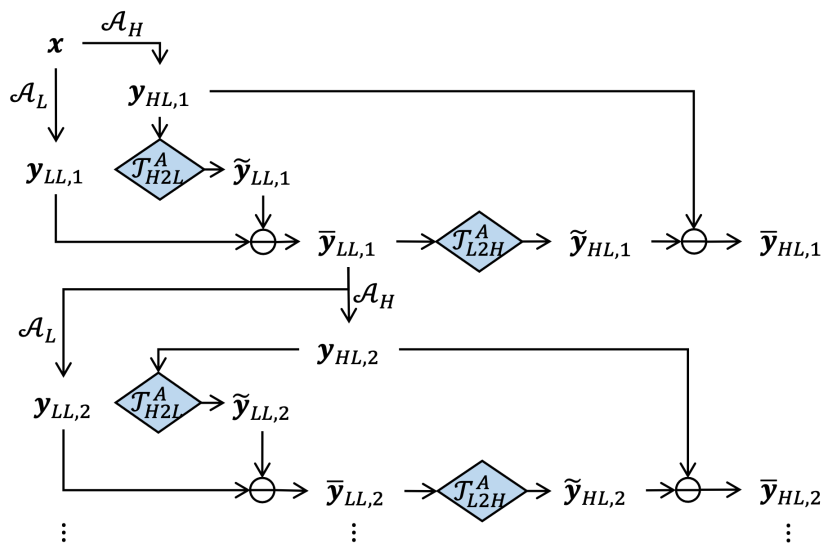
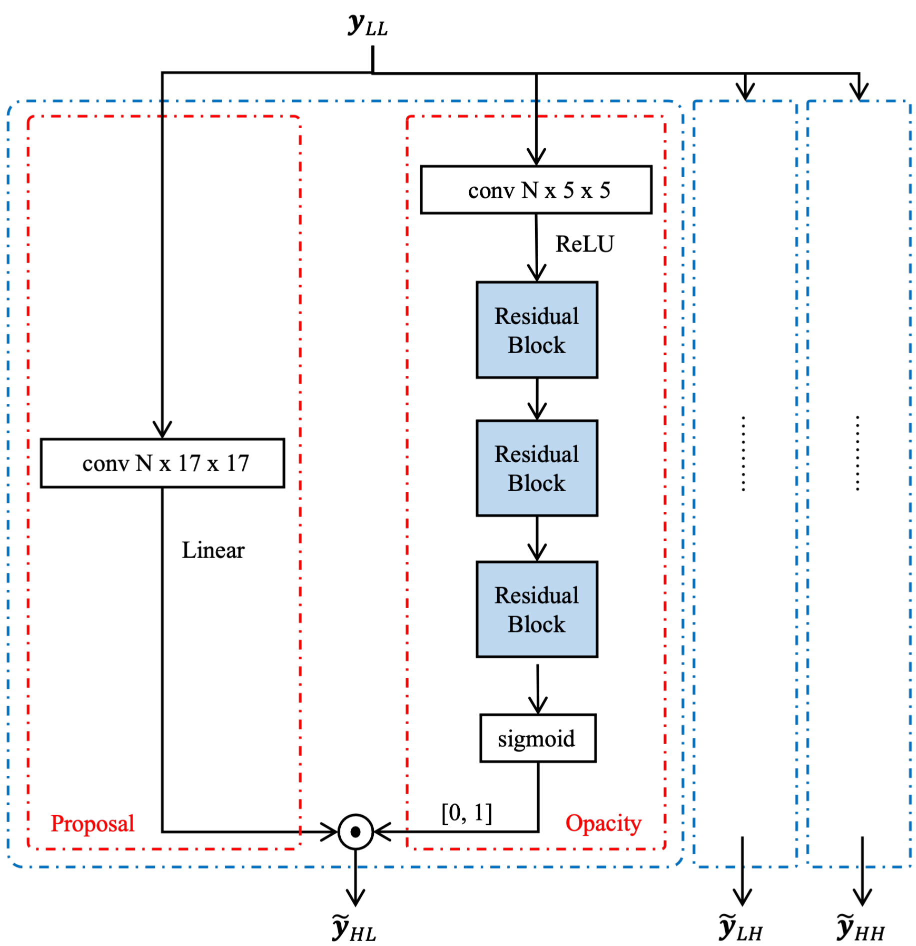


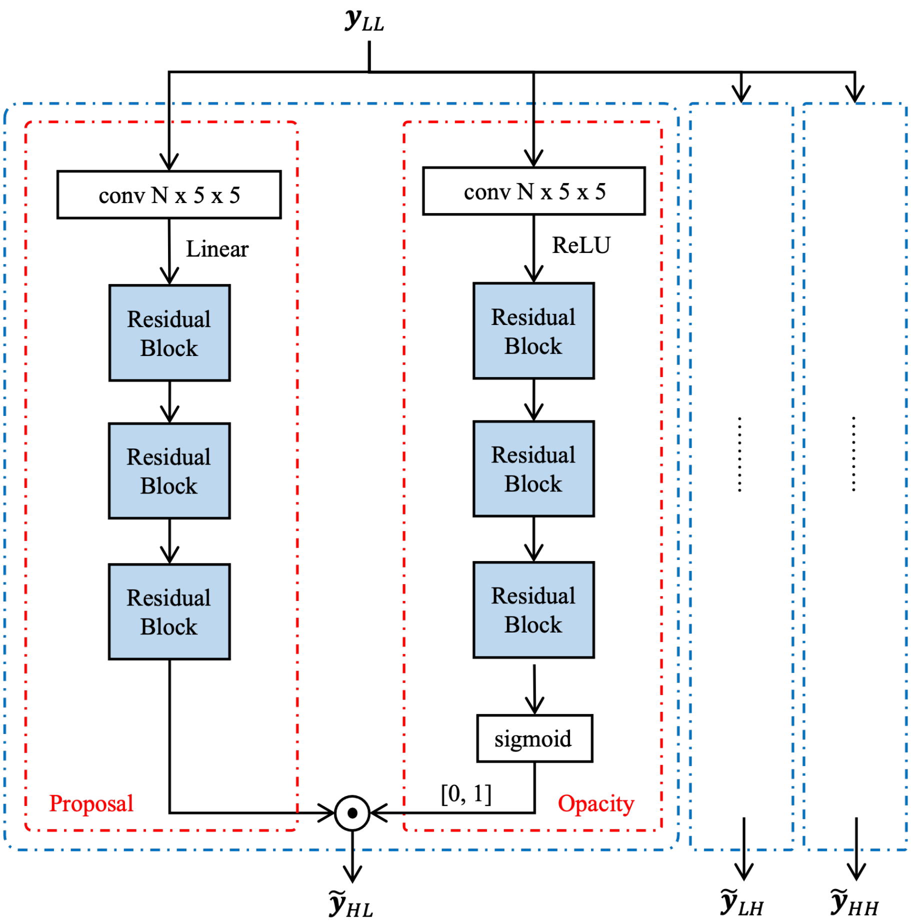
Figure 9.6: The idea to generate the training objective for the low-to-high network $\mathcal{T}^A_{L2H}$. We use HL band as an example here; the same methodology can be adopted for the LH and HH band.

From Fig. 9.9 we can see that by applying $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ to only the finest resolution in the open-loop setting, the linear proposal structure is actually better than the non-linear one in terms of rate-distortion performance. This empirically confirms that a classic set of Wiener filters attenuated by corresponding opacities (or likelihoods) is competitive with and even superior to a fully non-linear solution, which reinforces the theoretical arguments presented in Chapter 7.



(a)

Figure 9.7: The low-to-high network structure with linear proposals, where N x K x K denotes N filters (or channels) with kernel support K x K.



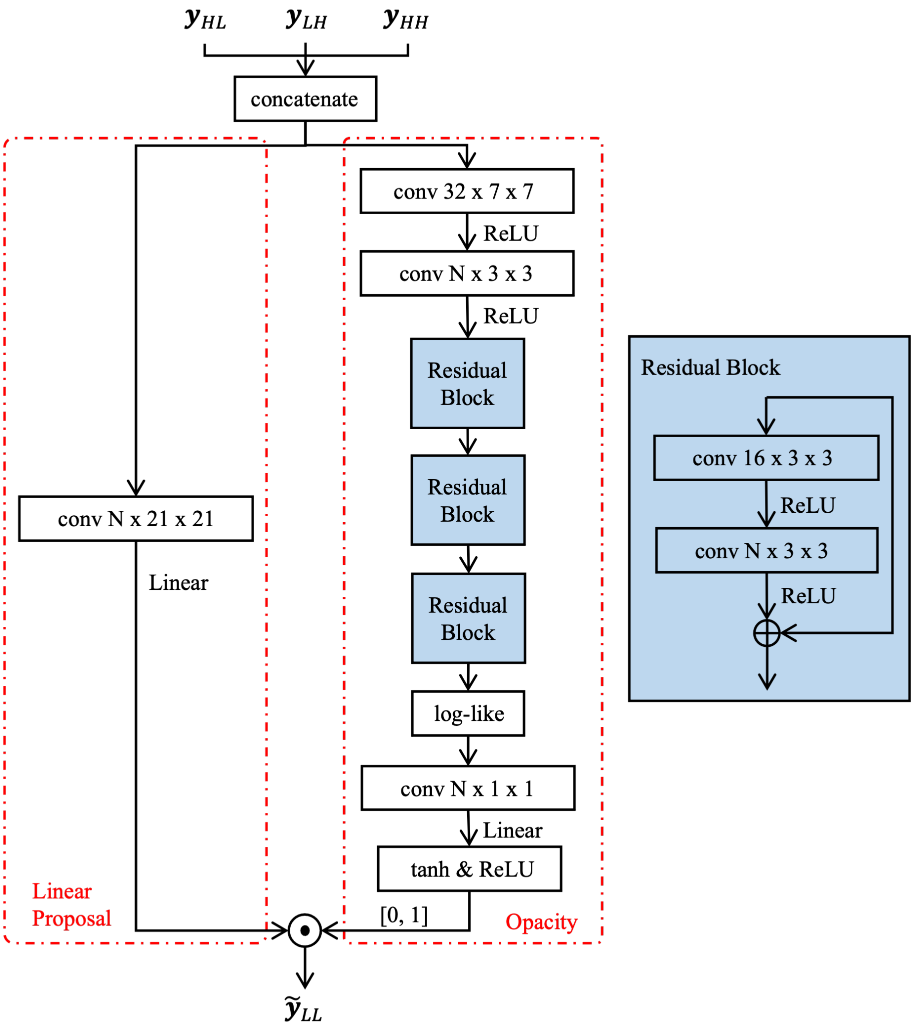
(b)

Figure 9.7: The low-to-high network structure with nonlinear proposals, where N x K x K denotes N filters (or channels) with kernel support K x K.

9.3 Appropriate Activation Functions

In this section, we consider the opacity network more carefully. Following the underlying theory elaborated in Chapter 7, we expect the opacity network to model geometric features in the scene, which should be invariant to absolute image intensity and contrast. Unfortunately, the conventional sigmoid function shown in Fig. 9.3, Fig. 9.5 and Fig. 9.7 does not have this property. We expect to do better, therefore, by replacing the sigmoid function with a log-like activation function.

In particular, we evolve the proposal-opacity structure as seen in Fig. 9.8. The log-like



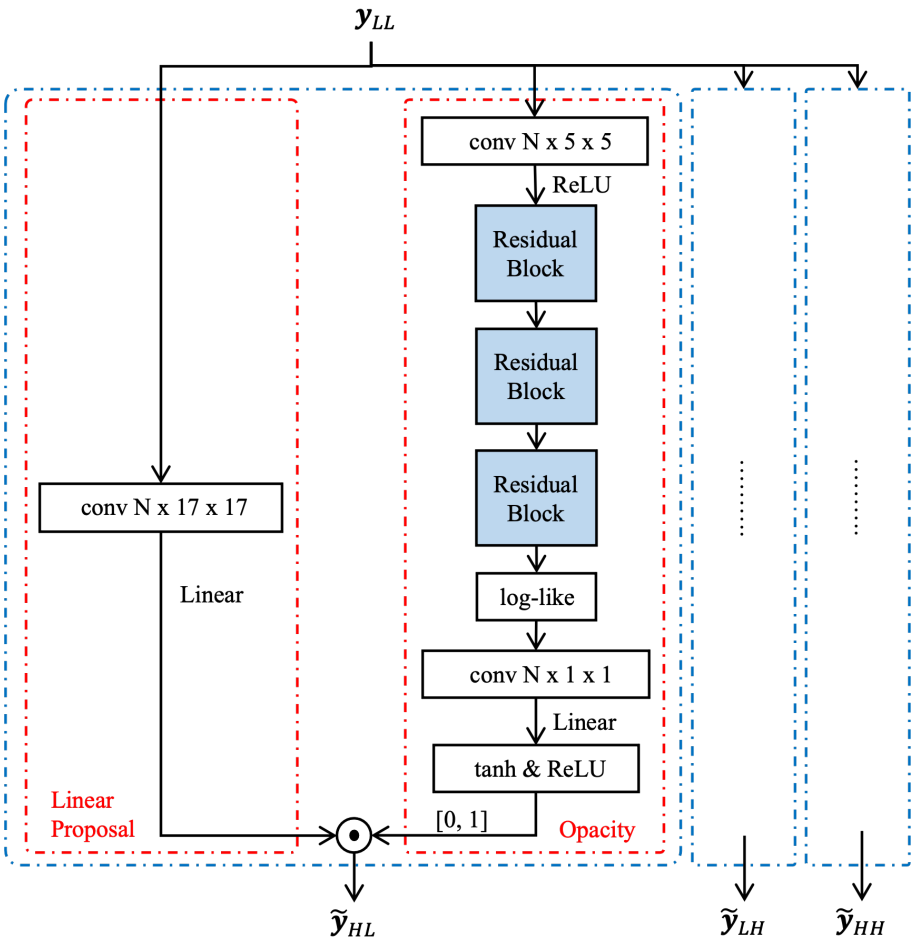
(a)

Figure 9.8: The high-to-low network structure with linear proposals and log-like activation function, where N x K x K denotes N filters (or channels) with kernel support K x K.

function that we adopt is

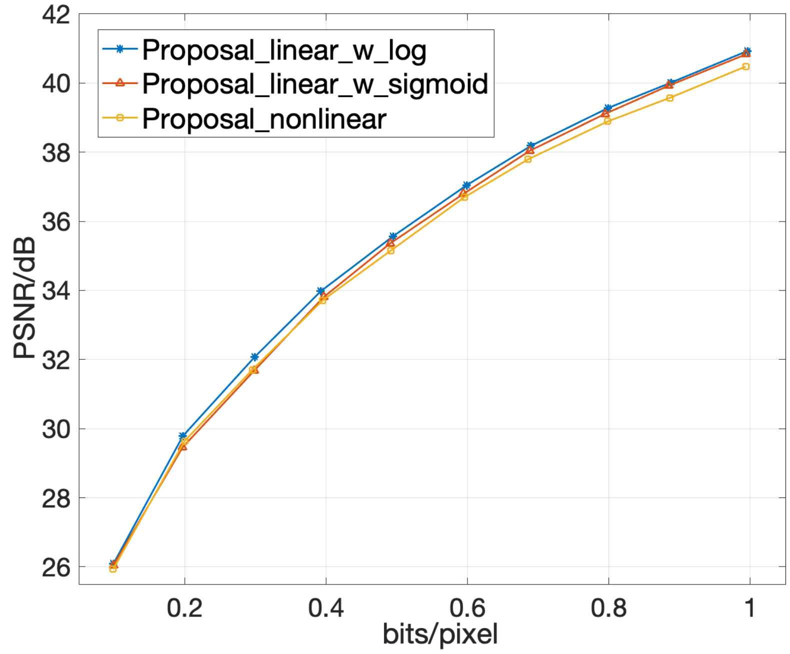
$$y = \begin{cases} \log(x + \text{offset}), & x > -\text{offset}/2 \\ \log(\text{offset}/2), & \text{otherwise} \end{cases}$$

where offset $= 0.01$ is chosen to define the derivative of the function at the origin. This log-like activation function is followed by a linear convolution layer, which is expected to choose the dominant geometric feature. In the end, *tanh* and *ReLU* are concatenated to cap the opacities within the range $[0, 1]$. Interestingly, we see the structure with the log-like activation function does perform better than that with the sigmoid function in the open-loop encoding system, even with fewer channels ($N = 4$). Meanwhile, the visual quality of the "cleaned" LL band is still maintained; see Fig. 9.4 and Fig. 9.9 for more details.

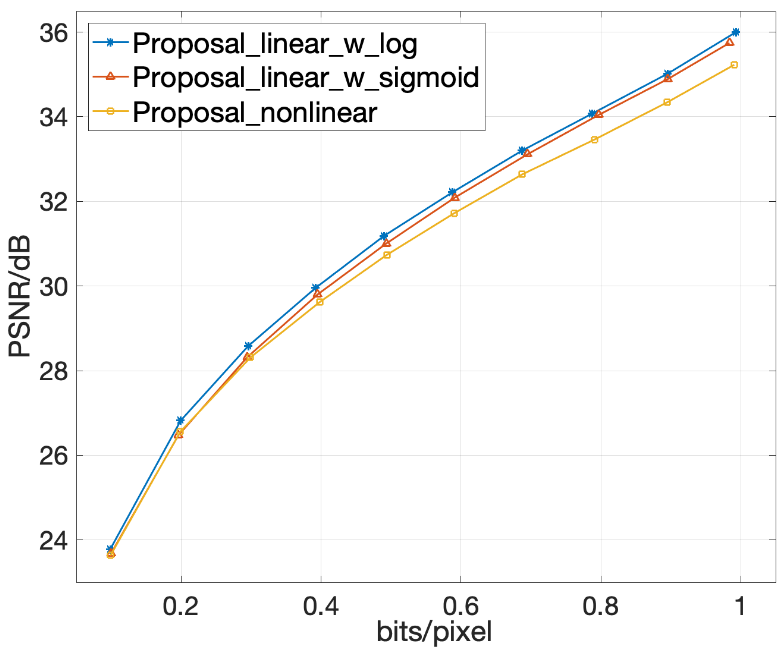


(b)

Figure 9.8: The low-to-high network structure with linear proposals and log-like activation function, where N x K x K denotes N filters (or channels) with kernel support K x K.



(a) image 846 from DIV2K



(b) image 821 from DIV2K

Figure 9.9: The rate-distortion performance under the primitive open-loop setting for different proposal-opacity network structures: linear proposals with sigmoid as the activation function as shown in Fig. 9.3 and Fig. 9.7(a); linear proposals with log-like activation function as shown in Fig. 9.8; and non-linear proposals as shown in Fig. 9.5 and Fig. 9.7(b).

Chapter 10

Contribution: End-to-end Learning Strategy with Backward Annealing

In this chapter, we introduce the learning strategy to jointly train the high-to-low and low-to-high networks for multiple levels of the DWT decomposition, along with the extra distortion gains introduced by these inference machines in addition to the base wavelet transform. The entire end-to-end optimization framework is depicted in Fig. 10.1. Interestingly, we eventually discover that a single pair of jointly trained high-to-low and low-to-high networks can be employed at all levels in the DWT decomposition hierarchy – that is, there is no need to learn and store separate network weights for each decomposition level.

In Chapter 9, aliasing suppression was our sole training objective in the initial exploration, since propagation of aliasing from high to low levels in the hierarchy would destroy the properties required for successful deployment of the approach at lower levels. As explained earlier, aliasing removal should be a reasonable proxy training objective when a single level of the hierarchy is considered in isolation. Now that we are embarking on an en-to-

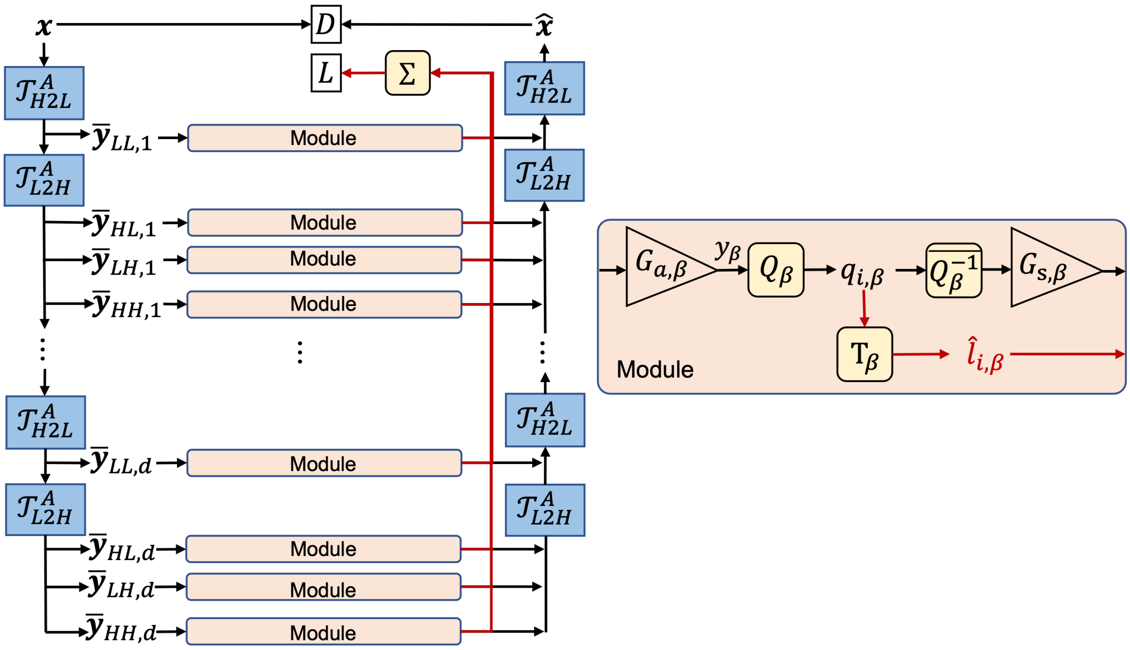


Figure 10.1: The proposed end-to-end optimization framework. $G_{a,\beta}$ and $G_{s,\beta}$ denote the extra analysis and synthesis gains introduced by the neural networks in addition to the base wavelet transform for subband B_β. By evolving $G_{a,\beta}$ and $G_{s,\beta}$ during training, we effectively optimize the quantization step size of the quantizer Q_β and the dequantizer $\overline{Q_\beta^{-1}}$ for subband B_β. Moreover, $q_{i,\beta}$ represents the quantization indices q_i within subband B_β, while T_β represents the look-up table that we use to map the quantization indices $q_{i,\beta}$ to the corresponding coded length $\hat{l}_{i,\beta}$ for subband B_β.

end learning strategy for $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$, taking all levels of the hierarchy into account together, it is possible to replace our training objective with one that focuses exclusively on rate-distortion performance. As we shall see, however, the aliasing suppression objective is quite compatible with end-to-end rate-distortion optimization. To expose this fact, we retain an aliasing suppression term as one part of the training objective, which can be selectively included to explore the role it plays in our final solution.

To be more specific, our objective is to minimize:

$$J(\phi) = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}(\phi)\|^2}_{D} + \lambda_1 \underbrace{\sum_{\beta} \sum_{i \in B_\beta} l_{i,\beta}}_{L} + \lambda_2 \underbrace{\sum_{d} \left\| \tilde{\mathbf{y}}_{LL,d}(\phi) - \tilde{\mathbf{y}}^t_{LL,d} \right\|_2^2}_{\text{aliasing constraint term}} \tag{10.1}$$

where

$$l_{i,\beta} = \log_2 \frac{1}{P_{V_\beta}(q_{i,\beta}; \boldsymbol{\phi})} = \log_2 \frac{1}{\text{Prob}(V_\beta = q_{i,\beta}; \boldsymbol{\phi})} \tag{10.2}$$

In (10.1), the total distortion term D represents the sum of squared errors between the input image $\mathbf{x}$ and its reconstructed counterpart $\hat{\mathbf{x}}$; $\boldsymbol{\phi}$ represents the vector of all network weights. The total coded length term L is the sum of all coded lengths $l_{i,\beta}$, resulting from the coding of quantization indices $q_{i,\beta}$ for all subbands B_β. We write V_β for the random variable from which the quantization indices $q_{i,\beta}$ are drawn; then, the coded length $l_{i,\beta}$ is modelled by (10.2). The LL band aliasing constraint term in (10.1) measures the sum of squared errors between $\widetilde{\mathbf{y}}_{LL,d}$ and $\widetilde{\mathbf{y}}^t_{LL,d}$ across all levels of decomposition d, as described in Section 9.1 and depicted in Fig. 9.2. The Lagrange multiplier λ_1 controls the trade-off between distortion D and coded length L, while the other Lagrange multiplier λ_2 controls the level of emphasis on visual quality of reconstructed images at different scales.

Eventually, we discover that constraining the aliasing term only at the finest resolution is sufficient for all intermediate resolutions to look good; this makes sense considering that we use the same set of network weights at all levels. In the training phase, we explore three settings of λ_2: 1) $\lambda_2 = 0$ to target rate-distortion performance alone; 2) $\lambda_2 = 1$ to encourage enhanced visual quality of LL bands within the rate-distortion optimization framework; and 3) λ_2 decreasing progressively from 1 to 0 through the training regime, so as to steer the training toward solutions that with visually appealing LL bands, while ultimately targeting rate-distortion performance alone.

To train this end-to-end optimized system for the objective in (10.1), most of the machine-learning optimization techniques, e.g. gradient descent, rely on differentiability for back-propagation. However, both the total distortion D and the total coded length L depend on the quantizer, whose derivative is either zero or infinity everywhere. For this reason, additive noise approaches, the straight-through estimator and soft-to-hard annealing methods have been developed in the literature, each of which has its own weakness and benefit as discussed in Section 6.3.

To take advantage of these existing methods, we propose a backward annealing approach, which essentially interpolates the discontinuous function using a sliding Gaussian to form a continuous relaxation of the non-differentiable step function in the backward pass, whereas the forward pass retains its original discontinuous quantization behavior, as seen in Fig. 10.2. In this way, our method avoids the train-test discrepancy, which however exists in additive noise approaches. By gradually reducing the standard deviation σ of the sliding Gaussian during training, the fitness of the continuous relaxation to the true discontinuous operator can be easily annealed. This means that we can gradually eliminate the forward-backward discrepancy while still provide the networks an accurate visibility to real quantized data early on during training; this is in contrast to the STE and soft-to-hard annealing approaches.

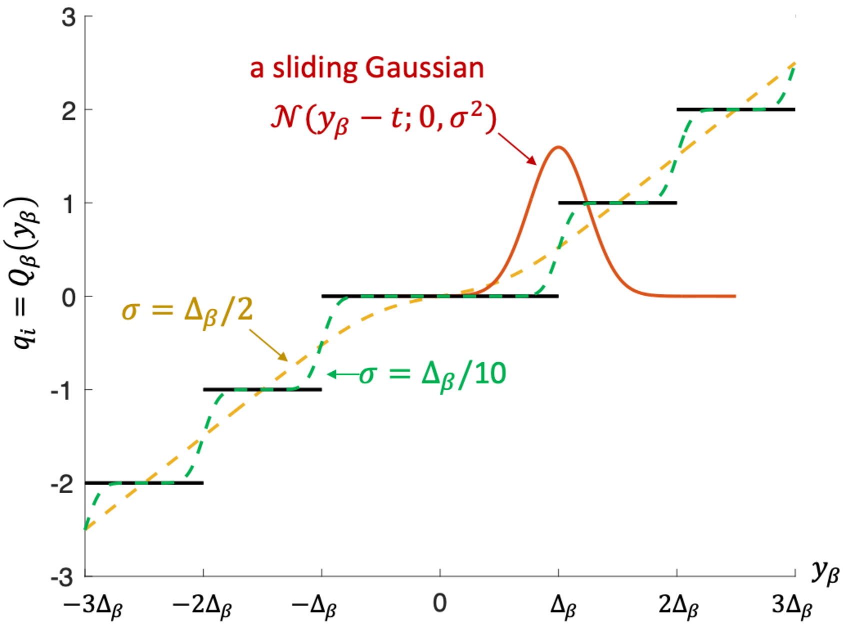


Figure 10.2: The proposed backward annealing approach for back-propagation, which interpolates the discontinuous step function Q_β (solid black lines) using a sliding Gaussian (red solid curves) to form a differentiable relaxation $\widetilde{Q}_\beta$ (green or yellow dashed curves).

Specifically, assume we wish to develop a differentiable approximation function $\widetilde{Q}_\beta$ for the quantizer Q_β of subband B_β, which is a uniform scalar quantizer with deadzone as employed in any JPEG 2000 compression framework:

$$q_{i,\beta} = Q_\beta(y_\beta) = \begin{cases} \text{sign}(y_\beta)\left\lfloor \frac{|y_\beta|}{\Delta_\beta} + \xi \right\rfloor, & \frac{|y_\beta|}{\Delta_\beta} + \xi > 0 \\ 0, & \text{otherwise} \end{cases} \tag{10.3}$$

where Δ_β denotes the quantization step size of subband B_β while ξ controls the width of the deadzone. In this thesis, ξ is set to be 0, which results in a zero-bin width of $2\Delta_\beta$.

Using Fig. 10.2 as guidance, we propose to convolve the discontinuous quantization function Q_β with a sliding Gaussian function $\mathcal{N}(t;\mu,\sigma^2)|_{\mu=0}$. This convolution relaxes the non-differentiable quantization function Q_β, producing the continuous relaxation counterpart $\widetilde{Q}_\beta$, which is more suitable for back-propagation; more concretely, we have

$$\begin{aligned} \widetilde{Q}_\beta(y_\beta) &= \int_{-\infty}^{+\infty} Q_\beta(t)\mathcal{N}(y_\beta - t; 0, \sigma^2)dt \\ &= \sum_{q_i=-M}^{-1} q_i \int_{(q_i-1)\Delta}^{q_i\Delta} \mathcal{N}(y_\beta - t; 0, \sigma^2)dt \\ &+ \sum_{q_i=1}^{M} q_i \int_{q_i\Delta}^{(q_i+1)\Delta} \mathcal{N}(y_\beta - t; 0, \sigma^2)dt \\ &+ \sum_{q_i=0} q_i \int_{(q_i-1)\Delta}^{(q_i+1)\Delta} \mathcal{N}(y_\beta - t; 0, \sigma^2)dt \end{aligned} \tag{10.4}$$

In practice, we limit the integration to the interval $\pm 3\sigma$, since a normal distribution decays to approximately zero at the endpoints of this interval. With a relatively big σ, our approach draws a straight line through the quantization step function for the purpose of back-propagation, as exemplified in yellow dashed curve in Fig. 10.2; this is essentially the concept of the STE. By decreasing the value of σ along with the learning rate during training, the method ensures a smooth transition from the STE to soft-to-hard annealing with a controllable "cooling" coefficient σ; an example is given as the green dashed curve seen in Fig. 10.2.

Now we move on to the calculation of the coded length $l_{i,\beta}$ in (10.1). Conceptually, if we knew the statistical distribution P_{V_β}, then $l_{i,\beta}$ can be calculated directly using (10.2). The challenge is that P_{V_β} is data-dependent, and depends weakly on the choice of the weights in the high-to-low and the low-to high networks. This weak dependency allows us, in practice, to estimate and update each P_{V_β} periodically using a histogram. This histogram, containing the number of occurrence and so the probability of each $q_{i,\beta}$, can then be converted to the coded length $\hat{l}_{i,\beta}$ as an estimate of $l_{i,\beta}$. All the coded lengths $\hat{l}_{i,\beta}$ for all $i \in B_\beta$ eventually form a look-up table T_β for each subband B_β, as depicted in Fig. 10.1. In terms of the back-propagation, each look-up table T_β together with its respective quantizer Q_β can be treated like one discontinuous operator, which maps each input y_β to its coded length $\hat{l}_{i,\beta}$; this discontinuous operator can be treated exactly the same way as described in Fig. 10.2.

It is important to highlight the fact that this periodic update of the histograms, and so the look-up tables $\{T_\beta\}_\beta$, does not introduce instability into the training. Essentially, the training process with periodic update alternates between two steps, each of which reduces the following modified version of the cost function in (10.1):

$$J'(\boldsymbol{\phi}, \{T_\beta\}_\beta) = \|x - \hat{x}(\boldsymbol{\phi})\|^2 + \lambda_1 \sum_\beta \sum_{i \in B_\beta} \hat{l}_{i,\beta} + \lambda_2 \sum_d \left\|\tilde{\mathbf{y}}_{LL,d}(\boldsymbol{\phi}) - \tilde{\mathbf{y}}^t_{LL,d}\right\|_2^2 \tag{10.5}$$

where

$$\hat{l}_{i,\beta} = T_\beta(q_{i,\beta}) \tag{10.6}$$

in which $\boldsymbol{\phi}$ represents the vector of all network weights; we remind the reader that we use the same set of weights for all levels of decomposition.

This modified cost function involves jointly optimizing the look-up tables $\{T_\beta\}_\beta$ and the weights $\boldsymbol{\phi}$, which proceeds in alternating steps. At any given point during training, we have *Step 1* that optimizes the network weights $\boldsymbol{\phi}$, assuming fixed look-up tables $\{T_\beta\}_\beta$. In *Step 2*, we adjust the look-up tables $\{T_\beta\}_\beta$ using histograms, given the set of network weights $\boldsymbol{\phi}$. Each step progressively reduces the same finite bounded objective in (10.5), therefore the entire system must converge. Although the global objective in (10.5) that we minimize is not exactly the same as (10.1), the difference gradually reduces as the periodic update progresses.

Chapter 11

Experimental Results: Part I

To explore the merits of our method, we develop a sequence of experiments to test the significance of the aliasing suppression term in (10.1). In addition, we explore our method with different base wavelet transforms – the LeGall 5/3 and the CDF 9/7 bi-orthogonal wavelet transforms. To put these results in context, we also compare them with some existing works. Note that the source code of our method, along with all training and testing datasets, are available on GitHub[1].

11.1 Experimental Settings

11.1.1 Training Phase

We employ 5 levels of the DWT decomposition during training, and aim to jointly train only a single pair of the high-to-low and low-to-high networks, which can be progressively applied to all levels of decomposition, as well as a wide range of compression ratio. This goal is explicitly chosen, because it is more sensible for practical applications to employ a method which only uses one set of weights for all levels. This is especially important for

[1] https://github.com/xinyue-li3/hybrid-lifting-structure/

scalable codecs, where the number of levels received at the decoder may not be the same as the encoder.

Now we begin by discussing the initialization of our training process. As explained in Chapter 10, there are two alternating update steps during training to minimize the modified objective (10.5). To start with *Step 1*, we first find the initial look-up tables $\{T_\beta\}_\beta$ using the weights of the high-to-low and the low-to-high networks $\boldsymbol{\phi}$ as trained in the exploration phase in Chapter 9. Then the network weights $\boldsymbol{\phi}$ can be optimized in *Step 2* given the initial lookup tables $\{T_\beta\}_\beta$, employing the training strategy with backward annealing in Chapter 10. *Step 1* and *Step 2* alternate periodically (in this thesis every 200 epochs), so that (10.5) gradually converges to (10.1).

Our initial choices of parameters are based upon the base wavelet transform that we are attempting to improve during the learning process. This base wavelet transform has extra distortion gains $G_{a,\beta} = G_{s,\beta} = 1.0$ as depicted in Fig. 10.1; therefore, we start from this point. In addition, this base wavelet transform also involves different quantization step sizes Δ_β for each subband B_β as seen in (10.3); we initialize these $\{\Delta_\beta\}_\beta$ in a way which typically results in a compression bit-rate around 1.0 bpp for training images. This bit-rate 1.0 bpp also corresponds to a particular rate-distortion slope λ_1 as seen in (10.1); therefore, this becomes the starting point of our λ_1 during training. Subsequently, we allow $G_{a,\beta}$ and $G_{s,\beta}$ to evolve during training while keeping Δ_β and λ_1 fixed. If the change in $G_{a,\beta}$ and $G_{s,\beta}$ are not too substantial, then we expect the compression bit-rate to still wind up in the vicinity of 1.0 bpp at the end of training.

In this thesis, Keras with TensorFlow backend and the Adam algorithm [194] are employed for training, with 75 image batches comprising 16 patches of size 256 x 256 from the DIV2K image dataset[2]. In total 1200 epochs are used for training in this paper, while periodic update of the look-up tables $\{T_\beta\}_\beta$ occurs every 200 epochs, as mentioned before. Within each 200 epochs, we progressively reduce the controllable "cooling" coefficient σ for back-propagation as explained in Chapter 10. This σ is empirically initialized as $\frac{\Delta}{2}$ and decays

[2] https://data.vision.ee.ethz.ch/cvl/DIV2K/

steadily until the change in the solution is negligible; in our case, this end point yields $\sigma \approx \frac{\Delta}{10}$. The learning rate is empirically set to 0.0001 and decays exponentially with $decay\_steps = 20$ and $decay\_rate = 0.96$. This might not be the optimal training schedule for σ, but it turns out that other more natural training schedules are hard to realize within the TensorFlow backend.

11.1.2 Testing Phase

We choose four datasets categorized into three classes during testing, in order to demonstrate the merits of our method in different scenarios. Note that none of these images are used during training.

Category 1: All images within this class have highly structured features, i.e. edges are either consistently oriented or significantly distinct from background textures. Two datasets are included in this class: a) Tecnick Sampling Dataset[3], from which 20 images are chosen with size 480 x 480; b) DIV2K Dataset[4], from which 30 images are chosen with size 1024 x 2048. We name these two dataset as *Tecnick-Cat1* and *DIV2K-Cat1*, respectively.

Category 2: All images in this category come with reasonably clear edges, while background textures are more complicated than those in Category 1. The dataset employed in this class is DIV2K dataset, from which another 70 images of size 1024 x 2048 are chosen; it is denoted by *DIV2K-Cat2*.

Category 3: All images in this category are considered to be “hard-to-code”, with one or more following properties: nearly no clear orientations; majority of the image is excessively blurred; and/or most orientations are horizontal or vertical, which are well handled by the wavelet transform. The dataset employed is Challenges on Learned Image Compression

[3] https://testimages.org/

[4] https://data.vision.ee.ethz.ch/cvl/DIV2K/

2019 test set[5], from which 15 images of size 1024 x 2048 are chosen; this is denoted as *CLIC2019-Cat3*.

Moreover, the Kodak dataset[6], which is commonly used as the benchmark for image compression, is also tested here to serve two purposes: 1) to demonstrate the effectiveness of our method in an entire dataset, which is not explicitly chosen nor altered; 2) to put our results in context with other existing works, even if the source codes are unavailable or hard to reproduce the inferences given their codes.

It is worthwhile pointing out that all images within all datasets are converted to grayscale before any training or testing. The reason for this is to avoid confusing our spatial transforms with color dependent questions, such as the optimal choice of color transform and the dependence of the wavelet transform on different color components.

11.2 Methods Explored

We first explore the following variations of our method:

- the effectiveness of replacing the adaptive operator $\mathcal{T}^{A}_{L2H}$ with the linear $\mathcal{T}^{W}_{L2H}$ in the hybrid architecture; As suggested in Chapter 8, $\mathcal{T}^{W}_{L2H}$ might be sufficient.
- three variations of the aliasing constraint parameter in (10.1): $\lambda_2 = 0$, $\lambda_2 = 1$ and λ_2 decreasing gradually from 1 to 0 throughout training; although λ_2 is not directly coupled with coding efficiency, it is driven by visual considerations as explained in Chapter 10.
- two different base wavelet transforms: the LeGall 5/3 [196] and the CDF 9/7 bi-orthogonal wavelet transforms [197], as they come with different levels of complexity and spatial supports.

[5] http://clic.compression.cc/2019/challenge/

[6] http://www.cs.albany.edu/ xypan/research/snr/Kodak.html

To put these results in context, they are also compared with some other existing works from the following categories: i) non-learning based compression standards; ii) learning-based, wavelet-like lossy image compression frameworks; and iii) variants of end-to-end optimized, non-wavelet-like lossy image compression with neural networks.

11.3 Evaluation Metrics

We consider evaluating the performance of all methods presented within this section both quantitatively and qualitatively. In terms of quantitative measurements, three widely used metrics are employed – Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), Multi-Scale Structural Similarity (MS-SSIM). All these metrics are measured and averaged for each dataset, from which Bjøntegaard (BD) rate savings (in %) are obtained.

With regard to qualitative assessment, we provide examples for both the "cleaned" LL bands at different scales and the full reconstructed images in this thesis. We remind the readers that the quality of the "cleaned" LL bands is dependent on λ_2 in (10.1), which controls the amount of aliasing as explained in Chapter 10.

11.4 Results and Discussions

11.4.1 Significance of the adaptive operator

We first empirically study the value of employing an adaptive low-to-high operator $\mathcal{T}^{A}_{L2H}$ rather than the linear operator $\mathcal{T}^{W}_{L2H}$ in the hybrid architecture. As explained in Chapter 8, $\mathcal{T}^{W}_{L2H}$ is conceptually sufficient to suppress redundancy within the detail bands, only if $\mathcal{T}^{A}_{H2L}$ is completely successful in cleaning all aliasing from the LL band; however, we do not expect it to be sufficient in practice.

To study this, the adaptive high-to-low and low-to-high networks ($\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$) are set as seen in Section 9.3, while the operator $\mathcal{T}^{W}_{L2H}$ is simply a linear filter that has the same region of support as $\mathcal{T}^{A}_{L2H}$. $\mathcal{T}^{A}_{H2L}$ is jointly trained with either $\mathcal{T}^{A}_{L2H}$ or $\mathcal{T}^{W}_{L2H}$ to improve the LeGall 5/3 wavelet transform, targeting the standard rate-distortion objective for MSE by setting $\lambda_2 = 0$ in (10.1). Further studies on different λ_2 and base wavelet transforms are provided shortly.

Table 11.1: Comparison between the adaptive operator $\mathcal{T}^{A}_{L2H}$ and the linear operator $\mathcal{T}^{W}_{L2H}$. The table shows BD-rate improvements for PSNR, SSIM and MS-SSIM metrics over the LeGall 5/3 wavelet transform for each dataset. Results are obtained with bit-rates between 0.1bpp to 1.0bpp.

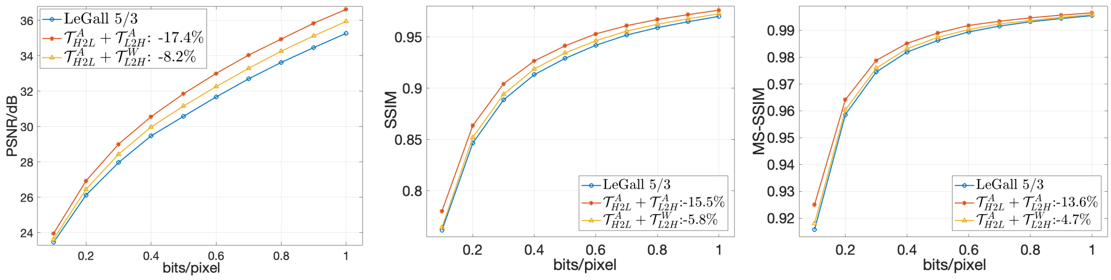
		BD-rate for PSNR		BD-rate for SSIM		BD-rate for MS-SSIM	
		$\mathcal{T}^{A}_{H2L}+\mathcal{T}^{A}_{L2H}$	$\mathcal{T}^{A}_{H2L}+\mathcal{T}^{W}_{L2H}$	$\mathcal{T}^{A}_{H2L}+\mathcal{T}^{A}_{L2H}$	$\mathcal{T}^{A}_{H2L}+\mathcal{T}^{W}_{L2H}$	$\mathcal{T}^{A}_{H2L}+\mathcal{T}^{A}_{L2H}$	$\mathcal{T}^{A}_{H2L}+\mathcal{T}^{W}_{L2H}$
LeGall 5/3	Tecknick-Cat1	−17.4%	−8.2%	−15.5%	−5.8%	−13.6%	−4.7%
	DIV2K-Cat1	−14.4%	−7.3%	−13.8%	−5.0%	−13.1%	−4.5%
	DIV2K-Cat2	−12.5%	−6.0%	−12.8%	−5.6%	−12.8%	−5.9%
	CLIC2019-Cat3	−7.3%	−3.4%	−7.5%	−2.7%	−8.9%	−4.0%

The BD rate savings (in %) and the rate-distortion curves for average PSNR, SSIM and MS-SSIM over the range of bit-rates from 0.1bpp to 1.0bpp across all four datasets are provided in Table. 11.1; the complete rate-distortion curves can be found in Fig 11.1. It can be observed consistently that both $\mathcal{T}^{A}_{L2H}$ and $\mathcal{T}^{W}_{L2H}$ are capable of improving coding efficiency of the conventional LeGall 5/3 wavelet transform, regardless the amount of distinct edges presented in images, so long as the adaptive high-to-low operator $\mathcal{T}^{A}_{H2L}$ is employed. More importantly, $\mathcal{T}^{A}_{L2H}$ performs significantly better than $\mathcal{T}^{W}_{L2H}$ across all datasets, achieving up to 17.4% average BD rate saving over the LeGall 5/3 wavelet transform, while $\mathcal{T}^{W}_{L2H}$ only reaches up to 8.2% average BD bit-rate saving.

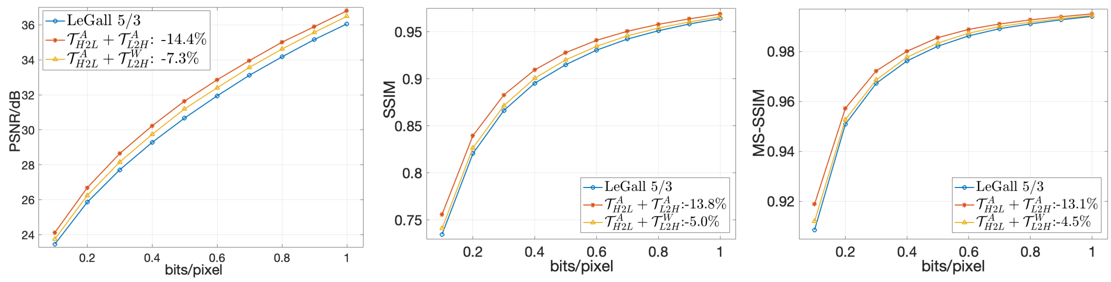
This observation aligns with the underlying theory presented in Chapter 8; by introducing an adaptive low-to-high operator, the hybrid approach can maintain the benefits of coding efficiency even if $\mathcal{T}^{A}_{H2L}$ is unable to fully clean aliasing from the low-pass band.

11.4.2 Role of the aliasing constraint term

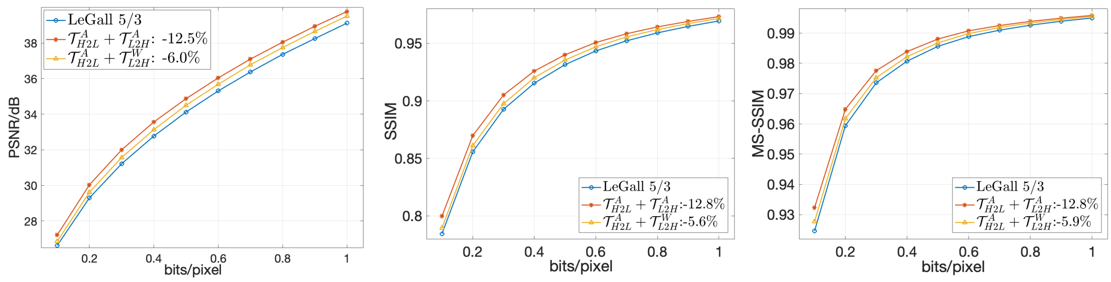
We now examine the role that the aliasing constraint term, $\lambda_2 \sum_d \left\| \tilde{\mathbf{y}}_{LL,d}(\boldsymbol{\phi}) - \tilde{\mathbf{y}}^{t}_{LL,d} \right\|_2^2$, plays in our training objective function seen in (10.1). Specifically, we explore three



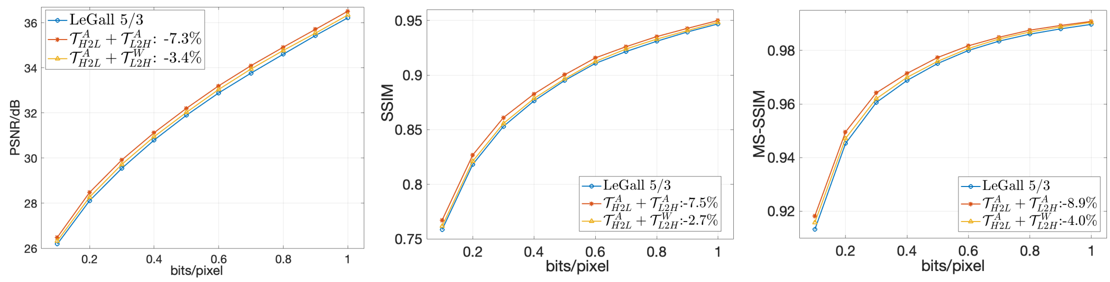
(a) Tecnick-Cat1



(b) DIV2K-Cat1



(c) DIV2K-Cat2



(d) CLIC2019-Cat3

Figure 11.1: Comparisons of the average PSNR, SSIM and MS-SSIM improvement across each dataset to illustrate the importance of the adaptive low-to-high operator $\mathcal{T}^{A}_{L2H}$ over the simple linear operator $\mathcal{T}^{W}_{L2H}$; the proposed method is trained to improve the LeGall 5/3 wavelet transform with $\lambda_2 = 0$. The Bjøntegaard (BD) rate savings are displayed in % next to the legend.

settings of the aliasing constraint parameter: $\lambda_2 = 0$, $\lambda_2 = 1$ and λ_2 decreasing from 1 to 0 during training; details have been given under (10.1) in Chapter 10. The operators employed here are the adaptive networks $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ as depicted in Section 9.3, whose benefit has been verified. The two operators are jointly trained to improve the LeGall 5/3 wavelet transform at this stage; extension to larger wavelet transforms will be given shortly.

Since λ_2 explicitly conditions the visual quality of the "cleaned" LL band at each level of decomposition, we now add this additional qualitative assessment into consideration when evaluating the performance of all methods in this section. Using the examples in Fig. 11.2 as guidance, we can observe that forcing aliasing suppression, i.e. $\lambda_2 = 1$ during training, does indeed ensure higher visual quality of the "cleaned" LL bands across multiple levels of decomposition. Moreover, we observe that employing $\lambda_2 = 1$ or annealing λ_2 from 1 to 0 produces substantially similar results.

In addition, we provide the BD bit-rate saving (in %) under average PSNR, SSIM and MS-SSIM over the range of bit-rates from 0.1bpp to 1.0bpp for different λ_2 settings across all datasets in Table. 11.2; the complete rate-distortion curves can be found in Fig 11.3. We can first see that, although the two networks $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ are jointly optimized to minimize MSE during training, they also work surprisingly well under the SSIM and the MS-SSIM metrics for all datasets. Not surprisingly, we observe the highest BD bit-rate saving when $\lambda_1 = 0$. However, the loss in coding efficiency associated with $\lambda_2 = 1$ and annealing λ_2 is not significant, in exchange for clear benefits obtained in visual quality at reduced resolutions.

It is also worthwhile to point out that the performance of our method does vary for different types of images. For Tecnick-Cat1 and DIV2K-Cat1 datasets that have consistent orientations or distinct edges, our method has the highest performance, achieving 17.4% average BD bit-rate saving over the LeGall 5/3 wavelet transform over the range of bit-rates from 0.1bpp to 1.0bpp. For DIV2K-Cat2 dataset with richer textures, the proposed method also manages to reach 12.8% average BD bit-rate saving over the range of bit-rate from 0.1bpp to 1.0bpp. Surprisingly, for images in CLIC2019-Cat3 dataset, which come

Table 11.2: Comparison between different aliasing constraint parameters λ_2 in (10.1) during training. The table shows BD-rate improvements for PSNR, SSIM and MS-SSIM metrics over the LeGall 5/3 and the CDF 9/7 wavelet transform. Results are obtained with bit-rates between 0.1bpp to 1.0bpp.

		BD-rate for PSNR			BD-rate for SSIM			BD-rate for MS-SSIM		
		$\lambda_2=0$	$\lambda_2=1$	anneal λ_2	$\lambda_2=0$	$\lambda_2=1$	anneal λ_2	$\lambda_2=0$	$\lambda_2=1$	anneal λ_2
LeGall 5/3	Tecknick-Cat1	−17.4%	−13.8%	−13.6%	−15.5%	−12.5%	−12.3%	−13.6%	−10.3%	−10.1%
	DIV2K-Cat1	−14.4%	−10.6%	−10.6%	−13.8%	−10.8%	−10.9%	−13.1%	−8.9%	−9.0%
	DIV2K-Cat2	−12.5%	−9.8%	−9.8%	−12.8%	−10.6%	−10.5%	−12.8%	−9.8%	−9.8%
	CLIC2019-Cat3	−7.3%	−5.8%	−5.7%	−7.5%	−5.8%	−5.9%	−8.9%	−5.7%	−5.8%
CDF 9/7	Tecknick-Cat1	−11.4%	−9.8%	−9.7%	−11.5%	−10.6%	−10.5%	−9.0%	−7.9%	−7.9%
	DIV2K-Cat1	−9.7%	−6.3%	−6.3%	−10.1%	−8.4%	−8.3%	−7.6%	−5.3%	−5.3%
	DIV2K-Cat2	−7.6%	−6.0%	−6.0%	−7.5%	−6.7%	−6.7%	−5.9%	−4.9%	−5.0%
	CLIC2019-Cat3	−4.2%	−3.4%	−3.3%	−4.1%	−3.5%	−3.4%	−1.7%	−0.8%	−0.7%

with hardly any clear orientations and edges, our method is still capable of achieving 8.94% average BD bit-rate saving. These observations again align with our underlying assumption in Chapter 7, that orientation is the key factor to reduce redundancy (notably aliasing) from the wavelet subbands to improve coding efficiency.

In the end, we also inspect the perceptual quality of full reconstructed images at different bit-rates from various images. Some examples are given in Fig. 11.4. We see that the proposed method produces significantly better reconstructed images, with less ringing around edges and more recovered textures than the conventional LeGall 5/3 wavelet transform at similar bit-rates.

11.4.3 Extension to larger wavelet transform

Although the structures of our neural networks have been developed in the first instance for the LeGall 5/3 wavelet transform, exactly the same network structures turn out to be also effective with the CDF 9/7 wavelet transform.

Specifically, we jointly train the two adaptive operators $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ to improve the CDF 9/7 wavelet transform, with all three settings of the aliasing constraint parameter λ_2. Similar as before, we evaluate the performance of the proposed method both quantitatively and qualitatively.

Examples of the visual quality of the “cleaned” LL bands are shown in Fig. 11.5. Although the CDF 9/7 wavelet transform already produces less aliased LL bands, our method still manages to reduce the remaining aliasing and produce more visually appealing LL bands. From Fig. 11.7 we also see that the visual quality of the reconstructed images from our method is significantly better than that from the CDF 9/7 wavelet transform, with much less ringing around edges.

Regarding the quantitative performance, our method can still achieve up to 11.5% average BD bit-saving for PSNR over the range of bit-rate from 0.1bpp to 1.0bpp, as seen in Table. 11.2. The story is fairly consistent for the SSIM and MS-SSIM metrics as well; the complete rate-distortion curves can be found in Fig. 11.6. All these results align with our previous conclusions with the LeGall 5/3 wavelet transform.

11.4.4 Comparison with existing works

To put our method in context, we compare the variations of the proposed approach with some existing works. These well-known works are: i) the JPEG2000 (with the LeGall 5/3 and the CDF 9/7 wavelets) and the WebP compression standards, which do not involve any machine learning; ii) iWave [23] and Dardouri [25], which are machine-learning optimized lifting schemes for wavelet-like lossy image compression; iii) Theis [29], Toderici [8] and Johnston [31], which are variants of the end-to-end optimized, non-wavelet-like learned lossy image compression systems.

For the sake of this comparison, we prefer to avoid including methods that employ a dedicated post-processing step on reconstructed data to reduce artifacts like in [22], or very sophisticated context modelling for entropy coding, such as [28] [198]. Moreover, to the best of our knowledge, all the end-to-end optimized image compression systems, including Theis [29], Toderici [8] and Johnston [31], lack important attributes of our method, such as resolution scalability, quality scalability and accessibility to region-of-interest. The resolution scalability feature, however, is found in iWave [23] and Dardouri [25], making comparisons with these methods particularly interesting. At the same time, these wavelet-like methods do not explicitly consider the visual quality of the LL bands at different resolutions, and they do not propose an end-to-end training strategy to directly optimize their methods for rate-distortion objective.

Fig. 11.8 provides the average PSNR and MS-SSIM results using the commonly tested Kodak dataset. The proposed method appears to be very competitive with other existing methods. Interestingly, the PSNR performance of iWave [23] is very close to our method, further confirming that wavelet-like compression schemes can be competitive with end-to-end optimized non-wavelet-like methods. For the other wavelet-like method Dardouri [25], we are unable to execute their inference procedure available to us. However, we observe from [25] that they are unable to present competitive PSNR and MS-SSIM results with respective to JPEG2000.

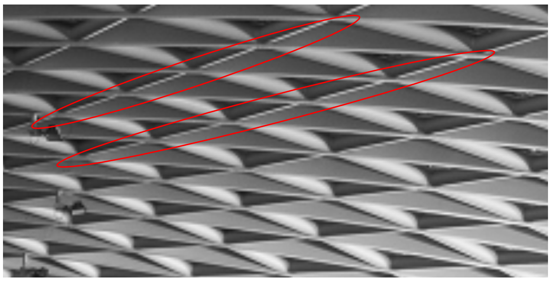
11.4.5 Computational Complexity

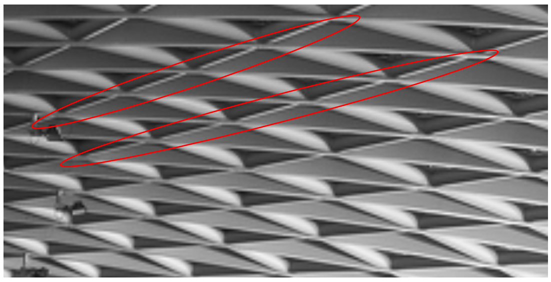
Finally, we evaluate computational complexity as well as region of support [7] associated with our method, in comparison with other existing works. From Table 11.3 we can see, our method comes with the fewest number of parameters and relatively small region of support.

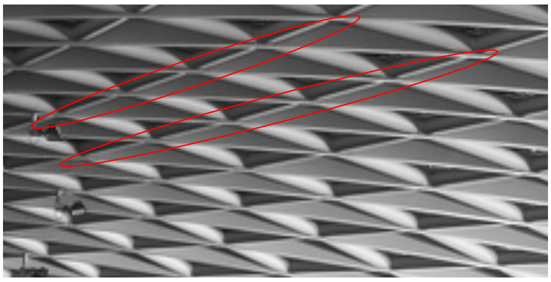
Table 11.3: Comparisons of computational complexity and region of support

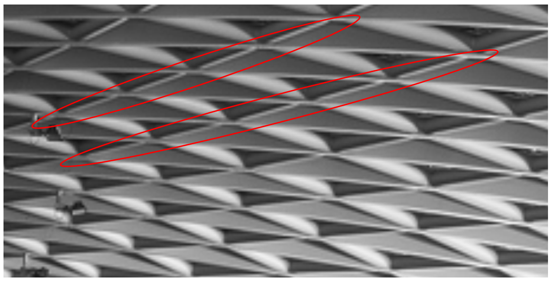
	Number of Parameters	Region of Support
Our method	$33K$	37 x 37
JPEG2000 [1]	-	9 x 7 or 5 x 3
WebP	-	-
iWave [23]	$97K$	≈ 21 x 21
Dardouri [25]	$167K$	≈ 253 x 125
Toderici [8]	$5.4M$	≈ 250 x 250
Theis [29]	$3.3M$	≈ 43 x 43
Johnston [31]	$9.9M$	≈ 300 x 300

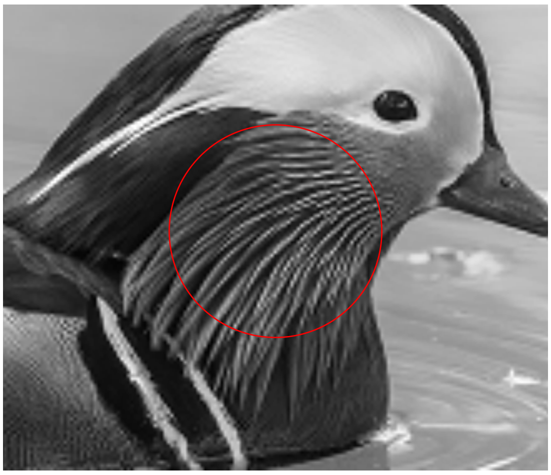
[7]Here region of support refers to the total receptive field of all networks involved in a certain approach.

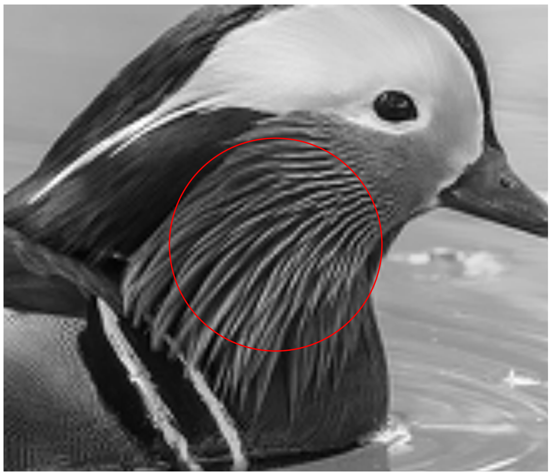

(a) LeGall 5/3, finest resolution

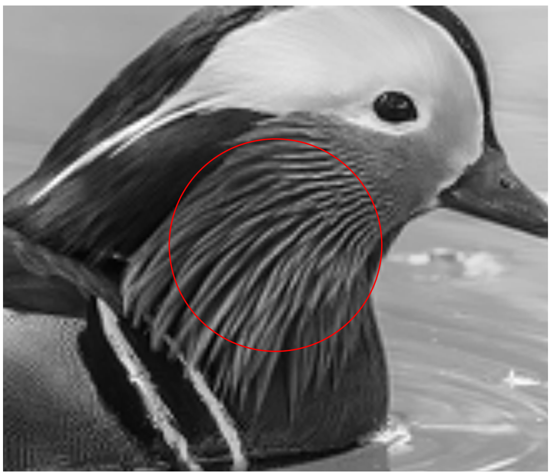

(b) λ_2=0


(c) λ_2=1, less staircases/aliasing


(d) anneal λ_2


(e) LeGall 5/3, 2^{nd} finest level


(f) λ_2=0


(g) λ_2=1, less staircases/aliasing

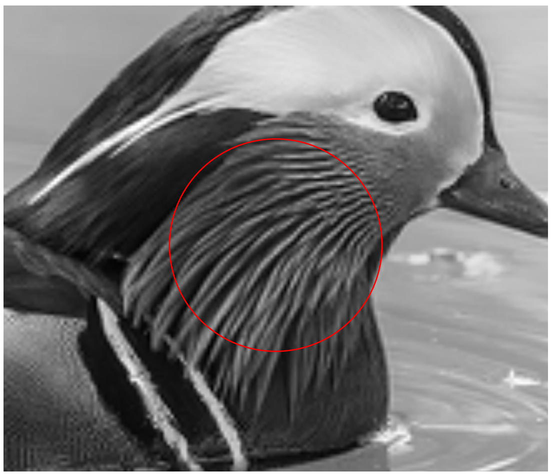
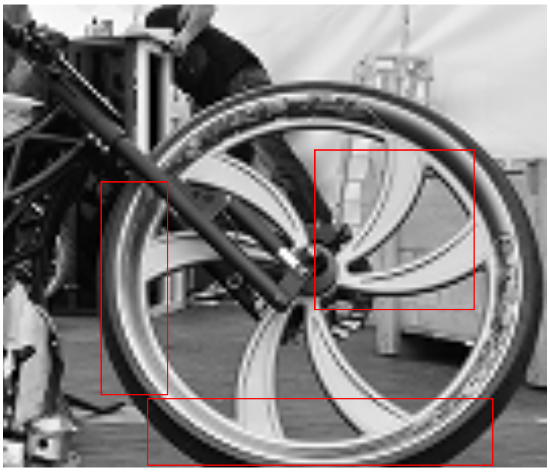
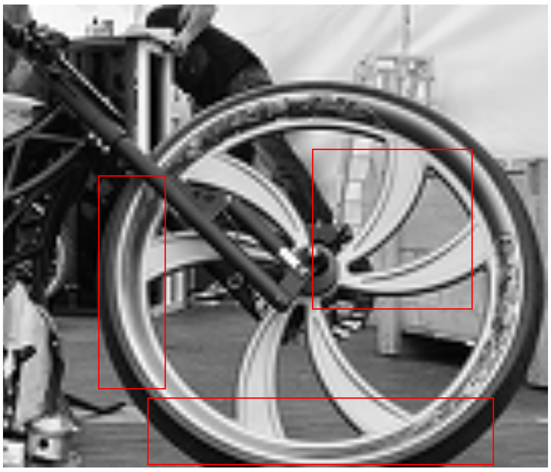

(h) anneal λ_2

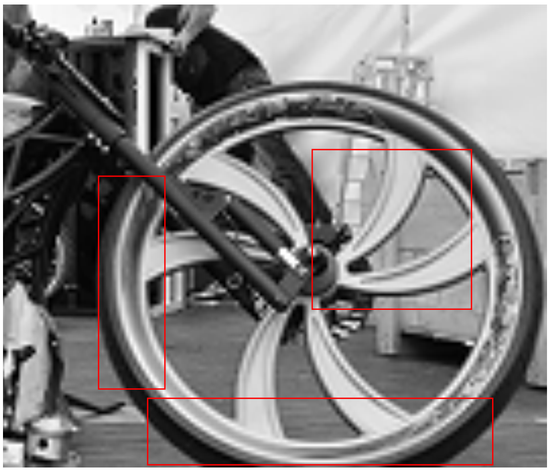
Figure 11.2: Continue to the next page...



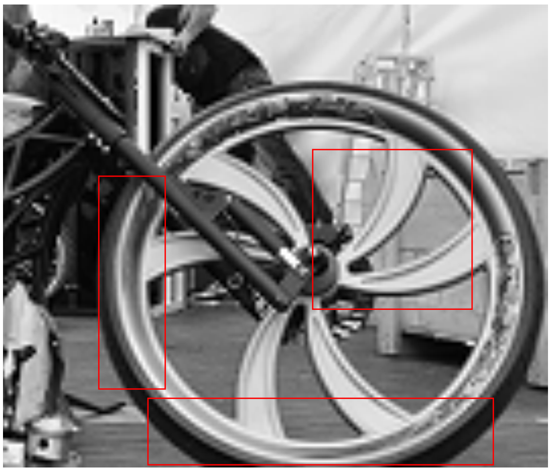
(i) LeGall 5/3, 3^{rd} finest resolution



(j) λ_2=0

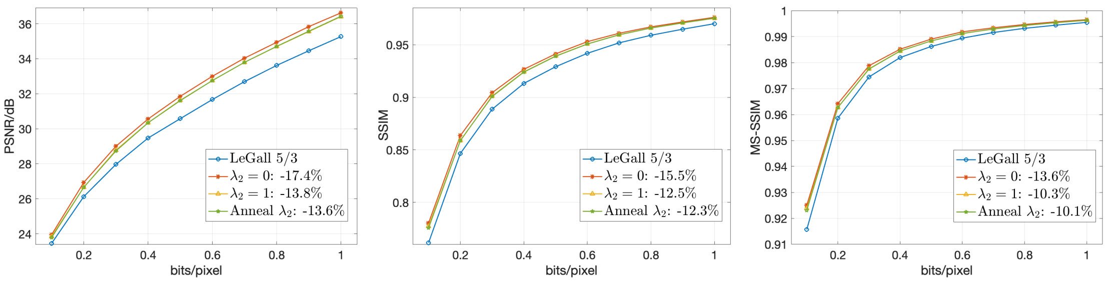


(k) λ_2=1, less staircases/aliasing

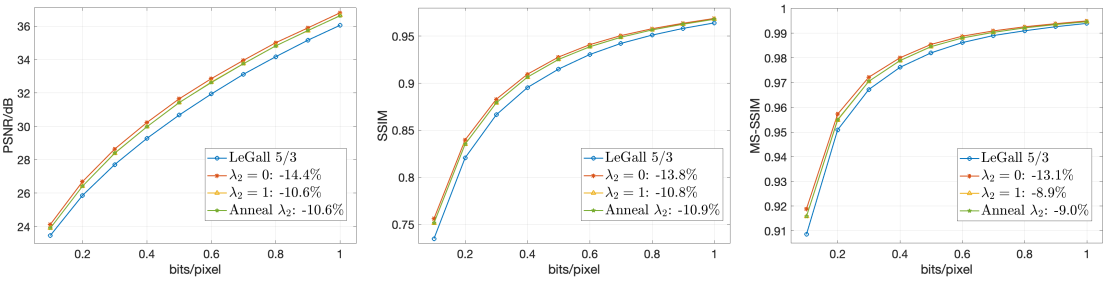


(l) anneal λ_2

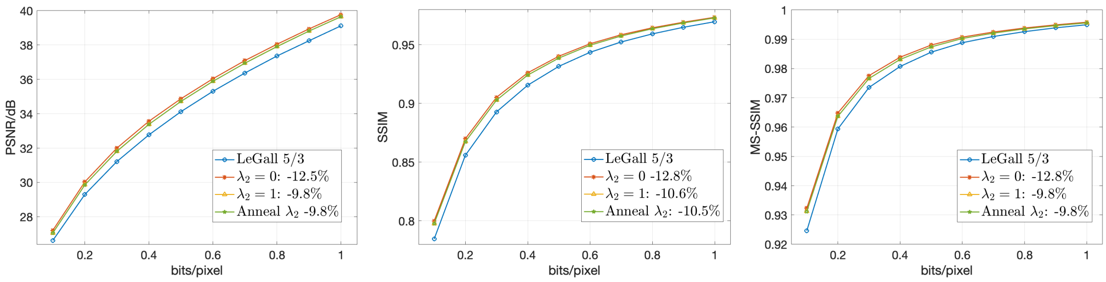
Figure 11.2: Visual quality of the "cleaned" LL bands at different scales from various images, obtained using different λ_2 strategies during training, and optimized for the LeGall 5/3 wavelet transform. Note that by reducing aliasing, we produce smoother edges with less "staircase" artifacts; red boxes lead to areas of significant difference.



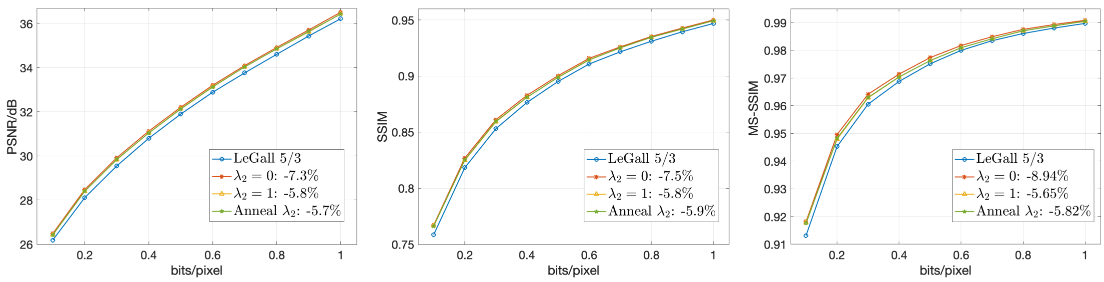
(a) Tecnick-Cat1



(b) DIV2K-Cat1



(c) DIV2K-Cat2



(d) CLIC2019-Cat3

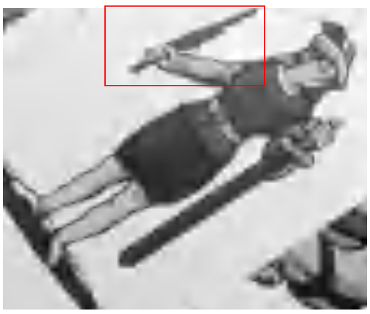
Figure 11.3: Comparisons of the average PSNR, SSIM and MS-SSIM improvements over the LeGall 5/3 wavelet transform across each dataset; the proposed method is trained to improve the LeGall 5/3 wavelet transform with various aliasing constraint parameters λ_2 during training. BD bit-rate savings (in %) are given next to the corresponding legends.

(a) the original image, cropped from image 28 of DIV2K-Cat2 dataset

(b) LeGall 5/3 at 0.199bpp, PSNR=29.53dB, SSIM=0.841, MS-SSIM=0.9585

(c) Proposed with λ_2=0,0.197bpp, PSNR=30.09dB,SSIM=0.854, MS-SSIM=0.9631

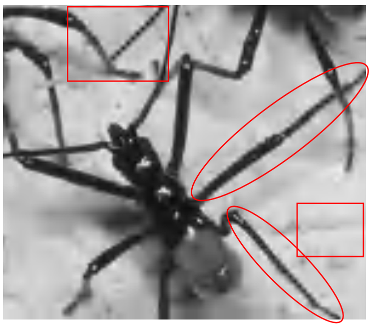
(d) the original image, cropped from image 9 of Tecnick-Cat1 dataset



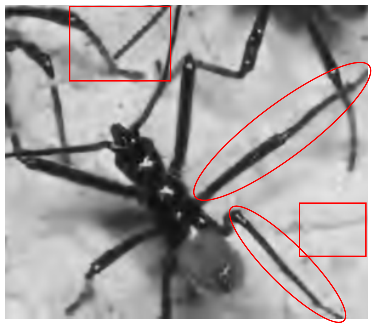
(e) LeGall 5/3 at 0.298bpp, PSNR=26.97dB, SSIM=0.862, MS-SSIM=0.9694

(f) Proposed with λ_2=0,0.295bpp, PSNR=28.19dB,SSIM=0.875, MS-SSIM=0.9718

(g) the original image, cropped from image 28 of DIV2K-Cat1 dataset

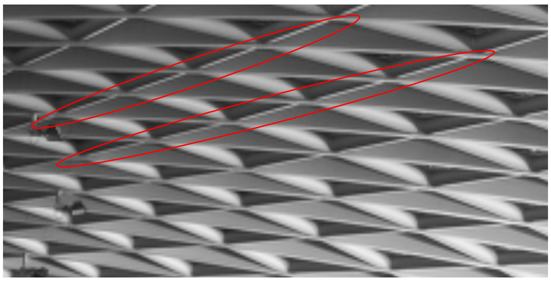


(h) LeGall 5/3 at 0.197bpp, PSNR=29.27dB, SSIM=0.831, MS-SSIM=0.9526

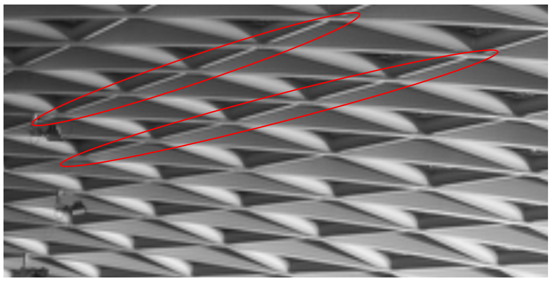


(i) Proposed with λ_2=0,0.196bpp, PSNR=30.39dB,SSIM=0.850, MS-SSIM=0.9592

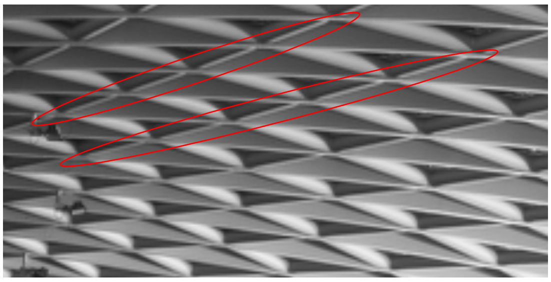
Figure 11.4: Examples of reconstructed images; red boxes lead to areas of significant difference. The proposed method is trained to improve the conventional LeGall 5/3 wavelet transform with $\lambda_2 = 0$.



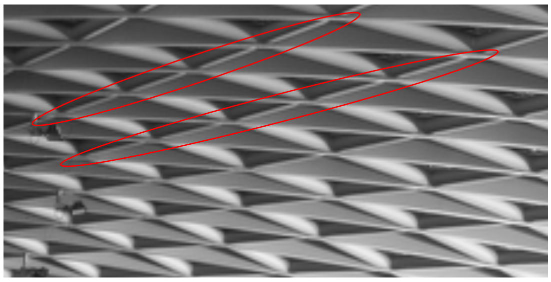
(a) CDF 9/7, finest resolution



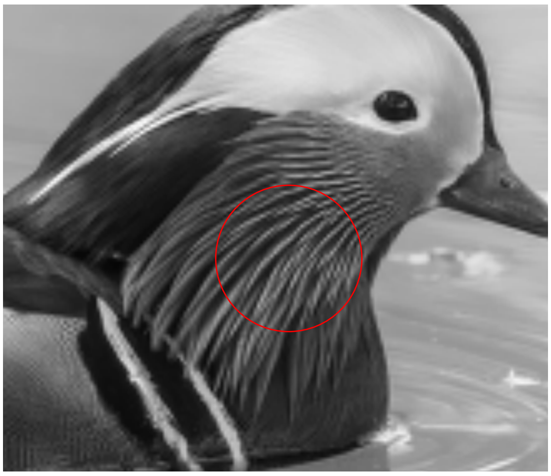
(b) λ_2=0



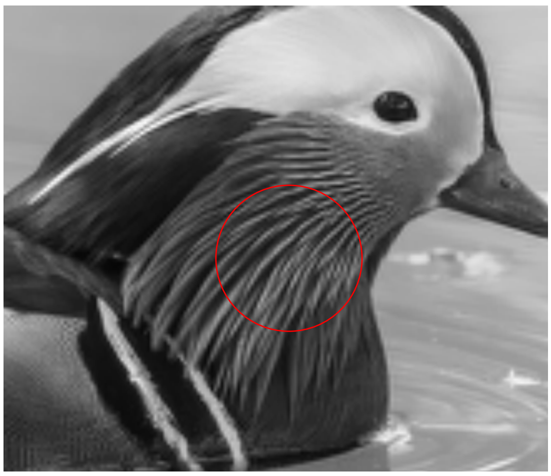
(c) λ_2=1, less staircases/aliasing



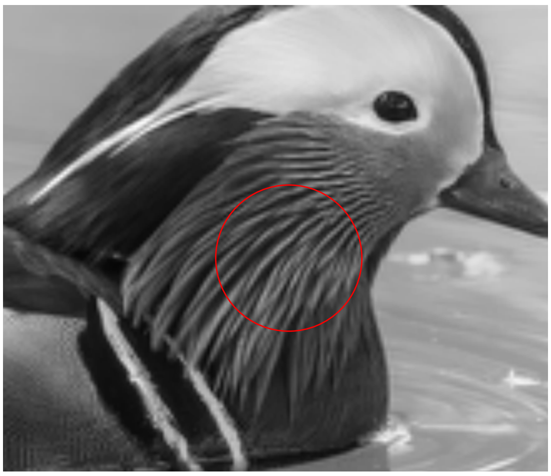
(d) anneal λ_2



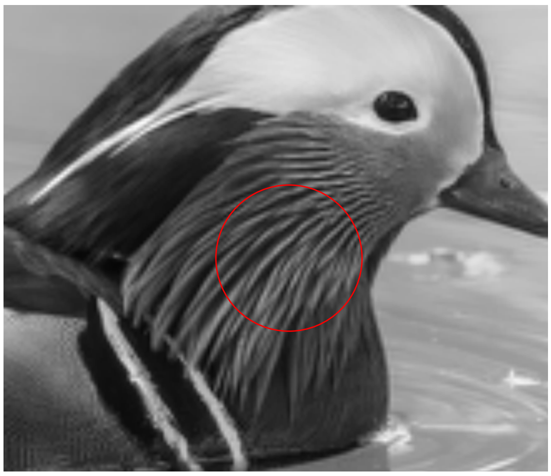
(e) CDF 9/7, 2^{nd} finest resolution



(f) λ_2=0

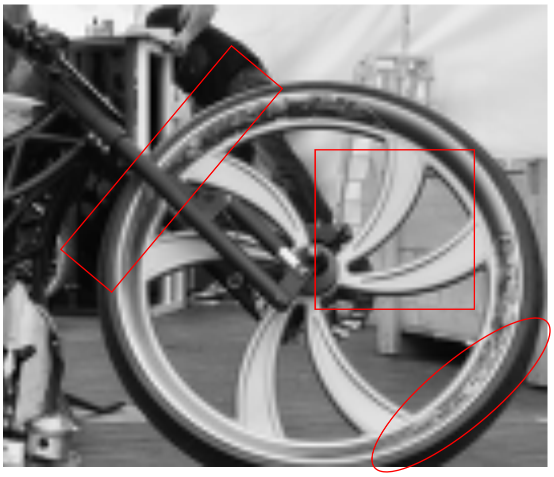


(g) λ_2=1, less staircases/aliasing

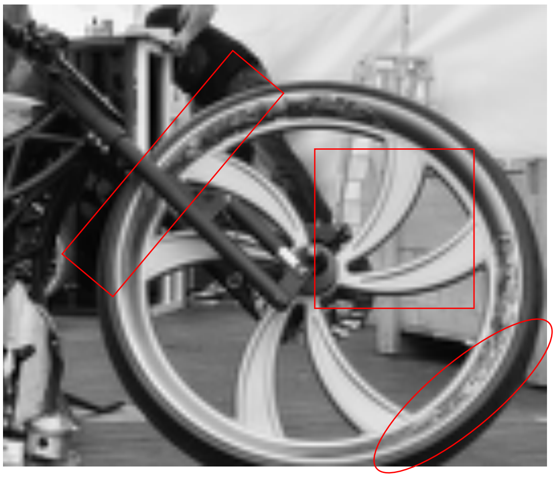


(h) anneal λ_2

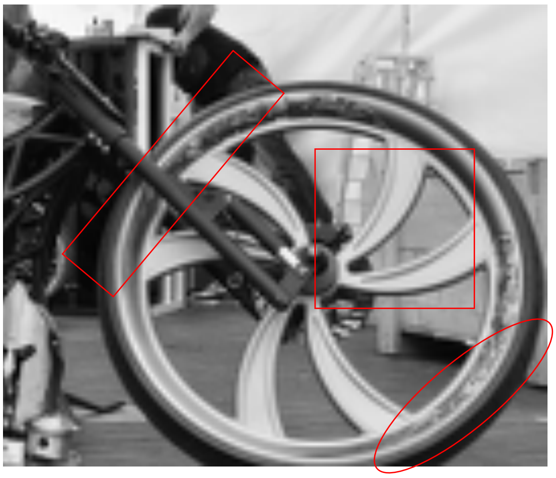
Figure 11.5: Continue to the next page...



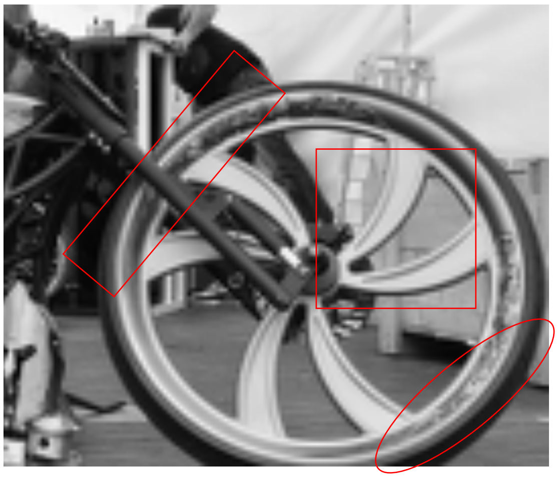
(i) CDF 9/7, 3^{rd} finest resolution



(j) λ_2=0

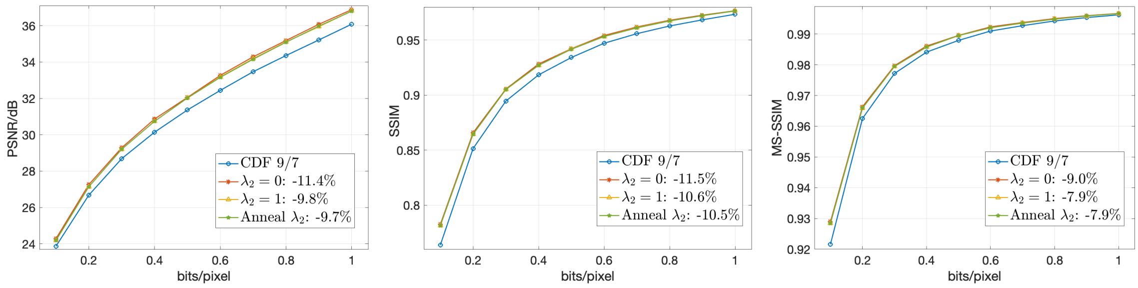


(k) λ_2=1, less staircases/aliasing

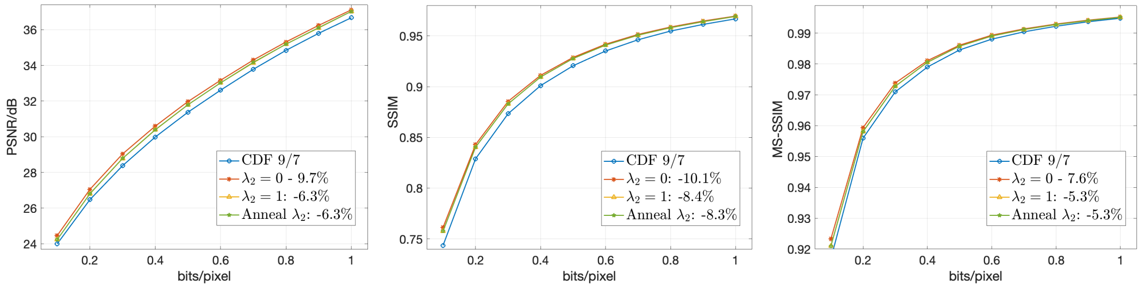


(l) anneal λ_2

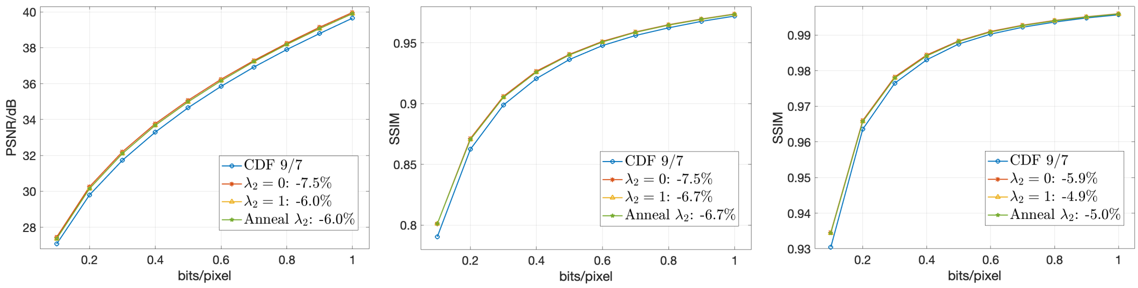
Figure 11.5: Visual quality of the “cleaned” LL bands at different scales from various images, obtained using different λ_2 during training and optimized for the CDF 9/7 wavelet transform. Note that by reducing aliasing, we produce smoother edges with less visible “staircase” artifacts; red boxes lead to areas of significant difference.



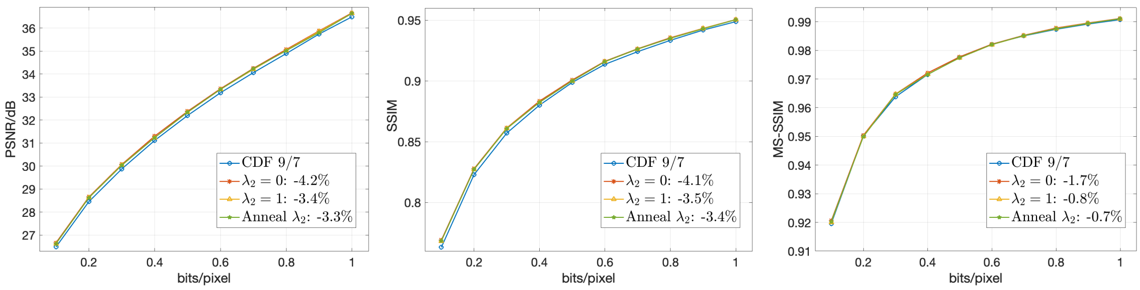
(a) Tecnick-Cat1



(b) DIV2K-Cat1



(c) DIV2K-Cat2



(d) CLIC2019-Cat3

Figure 11.6: Comparisons of the average PSNR, SSIM and MS-SSIM improvements over the CDF 9/7 wavelet transform across each dataset, with various aliasing constraint parameters λ_2 in (10.1) during training. BD bit-rate savings (in %) are given next to the corresponding legends.

(a) the original image, cropped from image 28 of DIV2K-Cat2 dataset

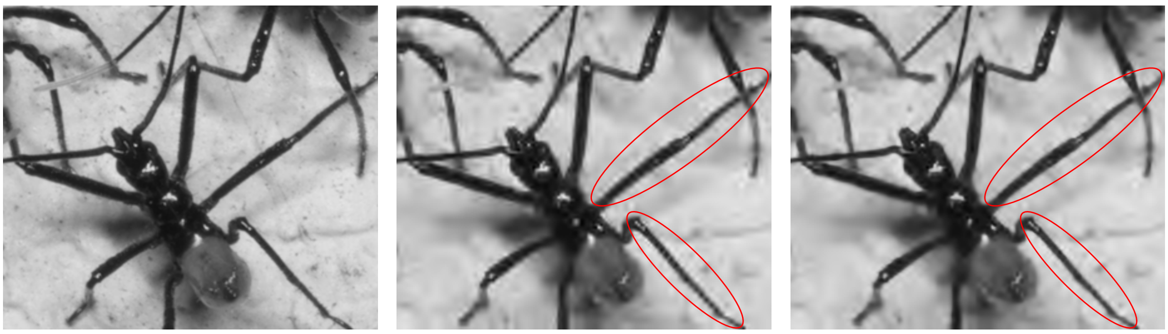
(b) CDF 9/7, bit-rate=0.196bpp, PSNR=29.91dB, SSIM=0.845, MS-SSIM=0.9614

(c) Proposed with $\lambda_2 = 0$, bit-rate=0.197bpp, PSNR=30.63dB, SSIM=0.859, MS-SSIM=0.9652

(d) the original image, cropped from image 9 of Tecnick-Cat1 dataset

(e) CDF 9/7, bit-rate=0.295bpp, PSNR=27.36dB, SSIM=0.854, MS-SSIM=0.9681

(f) Proposed with $\lambda_2 = 0$, bit-rate=0.295bpp, PSNR=28.41dB, SSIM=0.870, MS-SSIM=0.9711

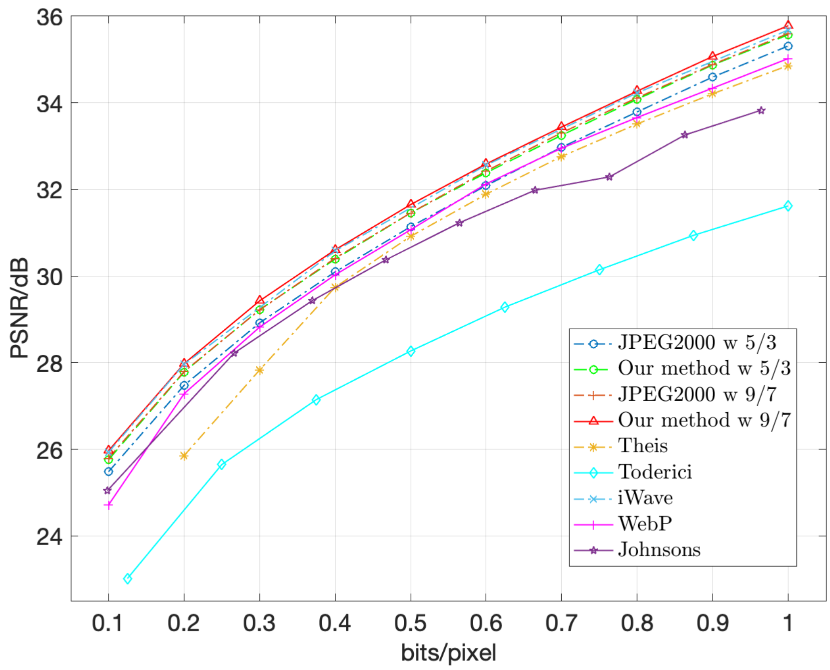


(g) the original image, cropped from image 28 of DIV2K-Cat1 dataset

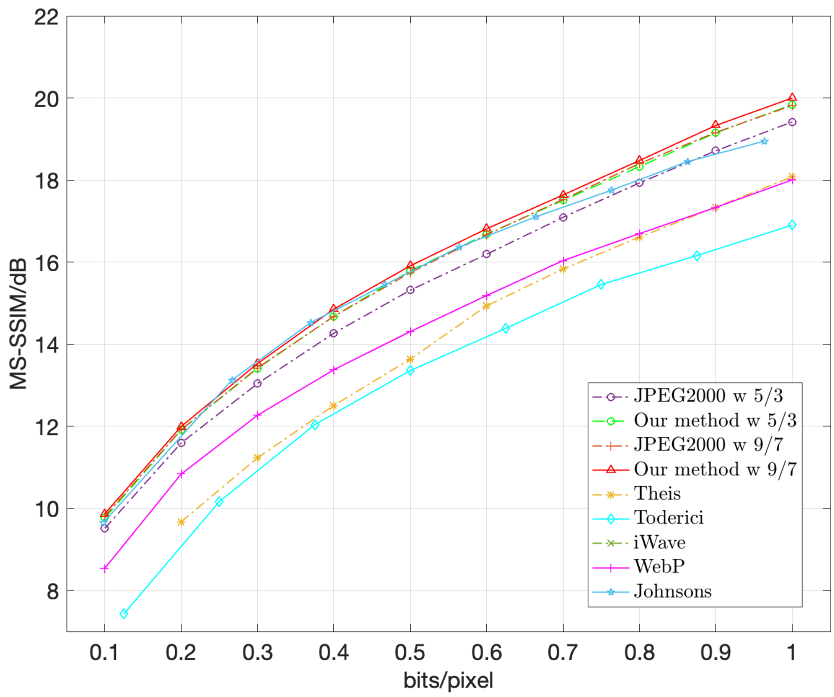
(h) CDF 9/7, bit-rate=0.199bpp, PSNR=30.13dB, SSIM=0.841, MS-SSIM=0.9573

(i) Proposed with $\lambda_2 = 0$, bit-rate=0.198bpp, PSNR=30.84dB, SSIM=0.856, MS-SSIM=0.9614

Figure 11.7: Examples of different reconstructed images, using the proposed method trained with $\lambda_2 = 0$ and the CDF 9/7 wavelet transform; red boxes lead to areas of significant difference.



(a) PSNR/dB



(b) MS-SSIM/dB

Figure 11.8: Comparisons of the average PSNR and MS-SSIM improvements between our methods and other existing works for the Kodak Dataset. MS-SSIM are calculated in dB as: $-10\log_{10}(1-\text{MS-SSIM})$. The results of other works are taken from the original papers without any reproduction.

Chapter 12

Contribution: Further Studies On Extensions of Neural Networks To The Existing Wavelet Transform

From Chapter 8 to Chapter 11, we have explored the merits of augmenting the existing base wavelet transform with two additional lifting steps $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$. Through this exploration, we have discovered that the proposal-opacity network architecture and the end-to-end optimization framework with backward annealing are particularly useful.

In this chapter, we leverage these two discoveries to bear upon a more comprehensive analysis on what can really be achieved by learning-based wavelet-like transforms in a critically sampled highly scalable compression system. In this more comprehensive analysis, we consider the merit of replacing all the fixed lifting steps that correspond to the base wavelet transform with neural networks. Since these lifting networks generally exhibit substantially larger region of support as well as higher computational complexity than the corresponding fixed lifting filters in the base wavelet transform, it is important to study whether or not the benefit in coding performance can be justified by the negative impacts on complexity and region of support.

Moreover, we also study the relationship between the depth of lifting structures (i.e. the number of learned lifting steps) and coding performance. Ultimately, all learned lifting steps in this thesis are employed in a critically sampled highly scalable compression system, in which only one trained model is applied to all levels of decomposition and for all bit-rates of interest. Therefore, it is not clear whether or not employing deep fully learned lifting structures would be beneficial in this context.

Furthermore, we study the impact of diversity (i.e. the number of channels in each lifting network) on coding performance, noting that the network diversity can be increased without incurring any cost to region of support. In the end, since compact region of support is one of the fundamental features of the base wavelet transform, and this depends on both the support of network kernels and the number of layers, we also examine whether or not similar coding performance can be achieved using lifting networks with smaller kernels and fewer layers.

Through this comprehensive study, we discover that developing a good training schedule becomes crucial, especially for the success of the work in this chapter. This arises because gradient descent, which underlies most of machine-learning based approaches (including the work in this chapter), can easily get stuck in local optima well before encountering a good solution when initialized with random conditions. To address this issue, we propose an oracle-opacity training schedule to progressively train all learned lifting steps in a disciplined way, as seen in Section 12.3.2; this is also a major contribution of this chapter.

The rest of this chapter is arranged as follows. We first introduce the lifting structures that we choose to investigate in Section 12.1. Subsequently, we explain how the proposal-opacity network architecture is leveraged and extended to all lifting steps that correspond to the base wavelet transform in Section 12.2. In the end, we give details on how to create oracle opacities for the proposed training schedule in Section 12.3.3.

12.1 Investigated Lifting Structures

This section introduces the lifting structures that we investigate in this chapter. These structures help us study more broadly the application of neural networks to all lifting steps that correspond to the existing base wavelet transform.

From our previous work in Chapter 8, the explored lifting structure is a mixture of fixed and learned lifting steps, as illustrated in Fig. 12.1; the fixed lifting steps correspond to the existing base wavelet transform, while the learned lifting steps correspond to the high-to-low and the low-to-high networks, $\mathcal{T}_{H2L}$ and $\mathcal{T}_{L2H}$ in the figure, which serve as additional lifting steps to exploit residual redundancy between successive levels of the base wavelet transform. Despite the success of this earlier work, the question remains whether

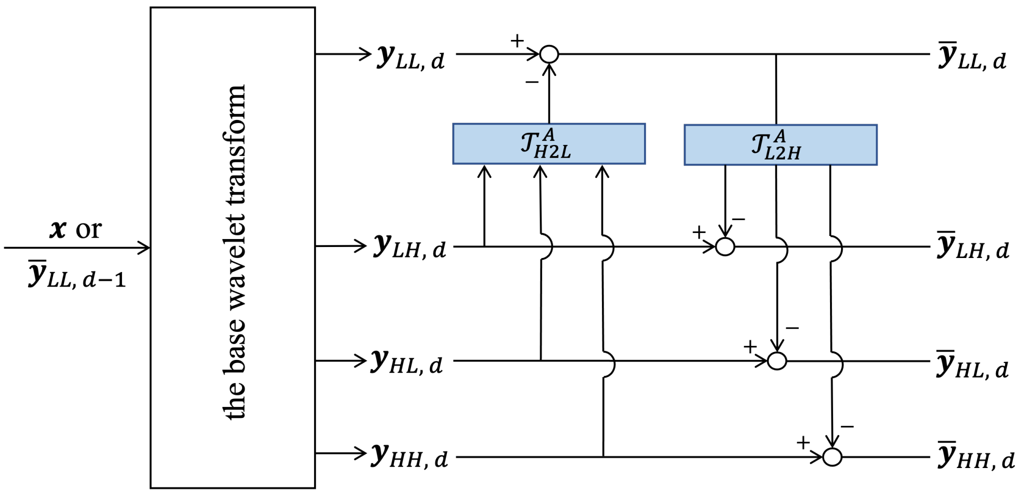


Figure 12.1: The lifting structure employed in our earlier work from Chapter 8, in which the base wavelet transform does not involve any learning; it is simply the conventional wavelet transform that has been commonly utilized in image compression applications, i.e. either the LeGall 5/3 or the CDF 9/7 bi-orthogonal wavelet transform. The high-to-low and low-to-high operators, $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$, serve as additional lifting steps to augment the base wavelet transform; these two additional steps employ neural networks and are considered to be adaptive to local scene geometry.

neural networks can be beneficially employed in the lifting steps that correspond to the base wavelet transform as well. One possible way to address this question is to replace the base wavelet transform with an arbitrary sequence of learned lifting steps. However, training of these learned lifting steps becomes very difficult. In fact, as we shall see in Section 13.4.1, these learned steps can easily wind up exhibiting worse coding efficiency than the base wavelet transform.

To address this, we adopt the following approach in this chapter: the base wavelet transform is first factorized into a sequence of two-dimensional lifting steps as seen in Fig. 12.2, each of which can then be replaced individually with a learnable equivalent. To be more specific, the input image $\mathbf{x}$ (or the low-pass band $\overline{\mathbf{y}}_{LL,d-1}$ at level $d-1$ of the wavelet decomposition) is first split into four cosets $\mathbf{y}_{ee,d}$, $\mathbf{y}_{oe,d}$, $\mathbf{y}_{eo,d}$ and $\mathbf{y}_{oo,d}$ as

$$\mathbf{y}_{ee,d}[n_1,n_2] = \begin{cases} \mathbf{x}[2n_1, 2n_2], & d=1 \\ \overline{\mathbf{y}}_{LL,d-1}[2n_1,2n_2], & d>1 \end{cases}$$

$$\mathbf{y}_{oe,d}[n_1,n_2] = \begin{cases} \mathbf{x}[2n_1+1, 2n_2], & d=1 \\ \overline{\mathbf{y}}_{LL,d-1}[2n_1+1,2n_2], & d>1 \end{cases}$$

$$\mathbf{y}_{eo,d}[n_1,n_2] = \begin{cases} \mathbf{x}[2n_1, 2n_2+1], & d=1 \\ \overline{\mathbf{y}}_{LL,d-1}[2n_1,2n_2+1], & d>1 \end{cases}$$

$$\mathbf{y}_{oo,d}[n_1,n_2] = \begin{cases} \mathbf{x}[2n_1+1, 2n_2+1], & d=1 \\ \overline{\mathbf{y}}_{LL,d-1}[2n_1+1,2n_2+1], & d>1 \end{cases} \tag{12.1}$$

The cosets with even rows $\mathbf{y}_{ee,d}$ and $\mathbf{y}_{eo,d}$ are then employed to predict the cosets with odd rows $\mathbf{y}_{oe,d}$ and $\mathbf{y}_{oo,d}$ respectively, using the vertical-predict operator $\mathcal{P}^V$. Subsequently, $\mathbf{y}_{oe,d}$ and $\mathbf{y}_{oo,d}$, along with their predictions from $\mathbf{y}_{ee,d}$ and $\mathbf{y}_{eo,d}$, are utilized to update $\mathbf{y}_{ee,d}$ and $\mathbf{y}_{eo,d}$ respectively, employing the vertical-update operator $\mathcal{U}^V$. Similarly, the horizontal-predict and the horizontal-update operators $\mathcal{P}^H$ and $\mathcal{U}^H$ can be applied to predict and then update the cosets with odd and even columns, as shown in Fig. 12.2. The results are the wavelet low-pass band $\mathbf{y}_{LL,d}$ and the detail bands $\mathbf{y}_{HL,d}$, $\mathbf{y}_{LH,d}$ and $\mathbf{y}_{HH,d}$ at level d of the two-dimensional decomposition.

In the conventional LeGall 5/3 wavelet transform, $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$ are simply one-

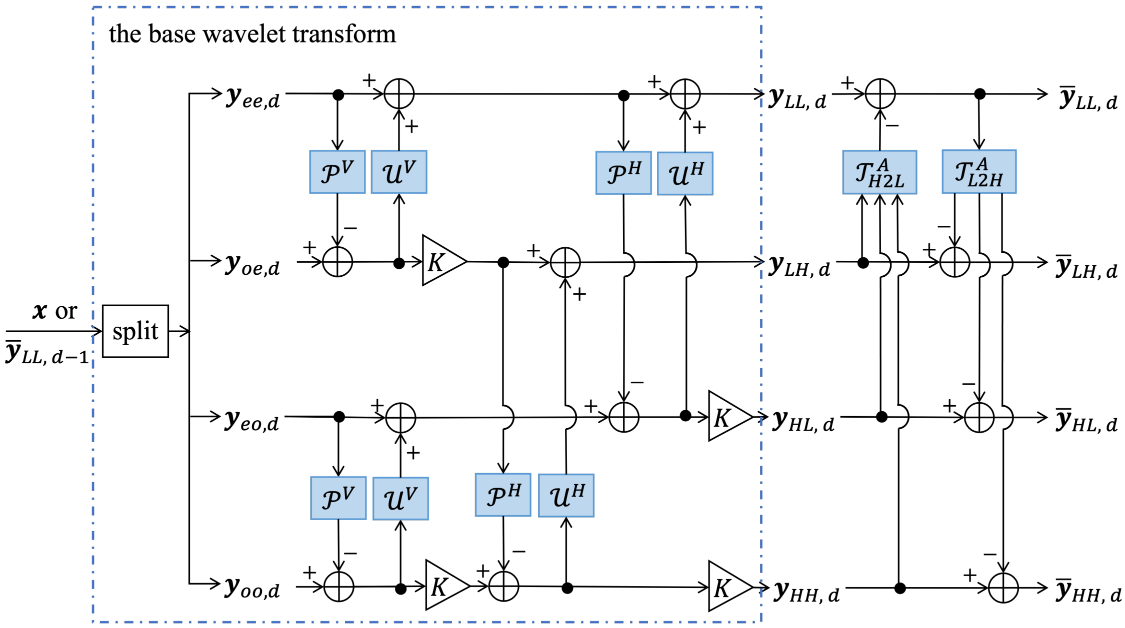


Figure 12.2: The factorization of the base wavelet transform shown in Fig 12.1 into a sequence of two-dimensional lifting steps; the operators $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$ represent the vertical-predict, the vertical-update, the horizontal-predict and the horizontal-update operators, respectively. K denotes the scaling factor; in the LeGall 5/3 wavelet transform, $K = 0.5$.

dimensional fixed separable lifting filters with transfer functions as

$$\mathcal{P}(z) = -\frac{1}{2}(1+z) \tag{12.2}$$

$$\mathcal{U}(z) = \frac{1}{4}(1+z^{-1}) \tag{12.3}$$

and so

$$\mathcal{P}^V(z_1, z_2) = \mathcal{P}(z_1), \quad \mathcal{P}^H(z_1, z_2) = \mathcal{P}(z_2) \tag{12.4}$$

$$\mathcal{U}^V(z_1, z_2) = \mathcal{U}(z_1), \quad \mathcal{U}^H(z_1, z_2) = \mathcal{U}(z_2) \tag{12.5}$$

In this chapter, we replace each of the operators $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$ individually with a learnable two-dimensional network. In addition, as we shall see in Section 12.2, we also choose a particular network architecture; it allows the more expressive learned lifting networks to discover the existing base wavelet transform as one possible solution, which already works well in terms of coding efficiency for compression. Moreover, when we put these learned lifting steps together with the two additional $\mathcal{T}_{H2L}$ and $\mathcal{T}_{L2H}$ as shown in

Fig. 12.2, the resulting lifting structure can also be capable of discovering the solution presented in our previous work in Chapter 8 and Chapter 11.

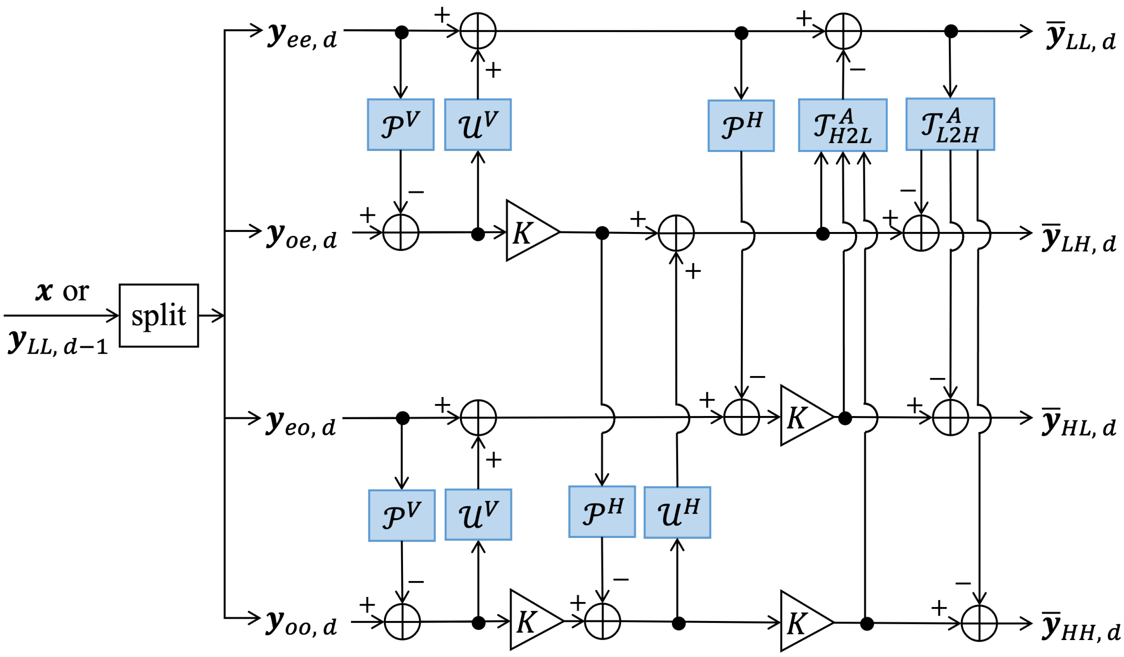
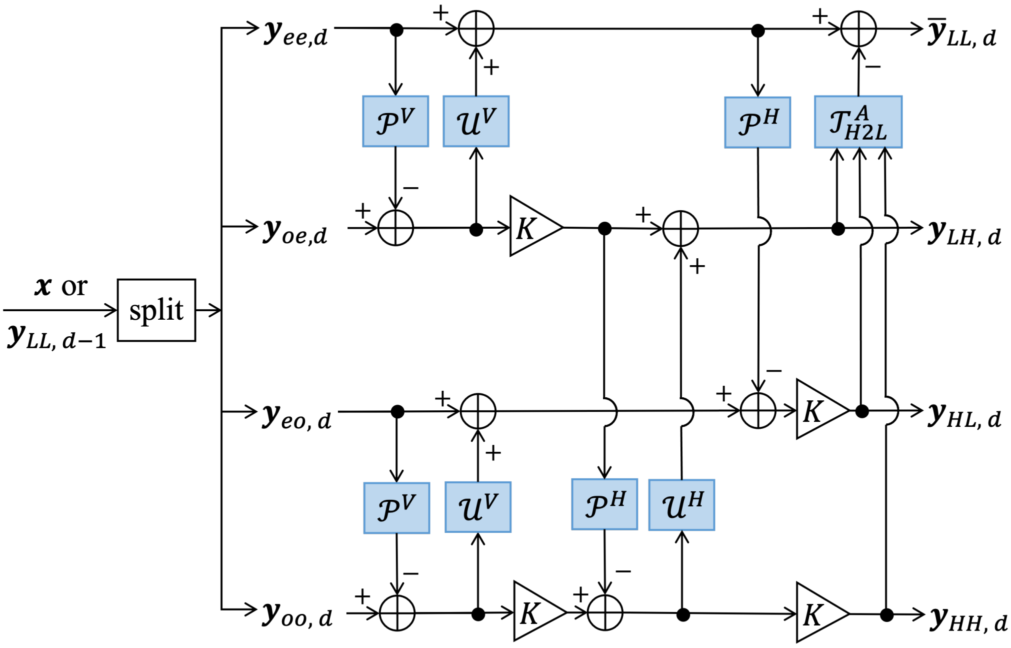


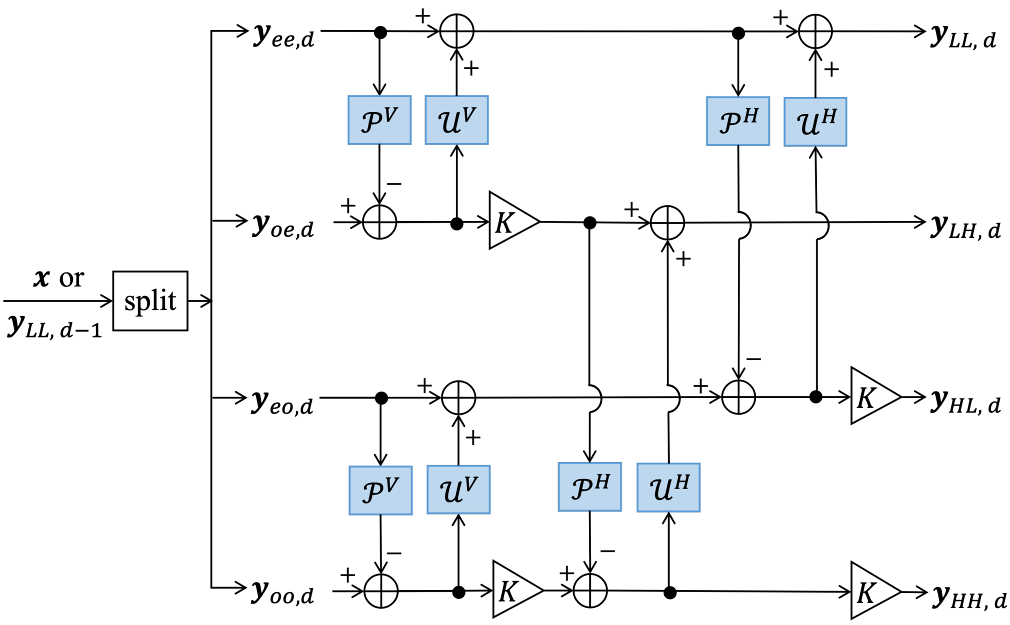
Figure 12.3: Fusing the last update operator $\mathcal{U}^H$ in Fig. 12.2 with the high-to-low network $\mathcal{T}^A_{H2L}$. The symbol K denotes the scaling factor; in the LeGall 5/3 wavelet transform, $K = 0.5$.

Furthermore, it is worthwhile to point out that the replacement of each individual lifting step in Fig. 12.2 with neural network comes with a cost. This is because each lifting network generally has a substantially larger region of support as well as higher computational complexity than the corresponding fixed lifting filter. More importantly, since each lifting network usually involves non-linearities, quantization errors can expand in an uncontrollable way through these non-linearities during synthesis. This is particularly important for the work in this chapter, because we aim to employ only one set of trained lifting networks for all levels of the wavelet decomposition and for all bit-rates of interest over a wide range, leading to a highly scalable compression system that preserves quality scalability, resolution scalability and region-of-interest accessibility.

As a result, it is natural to consider minimizing the number of learned lifting steps in



(a)



(b)

Figure 12.4: Progressively dropping the two additional lifting networks $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ shown in Fig 12.3. These two lifting structures allow us to study whether $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ can be absorbed into the previous learned lifting steps or not.

Fig. 12.2, so as to reduce the potential influence of non-linearities introduced by learned lifting networks in the synthesis path. This can be addressed by fusing the last update operator $\mathcal{U}^H$ with $\mathcal{T}^A_{H2L}$ as shown in Fig. 12.3, so that the total number of learned lifting steps can be reduced by one. In addition, we also consider progressively dropping the two additional steps $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$, resulting in other two structures as illustrated in Fig. 12.4. These two lifting structures allow us to study whether $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ can be absorbed into the previous learned lifting steps or not.

12.2 Proposal-opacity Network Architecture

12.2.1 Significance of the proposal-opacity topology

As we have mentioned in Section. 12.1, the underlying principle behind the development of our previous work in Chapter 8 is that residual redundancy between successive levels of the wavelet transform can be substantially removed by the introduction of additional lifting networks. A major form of this residual redundancy comes as aliasing information in the wavelet transform. Although we could build deep neural networks to exploit aliasing information amongst the wavelet subbands, deep networks also come with penalties in terms of computational complexity, region of support and robustness of a highly scalable compression system, as we have explained in Section 12.1.

In fact, our previous discussion in Chapter 7 has shown that if we know local orientations (i.e. geometric flow) a priori, then the solution to eliminating aliasing information from both the low- and high-pass subbands of the wavelet transform can be simply a linear filter. However, since geometric flow is a local property that is hard to accurately determine within an image, in reality, the untangling of aliasing from the wavelet subbands requires either an adaptive filtering solution or a bank of filters with an adaptive strategy for combining their responses.

Actually, after exploring various neural network architectures in Chapter 9, the best investigated network indeed turns out to be structured in terms of proposals and opacities; the proposals essentially form a bank of linear filters (or can be understood as candidate lifting steps) while the opacities provide a data-dependent blending of these linear proposals, as depicted in Fig. 12.5. We refer to this particular multiplicative architecture as a *proposal-opacity topology*. As we shall see in Section 12.2.3, this is also the network architecture that we decide to leverage in this chapter when studying the extension of neural networks to all the lifting steps $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$ within the base wavelet transform.

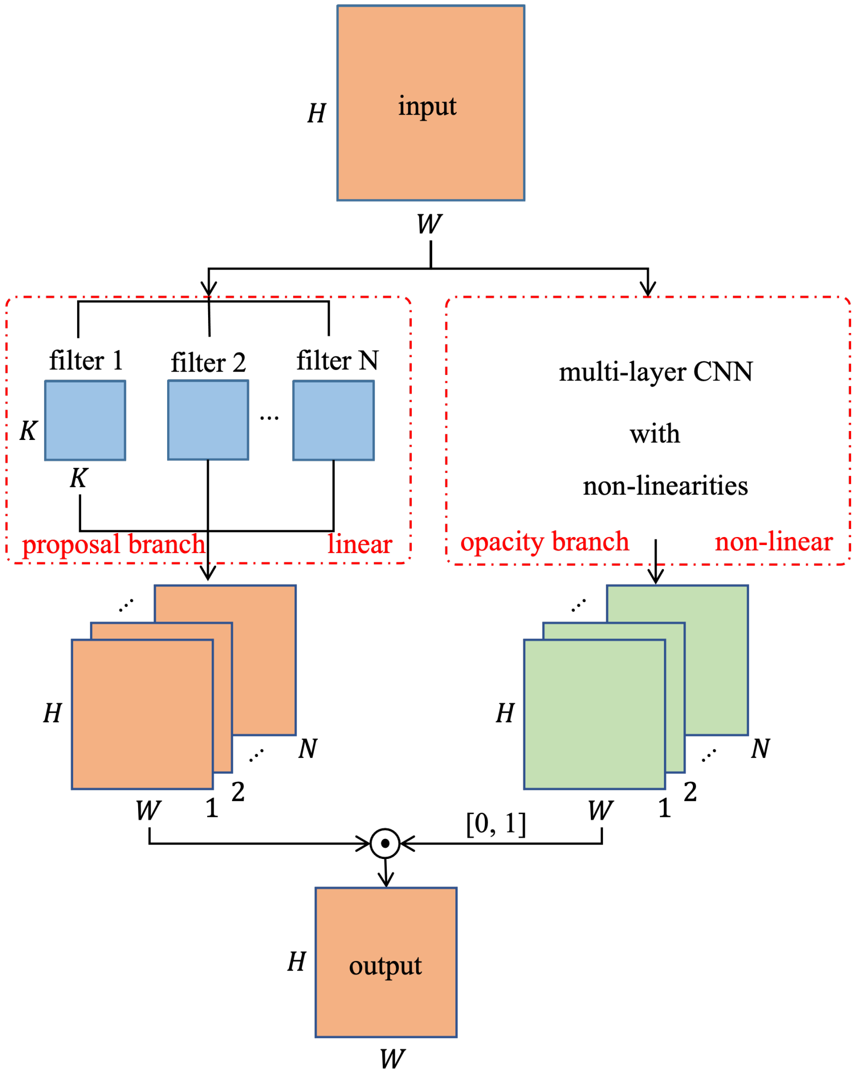


Figure 12.5: The proposal-opacity neural network topology proposed previously in Chapter 9. The symbol K x K denotes the filter support while N represents the number of filters (or equivalently the number of channels in the proposal/opacity branch).

Furthermore, this proposal-opacity network topology also offers the benefit that it can easily replicate the fixed lifting filters in the base wavelet transform. For instance, if the N filters in the proposal branch are all fixed as the one in (12.2), then the proposal-opacity network always produces the fixed lifting filter in (12.2) regardless of the outcomes of the opacity branch. This additional benefit is important for developing learned networks for $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$, as we expect the learned lifting networks to be still capable of discovering the base wavelet transform as one possible solution during training, as we have discussed in Section 12.1.

12.2.2 Particular properties of the opacity branch

Since the opacity branch in Fig. 12.5 can be generally understood as analysing local scene geometry to produce opacities (or likelihoods) that are used to blend linearly generated proposals, it is reasonable to constrain the outcomes of the opacity branch within the range from 0 to 1. Moreover, an important pre-requisite for the opacity branch is that it is capable of producing opacity maps that are invariant to absolute image intensity and contrast.

One way to preserve these properties is to adopt a log-like activation function after a succession of convolutional layers (no bias) with *ReLU*, as shown in Fig. 12.6(a); the employed log-like function is defined as

$$y = \begin{cases} \log\left(x + \text{offset}\right) & x > -\text{offset}/2 \\ \log\left(\text{offset}/2\right) & \text{otherwise} \end{cases} \tag{12.6}$$

where offset $= 0.01$ is chosen to define the derivative of the function at the origin for back-propagation. To explain Fig. 12.6(a) further, the log-like operator effectively converts intensity scaling factors into additive offsets. The following linear layer is expected to behave like an competitive operator, which can effectively remove these additive offsets. The final non-linearities *tanh* and *ReLU* are only placed to ensure that the outcomes of the opacity branch are within the range 0 to 1.

Another way to explicitly force this intensity-contrast independence of the opacity branch is to use a normalization block as shown in Fig. 12.6(b). The normalization function employed here is defined as

$$y = \frac{x_i + \text{offset}}{\sum_{i=1}^{N}(x_i + \text{offset})}, \quad \text{offset} = 0.01 \tag{12.7}$$

which is also capable of removing arbitrary intensity scaling factors from the input data.

Since Fig. 12.6(a) and (b) both preserve desirable properties for the opacity branch, it is natural to consider which one is more preferable. In our previous work in Chapter 9, we chose to utilize Fig. 12.6(a), because the two additional networks $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ merely modify the behaviour of the base wavelet transform. Therefore, it is not fundamentally problematic if all opacities turn out to be zero. However, utilizing such an approach may not be appropriate when replacing all the lifting steps with neural networks in the base wavelet transform. This is because at least one of the proposals needs to be in play in this scenario, which means the opacity branch should not deliver likelihoods that are all close to zero. As a result, we employ the approach in Fig. 12.6(b) when developing all the lifting networks $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$, $\mathcal{U}^H$, $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ in this chapter, as seen in Section 12.2.3.

12.2.3 Proposed lifting networks

Leveraging the proposal-opacity network topology in Section 12.2.1 and the general opacity architecture in Fig. 12.6(b), we construct the learned networks for the lifting steps $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$, as depicted in Fig. 12.7. The architecture of the opacity branch is heavily inspired by our previous development in Chapter 9, employing residual blocks that have been demonstrated to be useful in feature detection for the non-linear opacity branch. The linear proposal branch is chosen to have the same region of support as the opacity branch. We utilize this common network architecture for all four lifting networks $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$ in this chapter, except that the input to each network is different according to Fig. 12.2. For the sake of simplicity, the high-to-low and the low-to-high networks $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ also evolve from those in Chapter 9 into Fig. 12.8, employing Fig. 12.6(b) instead

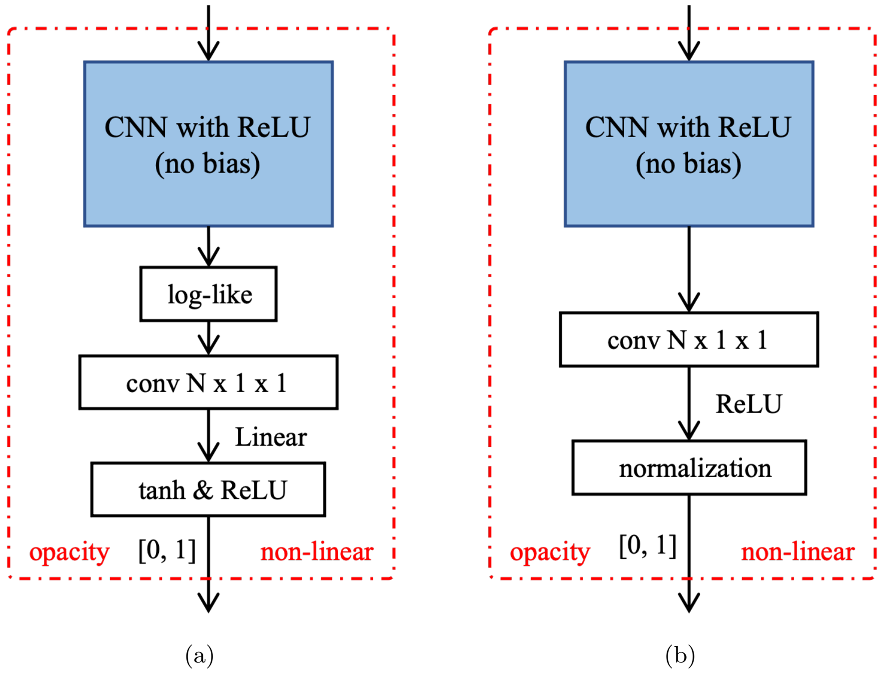


Figure 12.6: (a) depicts the log-like activation function after a succession of convolutional layers (no bias) with *ReLU*; this is the opacity architecture that we adopt in our previous work in Chapter 9. (b) illustrates the normalization block that we employ in this chapter to force the intensity-contrast independence of the opacity branch shown in Fig. 12.5.

of Fig. 12.6(a) to force the intensity-contrast independence for the opacity branches in this chapter.

Similar to our previous work in Chapter 11, we aim to employ only one set of learned lifting networks for all levels of the wavelet decomposition. This is goal is explicitly chosen because it essentially provides the learned lifting networks with an incentive to produce a compression system that naturally has a multi-scale, self-similar interpretation of images. More importantly, it is more sensible for practical applications to employ a method which only uses one trained model for all levels. This is especially important for scalable codecs, where the number of levels received at the decoder may not be the same as the encoder.

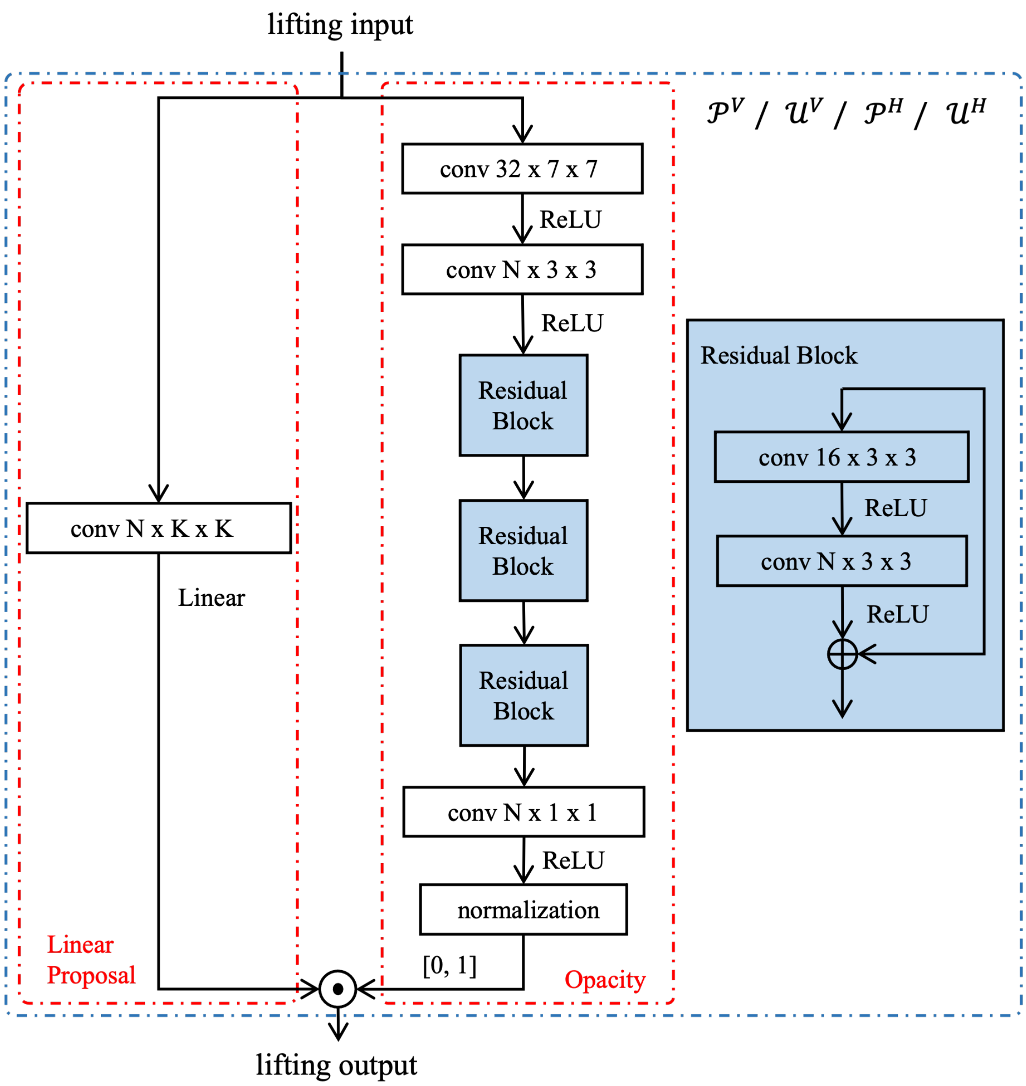
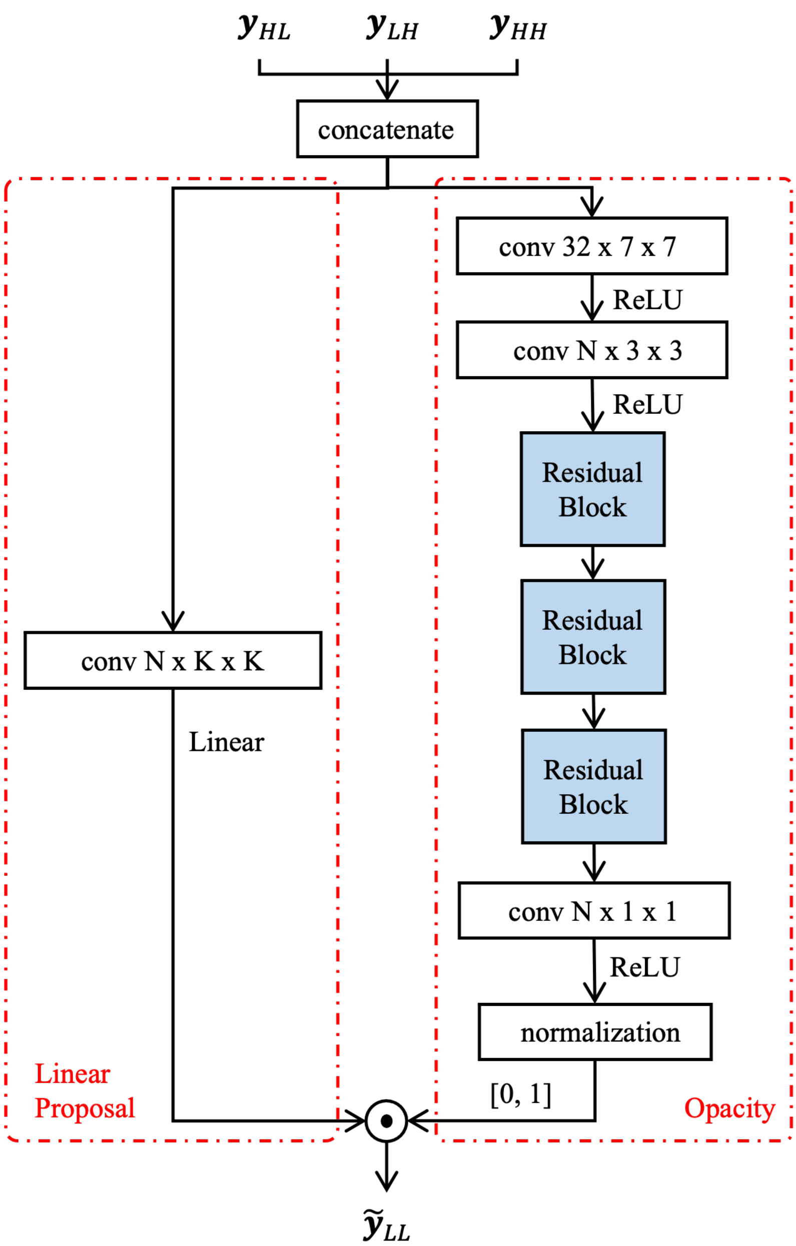
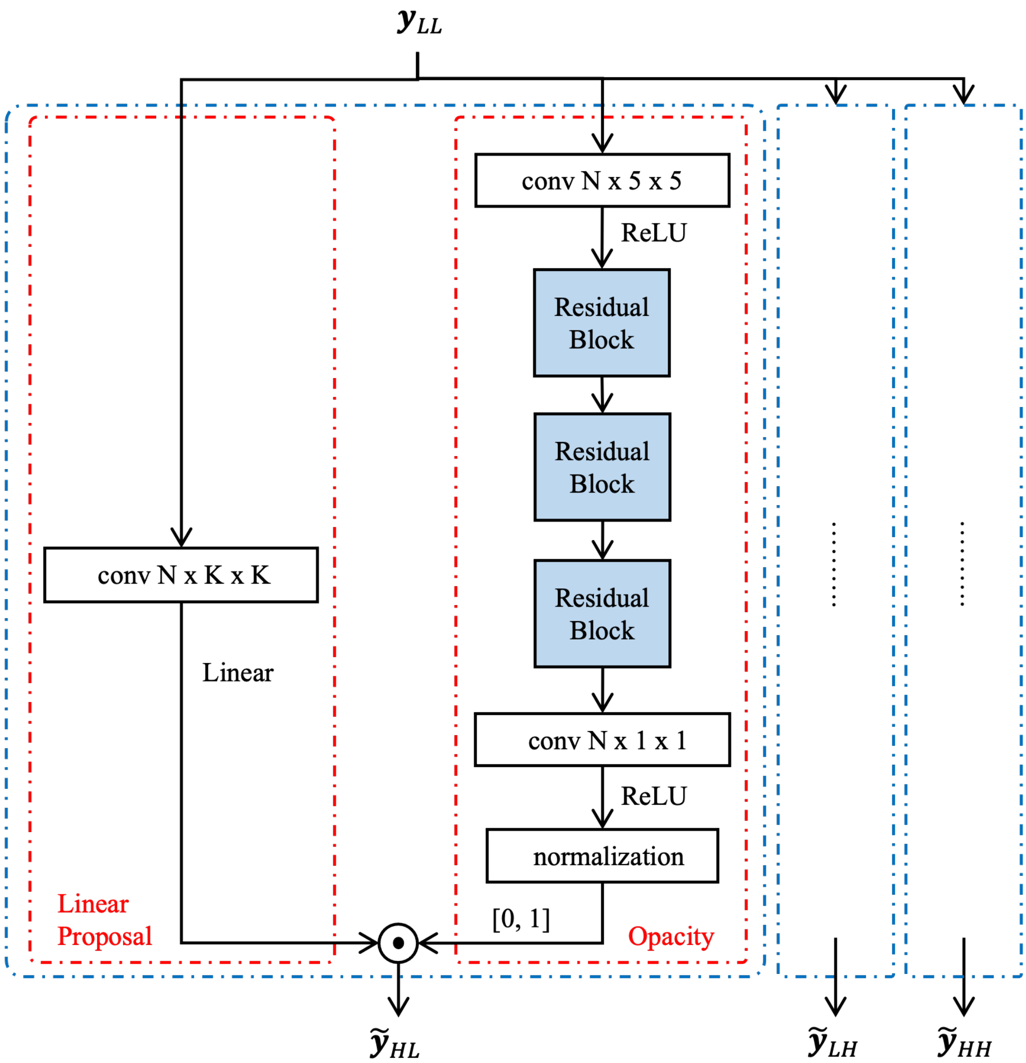


Figure 12.7: The common proposal-opacity network architecture that is utilised for all the lifting steps $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$ within the base wavelet transform seen in Fig. 12.2. The notation *conv* N x K x K represents the convolutional layer with N channels (or filters) and kernel support K x K. The linear proposal branch is chosen to have the same region of support as the opacity branch.



(a)

Figure 12.8: The high-to-low network $\mathcal{T}^{A}_{H2L}$ employed in this chapter; the residual blocks have the same structure as in Fig. 12.7. The notation *conv* N x K x K represents the convolutional layer with N channels (or filters) and kernel support K x K. The linear proposal branch is chosen to have the same region of support as the corresponding opacity branch.



(b)

Figure 12.8: The low-to-high network $\mathcal{T}^{A}_{L2H}$ employed in this chapter; the residual blocks have the same structure as in Fig. 12.7. The notation *conv* N x K x K represents the convolutional layer with N channels (or filters) and kernel support K x K. The linear proposal branch is chosen to have the same region of support as the corresponding opacity branch.

12.3 End-to-end Optimization Framework and Oracle-opacity Training Schedule

12.3.1 End-to-end learning strategy with a backward annealing approach

Now we have explained the lifting structures to investigate and the architectures of all learned lifting networks. The challenge that we immediately encounter is how to jointly train all learned lifting steps across multiple levels of decomposition for rate-distortion objectives, so as to optimize coding efficiency for compression.

In our previous work in Chapter 10, we consider selectively including an aliasing suppression term as part of the rate-distortion training objective, which is given by

$$J(\boldsymbol{\phi}) = \underbrace{\|\mathbf{x} - \hat{\mathbf{x}}(\boldsymbol{\phi})\|^2}_{D} + \lambda_1 \underbrace{\sum_{\beta} \sum_{i \in B_\beta} l_{i,\beta}}_{L} + \lambda_2 \underbrace{\sum_{d} \left\| \tilde{\mathbf{y}}_{LL,d}(\boldsymbol{\phi}) - \tilde{\mathbf{y}}^t_{LL,d} \right\|_2^2}_{\text{aliasing constraint term}} \tag{12.8}$$

where

$$l_{i,\beta} = \log_2 \frac{1}{P_{V_\beta}(q_{i,\beta}; \boldsymbol{\phi})} = \log_2 \frac{1}{\text{Prob}(V_\beta = q_{i,\beta}; \boldsymbol{\phi})} \tag{12.9}$$

In (12.8), the total distortion term D represents the sum of squared errors between the input image $\mathbf{x}$ and its reconstructed counterpart $\hat{\mathbf{x}}$; $\boldsymbol{\phi}$ represents the vector of all network weights. The total coded length term L is the sum of all coded lengths $l_{i,\beta}$, resulting from the coding of quantization indices $q_{i,\beta}$ for all subbands B_β. We write V_β for the random variable from which the quantization indices $q_{i,\beta}$ are drawn; then, the coded length $l_{i,\beta}$ is modelled by (12.9). The LL band aliasing constraint term in (12.8) measures the sum of squared errors between $\tilde{\mathbf{y}}_{LL,d}$ and $\tilde{\mathbf{y}}^t_{LL,d}$ at level d of the wavelet decomposition; $\tilde{\mathbf{y}}_{LL,d}$ is the aliasing prediction from the high-to-low network $\mathcal{T}^A_{H2L}$, while $\tilde{\mathbf{y}}^t_{LL,d}$ denotes the target aliasing model constructed in Section 9.1.

The Lagrange multiplier λ_1 controls the trade-off between distortion D and coded length L, while the other Lagrange multiplier λ_2 controls the level of emphasis on visual quality of reconstructed images at different scales. In Chapter 11, we have demonstrated that λ_2 can have a beneficial impact on perceptual quality of intermediate low-resolution images across different scales, without significant loss in coding efficiency. To simplify the experimental conditions of this chapter, which focus exclusively on coding efficiency, we stick to the case where $\lambda_2 = 0$.

As recognised by existing works in the literature, end-to-end learning targeting the objective in (12.8) requires a good strategy to model the quantization and the entropy coding processes, which are both discontinuous. For this purpose, we choose to adopt the end-to-end optimization framework developed in Chapter 10 for the work in this chapter. This particular end-to-end optimization framework employs a backward annealing approach, which has certain advantages over additive noise approaches [11,188], the straight-through estimator [189,190] and soft-to-hard annealing approaches [191,192], as we have elaborated in Chapter 10.

12.3.2 Oracle-opacity training schedule

As we just explained in Section 12.3.1, the end-to-end optimization framework with backward annealing is sufficient to model the discontinuous quantization and entropy coding processes. However, this optimization framework itself does not necessarily guarantee the convergence to a good solution, because it ultimately relies on stochastic gradient descent. As a result, developing a good training schedule becomes crucial; this is particularly important for the work in this chapter for two reasons.

The first reason is because the lifting structure in Fig. 12.2 becomes *unstructured* once we replace all the fixed lifting steps within the base wavelet transform with neural networks. If we employ random initialization as in Chapter 11 for all learned lifting networks in this chapter, we essentially forfeit the base wavelet transform that otherwise exits in our previous work in Chapter 9, which serves as a stable starting point for training. As a result, instead of improving upon the conventional fixed wavelet transform, we can easily wind up with coding performance that is below the conventional fixed wavelet transform, regardless of the employed network architectures, as we shall see in Section 13.4.1.

The second issue, which highlights the importance of a good training schedule for the work in this chapter, arises from the multiplicative proposal-opacity topology shown in Fig. 12.5. When optimizing this architecture, it is very easy for the optimizer to fall inside a local optima that stochastic gradient descent cannot escape from, and even random initialization does not necessarily solve the problem. To understand this, suppose all the linear filters in the proposal branch correspond to the fixed lifting filter in either (12.2) or (12.3), then the opacity branch that is randomly initialized may not learn anything useful, because the fixed lifting filter is already close to a local optimum for compression. Similarly, if one of the opacities is close to zero during the optimization procedure, then the corresponding proposal can hardly learn anything useful as well. This essentially means that the proposal and the opacity branches are so interdependent that each one ultimately establishes the gradient experienced by the other one during back-propagation.

To address these obstacles, we propose an oracle-opacity training schedule, which consists of three steps as summarised in Table 12.1. Each step within this particular training schedule is trained using the end-to-end optimization framework with backward annealing as developed in Chapter 10, directly targeting the rate-distortion objective in (12.8).

Table 12.1: The Proposed Oracle-opacity Training Schedule

	Proposals	Opacities
Step 1: freeze opacities to train proposals	• one proposal: frozen as the corresponding fixed lifting filter (as in either (12.2) or (12.3)) • the rest $N-1$ proposals: stay trainable	oracle opacities: frozen (obtained off-line)
Step 2: freeze proposals to train opacities	frozen as trained in Step 1	trainable from scratch
Step 3: free all proposals and opacities	trainable, starting from Step 2	trainable, starting from Step 2

To be more specific, *Step 1* employs a distinct set of N opacities that are likely to be useful in order to train N sensible and diverse proposals. These N distinct opacities are frozen throughout training in this step, and are not compatible with any inference machine; we can imagine them to be received externally by some means as part of the compression system. This is completely impossible in practice, because the decoder cannot receive the external knowledge to recover these N distinct opacities from the inference machine. For this reason, we refer to these opacities as *oracle opacities*. Moreover, in *Step 1* we freeze one of the filters in the proposal branch as the corresponding fixed lifting filter (either (12.2) or (12.3)) in the base wavelet transform, whereas the rest $N-1$ proposals stay trainable in this step. In this way, the lifting network is at least capable of discovering the corresponding fixed lifting filter as one possible solution, which already works well in terms of coding efficiency. Subsequently, in *Step 2* the proposals are frozen to train the corresponding opacities from scratch. In *Step 3*, we free both the proposal and the opacity branches for training, starting from the weights found in *Step 2*.

In practice, we first utilize this oracle-opacity training schedule to jointly learn all four lifting networks $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$ for the lifting structure shown in Fig. 12.4(b). This means that all the proposal branches of these networks are jointly learned in *Step 1*, followed by the joint optimization of all the opacity branches in *Step 2*. In *Step 3* we free all four lifting networks $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$ for training. Since the lifting structure in Fig. 12.4(a) adds the new element $\mathcal{T}^A_{H2L}$ to the configuration in Fig. 12.4(b), we then freeze the already trained networks $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$, and only train $\mathcal{T}^A_{H2L}$ using the proposed oracle-opacity training schedule. Afterwards, we free all five lifting networks $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$, $\mathcal{U}^H$ and $\mathcal{T}^A_{H2L}$ for training. The same methodology is adopted for training the lifting structure in Fig. 12.3; we only first train $\mathcal{T}^A_{L2H}$ and then free all lifting networks for training. In order to make fair comparisons in Chapter 13, we also apply the proposed oracle-opacity training schedule to learn the two additional lifting networks in Fig. 12.1; we choose to learn $\mathcal{T}^A_{H2L}$ first and then train the additional $\mathcal{T}^A_{L2H}$.

12.3.3 Proposed method to create oracle opacities

As we have seen in Section 12.3.2, oracle opacities are at the heart of the proposed training schedule in Table 12.1. Since we have revealed the importance of local geometric flow to eliminating redundancy in the wavelet transform in Chapter 7, we expect the pre-defined oracle opacities to explicitly carry orientation information, which can then serve as useful prior knowledge to train the corresponding proposals.

In this chapter, the oracle opacities are derived from the original image as shown in

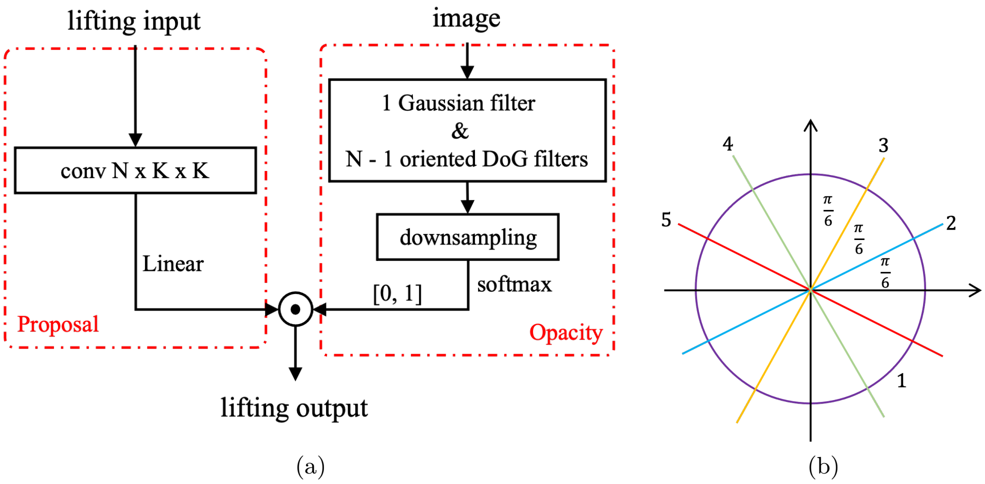


Figure 12.9: (a) illustrates the proposed method to create oracle opacities. (b) exemplifies the orientations of one Gaussian filter and four oriented DoG filters. The notation *conv* N x K x K represents the convolutional layer with N channels (or filters) and kernel support K x K.

Fig. 12.9(a), using a Gaussian filter and a collection of oriented derivative of Gaussian (DoG) filters, as exemplified in Fig. 12.9(b); the results consist of one non-oriented opacity and $N-1$ oriented oracle opacities. The non-oriented oracle opacity always corresponds to the frozen proposal in *Step 1* of the proposed training schedule seen in Table 12.1; this is because it is reasonable to utilize the fixed lifting filter when there is no specific geometric flow. Since the oracle opacities need to be dyadically down-sampled at different levels of decomposition as seen in Fig. 12.9(a), it is important to choose an appropriate standard deviation σ_d for the Gaussian and the DoG filters at each resolution d. In this chapter, σ_d increases exponentially as the decomposition level d goes deeper; that is $\sigma_d = 2^{d-1}, d = 1, 2, 3, \cdots$, where $d = 1$ denotes the finest level of decomposition.

As explained in Section 12.3.2, these pre-defined opacities are oracle values that in general the true opacity branch may not discover from the data available. However, surprisingly the true opacity branch turns out to be able to discover opacities whose utility is comparable with and even slightly superior to the utility of oracle opacities, as we shall see in Section 13.4.1. This indicates that these oracle opacities can indeed serve as a reasonable starting point to train the corresponding proposals.

Chapter 13

Experimental Results: Part II

In this chapter, we first empirically demonstrate the significance and the rationale of the oracle-opacity training schedule proposed in Section 12.3.2. Employing this particular training schedule, we then develop a sequence of experiments to study different aspects (depth, diversity and region of support) of the investigated lifting structures shown in Fig. 12.1, Fig. 12.3, Fig. 12.4(a) and Fig. 12.4(b). Ultimately, these experiments give us guidance on how to strategically deploy neural networks to enhance the base wavelet transform for compression, balancing coding performance with computational complexity and spatial support. Note that the source code of the work in this thesis, along with all the training and testing datasets, are available on GitHub[1].

13.1 Experimental Settings

13.1.1 Training Phase

In this chapter, Keras with TensorFlow backend and the Adam algorithm [194] are employed for training, with 75 image batches comprising 16 patches of size 256 x 256 from the

[1] https://github.com/xinyue-li3/learned-wavelet-like-transforms/

DIV2K image dataset[2]. We employ 5 levels of the wavelet decomposition during training, and aim to jointly train only one set of learned lifting networks, which can be applied to all levels of decomposition as well as a wide range of bit-rates. This goal is explicitly chosen for the reasons elaborated in Section 12.2.3.

As explained in Section 12.3.2, the proposed oracle-opacity training schedule is utilized to jointly learn lifting networks to minimize the objective in (12.8) with $\lambda_2 = 0$; the training progresses from the lifting structure shown in Fig. 12.4(b) to the one seen in Fig. 12.3. This particular training schedule consists of three steps as summarized in Table 12.1, and each step is trained using the end-to-end optimization framework with backward annealing, as developed in Chapter 10. In this chapter, the end-to-end optimization framework with backward annealing is initialized the same way as in Chapter 11, so that each step of the oracle-opacity training schedule requires 1200 epochs to complete.

13.1.2 Testing Phase

In this chapter, four categorized datasets are used during testing, in order to demonstrate the merits of different lifting structures in various scenarios. These four datasets are identical to those in Chapter 11, each of which contains different amount of distinct edges; we refer to them as *Tecnick-Cat1*, *DIV2K-Cat1*, *DIV2K-Cat2* and *CLIC2019-Cat3*. Note that none of these images are used during training.

13.2 Methods Explored

To study the relationship between the depth of lifting structures (i.e., the number of learned lifting steps) and coding performance, we explore the following variations:

- Hybrid(5/3)-5c: the hybrid lifting structure with two learned steps shown in Fig. 12.1, using the LeGall 5/3 bi-orthogonal base wavelet transform, and $N = 5$ for $\mathcal{T}^{A}_{H2L}$ and

[2] https://data.vision.ee.ethz.ch/cvl/DIV2K/

$\mathcal{T}^A_{L2H}$ networks seen in Fig. 12.8.

- Hybrid(9/7)-5c: the hybrid lifting structure with two learned steps shown in Fig. 12.1, using the CDF 9/7 bi-orthogonal base wavelet transform, and $N = 5$ for $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ networks seen in Fig. 12.8.
- Custom-4S-5c: the lifting structure with four learned lifting steps as depicted in Fig. 12.4(b), using $N = 5$ for all lifting networks $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$ and $\mathcal{U}^H$ seen in Fig. 12.7.
- Custom-4MS-5c: the lifting structure with four modified learned lifting steps as depicted in Fig. 12.4(a), using $N = 5$ for all lifting networks $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$, $\mathcal{U}^H$ and $\mathcal{T}^A_{H2L}$ seen in Fig. 12.7 and Fig. 12.8(a).
- Custom-5S-5c: the lifting structure with five learned lifting steps as depicted in Fig. 12.3, using $N = 5$ for all lifting networks $\mathcal{P}^V$, $\mathcal{U}^V$, $\mathcal{P}^H$, $\mathcal{U}^H$, $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ seen in Fig. 12.7 and Fig. 12.8.

To study the merits of increasing the diversity of learned lifting steps (i.e., the number of channels N in each learned lifting network), we employ $N = 9$ instead of $N = 5$ for all above-mentioned lifting structures. The resulting variations are referred to as Hybrid(5/3)-9c, Hybrid(9/7)-9c, Custom-4S-9c, Custom-4MS-9c and Custom-5S-9c.

To study the impacts of spatial support of learned lifting steps on coding efficiency, we explore the following variation:

- Hybrid(9/7)-9c-compact: the hybrid lifting structure shown in Fig. 12.1, using the CDF 9/7 bi-orthogonal base wavelet transform and $N = 9$ for $\mathcal{T}^A_{H2L}$ and $\mathcal{T}^A_{L2H}$ networks, setting all convolutional kernels in Fig. 12.8 to be 3 x 3 and removing the last residual block.

13.3 Evaluation Metrics

Identical to Chapter 11, we consider evaluating the performance of all lifting structures both quantitatively and qualitatively. In terms of quantitative measurements, three widely used metrics are employed – Peak Signal-to-Noise Ratio (PSNR), Structural Similarity (SSIM), Multi-Scale Structural Similarity (MS-SSIM). All these metrics are measured and averaged for each dataset, from which Bjøntegaard (BD) rate savings (in %) are obtained. With regard to qualitative assessment, we provide examples for both the LL bands produced by each lifting structure at different scales and the full reconstructed images in this chapter.

13.4 Results and Discussions

13.4.1 Significance and rationale of the proposed oracle-opacity training schedule

We first empirically study the significance and the rationale of the oracle-opacity training schedule proposed in Section 12.3.2. For the sake of simplicity, we focus only on learning the lifting structure Custom-4S-5c in this section. Further studies on other investigated lifting structures are provided shortly.

In our earlier work, the two additional lifting networks $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$ are randomly initialized and pre-trained, targeting the aliasing model $\tilde{\mathbf{y}}^{t}_{LL,d}$ for the LL band and energy compaction for the detail bands, as seen in Chapter 9. The pre-trained network weights are then utilized as the starting point for the end-to-end optimization with backward annealing, as seen in Chapter 11. However, as we can see in Fig. 13.2, if we adopt the same approach to learn the lifting networks within the base wavelet transform, they can easily wind up exhibiting much worse coding performance than the conventional wavelet transforms across all four datasets for average PSNR, SSIM and MS-SSIM over the range of bit-rates from 0.1bpp to 1.0bpp. We believe that this remarkable result is in large part due to the multiplicative structure within our proposal-opacity network architecture shown in Section 12.2. This observation reinforces the need for a robust training schedule, when studying more broadly the application of neural networks to all lifting steps within the base wavelet transform, as explained in Section 12.3.2.

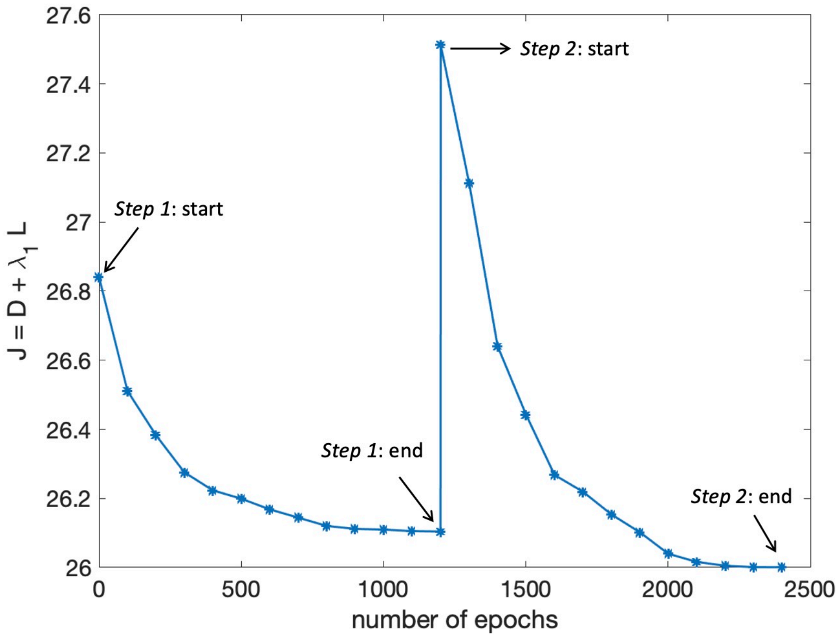


Figure 13.1: The envelope of the rate-distortion objective in (12.8) with $\lambda_2 = 0$ for learning the lifting structure Custom-4S-5c throughout *Step 1* and *Step 2* of the proposed oracle-opacity training schedule.

As we have elaborated in Section 12.3.3, the proposed training schedule employs pre-defined oracle opacities, which can be understood as external knowledge that is received off-line as part of the compression system. These oracle opacities are incompatible with any inference machine; however, they are expected to be useful as guidance to train the corresponding proposals in *Step 1* of the proposed training schedule. One fundamental question then arises as to whether or not the true opacity branch is capable of discovering opacities that are as useful as, or even more useful than these pre-defined oracle opacities, using the actual data available in *Stage 2* of the proposed training schedule.

To address this question, we plot the envelope of the rate-distortion objective in (12.8) with $\lambda_2 = 0$ throughout *Step 1* and *Step 2* of the proposed training schedule, when learning the lifting structure Custom-4S-5c. Fig. 13.1 demonstrates that the rate-distortion value J in (12.8) continuously decreases until reaching a steady point at the end of *Step 1*. At the beginning of *Step 2*, the rate-distortion cost function J rises suddenly; this is expected as we are training the true opacity branches from scratch while freezing the proposals. The cost function J then decreases quickly as the networks learn the true opacities.

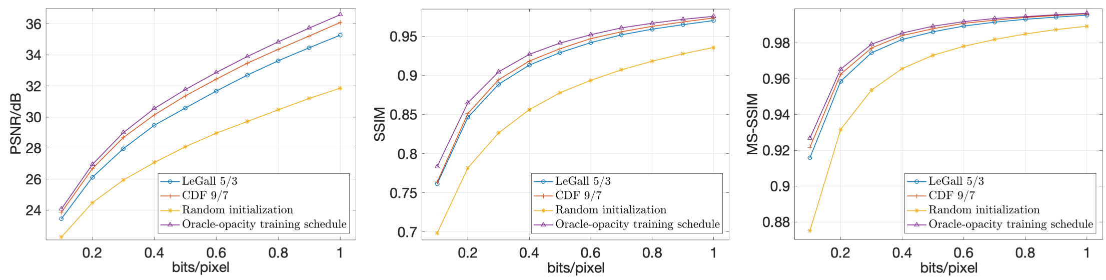
We can see surprisingly that the true opacity branches are indeed capable of discovering opacities whose utility is comparable with and even superior to the utility of oracle opacities, producing a slightly lower rate-distortion value at the end of *Step 2* compared with that of *Step 1*. This observation strongly reinforces the appropriateness of the proposed oracle-opacity training schedule.

13.4.2 Impacts of increasing the depth of lifting structures

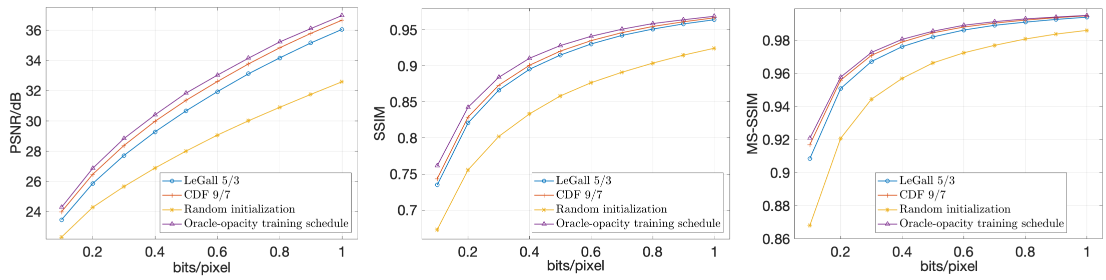
Now we examine the relationship between the depth of lifting structures (i.e., the number of learned lifting steps) and coding performance. As we have mentioned in Section 12.1, each lifting network generally has a substantially larger region of support as well as higher computational complexity than the corresponding fixed lifting steps in the base wavelet transform. Therefore, it is important to study whether or not the benefit in coding performance can be justified by the negative impacts on complexity and spatial support.

Specifically, we examine the performance of the following lifting structures: Hybrid(5/3)-5c, Hybrid(9/7)-5c, Custom-4S-5c, Custom-4MS-5c and Custom-5S-5c. To make fair comparisons, we utilize the proposed oracle-opacity training schedule to learn all these lifting structures, as explained in Section 12.3.2. The BD-rate savings (in %) for average PSNR, SSIM and MS-SSIM over the range of bit-rates from 0.1bpp to 1.0bpp across all four datasets are provided in Table 13.1; the complete rate-distortion curves can be found in Fig. 13.3.

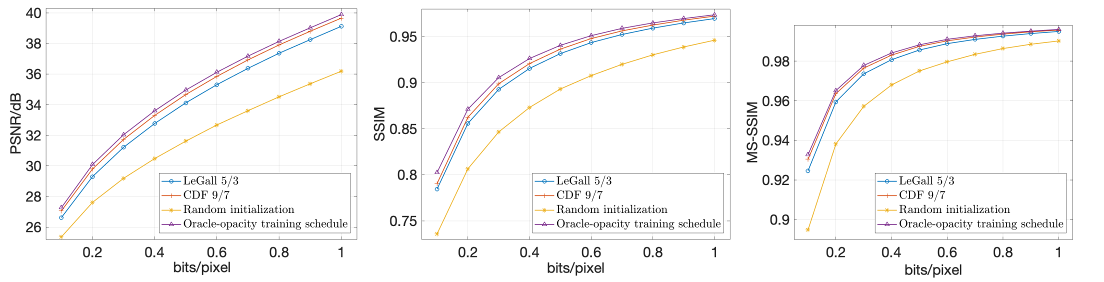
The overall observation is that all the investigated lifting structures with learnable steps ($N = 5$) are capable of improving coding efficiency of the conventional LeGall 5/3 and CDF



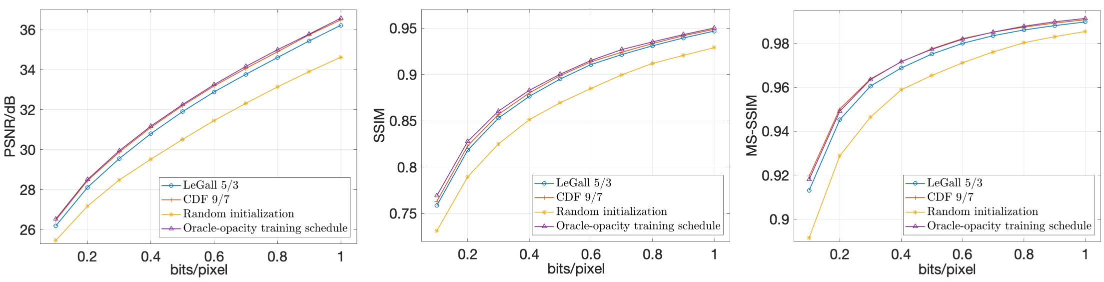
(a) Tecnick-Cat1



(b) DIV2K-Cat1



(c) DIV2K-Cat2



(d) CLIC2019-Cat3

Figure 13.2: Comparisons of the average PSNR, SSIM and MS-SSIM improvements over the LeGall 5/3 and the CDF 9/7 wavelet transforms across each dataset, using pre-training with random initialization followed by end-to-end optimization with backward annealing as in Chapter 11 versus the proposed oracle-opacity training schedule in Chapter 12. We focus only on learning the lifting structure Custom-4S-5c in this comparison.

9/7 wavelet transforms, regardless of the amount of distinct edges presented in images. Although all these lifting structures are optimized to minimize MSE during training, they also work surprisingly well under the SSIM and the MS-SSIM metrics for all datasets. It is also worthwhile to point out that all the investigated lifting structures exhibit higher coding efficiency for datasets that have consistent or distinct edges, such as Tecnick-Cat1 and DIV2K-Cat1 datasets. This reinforces our underlying assumption about geometric flow in Chapter 7, and also aligns with our previous discoveries in Chapter 11.

More interestingly, we can see that increasing the number of learned lifting steps does not lead to significantly higher coding efficiency in terms of PSNR, SSIM and MS-SSIM, whereas computational complexity and region of support increase dramatically as demonstrated in Table 13.4. In addition, employing more learned lifting steps does not actually improve the visual quality of LL bands across different scales, as seen in Fig. 13.6(a)(b)(c), Fig. 13.7(a)(b)(c) and Fig. 13.8(a)(b)(c). Similarly, perceptual quality of the reconstructed images at full resolution is not significantly improved by using more learned lifting steps, as illustrated in Fig. 13.9(d)(e)(f), Fig. 13.10(d)(e)(f) and Fig. 13.11(d)(e)(f).

These observations suggest that to improve the conventional wavelet transform with neural networks, it may not be worthwhile to develop deep fully learned lifting structures. Instead, it appears to be more profitable to augment a larger base wavelet transform with two additional learned lifting steps $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$, as shown in Fig. 12.1; this approach represents competitive coding performance across various evaluation metrics for different types of images, while exhibiting much lower computational complexity and more compact region of support. This is particularly valuable for practical applications, in which computational complexity and spatial support are considered to be very important.

13.4.3 Merits of increasing the diversity of learned lifting networks

Now we move on to study the merits of increasing the diversity (i.e., the number of channels N) of learned lifting networks seen in Fig. 12.7 and Fig. 12.8. It is worthwhile to point out that increasing the diversity of learned lifting networks does not incur any cost to region

Table 13.1: Impacts of increasing the depth of lifting structures on coding efficiency. The table shows BD-rate improvements for PSNR, SSIM and MS-SSIM metrics over the LeGall 5/3 and the CDF 9/7 wavelet transform. Results are obtained with bit-rates between 0.1bpp to 1.0bpp.

		BD-rate for PSNR				
		Hybrid (5/3)-5c	Hybrid (9/7)-5c	Custom -4S-5c	Custom -4MS-5c	Custom -5S-5c
LeGall 5/3	Tecknick-Cat1	−17.4%	−21.9%	−17.3%	−18.4%	−20.3%
	DIV2K-Cat1	−14.4%	−19.6%	−17.1%	−18.3%	−19.7%
	DIV2K-Cat2	−12.5%	−15.9%	−13.4%	−14.2%	−15.6%
	CLIC2019-Cat3	−7.3%	−10.8%	−8.2%	−8.7%	−9.8%
CDF 9/7	Tecknick-Cat1	−6.5%	−11.4%	−6.3%	−7.5%	−9.5%
	DIV2K-Cat1	−3.8%	−9.7%	−7.0%	−8.2%	−9.9%
	DIV2K-Cat2	−3.8%	−7.5%	−4.9%	−5.7%	−7.2%
	CLIC2019-Cat3	−0.5%	−4.2%	−1.5%	−1.8%	−3.2%

		BD-rate for SSIM				
		Hybrid (5/3)-5c	Hybrid (9/7)-5c	Custom -4S-5c	Custom -4MS-5c	Custom -5S-5c
LeGall 5/3	Tecknick-Cat1	−15.5%	−16.9%	−15.8%	−16.7%	−18.6%
	DIV2K-Cat1	−13.8%	−15.7%	−15.1%	−16.4%	−18.4%
	DIV2K-Cat2	−12.8%	−13.8%	−13.6%	−14.3%	−16.1%
	CLIC2019-Cat3	−7.5%	−8.2%	−7.8%	−8.1%	−9.5%
CDF 9/7	Tecknick-Cat1	−10.0%	−11.5%	−10.2%	−11.1%	−13.2%
	DIV2K-Cat1	−8.2%	−10.1%	−9.5%	−10.9%	−12.9%
	DIV2K-Cat2	−6.3%	−7.5%	−7.2%	−7.9%	−9.8%
	CLIC2019-Cat3	−3.3%	−4.1%	−3.6%	−4.0%	−5.4%

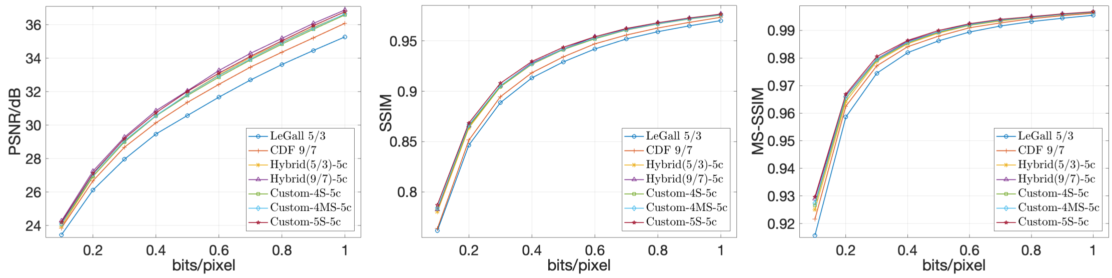
		BD-rate for MS-SSIM				
		Hybrid (5/3)-5c	Hybrid (9/7)-5c	Custom -4S-5c	Custom -4MS-5c	Custom -5S-5c
LeGall 5/3	Tecknick-Cat1	−13.6%	−17.2%	−15.1%	−16.3%	−18.8%
	DIV2K-Cat1	−13.1%	−16.8%	−14.5%	−15.9%	−18.5%
	DIV2K-Cat2	−12.7%	−15.0%	−13.8%	−14.6%	−17.2%
	CLIC2019-Cat3	−8.9%	−11.0%	−8.9%	−9.5%	−12.5%
CDF 9/7	Tecknick-Cat1	−5.1%	−9.0%	−6.7%	−8.0%	−10.6%
	DIV2K-Cat1	−9.9%	−7.5%	−5.1%	−6.6%	−9.5%
	DIV2K-Cat2	−3.4%	−5.9%	−4.7%	−5.5%	−8.3%
	CLIC2019-Cat3	0.6%	−1.7%	0.5%	−0.2%	−3.4%

Table 13.2: Merits of increasing the diversity of lifting structures (i.e., the number of channels N in each learned lifting network) on coding efficiency. The table shows BD-rate improvements for PSNR, SSIM and MS-SSIM metrics over the LeGall 5/3 and the CDF 9/7 wavelet transform. Results are obtained with bit-rates between 0.1bpp to 1.0bpp.

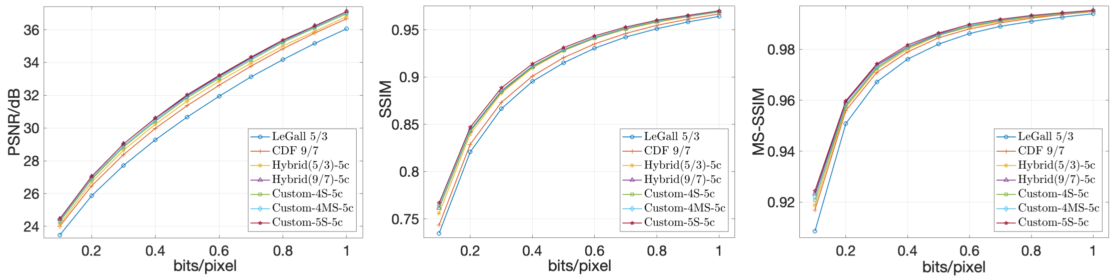
		BD-rate for PSNR				
		Hybrid (5/3)-9c	Hybrid (9/7)-9c	Custom -4S-9c	Custom -4MS-9c	Custom -5S-9c
LeGall 5/3	Tecknick-Cat1	−20.7%	−24.0%	−20.2%	−20.9%	−22.4%
	DIV2K-Cat1	−18.2%	−22.2%	−19.8%	−20.7%	−22.0%
	DIV2K-Cat2	−15.4%	−17.8%	−15.6%	−16.2%	−17.4%
	CLIC2019-Cat3	−8.9%	−12.2%	−9.0%	−9.5%	−10.6%
CDF 9/7	Tecknick-Cat1	−10.1%	−13.7%	−9.5%	−10.2%	−11.9%
	DIV2K-Cat1	−8.1%	−12.6%	−10.0%	−10.9%	−12.4%
	DIV2K-Cat2	−7.0%	−9.7%	−7.3%	−7.9%	−9.2%
	CLIC2019-Cat3	−2.2%	−5.7%	−2.3%	−2.9%	−4.0%

		BD-rate for SSIM				
		Hybrid (5/3)-9c	Hybrid (9/7)-9c	Custom -4S-9c	Custom -4MS-9c	Custom -5S-9c
LeGall 5/3	Tecknick-Cat1	−19.8%	−19.9%	−19.7%	−20.3%	−21.8%
	DIV2K-Cat1	−17.9%	−18.9%	−19.2%	−20.2%	−21.9%
	DIV2K-Cat2	−16.1%	−16.4%	−16.3%	−16.8%	−18.3%
	CLIC2019-Cat3	−9.0%	−10.1%	−8.7%	−9.1%	−10.4%
CDF 9/7	Tecknick-Cat1	−14.5%	−14.5%	−14.4%	−15.0%	−16.6%
	DIV2K-Cat1	−12.5%	−13.4%	−13.9%	−14.9%	−16.7%
	DIV2K-Cat2	−9.8%	−10.1%	−10.1%	−10.5%	−12.1%
	CLIC2019-Cat3	−4.9%	−6.0%	−4.6%	−5.0%	−6.4%

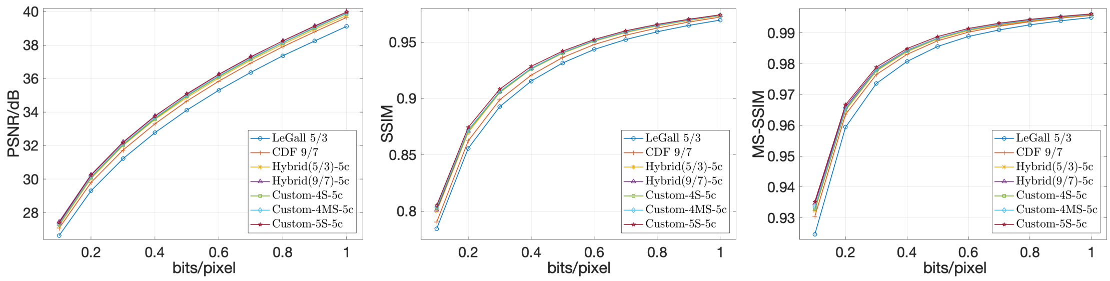
		BD-rate for MS-SSIM				
		Hybrid (5/3)-9c	Hybrid (9/7)-9c	Custom -4S-9c	Custom -4MS-9c	Custom -5S-9c
LeGall 5/3	Tecknick-Cat1	−17.6%	−19.6%	−18.8%	−19.8%	−21.9%
	DIV2K-Cat1	−16.5%	−19.0%	−17.9%	−19.4%	−21.7%
	DIV2K-Cat2	−15.7%	−17.1%	−16.3%	−17.2%	−19.5%
	CLIC2019-Cat3	−10.0%	−12.3%	−9.4%	−10.7%	−13.5%
CDF 9/7	Tecknick-Cat1	−9.5%	−11.5%	−10.7%	−11.8%	−14.1%
	DIV2K-Cat1	−7.2%	−10.0%	−8.9%	−10.5%	−13.0%
	DIV2K-Cat2	−6.7%	−8.2%	−7.4%	−8.3%	−10.8%
	CLIC2019-Cat3	−0.7%	−3.1%	−0.1%	−1.5%	−4.5%



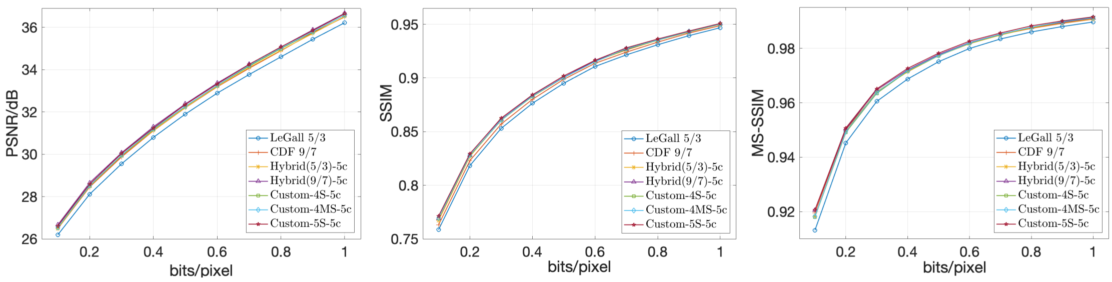
(a) Tecnick-Cat1



(b) DIV2K-Cat1

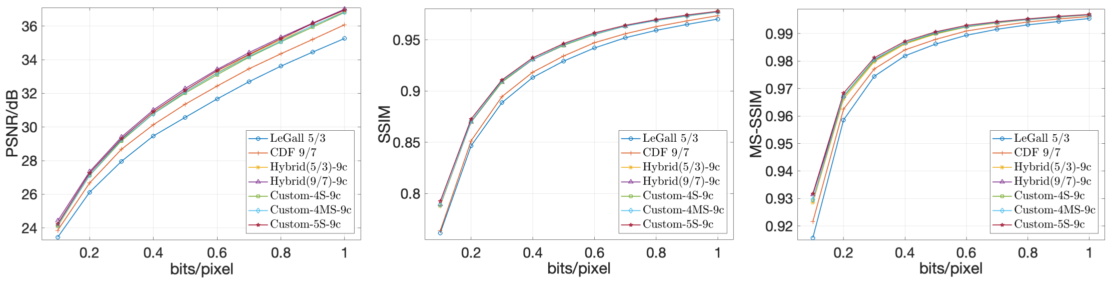


(c) DIV2K-Cat2

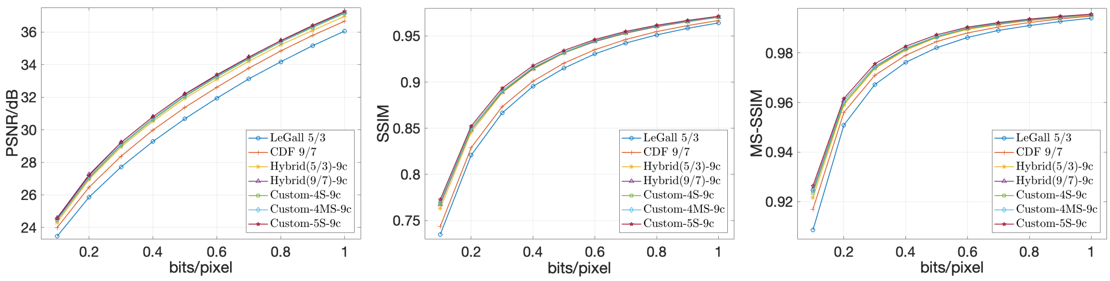


(d) CLIC2019-Cat3

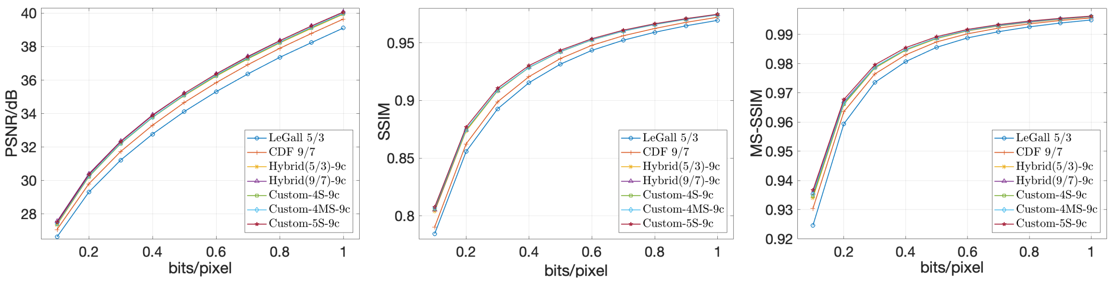
Figure 13.3: Comparisons of the average PSNR, SSIM and MS-SSIM improvements over the LeGall 5/3 and the CDF 9/7 wavelet transforms across each dataset, using lifting structures with different numbers of learned lifting steps – Hybrid(5/3)-5c, Hybrid(9/7)-5c, Custom-4S-5c, Custom-4MS-5c and Custom-5S-5c.



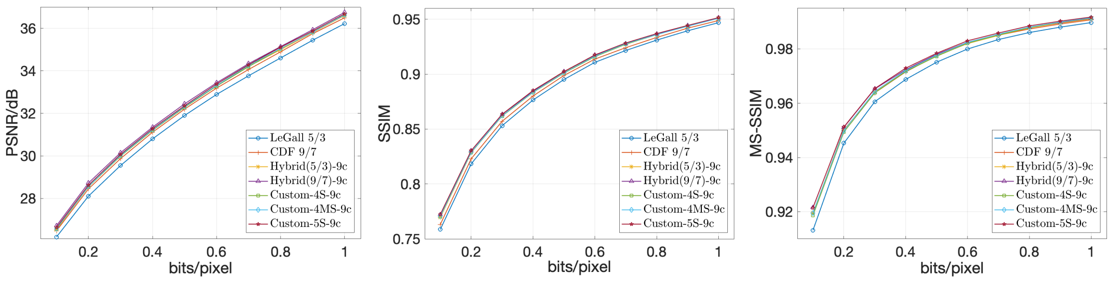
(a) Tecnick-Cat1



(b) DIV2K-Cat1



(c) DIV2K-Cat2



(d) CLIC2019-Cat3

Figure 13.4: Comparisons of the average PSNR, SSIM and MS-SSIM improvements over the LeGall 5/3 and the CDF 9/7 wavelet transforms across each dataset, using lifting structures with more channels N in each learned lifting step – Hybrid(5/3)-9c, Hybrid(9/7)-9c, Custom-4S-9c, Custom-4MS-9c and Custom-5S-9c.

of support, but does increase computational complexity.

In this section, we proceed exactly the same as in Section 13.4.2, but employing $N = 9$ instead of $N = 5$ for all learned networks. These lifting networks are all trained using the proposed oracle-opacity training schedule, as explained in Section 12.3.2. The BD-rate savings (in %) in average PNSR, SSIM and MS-SSIM over the range of bit-rates from 0.1bpp to 1.0bpp is provided in Table 13.2; the complete rate-distortion curves can be found in Fig. 13.4.

Table 13.1 and Table 13.2 reveal that increasing the diversity of learned lifting networks further improves coding efficiency of the corresponding lifting structures, at the cost of additional computational complexity as seen in Table 13.4. More interestingly, we observe that the hybrid lifting structure Hybrid(9/7)-9c exhibits the highest coding efficiency, especially for PSNR metric, amongst other configurations with more learned lifting steps. This configuration also produces better visual quality of LL bands across different scales as compared in Fig. 13.6(d)(e), Fig. 13.7(d)(e) and Fig. 13.8(d)(e), as well as comparable full reconstructed images as compared in Fig. 13.9(g)(h), Fig. 13.10(g)(h) and Fig. 13.11(g)(h).

This observation aligns with the conclusion in Section 13.4.2 that it is more profitable to employ learned $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$ steps to improve the conventional base wavelet transform, rather than developing deep fully learned lifting structures. Moreover, if we can afford additional computational complexity and are only interested in coding efficiency, we can choose to increase the diversity of $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$ networks for higher coding efficiency.

13.4.4 Study on spatial support of learned lifting steps

Compact region of support is one of the fundamental features of the conventional wavelet transform. This feature, however, is damaged by augmenting or replacing the fixed lifting steps that correspond to the base wavelet transform with neural networks, which generally have substantially larger spatial supports. Therefore, it is important to study whether similar coding performance can be achieved using lifting networks with more compact region of support or not.

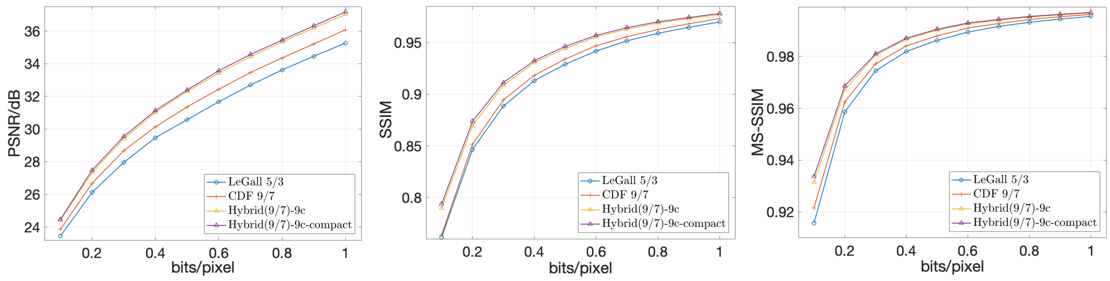
To study this, we start with the hybrid lifting structure Hybrid(9/7)-9c, whose superiority over other configurations has been demonstrated in Section 13.4.3. Specifically, we set all convolutional kernels in $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$ to be 3 x 3, and remove the last residual block shown in Fig. 12.8. The resulting lifting structure Hybrid(9/7)-9c-compact then has significantly smaller region of support (23 x 23 instead of 37 x 37), together with lower computational complexity than Hybrid(9/7)-9c, as demonstrated in Table 13.4.

To evaluate the performance of this configuration, the BD-rate savings (in %) in average PNSR, SSIM and MS-SSIM over the range of bit-rates from 0.1bpp to 1.0bpp is provided in Table 13.3; the complete rate-distortion curves can be found in Fig. 13.5. We can see that competitive (even slightly superior) coding performance can be achieved using more diverse $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$ networks with smaller region of support. This configuration also produces slightly enhanced full reconstructed images as compared in Fig. 13.9(g)(i), Fig. 13.10(g)(i) and Fig. 13.11(g)(i), while maintaining visual quality of LL bands across different resolutions as illustrated in Fig. 13.6(d)(f), Fig. 13.7(d)(f) and Fig. 13.8(d)(f).

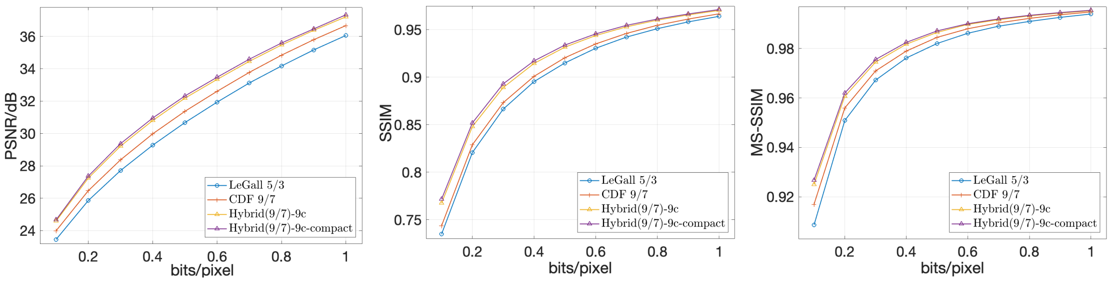
These observations reinforce the statement in Section 12.1 that it is important for a highly scalable compression system to have limited number of non-linearities; otherwise, quantization errors can expand in an uncontrollable way during synthesis.

Table 13.3: Study on spatial support of learned lifting steps on coding efficiency. The table shows BD-rate improvements for PSNR, SSIM and MS-SSIM metrics over the LeGall 5/3 and the CDF 9/7 wavelet transform, with bit-rates between 0.1bpp to 1.0bpp.

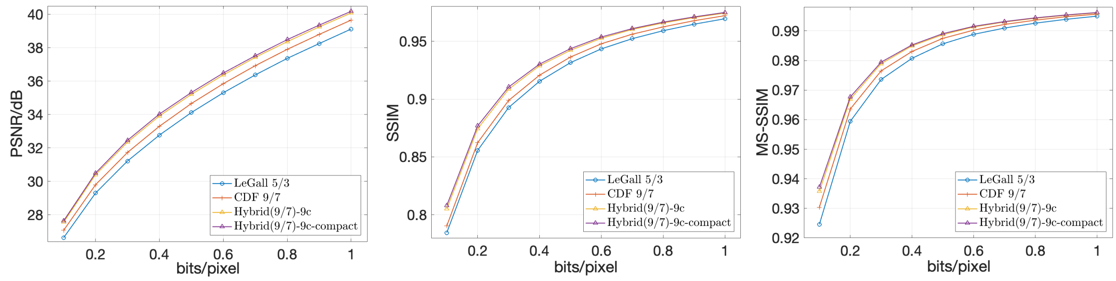
		BD-rate for PSNR	
		Hybrid (9/7)-9c	Hybrid (9/7)-9c-compact
LeGall 5/3	Tecknick-Cat1	−24.0%	−25.6%
	DIV2K-Cat1	−22.2%	−24.0%
	DIV2K-Cat2	−17.9%	−19.4%
	CLIC2019-Cat3	−12.2%	−12.8%
CDF 9/7	Tecknick-Cat1	−13.7%	−15.5%
	DIV2K-Cat1	−12.6%	−14.6%
	DIV2K-Cat2	−9.7%	−11.4%
	CLIC2019-Cat3	−5.7%	−6.4%
		BD-rate for SSIM	
		Hybrid (9/7)-9c	Hybrid (9/7)-9c-compact
LeGall 5/3	Tecknick-Cat1	−19.9%	−22.5%
	DIV2K-Cat1	−18.9%	−21.5%
	DIV2K-Cat2	−16.4%	−18.3%
	CLIC2019-Cat3	−10.1%	−10.9%
CDF 9/7	Tecknick-Cat1	−14.5%	−17.3%
	DIV2K-Cat1	−13.5%	−16.2%
	DIV2K-Cat2	−10.1%	−12.2%
	CLIC2019-Cat3	−6.0%	−6.9%
		BD-rate for MS-SSIM	
		Hybrid (9/7)-9c	Hybrid (9/7)-9c-compact
LeGall 5/3	Tecknick-Cat1	−19.5%	−21.9%
	DIV2K-Cat1	−19.0%	−21.5%
	DIV2K-Cat2	−17.1%	−18.8%
	CLIC2019-Cat3	−12.3%	−13.1%
CDF 9/7	Tecknick-Cat1	−11.5%	−14.1%
	DIV2K-Cat1	−10.0%	−12.7%
	DIV2K-Cat2	−8.2%	−10.0%
	CLIC2019-Cat3	−3.1%	−4.1%



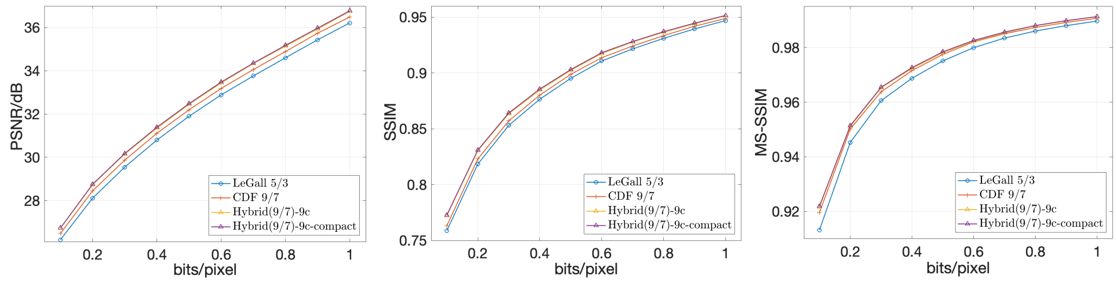
(a) Tecnick-Cat1



(b) DIV2K-Cat1

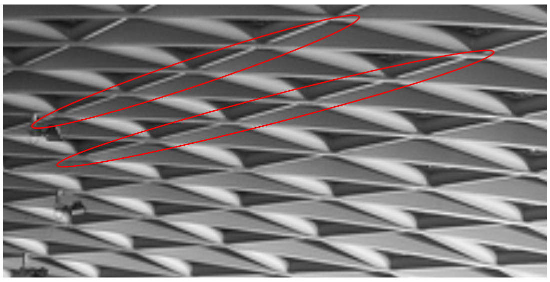


(c) DIV2K-Cat2

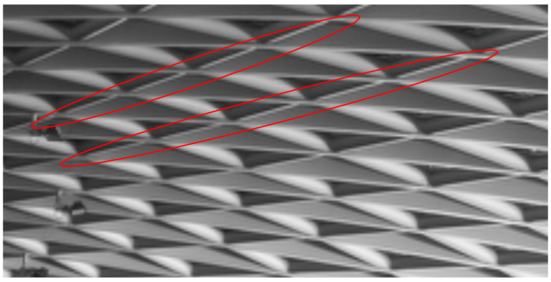


(d) CLIC2019-Cat3

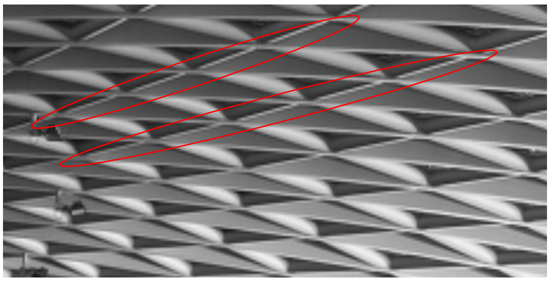
Figure 13.5: Comparisons of the average PSNR, SSIM and MS-SSIM improvements over the LeGall 5/3 and the CDF 9/7 wavelet transforms across each dataset, using lifting structures with different regions of support – Hybrid(9/7)-9c and Hybrid(9/7)-9c-compact.



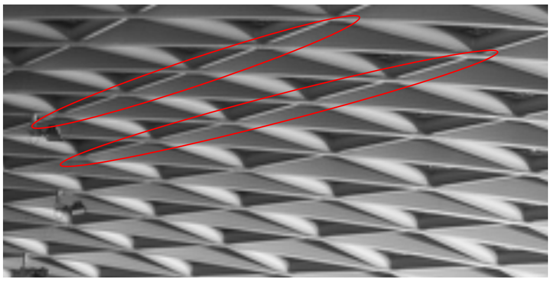
(a) Hybrid(5/3)-5c



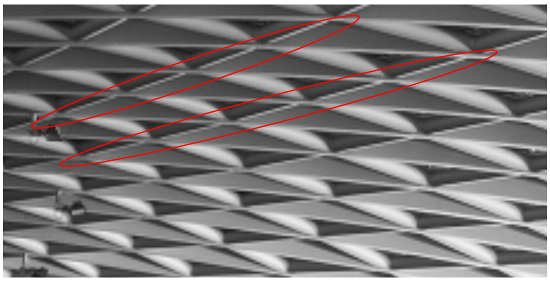
(b) Hybrid(9/7)-5c



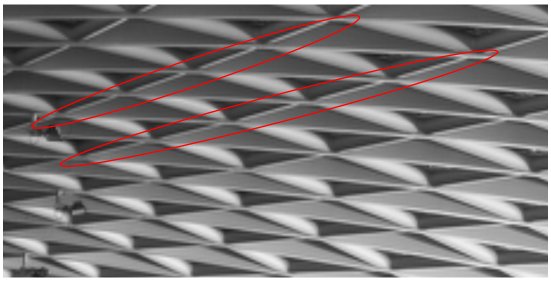
(c) Custom-5S-5c



(d) Hybrid(9/7)-9c

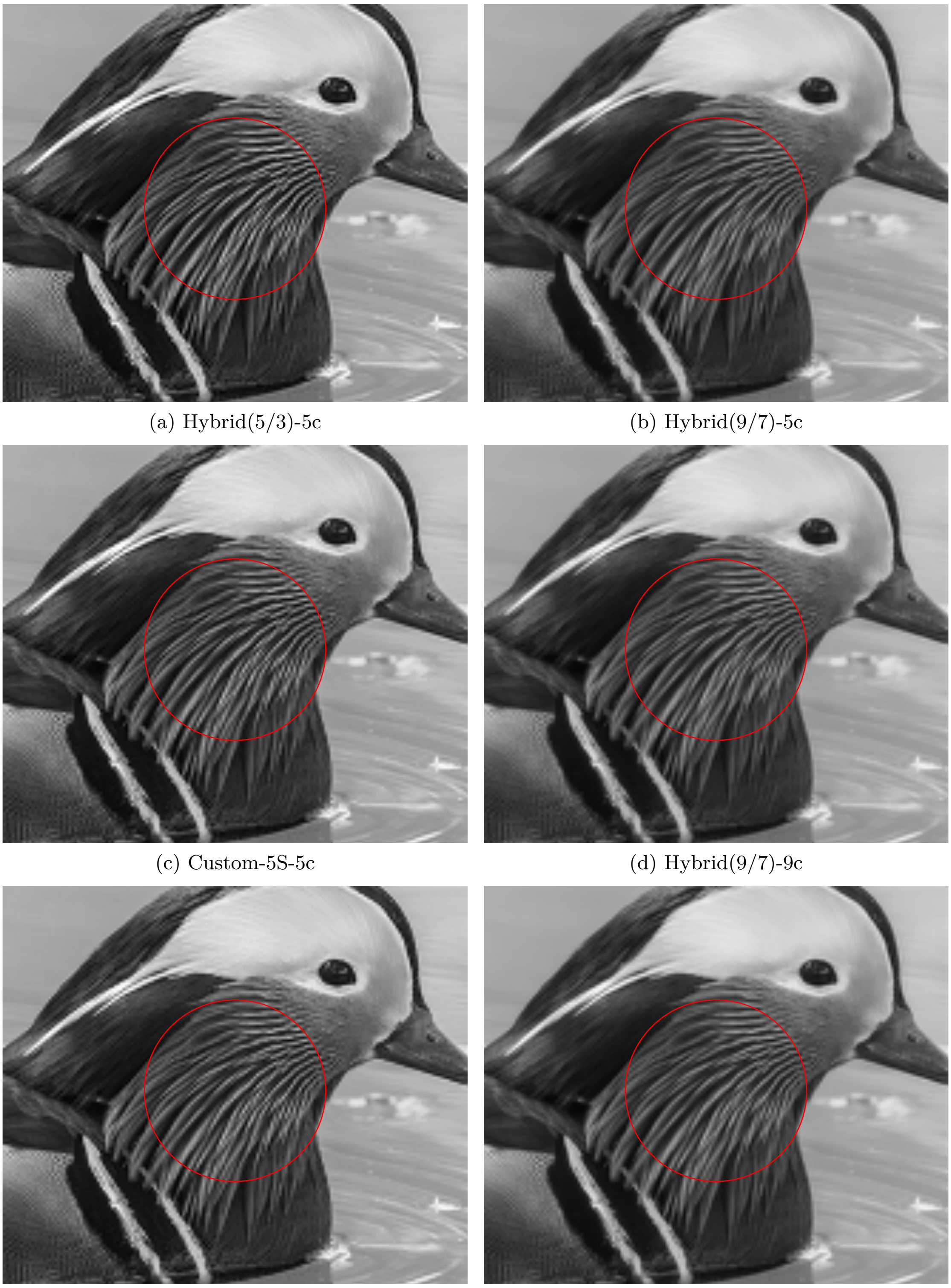


(e) Custom-5S-9c



(f) Hybrid(9/7)-9c-compact

Figure 13.6: Visual quality of the LL bands at the finest resolution obtained from different lifting structures with various numbers of learned steps and diversities.



(a) Hybrid(5/3)-5c

(b) Hybrid(9/7)-5c

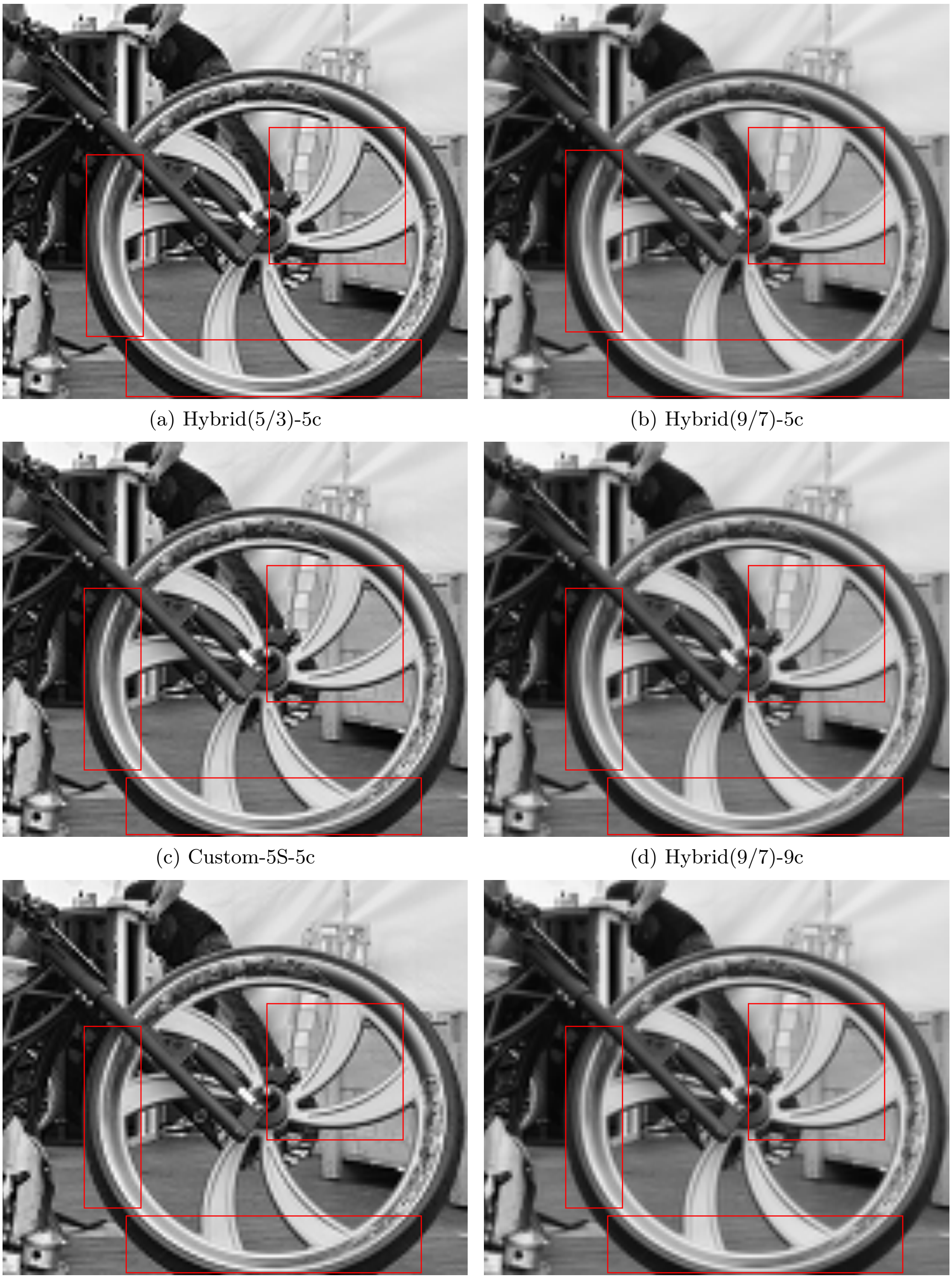
(c) Custom-5S-5c

(d) Hybrid(9/7)-9c

(e) Custom-5S-9c

(f) Hybrid(9/7)-9c-compact

Figure 13.7: Visual quality of the LL bands at the 2^{nd} finest resolution obtained from different lifting structures with various numbers of learned steps and diversities.



(a) Hybrid(5/3)-5c

(b) Hybrid(9/7)-5c

(c) Custom-5S-5c

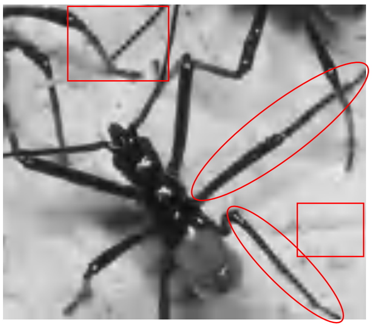
(d) Hybrid(9/7)-9c

(e) Custom-5S-9c

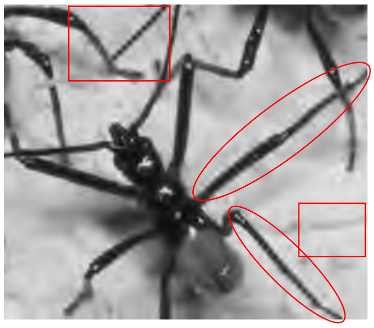
(f) Hybrid(9/7)-9c-compact

Figure 13.8: Visual quality of the LL bands at the 3^{rd} finest resolution obtained from different lifting structures with various numbers of learned steps and diversities.

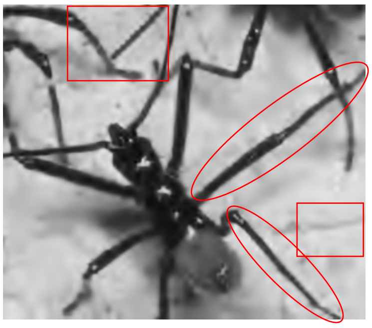
(a) the original image, cropped from image 28 of DIV2K-Cat1 dataset



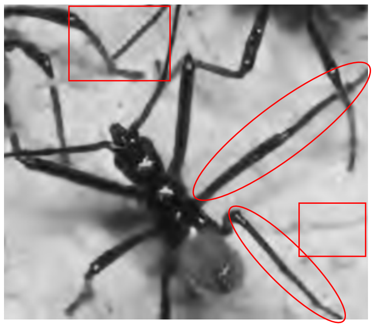
(b) LeGall 5/3 at 0.197bpp, PSNR=29.27dB,SSIM=0.831,MS-SSIM=0.9526



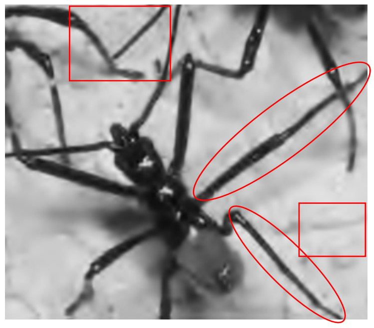
(c) CDF 9/7 at 0.199bpp, PSNR=30.13dB,SSIM=0.841,MS-SSIM=0.9573



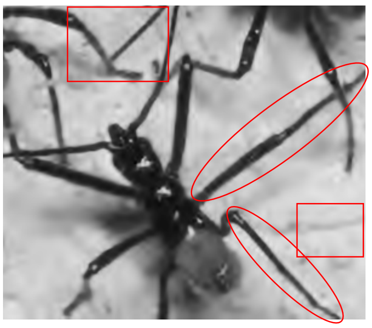
(d) Hybrid(5/3)-5c at 0.196bpp, PSNR=30.39dB,SSIM=0.850,MS-SSIM=0.959



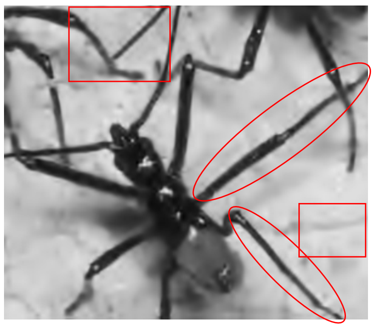
(e) Hybrid(9/7)-5c at 0.198bpp, PSNR=30.84dB,SSIM=0.856,MS-SSIM=0.962



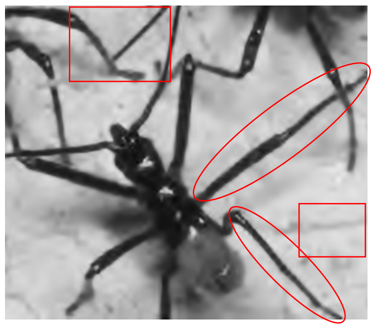
(f) Custom-5S-5c at 0.196bpp, PSNR=30.76dB,SSIM=0.871,MS-SSIM=0.963



(g) Hybrid(9/7)-9c at 0.196bpp, PSNR=30.96dB,SSIM=0.856,MS-SSIM=0.962



(h) Custom-5S-9c at 0.198bpp, PSNR=30.96dB,SSIM=0.876,MS-SSIM=0.965



(i) Hybrid(9/7)-9c-compact, 0.199bpp,PSNR=31.13dB, SSIM=0.864,MS-SSIM=0.966

Figure 13.9: Examples of reconstructed images, obtained from different lifting structures with various numbers of learned steps and diversities. We compare these results with two conventional wavelet transforms: the LeGall 5/3 and the CDF 9/7 wavelet transforms.

(a) the original image, cropped from image 28 of DIV2K-Cat2 dataset

(b) LeGall 5/3 at 0.199bpp, PSNR=29.53dB,SSIM=0.841,MS-SSIM=0.9585

(c) CDF 9/7 at 0.196bpp, PSNR=29.91dB,SSIM=0.845,MS-SSIM=0.9614

(d) Hybrid(5/3)-5c at 0.197bpp, PSNR=30.09dB,SSIM=0.854,MS-SSIM=0.963

(e) Hybrid(9/7)-5c at 0.197bpp, PSNR=30.63dB,SSIM=0.859,MS-SSIM=0.965

(f) Custom-5S-5c at 0.199bpp, PSNR=30.6dB,SSIM=0.872,MS-SSIM=0.9662

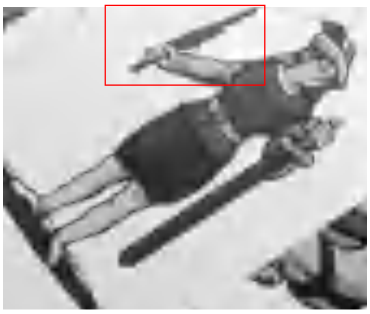
(g) Hybrid(9/7)-9c at 0.197bpp, PSNR=30.8dB, SSIM=0.863,MS-SSIM=0.967

(h) Custom-5S-9c at 0.197bpp, PSNR=30.8dB,SSIM=0.877,MS-SSIM=0.967

(i) Hybrid(9/7)-9c-compact, 0.197bpp,PSNR=30.8dB, SSIM=0.867,MS-SSIM=0.968

Figure 13.10: Examples of reconstructed images, obtained from different lifting structures with various numbers of learned steps and diversities. We compare these results with two conventional wavelet transforms: the LeGall 5/3 and the CDF 9/7 wavelet transforms.

(a) the original image, cropped from image 9 of Tecnick-Cat1 dataset



(b) LeGall 5/3 at 0.298bpp, PSNR=26.97dB,SSIM=0.862,MS-SSIM=0.9694

(c) CDF 9/7 at 0.295bpp, PSNR=27.36dB,SSIM=0.854,MS-SSIM=0.9681

(d) Hybrid(5/3)-5c at 0.295bpp, PSNR=28.19dB,SSIM=0.875,MS-SSIM=0.972

(e) Hybrid(9/7)-5c at 0.295bpp, PSNR=28.41dB,SSIM=0.870,MS-SSIM=0.971

(f) Custom-5S-5c at 0.298bpp, PSNR=28.31dB,SSIM=0.894,MS-SSIM=0.975

(g) Hybrid(9/7)-9c at 0.298bpp, PSNR=28.8dB,SSIM=0.878,MS-SSIM=0.973

(h) Custom-5S-9c at 0.299bpp, PSNR=28.8dB,SSIM=0.899,MS-SSIM=0.973

(i) Hybrid(9/7)-9c-compact, 0.297bpp,PSNR=29.1dB, SSIM=0.881,MS-SSIM=0.973

Figure 13.11: Examples of reconstructed images, obtained from different lifting structures with various numbers of learned steps and diversities. We compare these results with two conventional wavelet transforms: the LeGall 5/3 and the CDF 9/7 wavelet transforms.

13.4.5 Computational Complexity

Finally, we evaluate the computational complexity as well as the region of support [3] associated with different lifting structures that are investigated in this chapter. We can see that the hybrid lifting structure Hybrid(9/7)-9c-compact provides the lowest computational complexity and the most compact region of support. As we have demonstrated in Section 13.4.2, Section 13.4.3 and Section 13.4.4, this configuration also exhibits the highest coding performance and produces the best visual quality, for full-resolution reconstructed images and LL bands, amongst all the investigated structures, including the conventional wavelet transform. Ultimately, this is our recommended approach to enhance the conventional wavelet transform.

Table 13.4: Comparisons of computational complexity and region of support

	Number of Parameters	Region of Support
Hybrid(5/3 or 9/7)-5c	$35K$	37 x 37
Hybrid(5/3 or 9/7)-9c	$63K$	37 x 37
Hybrid(9/7)-9c-compact	$35K$	23 x 23
Custom-4S-5c	$38K$	81 x 81
Custom-4MS-5c	$55K$	101 x 101
Custom-5S-5c	$73K$	117 x 117
Custom-4S-9c	$69K$	81 x 81
Custom-4MS-9c	$98K$	101 x 101
Custom-5S-9c	$133K$	117 x 117

[3] Here region of support refers to the total receptive field of all networks involved in a certain approach.

Chapter 14

Conclusion and Future Directions

14.1 Conclusion on augmenting the conventional wavelet transform

In this thesis, we first propose two networks, the high-to-low and low-to-high networks, as additional lifting steps to augment the conventional wavelet transforms, improving coding efficiency and visual quality of LL bands across multiple levels of decomposition. The high-to-low network serves to clean aliasing and perhaps other redundancies from the low-pass band produced at each successive level of the decomposition, while the low-to-high network aims to further reduce redundancy amongst the detail bands.

The proposal of the high-to-low and low-to-high networks is inspired and guided by a specific theoretical argument related to the opportunity presented by geometric flow and connected to super resolution. Specifically, this argument reveals the role that geometric flow can play in untangling redundant information from the low- and the high-pass sub-bands. Following the ablation study of different network structures, we eventually find that the best investigated network topology does indeed involve banks of optimized linear filters controlled dynamically by an opacity network, as suggested by the underlying theory.

More importantly, the high-to-low and low-to-high networks driven by our hypothesis are compact and with limited non-linearities, allowing high coding efficiency and scalability over a wide range of bit-rates and multiple resolutions in a critically sampled self-similar compression system. This means that all coded *wavelet* coefficients have a relatively small region of influence in the image domain, and there is no need to learn and store separate network weights for each decomposition level and for each bit-rate of interest. In addition, since the structure involves a collection of purely linear filters, these two additional networks come with fairly low computational complexity.

Apart from the networks themselves, we also propose a backward annealing approach to manage the discontinuities in quantization and cost functions during training, so as to jointly train the proposed networks in an end-to-end optimization framework. By employing this backward annealing approach and selectively including the aliasing suppression term in the training objective, we demonstrate that augmenting the conventional wavelet transform with the high-to-low and low-to-high networks achieves up to 17.4% average BD rate saving over the LeGall 5/3 wavelet transform in a wide range of bit-rates. Moreover, the high-to-low and low-to-high networks also manage to reduce aliasing at intermediate resolutions, producing a more visually appealing multi-scale wavelet representation. Overall, the coding efficiency of the proposed hybrid lifting scheme appears to be very competitive with other related works, meanwhile offering other desirable features like resolution and quality scalability and region-of-interest accessibility.

14.2 Conclusion on extensions of neural networks to the base wavelet transform

Built upon the success of the high-to-low and low-to-high networks, we then bear upon a more comprehensive analysis on what can really be achieved by learning-based wavelet-like transforms in a critically sampled self-similar highly scalable compression system. In this comprehensive study, we first consider extending neural networks to all lifting steps that correspond to the base wavelet transform. Rather than replacing the base wavelet transform with an arbitrary sequence of learned lifting steps, we take a more disciplined approach in this thesis, which factorizes the base wavelet transform in the way that each fixed two-dimensional lifting step can be replaced individually with a learnable equivalent. By leveraging the proposed proposal-opacity network topology, the factorized lifting networks have the advantage to discover the conventional wavelet transform as one possible solution during training, which already works well in terms of coding efficiency for compression.

Subsequently, we study the relationship between the depth of lifting structures (i.e. the number of learned lifting steps) and coding performance, considering computational complexity and region of support as well. Furthermore, we analyse the merits of increasing the diversity (i.e. the number of channels in each lifting network) on coding performance, noting that this does not incur any cost to spatial support. In the end, we also study whether or not similar coding performance can be achieved using more compact region of support for learned lifting networks.

Through this comprehensive study, we discover that developing a good training schedule becomes crucial, especially when we extend neural networks to all lifting steps within the base wavelet transform. To facilitate the convergence to a good solution, we propose an oracle-opacity training schedule in addition to the end-to-end optimization framework with backward annealing. This particular training schedule utilizes oracle opacities that are derived externally from source images, so as to initialize and progressively train the multiplicative proposal-opacity lifting networks for optimized rate-distortion performance.

Employing the proposed oracle-opacity training strategy, experimental results suggest that to improve the conventional wavelet transform, it is more profitable to augment a larger base wavelet transform with two additional lifting networks $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$, rather than developing deep fully learnable lifting structures with more lifting steps. If we can afford additional computational complexity and are only interested in coding performance, we recommend to only increase the diversity of $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$ networks for higher coding efficiency. In the scenario where computational complexity is restricted, we recommend to employ more diverse $\mathcal{T}^{A}_{H2L}$ and $\mathcal{T}^{A}_{L2H}$ networks but with more compact region of support;

this can lead to comparable or even slightly better coding performance.

14.3 Future directions

There are many possibilities to improve upon the work in this thesis. We restrict our attention to grayscale images having only one image component in this work; however, it is possible to incorporate learned color transforms with our methods to deal with color images. Moreover, for the work in this thesis, we utilize marginal entropy as the model during training, while the actual coding techniques employed for performance evaluation use conditional arithmetic coding, as mentioned in Section 3.3. Therefore, we can also consider incorporating collections of rich context models with our approach, so that statistical dependencies between quantized sample values can be taken into account during the training process as well. In addition, we can also investigate learned post-processing strategies to further enhance coding performance as well as reconstruction quality of the work in this thesis. It is worthwhile to point out that all of these additional features increases computational complexities and may have an adverse impact on important attributes of the proposed compression scheme, such as quality scalability, resolution scalability and region of interest accessibility, therefore may not be valuable for practical applications.

Furthermore, there are many possibilities to extend the work in this thesis for other signal processing applications. For instance, the proposed method can be utilized for scalable wavelet-based video compression applications, so as to reduce disturbing visual artifacts caused by excessive levels of aliasing from the spatial subband transform. Since the work in this thesis is closely related to inverse image processing problems, we can consider extending the concept of the proposal-opacity topology to denoising, demosaicing and deblurring applications as well.

References

[1] D. S. Taubman and M. W. Marcellin, *JPEG 2000: Image Compression Fundamentals, Standards and Practice.* Norwell, MA, USA: Kluwer Academic Publishers, 2001.

[2] G. K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[3] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (hevc)," in *Integrated circuit and systems, algorithms and architectures.* Springer, 2014, vol. 39, p. 40.

[4] 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.* Springer, 2014, pp. 184–199.

[5] Sonehara, Kawato, Miyake, and Nakane, "Image data compression using a neural network model," in *International 1989 Joint Conference on Neural Networks*, 1989, pp. 35–41 vol.2.

[6] A. Namphol, S. H. Chin, and M. Arozullah, "Image compression with a hierarchical neural network," *IEEE transactions on Aerospace and Electronic Systems*, vol. 32, no. 1, pp. 326–338, 1996.

[7] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[8] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2017, pp. 5306–5314.

[9] D. Minnen, G. Toderici, M. Covell, T. Chinen, N. Johnston, J. Shor, S. J. Hwang, D. Vincent, and S. Singh, "Spatially adaptive image compression using a tiled deep network," in *2017 IEEE International Conference on Image Processing (ICIP)*, 2017, pp. 2796–2800.

[10] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proceedings of the 34th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, D. Precup and Y. W. Teh, Eds., vol. 70. PMLR, 06–11 Aug 2017, pp. 2922–2930. [Online]. Available: https://proceedings.mlr.press/v70/rippel17a.html

[11] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," in *2016 Picture Coding Symposium (PCS)*, 2016, pp. 1–5.

[12] X. Li, A. Naman, and D. Taubman, "Machine-learning based secondary transform for improved image compression in jpeg2000," in *2021 IEEE International Conference on Image Processing (ICIP)*, 2021, pp. 3752–3756.

[13] ISO/IEC, "15444-1:2016 Information technology – JPEG 2000 image coding system: Core coding system," 2016.

[14] ——, "21122-1:2019 Information technology — JPEG XS low-latency lightweight image coding system — Part 1: Core coding system," 2019.

[15] E. J. Candes and D. L. Donoho, "Curvelets: A surprisingly effective nonadaptive representation for objects with edges," Stanford Univ Ca Dept of Statistics, Tech. Rep., 2000.

[16] M. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[17] ——, "The finite ridgelet transform for image representation," *IEEE Transactions on Image Processing*, vol. 12, no. 1, pp. 16–28, 2003.

[18] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on image processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[19] E. Le Pennec and S. Mallat, "Bandelet representations for image compression," in *Proceedings 2001 International Conference on Image Processing (Cat. No. 01CH37205)*, vol. 1. IEEE, 2001, pp. 12–vol.

[20] P. L. Dragotti, V. Velisavljevic, M. Vetterli, and B. Beferull-Lozano, "Discrete directional wavelet bases for image compression," in *Visual Communications and Image Processing 2003*, T. Ebrahimi and T. Sikora, Eds., vol. 5150, International Society for Optics and Photonics. SPIE, 2003, pp. 1287 – 1295. [Online]. Available: https://doi.org/10.1117/12.509905

[21] P. Vandergheynst and J.-F. Gobbers, "Directional dyadic wavelet transforms: Design and algorithms," *IEEE transactions on image processing*, vol. 11, no. 4, pp. 363–372, 2002.

[22] H. Ma, D. Liu, R. Xiong, and F. Wu, "A CNN-based image compression scheme compatible with JPEG-2000," in *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 704–708.

[23] ——, "iWave: CNN-based wavelet-like transform for image compression," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1667–1679, 2020.

[24] T. Dardouri, M. Kaaniche, A. Benazza-Benyahia, and J.-C. Pesquet, "Optimized lifting scheme based on a dynamical fully connected network for image coding," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3329–3333.

[25] ——, "Dynamic neural network for lossy-to-lossless image coding," *IEEE Transactions on Image Processing*, vol. 31, pp. 569–584, 2022.

[26] S. Li, W. Dai, Z. Zheng, C. Li, J. Zou, and H. Xiong, "Reversible autoencoder: A CNN-based nonlinear lifting scheme for image reconstruction," *IEEE Transactions on Signal Processing*, vol. 69, pp. 3117–3131, 2021.

[27] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[28] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[29] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," *arXiv preprint arXiv:1703.00395*, 2017.

[30] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *arXiv preprint arXiv:1511.06085*, 2015.

[31] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. J. Hwang, J. Shor, and G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4385–4393.

[32] N. Ahmed, T. Natarajan, and K. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, 1974.

[33] S. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, 1989.

[34] M. Unser, “On the approximation of the discrete karhunen-loeve transform for stationary processes,” *Signal Processing*, vol. 7, no. 3, pp. 231–249, 1984. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0165168484900021

[35] R. Dony *et al.*, “Karhunen-loeve transform,” *The transform and data compression handbook*, vol. 1, no. 1-34, p. 29, 2001.

[36] J. Roese, “Interframe coding of digital images using transforms and transforms/predictive techniques,” *USCIPI Rep*, vol. 700, 1976.

[37] J. Roese, W. Pratt, and G. Robinson, “Interframe cosine transform image coding,” *IEEE Transactions on communications*, vol. 25, no. 11, pp. 1329–1339, 1977.

[38] A. Netravali and J. Stuller, “Motion-compensated transform coding,” *Bell System Technical Journal*, vol. 58, no. 7, pp. 1703–1718, 1979.

[39] I. Daubechies, “Orthonormal bases of compactly supported wavelets,” *Communications on pure and applied mathematics*, vol. 41, no. 7, pp. 909–996, 1988.

[40] ——, *Ten lectures on wavelets*. SIAM, 1992.

[41] A. Cohen, I. Daubechies, and J.-C. Feauveau, “Biorthogonal bases of compactly supported wavelets,” *Communications on pure and applied mathematics*, vol. 45, no. 5, pp. 485–560, 1992.

[42] C. Redies, J. Hasenstein, and J. Denzler, “Fractal-like image statistics in visual art: similarity to natural scenes,” *Spatial vision*, vol. 21, no. 1-2, pp. 137–148, 2008.

[43] W. Sweldens, “Lifting scheme: a new philosophy in biorthogonal wavelet constructions,” in *Wavelet Applications in Signal and Image Processing III*, A. F. Laine and M. A. Unser, Eds., vol. 2569, International Society for Optics and Photonics. SPIE, 1995, pp. 68 – 79. [Online]. Available: https://doi.org/10.1117/12.217619

[44] ——, “The lifting scheme: A custom-design construction of biorthogonal wavelets,” *Applied and Computational Harmonic Analysis*, vol. 3, no. 2, pp.

186–200, 1996. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1063520396900159

[45] ——, "The lifting scheme: A construction of second generation wavelets," *SIAM Journal on Mathematical Analysis*, vol. 29, no. 2, pp. 511–546, 1998. [Online]. Available: https://doi.org/10.1137/S0036141095289051

[46] I. Daubechies and W. Sweldens, "Factoring wavelet transforms into lifting steps," *Journal of Fourier analysis and applications*, vol. 4, no. 3, pp. 247–269, 1998.

[47] F. Fernandes, R. van Spaendonck, M. J. Coates, and C. S. Burrus, "Directional complex-wavelet processing," in *Wavelet Applications in Signal and Image Processing VIII*, vol. 4119. SPIE, 2000, pp. 536–546.

[48] F. C. Fernandes, R. L. van Spaendonck, and C. S. Burrus, "A new framework for complex wavelet transforms," *IEEE Transactions on signal processing*, vol. 51, no. 7, pp. 1825–1837, 2003.

[49] F. C. Fernandes, M. B. Wakin, and R. G. Baraniuk, "Non-redundant, linear-phase, semi-orthogonal, directional complex wavelets [image/video processing applications]," in *2004 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2. IEEE, 2004, pp. ii–953.

[50] I. W. Selesnick, R. G. Baraniuk, and N. C. Kingsbury, "The dual-tree complex wavelet transform," *IEEE signal processing magazine*, vol. 22, no. 6, pp. 123–151, 2005.

[51] N. G. Kingsbury, "The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters," in *IEEE digital signal processing workshop*, vol. 86. Citeseer, 1998, pp. 120–131.

[52] N. Kingsbury, "A dual-tree complex wavelet transform with improved orthogonality and symmetry properties," in *Proceedings 2000 international conference on image processing (Cat. No. 00CH37101)*, vol. 2. IEEE, 2000, pp. 375–378.

[53] N. Mehrseresht and D. Taubman, “Spatially continuous orientation adaptive discrete packet wavelet decomposition for image compression,” in *2006 International Conference on Image Processing*, 2006, pp. 1593–1596.

[54] W. Ding, F. Wu, and S. Li, “Lifting-based wavelet transform with directionally spatial prediction,” in *Picture Coding Symposium*, vol. 62, 2004, pp. 291–294.

[55] D. Taubman and A. Zakhor, “Orientation adaptive subband coding of images,” in *1993 IEEE International Symposium on Circuits and Systems*, 1993, pp. 271–274 vol.1.

[56] P. Carre, E. Andres, and C. Fernandez-Maloigne, “Discrete rotation for directional orthogonal wavelet packets,” in *Proceedings 2001 International Conference on Image Processing (Cat. No.01CH37205)*, vol. 2, 2001, pp. 257–260 vol.2.

[57] C.-N. Zhang and X. Wu, “A hybrid approach of wavelet packet and directional decomposition for image compression,” in *Engineering Solutions for the Next Millennium. 1999 IEEE Canadian Conference on Electrical and Computer Engineering (Cat. No.99TH8411)*, vol. 2, 1999, pp. 755–760 vol.2.

[58] D. Wang, L. Zhang, and A. Vincent, “Curved wavelet transform and overlapped extension for image coding,” in *2004 International Conference on Image Processing, 2004. ICIP’04.*, vol. 2. IEEE, 2004, pp. 1273–1276.

[59] R. Gray, “Vector quantization,” *IEEE ASSP Magazine*, vol. 1, no. 2, pp. 4–29, 1984.

[60] N. Nasrabadi and R. King, “Image coding using vector quantization: a review,” *IEEE Transactions on Communications*, vol. 36, no. 8, pp. 957–971, 1988.

[61] S. Arya and D. Mount, “Algorithms for fast vector quantization,” in *[Proceedings] DCC ‘93: Data Compression Conference*, 1993, pp. 381–390.

[62] M. Marcellin and T. Fischer, “Trellis coded quantization of memoryless and gauss-markov sources,” *IEEE Transactions on Communications*, vol. 38, no. 1, pp. 82–93, 1990.

[63] T. Fischer and M. Wang, "Entropy-constrained trellis-coded quantization," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 415–426, 1992.

[64] J. Kasner, M. Marcellin, and B. Hunt, "Universal trellis coded quantization," *IEEE Transactions on Image Processing*, vol. 8, no. 12, pp. 1677–1687, 1999.

[65] A. Krishnamurthy, S. Ahalt, D. Melton, and P. Chen, "Neural networks for vector quantization of speech and images," *IEEE Journal on Selected Areas in Communications*, vol. 8, no. 8, pp. 1449–1457, 1990.

[66] Y. Zhou, S.-M. Moosavi-Dezfooli, N.-M. Cheung, and P. Frossard, "Adaptive quantization for deep neural network," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, Apr. 2018. [Online]. Available: https://ojs.aaai.org/index.php/AAAI/article/view/11623

[67] A. T. Elthakeb, P. Pilligundla, A. Yazdanbakhsh, S. Kinzer, and H. Esmaeilzadeh, "Releq: A reinforcement learning approach for deep quantization of neural networks," *CoRR*, vol. abs/1811.01704, 2018. [Online]. Available: http://arxiv.org/abs/1811.01704

[68] C. E. Shannon, "A mathematical theory of communication," *The Bell System Technical Journal*, vol. 27, no. 3, pp. 379–423, 1948.

[69] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," 2018. [Online]. Available: https://arxiv.org/abs/1809.10452

[70] Z. Guo, Y. Wu, R. Feng, Z. Zhang, and Z. Chen, "3-d context entropy model for improved practical image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[71] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, "End-to-end learnt image compression via non-local attention optimization and improved context modeling," *IEEE Transactions on Image Processing*, vol. 30, pp. 3179–3191, 2021.

[72] M. Li, K. Zhang, J. Li, W. Zuo, R. Timofte, and D. Zhang, "Learning context-based nonlocal entropy modeling for image compression," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–14, 2021.

[73] T. Berger, "Rate-distortion theory," *Wiley Encyclopedia of Telecommunications*, 2003.

[74] X. Li, A. Naman, and D. Taubman, "Adaptive secondary transform for improved image coding efficiency in jpeg2000," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 1216–1220.

[75] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[76] S. CCITT, "Video codec for audiovisual services at p$*$ 64 kbit/s recommendation h. 261," 1993.

[77] M. Whybray and W. Ellis, "H. 263-video coding recommendation for pstn videophone and multimedia," 1995.

[78] I. Rec, "H. 262| iso/iec 13818-2," *Information technology—Generic coding of moving pictures and associated audio information—Video*, 2000.

[79] G. J. Sullivan, P. N. Topiwala, and A. Luthra, "The h. 264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions," *Applications of Digital Image Processing XXVII*, vol. 5558, pp. 454–474, 2004.

[80] L. Fan, S. Ma, and F. Wu, "Overview of avs video standard," in *2004 IEEE International Conference on Multimedia and Expo (ICME)(IEEE Cat. No. 04TH8763)*, vol. 1. IEEE, 2004, pp. 423–426.

[81] D. Le Gall, "Mpeg: A video compression standard for multimedia applications," *Communications of the ACM*, vol. 34, no. 4, pp. 46–58, 1991.

[82] C. Fogg, D. J. LeGall, J. L. Mitchell, and W. B. Pennebaker, *MPEG video compression standard*. Springer Science & Business Media, 2007.

[83] A. Tarantola, *Inverse problem theory and methods for model parameter estimation.* SIAM, 2005.

[84] D. C. Youla and H. Webb, “Image restoration by the method of convex projections: Part 1 teory,” *IEEE transactions on medical imaging*, vol. 1, no. 2, pp. 81–94, 1982.

[85] H. Stark and P. Oskoui, “High-resolution image recovery from image-plane arrays, using convex projections,” *JOSA A*, vol. 6, no. 11, pp. 1715–1726, 1989.

[86] M. I. Sezan and H. J. Trussell, “Use of a priori knowledge in multispectral image restoration,” in *International Conference on Acoustics, Speech, and Signal Processing,*. IEEE, 1989, pp. 1429–1432.

[87] A. Zakhor, “Iterative procedures for reduction of blocking effects in transform image coding,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 2, no. 1, pp. 91–95, 1992.

[88] Y. Yang, N. P. Galatsanos, and A. K. Katsaggelos, “Regularized reconstruction to reduce blocking artifacts of block discrete cosine transform compressed images,” *IEEE transactions on circuits and systems for video technology*, vol. 3, no. 6, pp. 421–432, 1993.

[89] ——, “Projection-based spatially adaptive reconstruction of block-transform compressed images,” *IEEE transactions on image processing*, vol. 4, no. 7, pp. 896–908, 1995.

[90] Y. Yang and N. P. Galatsanos, “Removal of compression artifacts using projections onto convex sets and line process modeling,” *IEEE Transactions on Image Processing*, vol. 6, no. 10, pp. 1345–1357, 1997.

[91] C. Weerasinghe, A. W.-C. Liew, and H. Yan, “Artifact reduction in compressed images based on region homogeneity constraints using the projection onto convex sets algorithm,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 891–897, 2002.

[92] J. J. Zou, H. Yan, and D. C. Levy, "Reducing artifacts in block-coded images using adaptive constraints," *Optical Engineering*, vol. 42, no. 10, pp. 2900–2911, 2003.

[93] X. Gan, A. W.-C. Liew, and H. Yan, "A smoothness constraint set based on local statistics of bdct coefficients for image postprocessing," *Image and Vision Computing*, vol. 23, no. 8, pp. 731–737, 2005.

[94] L. Atzori, G. Ginesu, and A. Raccis, "Jpeg2000-coded image error concealment exploiting convex sets projections," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 487–498, 2005.

[95] S. Liu, J. Chen, Y. Ai, and S. Rahardja, "An optimized quantization constraints set for image restoration and its gpu implementation," *IEEE Transactions on Image Processing*, vol. 29, pp. 6043–6053, 2020.

[96] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE signal processing magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[97] A. Sandryhaila and J. M. Moura, "Discrete signal processing on graphs: Frequency analysis," *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, 2014.

[98] A. Kheradmand and P. Milanfar, "A general framework for kernel similarity-based image denoising," in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 415–418.

[99] W. Hu, X. Li, G. Cheung, and O. Au, "Depth map denoising using graph-based transform and group sparsity," in *2013 IEEE 15th international workshop on multimedia signal processing (MMSP)*. IEEE, 2013, pp. 001–006.

[100] A. Kheradmand and P. Milanfar, "A general framework for regularized, similarity-based image restoration," *IEEE Transactions on Image Processing*, vol. 23, no. 12, pp. 5136–5151, 2014.

[101] P. Wan, G. Cheung, D. Florencio, C. Zhang, and O. C. Au, "Image bit-depth enhancement via maximum a posteriori estimation of ac signal," *IEEE Transactions on Image Processing*, vol. 25, no. 6, pp. 2896–2909, 2016.

[102] X. Liu, G. Cheung, X. Wu, and D. Zhao, "Inter-block consistent soft decoding of jpeg images with sparsity and graph-signal smoothness priors," in *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2015, pp. 1628–1632.

[103] W. Hu, G. Cheung, and M. Kazui, "Graph-based dequantization of block-compressed piecewise smooth images," *IEEE Signal Processing Letters*, vol. 23, no. 2, pp. 242–246, 2015.

[104] X. Liu, G. Cheung, X. Wu, and D. Zhao, "Random walk graph laplacian-based smoothness prior for soft decoding of jpeg images," *IEEE Transactions on Image Processing*, vol. 26, no. 2, pp. 509–524, 2016.

[105] S. I. Young, A. T. Naman, R. K. Mathew, and D. Taubman, "Optimized decoding of images based on generalized graph laplacians," in *2016 Picture Coding Symposium (PCS)*. IEEE, 2016, pp. 1–5.

[106] S. I. Young, A. T. Naman, and D. Taubman, "Cogl: Coefficient graph laplacians for optimized jpeg image decoding," *IEEE Transactions on Image Processing*, vol. 28, no. 1, pp. 343–355, 2018.

[107] M. Elad and A. Feuer, "Restoration of a single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Transactions on Image Processing*, vol. 6, no. 12, pp. 1646–1658, 1997.

[108] N. Nguyen, P. Milanfar, and G. Golub, "Efficient generalized cross-validation with applications to parametric image restoration and resolution enhancement," *IEEE Transactions on image processing*, vol. 10, no. 9, pp. 1299–1308, 2001.

[109] R. R. Schultz and R. L. Stevenson, "A bayesian approach to image expansion for improved definition," *IEEE Transactions on Image Processing*, vol. 3, no. 3, pp. 233–242, 1994.

[110] T. Kasetkasem, M. K. Arora, and P. K. Varshney, “Super-resolution land cover mapping using a markov random field based approach,” *Remote Sensing of Environment*, vol. 96, no. 3-4, pp. 302–314, 2005.

[111] P. Cheeseman, B. Kanefsky, R. Kraft, J. Stutz, and R. Hanson, “Super-resolved surface reconstruction from multiple images,” in *Maximum entropy and bayesian methods*. Springer, 1996, pp. 293–308.

[112] R. R. Schultz and R. L. Stevenson, “Extraction of high-resolution frames from video sequences,” *IEEE transactions on image processing*, vol. 5, no. 6, pp. 996–1011, 1996.

[113] 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.

[114] 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.

[115] 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.

[116] R. Tsai, “Multiframe image restoration and registration,” *Advance Computer Visual and Image Processing*, vol. 1, pp. 317–339, 1984.

[117] A. Yuille and D. Kersten, “Vision as bayesian inference: analysis by synthesis,” *Trends in cognitive sciences*, vol. 10, no. 7, pp. 301–308, 2006.

[118] C. G. Bell, H. Fujisaki, J. M. Heinz, K. N. Stevens, and A. S. House, “Reduction of speech spectra by analysis-by-synthesis techniques,” *The Journal of the Acoustical Society of America*, vol. 33, no. 12, pp. 1725–1736, 1961.

[119] E. B. George and M. J. Smith, “Analysis-by-synthesis overlap-add sinusoidal modeling applied to the analysis and synthesis of musical tones,” *Journal of the Audio Engineering Society*, vol. 40, no. 6, pp. 497–516, 1992.

[120] ——, “Speech analysis synthesis and modification using an analysis-by-synthesis overlap-add sinusoidal model,” *IEEE transactions on speech and audio processing*, vol. 5, no. 5, pp. 389–406, 1997.

[121] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[122] S. Sharma, S. Sharma, and A. Athaiya, “Activation functions in neural networks,” *towards data science*, vol. 6, no. 12, pp. 310–316, 2017.

[123] T. M. Mitchell, “Artificial neural networks,” *Machine learning*, vol. 45, no. 81, p. 127, 1997.

[124] G. W. Cottrell, “Image compression by back-propagation: An example of extensional programming,” *Advances in cognitive science*, vol. 3, pp. 208–240, 1988.

[125] S. Carrato and S. Marsi, “Parallel structure based on neural networks for image compression,” *Electronics letters*, vol. 12, no. 28, pp. 1152–1153, 1992.

[126] S. A. Dianat, N. M. Nasrabadi, and S. Venkataraman, “A non-linear predictor for differential pulse-code encoder (dpcm) using artificial neural networks,” in *Acoustics, Speech, and Signal Processing, IEEE International Conference on.* IEEE Computer Society, 1991, pp. 2793–2794.

[127] J. Daugman, “Complete discrete 2-d gabor transforms by neural networks for image analysis and compression,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 36, no. 7, pp. 1169–1179, 1988.

[128] H. Abbas and M. Fahmy, “Neural model for karhunen-loeve transform with application to adaptive image compression,” *IEE Proceedings I (Communications, Speech and Vision)*, vol. 140, no. 2, pp. 135–143, 1993.

[129] R. D. Dony and S. Haykin, “Neural network approaches to image compression,” *Proceedings of the IEEE*, vol. 83, no. 2, pp. 288–303, 1995.

[130] S. Albawi, T. A. Mohammed, and S. Al-Zawi, "Understanding of a convolutional neural network," in *2017 International Conference on Engineering and Technology (ICET)*, 2017, pp. 1–6.

[131] C.-C. J. Kuo, "Understanding convolutional neural networks with a mathematical model," *Journal of Visual Communication and Image Representation*, vol. 41, pp. 406–413, 2016.

[132] M. Liu, J. Shi, Z. Li, C. Li, J. Zhu, and S. Liu, "Towards better analysis of deep convolutional neural networks," *IEEE Transactions on Visualization and Computer Graphics*, vol. 23, no. 1, pp. 91–100, 2017.

[133] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," *CoRR*, vol. abs/1511.08458, 2015. [Online]. Available: http://arxiv.org/abs/1511.08458

[134] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2285–2294.

[135] H. Lee and H. Kwon, "Going deeper with contextual cnn for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 26, no. 10, pp. 4843–4855, 2017.

[136] Y. Dong, Q. Liu, B. Du, and L. Zhang, "Weighted feature fusion of convolutional neural network and graph attention network for hyperspectral image classification," *IEEE Transactions on Image Processing*, vol. 31, pp. 1559–1572, 2022.

[137] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," *Advances in neural information processing systems*, vol. 28, 2015.

[138] Z. Cai and N. Vasconcelos, "Cascade r-cnn: Delving into high quality object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6154–6162.

[139] H.-C. Shin, H. R. Roth, M. Gao, L. Lu, Z. Xu, I. Nogues, J. Yao, D. Mollura, and R. M. Summers, “Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning,” *IEEE transactions on medical imaging*, vol. 35, no. 5, pp. 1285–1298, 2016.

[140] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[141] J. Ballé, V. Laparra, and E. P. Simoncelli, “Density modeling of images using a generalized normalization transformation,” *arXiv preprint arXiv:1511.06281*, 2015.

[142] D. Minnen, J. Ballé, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” *Advances in neural information processing systems*, vol. 31, 2018.

[143] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, “Learning convolutional networks for content-weighted image compression,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3214–3223.

[144] J. Lee, S. Cho, and S.-K. Beack, “Context-adaptive entropy model for end-to-end optimized image compression,” *arXiv preprint arXiv:1809.10452*, 2018.

[145] M. Li, K. Ma, J. You, D. Zhang, and W. Zuo, “Efficient and effective context-based convolutional entropy modeling for image compression,” *IEEE Transactions on Image Processing*, vol. 29, pp. 5900–5911, 2020.

[146] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, “End-to-end learnt image compression via non-local attention optimization and improved context modeling,” *IEEE Transactions on Image Processing*, vol. 30, pp. 3179–3191, 2021.

[147] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, “Checkerboard context model for efficient learned image compression,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 14 771–14 780.

[148] Z. Guo, Z. Zhang, R. Feng, and Z. Chen, "Causal contextual prediction for learned image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 4, pp. 2329–2341, 2022.

[149] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, "Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 5718–5727.

[150] C. Cai, L. Chen, X. Zhang, and Z. Gao, "End-to-end optimized roi image compression," *IEEE Transactions on Image Processing*, vol. 29, pp. 3442–3457, 2020.

[151] H. Liu, T. Chen, Q. Shen, T. Yue, and Z. Ma, "Deep image compression via end-to-end learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2575–2578.

[152] L. Cavigelli, P. Hager, and L. Benini, "Cas-cnn: A deep convolutional neural network for image compression artifact suppression," in *2017 International Joint Conference on Neural Networks (IJCNN)*, 2017, pp. 752–759.

[153] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Energy compaction-based image compression using convolutional autoencoder," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 860–873, 2020.

[154] C. Aytekin, X. Ni, F. Cricri, J. Lainema, E. Aksu, and M. Hannuksela, "Block-optimized variable bit rate neural image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2551–2554.

[155] Z. Tang and L. Luo, "Compression artifact removal using multi-scale reshuffling convolutional network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2567–2570.

[156] P. Akyazi and T. Ebrahimi, "Learning-based image compression using convolutional autoencoder and wavelet decomposition," in *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, no. CONF, 2019.

[157] D. Mishra, S. K. Singh, and R. K. Singh, "Wavelet-based deep auto encoder-decoder (wdaed)-based image compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 4, pp. 1452–1462, 2020.

[158] S. Ayzik and S. Avidan, "Deep image compression using decoder side information," in *European Conference on Computer Vision.* Springer, 2020, pp. 699–714.

[159] L. R. Medsker and L. Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, pp. 64–67, 2001.

[160] S. Grossberg, "Recurrent neural networks," *Scholarpedia*, vol. 8, no. 2, p. 1888, 2013.

[161] K. Doya, "Bifurcations of recurrent neural networks in gradient descent learning," *IEEE Transactions on neural networks*, vol. 1, no. 75, p. 218, 1993.

[162] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[163] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[164] F. A. Gers, J. Schmidhuber, and F. Cummins, "Learning to forget: Continual prediction with lstm," *Neural computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

[165] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[166] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 31, no. 5, pp. 855–868, 2008.

[167] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," 2014.

[168] X. Li and X. Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *2015 ieee international conference on acoustics, speech and signal processing (icassp)*. IEEE, 2015, pp. 4520–4524.

[169] K. Islam, L. M. Dang, S. Lee, and H. Moon, "Image compression with recurrent neural network and generalized divisive normalization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1875–1879.

[170] C. Lin, J. Yao, F. Chen, and L. Wang, "A spatial rnn codec for end-to-end image compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13 269–13 277.

[171] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *2020 IEEE International Conference on Image Processing (ICIP)*, 2020, pp. 3339–3343.

[172] R. Su, Z. Cheng, H. Sun, and J. Katto, "Scalable learned image compression with a recurrent neural networks-based hyperprior," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3369–3373.

[173] A. G. Ororbia, A. Mali, J. Wu, S. O'Connell, W. Dreese, D. Miller, and C. L. Giles, "Learned neural iterative decoding for lossy image compression systems," in *2019 Data Compression Conference (DCC)*, 2019, pp. 3–12.

[174] A. Mali, A. Ororbia, D. Kifer, and L. Giles, "An empirical analysis of recurrent learning algorithms in neural lossy image compression systems," *arXiv preprint arXiv:2201.11782*, 2022.

[175] K. Wang, N. Zhang, K. Hu, and T. Cao, "Multispectral image compression algorithm based on silced convolutional lstm," in *International Conference in Communications, Signal Processing, and Systems.* Springer, 2022, pp. 887–891.

[176] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[177] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[178] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," *Advances in neural information processing systems*, vol. 29, 2016.

[179] M.-Y. Liu and O. Tuzel, "Coupled generative adversarial networks," *Advances in neural information processing systems*, vol. 29, 2016.

[180] Y. Lu, Y.-W. Tai, and C.-K. Tang, "Conditional cyclegan for attribute guided face image generation," *arXiv preprint arXiv:1705.09966*, vol. 2, 2017.

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

[182] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, "Generative adversarial networks for extreme learned image compression," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 221–231.

[183] C. Jia, X. Zhang, S. Wang, S. Wang, and S. Ma, "Light field image compression using generative adversarial network-based view synthesis," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 177–189, 2018.

[184] D. Liu, X. Huang, W. Zhan, L. Ai, X. Zheng, and S. Cheng, "View synthesis-based light field image compression using a generative adversarial network," *Information Sciences*, vol. 545, pp. 118–131, 2021.

[185] C. Huang, H. Liu, T. Chen, Q. Shen, and Z. Ma, "Extreme image coding via multi-scale autoencoders with generative adversarial optimization," in *2019 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2019, pp. 1–4.

[186] J. Yang, C. Yang, Y. Ma, S. Liu, and R. Wang, "Learned low bit-rate image compression with adversarial mechanism," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 140–141.

[187] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Extreme learned image compression with gans," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 2587–2590.

[188] E. Agustsson and L. Theis, "Universally quantized neural compression," *Advances in neural information processing systems*, vol. 33, pp. 12 367–12 376, 2020.

[189] Y. Bengio, N. Léonard, and A. Courville, "Estimating or propagating gradients through stochastic neurons for conditional computation," *arXiv preprint arXiv:1308.3432*, 2013.

[190] P. Yin, J. Lyu, S. Zhang, S. Osher, Y. Qi, and J. Xin, "Understanding straight-through estimator in training activation quantized neural nets," *arXiv preprint arXiv:1903.05662*, 2019.

[191] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," *Advances in neural information processing systems*, vol. 30, 2017.

[192] Y. Yang, R. Bamler, and S. Mandt, "Improving inference for neural image compression," *Advances in Neural Information Processing Systems*, vol. 33, pp. 573–584, 2020.

[193] X. Li, A. Naman, and D. Taubman, "A neural network lifting based secondary transform for improved fully scalable image compression in jpeg 2000," in *2022 IEEE International Conference on Image Processing (ICIP)*, 2022, pp. 1606–1610.

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

[195] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3214–3223.

[196] D. Le Gall and A. Tabatabai, "Sub-band coding of digital images using symmetric short kernel filters and arithmetic coding techniques," in *ICASSP-88., International Conference on Acoustics, Speech, and Signal Processing*, 1988, pp. 761–764 vol.2.

[197] A. Cohen, I. Daubechies, and J.-C. Feauveau, "Biorthogonal bases of compactly supported wavelets," *Communications on pure and applied mathematics*, vol. 45, no. 5, pp. 485–560, 1992.

[198] H. Ma, D. Liu, N. Yan, H. Li, and F. Wu, "End-to-end optimized versatile image compression with wavelet-like transform," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 3, pp. 1247–1263, 2022.