Color Optimization and Diffusion-based Post-processing to Obtain Sharper Images without Compromising R-D Performance in Learned Image Compression

by

O. Ugur Ulas

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Electrical and Electronics Engineering

KOÇ ÜNİVERSİTESİ

July 29, 2024

Color Optimization and Diffusion-based Post-processing to Obtain Sharper Images without Compromising R-D Performance in Learned Image Compression

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

O. Ugur Ulas

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

_____

Prof. A. Murat Tekalp

_____

Assoc. Prof. Aykut Erdem

_____

Prof. Erkut Erdem

Date: _____

To My Family

ABSTRACT

Color Optimization and Diffusion-based Post-processing to Obtain Sharper Images without Compromising R-D Performance in Learned Image Compression

O. Ugur Ulas

Master of Science in Electrical and Electronics Engineering

July 29, 2024

In the digital era, efficient storage and transmission of visual signals have become paramount due to the explosive growth in multimedia content. The need for advanced image compression methods is driven by increasing image resolutions and the limitations of traditional codecs in terms of flexibility and adaptability.

In the first part of this thesis, we introduce a flexible method for coding color images in the YCrCb space, addressing the human visual system's greater sensitivity to the luma component over chroma components. We extend the variable-rate image coding approach to YCrCb images, enabling separate rate adjustments for luma and chroma components. By implementing image-adaptive luma-chroma bit allocation during inference, we can increase Y PSNR at the expense of slightly lower chroma PSNR, resulting in sharper images without introducing color artifacts. This strategy enhances image sharpness more effectively than optimizing for RGB PSNR alone. Our experimental results demonstrate that sharper images with better VMAF and Y PSNR can be obtained by optimizing models for YCrCb MSE compared to state-of-the-art models optimizing RGB MSE at the same bpp.

In the second part, we explore the use of diffusion models for post-processing in wavelet-based image codecs. Diffusion models, a type of deep generative models, have shown great promise in various domains, including inverse problems in image processing. They are particularly effective at producing visually pleasing textures. By integrating a fixed, invertible transform with a learned entropy model and a diffusion-based post-processing module, we demonstrate enhanced visual quality without compromising the rate-distortion performance. Our experimental results show that sharper images with better perceptual quality and YCrCb PSNR can be obtained compared to state-of-the-art classic and learned codecs.

ÖZETÇE

Öğrenilmiş Görüntü Sıkıştırmada R-D Performansını Bozmadan Daha Keskin Görüntüler Elde Etmek İçin Renk Optimizasyonu ve Difüzyon Tabanlı Son İşleme

O. Ugur Ulas

Elektrik ve Elektronik Mühendisliği, Yüksek Lisans

29 Temmuz 2024

Dijital çağda, görsel sinyallerin verimli bir şekilde depolanması ve iletilmesi, multimedya içeriğindeki patlayıcı artış nedeniyle büyük önem kazanmıştır. Artan görüntü çözünürlükleri ve geleneksel kodlayıcıların esneklik ve uyum sağlama konusundaki sınırlamaları, gelişmiş görüntü sıkıştırma yöntemlerine olan ihtiyacı artırmaktadır.

Bu tezin ilk bölümünde, insan görsel sisteminin luma bileşenine olan hassasiyetinin chroma bileşenlerine kıyasla daha fazla olduğunu göz önünde bulundurarak YCrCb uzayında görüntülerin kodlanması için esnek bir yöntem sunuyoruz. Değişken oranlı görüntü kodlama yaklaşımını YCrCb görüntülerine genişleterek luma ve chroma bileşenleri için ayrı oran ayarlamaları yapılmasını sağlıyoruz. Çıkarım sırasında görüntüye uyarlanabilir luma-chroma bit tahsisi yaparak, Y PSNR'yi artırırken, hafifçe daha düşük chroma PSNR pahasına, renk bozulmalarına yol açmadan daha keskin görüntüler elde edebiliyoruz. Bu strateji, yalnızca RGB PSNR'yi optimize etmekten daha etkili bir şekilde görüntü keskinliğini artırmaktadır. Deneysel sonuçlarımız, YCrCb MSE'yi optimize eden modellerin, aynı bpp oranında RGB MSE'yi optimize eden en güncel modellerle karşılaştırıldığında, daha keskin görüntüler ve daha iyi VMAF ile Y PSNR sağladığını göstermektedir.

İkinci bölümde ise, wavelet tabanlı görüntü kodlayıcılarında difüzyon modellerinin son işlem için kullanımını inceliyoruz. Derin üretici modellerin bir türü olan difüzyon modelleri, görüntü işleme alanındaki ters problemler de dahil olmak üzere çeşitli alanlarda büyük bir potansiyel göstermiştir. Sabit, terslenebilir bir dönüşümü öğrenilmiş bir entropi modeli ve difüzyon tabanlı bir son işlem modülü ile entegre ederek, RD performansını bozmadan görsel kaliteyi artırabileceğimizi gösteriyoruz. Deneysel sonuçlarımız, klasik ve öğrenilmiş en güncel kodlayıcılara kıyasla daha keskin görüntüler ve daha iyi algısal kalite elde edilebileceğini göstermektedir.

ACKNOWLEDGMENTS

I would like to express my deepest gratitude to my advisor, Professor A. Murat Tekalp, for his invaluable guidance, insightful feedback, and unwavering support throughout the course of my research. At every stage of this work, his vast knowledge and experience have served as a guiding light, helping me navigate the complexities of my research.

I am also deeply grateful to my family, whose love and support have been my constant source of strength and motivation. Their belief in me has been a driving force throughout my academic journey.

Lastly, I want to thank my beloved girlfriend, Gökçe, for her endless patience, understanding, and encouragement. Her unwavering support has been a true pillar in my life, and I am incredibly fortunate to have her by my side.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

AE	Autoencoder
AR	Autoregressive
BPP	Bits per pixel
CCM	Checkerboard Context Model
CNN	Convolutional Neural Network
CVR	Continuously variable rate
DCT	Discrete Cosine Transform
DDRM	Denoising Diffusion Restoration Models
DISTS	Deep Image Structure and Texture Similarity
DVR	Discrete variable rate
GAN	Generative adversarial network
GDN	Generalized Divisive Normalization
GMM	Gaussian mixture model
HEVC	High Efficiency Video Coding
IGDN	Inverse Generalized Divisive Normalization
LIC	Learned Image Compression
LPIPS	Learned Perceptual Image Patch Similarity
LRP	Latent Residual Prediction
MSE	Mean Squared Error
MS-SSIM	Multi-Scale Structural Similarity Index
PSNR	Peak Signal-to-Noise Ratio
RD	Rate-Distortion
SCCTX	Space-Channel ConTeXt
SSIM	Structural Similarity Index
VAE	Variational autoencoder
VVC	Versatile Video Coding

Chapter 1

INTRODUCTION

1.1 Motivation

In the digital age, the efficient storage and transmission of visual data have become paramount due to the explosive growth in multimedia content. The necessity for advanced image compression methods is driven by several factors. First, the increasing resolution of digital images and the ubiquity of high-definition video require more effective compression techniques to efficiently manage the vast amounts of data. Second, the limitations of traditional codecs in terms of flexibility and adaptability call for more sophisticated solutions.

Learned image compression offers several advantages over traditional methods, primarily due to its use of advanced machine learning techniques, particularly deep learning. One of the key areas where learned image compression excels is using nonlinear transforms. Traditional image compression methods, like JPEG and JPEG2000, rely on linear transforms such as Discrete Cosine Transform (DCT) and Wavelet Transform. While these methods are effective, they have limitations in capturing complex patterns and structures in images. Learned image compression uses deep neural networks, which can model highly complex and nonlinear relationships in image data. These networks can learn to transform images into a more compact representation that better captures important features and details. Additionally, neural networks can adapt to different types of images and content, providing more flexibility and efficiency in compression compared to fixed linear transforms.

Another significant advantage of learned image compression is end-to-end optimization. Traditional methods often separate the stages of transformation, quantization, and entropy coding, optimizing each stage independently. In contrast, learned

image compression optimizes all stages of the compression process jointly. This end-to-end approach allows better coordination between different stages, leading to more efficient compression. By using differentiable components, the entire compression process can be trained using gradient descent, which allows for the use of sophisticated loss functions that can directly optimize for compression performance and image quality.

Perceptual metrics are another area where learned image compression surpasses traditional methods. Traditional image compression methods typically optimize for pixel-based metrics like Mean Squared Error (MSE) or Peak Signal-to-Noise Ratio (PSNR), which do not always correlate well with human perception of image quality. Learned image compression can incorporate perceptual loss functions that better align with human visual perception. These include losses based on features extracted by neural networks (e.g., VGG network features) and adversarial losses used in the generative adversarial networks (GAN). By focusing on perceptual quality, learned image compression methods can produce images that look better to the human eye at the same or lower bitrates compared to traditional methods.

However, most of the LIC architectures are optimized for RGB images, while it is known that the luma component is more important for the human visual system since the human eye is less sensitive to details in chrominance components than the luma component. In this thesis, we first explore the LIC for YCrCb images and offer flexibility for rate adjustments for luma-chroma components. Secondly, diffusion models have emerged as the most advanced type of deep generative models. These models have challenged the long-standing supremacy of GANs in tasks related to image synthesis and have demonstrated potential in various domains. Following these advancements, we explore the effectiveness of diffusion-based post-processing for wavelet-based image codecs.

1.2 Contribution and Organization

The main contributions of this thesis are as follows:

- We introduce a flexible method for coding images in the YCrCb color space. First, we extend the variable-rate image coding approach to YCrCb images, allowing for separate adjustments of the rate for luma and chroma components. To achieve this, we provide two distinct architectures. We demonstrate that by adjusting the rate of luma and chroma components during inference, we can significantly improve the perceptual quality of the images. The proposed method yields perceptually better images compared to the RGB-optimized counterparts, which serve as our baseline. This work is presented in PCS 2022 [Ulas and Tekalp, 2022]

- We demonstrate that diffusion models can significantly enhance the visual quality of images produced by wavelet-based codecs. Our approach integrates a fixed, invertible transform with a learned entropy model and a diffusion-based post-processing module. We show that learned entropy models can improve the performance of a wavelet-based image codec with a fixed transform, and we demonstrate that diffusion-based post-processing can enhance visual quality without compromising rate-distortion performance.

The rest of the thesis is organized as follows:

- Chapter 2 covers background and literature review. Section 2.1 explains the fundamental concepts of lossless data compression, including entropy and information, Huffman coding, and arithmetic coding. Section 2.2 delves into rate-distortion theory and optimization, providing a comprehensive overview of the theoretical foundations necessary for understanding the subsequent chapters. Lastly, Section 2.3 presents a literature review where we review and explain learned image compression methods, offering a critical assessment of existing approaches and their limitations.

- Chapter 3 introduces our flexible luma-chroma coding architecture. Section 3.1 provides an introduction to learned image compression in the YCrCb space, setting the stage for the innovations presented in this thesis. Section 3.2 covers

the proposed network architectures for flexible luma-chroma coding and the fine-tuning methods employed. Section 3.3 describes the training details and presents experimental results, showcasing the performance improvements and qualitative enhancements achieved by our method.

- Chapter 4 introduces our approach to learned image compression with diffusion-based post-processing. Section 4.1 details wavelet-domain image compression with a learned entropy model, including subband decomposition by a fixed transform and the integration of a learned entropy model. Section 4.2 discusses the application of diffusion models for post-processing to enhance visual quality, explaining how these models contribute to the overall performance of our method. Finally, Section 4.3 describes the experimental settings and results, including training details, experimental setup, and comparative results, demonstrating the effectiveness of our method in producing sharper images with better perceptual quality.

- Chapter 5 presents the conclusion.

Chapter 2

BACKGROUND AND RELATED WORK

2.1 Lossless Compression

2.1.1 Entropy and information

Entropy, denoted by $H[X]$, serves as a measure of the uncertainty or information content associated with a discrete random variable X given its probability distribution P. In essence, it quantifies how much unpredictability there is in the outcomes of X. Mathematically, entropy is defined as the expected value of the negative logarithm of the probability of X:

$$H[X] = \mathbb{E}_{X\sim P}[-\log_2 P(X)]$$

To illustrate, consider a fair six-sided die. In this case, the entropy is approximately 2.58 bits, reflecting the moderate uncertainty associated with rolling any of the six faces. Similarly, the entropy of a fair coin flip is 1 bit, as there are two equally likely outcomes. In contrast, if a die is heavily biased so that one number is almost certain, the entropy is significantly reduced, approaching zero for a deterministic event.

In the context of lossless data compression, entropy represents the theoretical limit on the average number of bits required to encode the data. When using an optimal prefix code, the expected length of the encoded message $\mathbb{E}[|C(X)|]$ satisfies the inequality:

$$H[X] \leq \mathbb{E}[|C(X)|] \leq H[X] + 1$$

Entropy coding, a fundamental technique in lossless data compression, leverages these principles. It involves encoding a sequence of outcomes from a discrete random

variable so that more frequent symbols are assigned shorter codewords, while less frequent symbols are given longer codewords. Hence, it reduces the overall length of the encoded message, approaching the entropy limit.

2.1.2 Arithmetic Coding

Among various entropy coding methods, Arithmetic coding is a form of entropy encoding commonly used in lossless data compression. Unlike Huffman coding [Huffman, 1952], which uses discrete symbols, arithmetic coding represents the entire message as a single number, a fraction n where $(0.0 \leq n < 1.0)$ [Witten et al., 1987].

The algorithm works by partitioning the range $[0, 1)$ into sub-intervals based on the probabilities of the symbols based on the cumulative distribution function (CDF) of the symbols. For a symbol with probability P and CDF value C, the sub-interval is defined as $[C, C+P)$. This ensures that the more frequent symbols are represented with larger sub-intervals, which corresponds to shorter code lengths. Each symbol of the input message narrows the interval, and the final interval represents the encoded message.

Algorithm 1: Arithmetic Coding

Procedure `Encode(`*symbols, cdf[]*`):`

Input: symbols, cdf[]

Output: low

$[low, high] \leftarrow [0, 1]$;

for *each* s **to** *symbols* **do**

$low \leftarrow low + (high - low) \cdot cdf[s-1]$;

$high \leftarrow low + (high - low) \cdot cdf[s]$;

return *low*;

Procedure `Decode(`*V, cdf[], N*`):`

Input: V: received bit-stream, *cdf*[], N: number of symbols

Output: data

$[low, high] \leftarrow [0, 1]$;

$data \leftarrow [\,]$;

for $i \leftarrow 1$ **to** N **do**

find s such that $cdf[s-1] \leq \frac{V-low}{high-low} < cdf[s]$;

$data$.append(s);

$low \leftarrow low + (high - low) \cdot cdf[s-1]$;

$high \leftarrow low + (high - low) \cdot cdf[s]$;

return *data*;

Consider a random variable X with a ternary alphabet $\{A, B, C\}$, assigned probabilities of 0.4, 0.4, and 0.2, respectively. Let the sequence to be encoded be $ACAA$. Initially, the input sequence is empty, and the corresponding interval is $[0, 1)$.

1. For the first symbol A, the interval is updated to $[0, 0.4)$.
2. For the second symbol C, the interval is narrowed down to $[0.32, 0.4)$.
3. For the third symbol A, the interval further shrinks to $[0.32, 0.352)$.
4. For the fourth symbol A, the interval becomes $[0.32, 0.3328)$.

Given the probability of this sequence is 0.0128, we use $\log(1/0.0128) + 2$ bits, which totals 9 bits. The detailed procedure is illustrated in Figure 2.1 and example is taken from [Cover and Thomas, 2006].

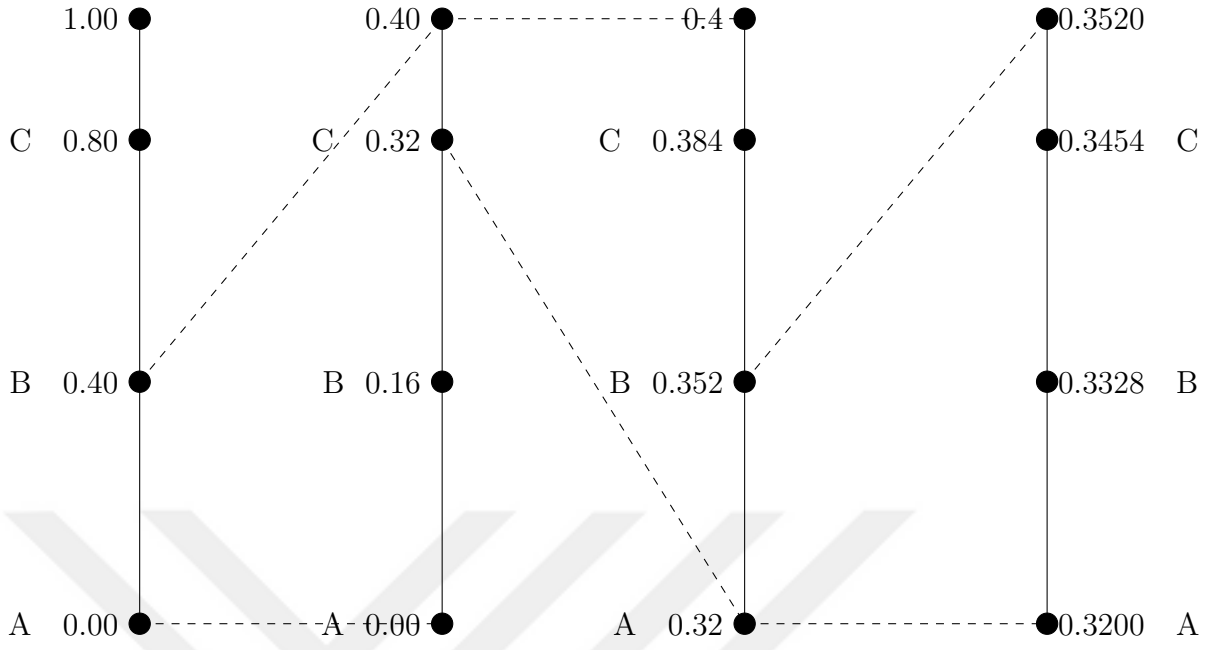


Figure 2.1: Arithmetic coding example.

2.2 End-to-End Rate-distortion Optimization

In neural image compression, the components of the transform coding approach are replaced with learned functions. These components are mainly analysis transform g_s to map input image to a latent y, entropy model P is used to losslessly transmit quantized latent $\hat{y}$ as bitstreams. Then the synthesis transform g_s is used to recover the reconstructed image from quantized latent $\hat{y}$,i.e. $\hat{y} = [\![y]\!]$. In rate-distortion optimization, the primary objective is to simultaneously minimize both the rate,

$$R := \mathbb{E}[-\log_2 P([\![g_a(X)]\!])]$$

and the distortion,

$$D := \mathbb{E}[\rho(X, g_s([\![g_a(X)]\!]))]$$

Then, by introducing the rate-distortion Lagrangian, the optimization problem becomes

$$\mathbb{E}[-\log_2 P(\llbracket g_a(X) \rrbracket)] + \lambda \mathbb{E}[\rho(X, g_s(\llbracket g_a(X) \rrbracket))],$$

where λ is the trade-off factor. Generally, LIC methods train a set of models with different λ values, one for each discrete R-D point. However, due to the non-differentiability of the quantization operator, since its derivative is undefined at integer points and zero everywhere else, this optimization cannot be performed directly.

To address this issue, several different approaches have been proposed [Ballé et al., 2016, Agustsson et al., 2017, Yang et al., 2020]. Among these, additive uniform noise (AUN) is most commonly used which replaces rounding operations with additive uniform noise for model training.

$$\lfloor \mathbf{y} \rceil \approx \mathbf{y} + \mathbf{u}, \quad \mathbf{u} \sim \mathcal{U}([-0.5, 0.5)^n)$$

After applying AUN, our optimization problem becomes

$$\mathbb{E}_{\mathbf{x} \sim P_{\text{data}}, \mathbf{u} \sim \mathcal{U}}[-\log_2 \tilde{p}(g_a(\mathbf{x}) + \mathbf{u}) + \lambda \rho(\mathbf{x}, g_s(g_a(\mathbf{x}) + \mathbf{u}))]$$

where

$$\tilde{p} := p * \mathcal{U}([-0.5, 0.5)^n)$$

The selection of the distortion function in LIC is critical, as neural networks tend to excel on the specific metric they are trained to optimize, often at the expense of performance on other metrics. Typically, the distortion function ρ is MSE. However, it is known that models optimized for MSE tend to produce images with suboptimal perceptual quality. This discrepancy arises because MSE does not align well with human visual perception, often resulting in overly smooth images that lack fine details.

Another commonly employed metric is the Structural Similarity Index (SSIM) [Wang et al., 2004], which better correlates with human perception by considering changes in structural information, luminance, and contrast. However, SSIM can

sometimes introduce artifacts in reconstructed images, and neural networks may overfit to this metric. Consequently, SSIM is often combined with MSE during training, or fine-tuning is performed after training models with MSE for several epochs. It is also observed that SSIM does not perform well with generative models. Therefore, distortion measures such as Learned Perceptual Image Patch Similarity (LPIPS) and Deep Image Structure and Texture Similarity (DISTS), which utilize neural network-based features to estimate distortion, are generally preferred for these models.

2.3 Literature Review on Learned Image Compression

The state-of-the-art in LIC has evolved significantly since the introduction of the variational autoencoder framework in the seminal work by [Ballé et al., 2017], which established a new paradigm for end-to-end rate-distortion optimized image compression. In the years that followed, the field has seen substantial improvements in rate-distortion performance through more sophisticated and complex entropy modeling techniques [Ballé et al., 2018, Minnen et al., 2018, Minnen and Singh, 2020, Cheng et al., 2020, He et al., 2021, He et al., 2022, Jiang and Wang, 2023]. Recent developments have also embraced invertible transform-based models, such as normalizing flows [Xie et al., 2021a] and learned wavelet transforms [Ma et al., 2020, Ma et al., 2022], which offer enhanced flexibility and efficiency in modeling the distribution of image data. While CNN architectures are commonly used due to their ability to effectively capture local features and patterns, recent advancements have also explored the use of transformers and attention mechanisms to further enhance compression efficiency by capturing global relationships within the image [Liu et al., 2023, Qian et al., 2022, Koyuncu et al., 2022, Zou et al., 2022, Koyuncu et al., 2024]. These approaches are particularly advantageous in entropy modeling, where understanding the broader context can lead to more accurate predictions and better compression rates. By leveraging the global attention capabilities of transformers, these methods can identify and utilize long-range dependencies and correlations across the image, resulting in improved compression performance and efficiency. This section delves

into a selected subset of these advancements, carefully chosen to illustrate their pivotal contributions to advancing the state of learned image compression.

2.3.1 Variational Image Compression with a Scale Hyper-prior

[Ballé et al., 2018] proposed a method that enhances traditional autoencoder-based image compression by introducing a hyperprior to effectively capture spatial dependencies within the latent representation, y. Conventional methods often fail to account for the complex relationships between pixel intensities and their spatial arrangement, leading to suboptimal compression efficiency. This inefficiency arises because [Ballé et al., 2017] approaches use a fully factorized entropy model that does not consider the statistical dependencies that remain in the latent representation. By introducing a hyperprior, each element of the latent representation is modeled as a zero-mean Gaussian with a standard deviation predicted by a parametric transformation applied to additional latent variables, making the entropy model adaptive to the image content. This end-to-end optimization allows the model to learn both the image representation and the entropy model, thereby balancing the amount of side information with the expected improvement in compression performance.

The network architecture of the variational autoencoder with a scale hyperprior is illustrated in Figure 2.2. The left side shows the image autoencoder architecture, while the right side corresponds to the autoencoder implementing the hyperprior. The autoencoder architecture employs an analysis transform g_a, which consists of a series of convolutional layers with kernel sizes $N \times 5 \times 5/2 \downarrow$, each followed by Generalized Divisive Normalization (GDN). This process maps the input RGB image of shape $H \times W \times 3$ into the latent representation y, which has a size of $H/16 \times W/16 \times C$. The latent representation y is then passed through a hyper-analysis transform, comprising additional convolutional layers and activation functions, to obtain the hyper-latent variable z.

The hyper-latent variable z is modeled using a non-parametric, fully factorized density model similar to [Ballé et al., 2017]. After quantization, the hyper-latent variable z is encoded and included in the bitstream as side information. The quan-

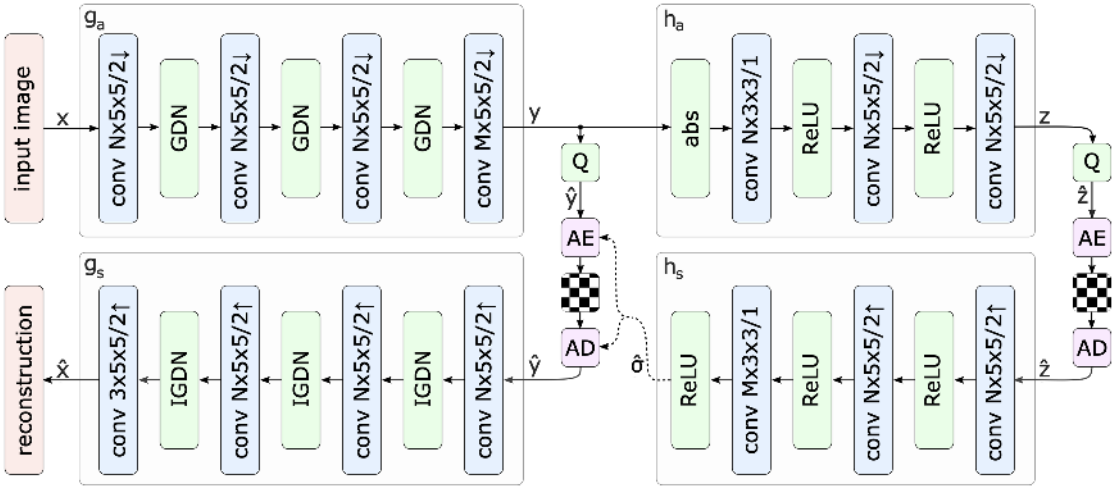


Figure 2.2: Network architecture of the scale hyperprior model. AE and AD denote arithmetic encoding and decoding, respectively. [Ballé et al., 2018]

tized hyper-latent variables are then processed through the hyper-synthesis transform h_s to produce the scale parameters. Finally, the quantized latent representation $\hat{y}$ is passed through the synthesis transform g_s, which consists of transposed convolutions and inverse GDN (IGDN) layers, to reconstruct the input image as $\hat{x}$.

The authors reported that the rate-distortion (PSNR) performance of their model approaches that of traditional image codecs like BPG [Bellard, 2015], while it surpasses BPG in terms of MS-SSIM.

2.3.2 Joint Autoregressive and Hierarchical Priors

[Minnen et al., 2018], inspired by the success of autoregressive priors in probabilistic generative models, advanced entropy modeling by incorporating joint autoregressive (AR) and hierarchical priors. This approach, referred to as backward adaptive entropy modeling, conditions each image latent y_i on previously decoded latents and jointly predicts μ_i and σ_i, thus allowing for better exploitation of spatial dependencies in the data and significantly improves the rate-distortion performance. For the AR context model, they use 5×5 masked convolution kernel similar to the [van den Oord et al., 2016], which allows causal context modeling by restricting information

flow from the future pixels. Their experimental results show that hierarchical priors and AR context model are complementary. The hierarchical priors capture global structures and dependencies in the data, while the AR context model component refines these predictions by accounting for local correlations.

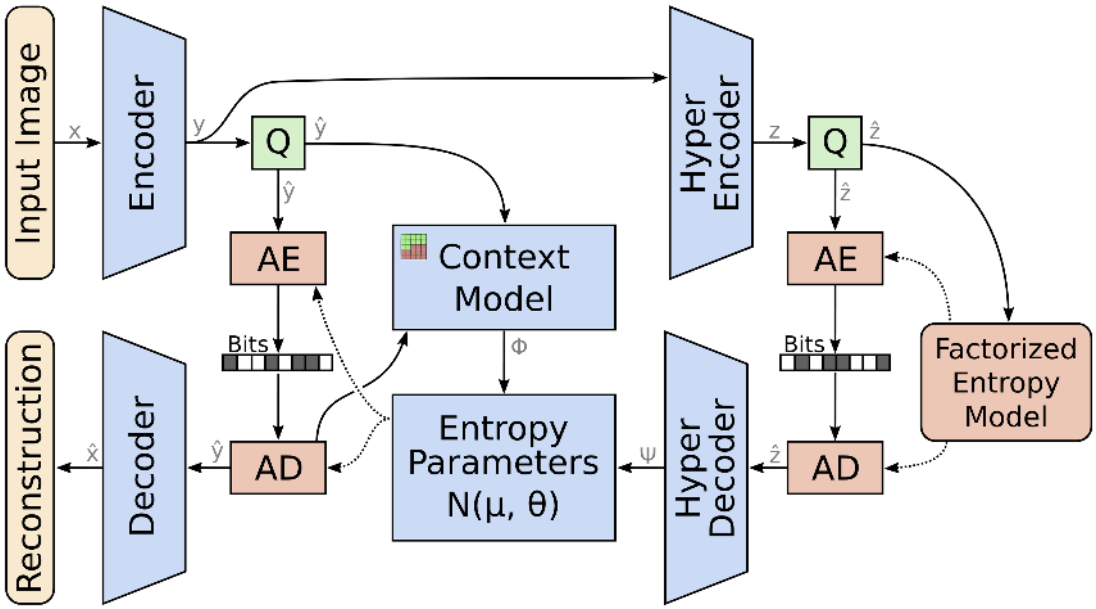


Figure 2.3: Model architecture that combines context model with hyper-prior from [Minnen et al., 2018]

Unlike scale-hyperprior architecture described in Section 2.3.1, that focused on scale prediction, this extended model predicts both the mean and scale parameters conditioned on the hyperprior $\hat{z}$ and the causal context $\hat{y}_{<i}$. The Gaussian parameters are determined through the hyper-decoder, context model, and entropy parameters networks, collectively encapsulated in the parameters $\theta_{hd}, \theta_{cm}, \theta_{ep}$:

$$p_{\hat{y}}(\hat{y} \mid \hat{z}, \theta_{hd}, \theta_{cm}, \theta_{ep}) = \prod_i \left(\mathcal{N}(\mu_i, \sigma_i^2) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right)(\hat{y}_i)$$

with $\mu_i, \sigma_i = g_{ep}(\psi, \phi_i; \theta_{ep}), \psi = g_h(\hat{z}; \theta_{hd})$, and $\phi_i = g_{cm}(\hat{y}_{<i}; \theta_{cm})$.

In [Cheng et al., 2020], this entropy model is enhanced by employing a discretized K-component Gaussian mixture model (GMM):

$$p_{\hat{y}|\hat{z}}(\hat{y}_i|\hat{z}) = \sum_{0<k<K} \pi_i^{(k)} \left[\mathcal{N}(\mu_i^{(k)}, (\sigma_i^{(k)})^2) * \mathcal{U}\left(-\frac{1}{2}, \frac{1}{2}\right) \right] (\hat{y}_i)$$

Here, the K groups of entropy parameters $(\pi^{(k)}, \mu^{(k)}, \sigma^{(k)})$ are determined by g_{ep}.

Moreover, [Cheng et al., 2020] further improves the performance by replacing transposed convolution layers with a pixel shuffler, adapting the attention mechanism, and incorporating residual blocks. As a result, this approach is the first to achieve a PSNR comparable with Versatile Video Coding (VVC) Intra [Bross et al., 2020].

2.3.3 Channel-wise Autoregressive Entropy Models

[Minnen and Singh, 2020] proposes a channel-wise AR entropy model. The primary motivation behind this research is to address the limitations of backward adaptive entropy models described in Section 2.3.2. Channel-wise AR entropy model particularly tries to solve their inefficiency in utilizing parallel processing capabilities of GPUs and TPUs due to the serial nature of backward adaptation. This inefficiency arises because backward adaptation typically requires processing based on the causal context of each symbol, necessitating sequential steps that hinder the effective use of parallel hardware.

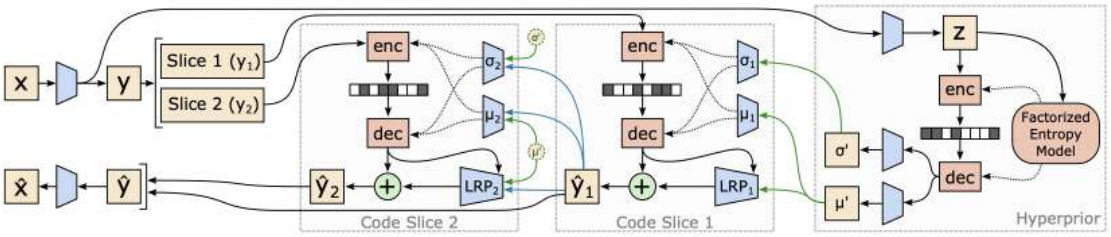


Figure 2.4: Channel-wise autoregressive entropy model [Minnen and Singh, 2020]

To overcome these challenges, the authors propose channel-conditioning, which involves splitting the latent tensor into several slices along the channel dimension, each encoded and decoded sequentially but processed in parallel within each slice.

This method models dependencies between channels, thereby reducing entropy and enhancing parallel processing capabilities. By conditioning on previously decoded slices, channel-conditioning effectively reduces the need for serial processing and improves the efficiency of GPU and TPU utilization during decoding.

Secondly, they introduce latent residual prediction (LRP). LRP aims to correct the quantization error in the latent space, by predicting and adjusting the residual error slice-by-slice leveraging the inter-slice correlations and the hyperprior. Reducing the quantization error in latent space not only improves the distortion but also reduces the bitrate of the slices since previous slices are used to estimate entropy parameters of remaining slices.

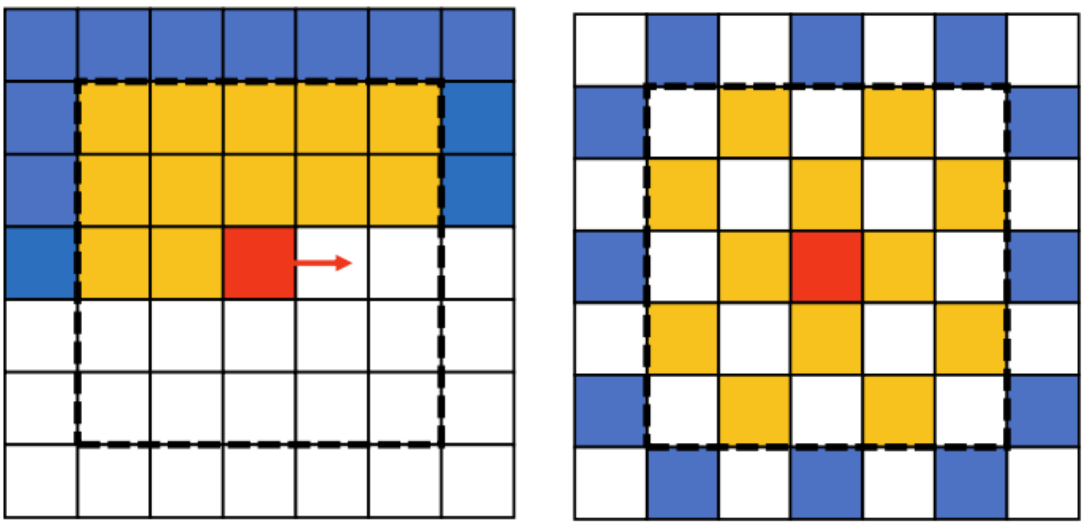
The authors also propose a mixed training approach where uniform noise simulates quantization during entropy model training, but rounding is applied when passing quantized values to the synthesis transform. This strategy also leads betters rate-distortion performance by addressing the gradient issue.

Empirically, these enhancements result in substantial improvements in compression efficiency. The combined effect of channel-conditioning and latent residual prediction yields an average rate savings of 6.7% on the Kodak image set and 11.4% on the Tecnick image set compared to [Ballé et al., 2018].

2.3.4 Checkerboard Context-model

[He et al., 2021] also addresses the limitations of existing AR context models by proposing a parallelizable checkerboard context model (CCM) and a two-pass decoding method. The CCM reorganizes the decoding order to eliminate spatial location limitations, thereby enabling parallel processing. This reorganization is achieved by decoding half of the latent representations (referred to as "anchors") independently of their spatial context, relying only on hyperprior information. The remaining latent representations (referred to as "non-anchors") are decoded in parallel, using the context information from the already decoded anchors.

Figure 2.6 shows the two-pass decoding process that CCM uses. Firstly, latent tensor is splitted into two interleaved sets of spatial locations, resembling a checker-



(a) (b)

Figure 2.5: Comparison of 5x5 serial and checkerboard context models. In the serial context model shown in (a), the red block is currently being encoded/decoded, and the yellow and blue blocks are visible. For the serial context model shown in (a) strict Z-order is required. In contrast, the checkerboard context model depicted in (b) allows for parallel computation of all non-anchors (red and white blocks) using the anchors (blue and yellow blocks), which are coded only using hyperprior information [He et al., 2021]

board pattern. In the first pass, the anchors are decoded using only the hyperprior, ensuring no dependency on yet-to-be-decoded latents. In the second pass, the non-anchors are decoded using the context from the already decoded anchors, which can be processed in parallel. This method allows the context model to utilize parallel processing capabilities fully, drastically improving the computational efficiency of the decoding process without significantly compromising RD performance.

The proposed CCM achieves more than a 40-fold speedup in the decoding process compared to traditional serial context models while maintaining comparable RD performance. The significant increase in computational efficiency makes this model particularly suitable for practical applications.

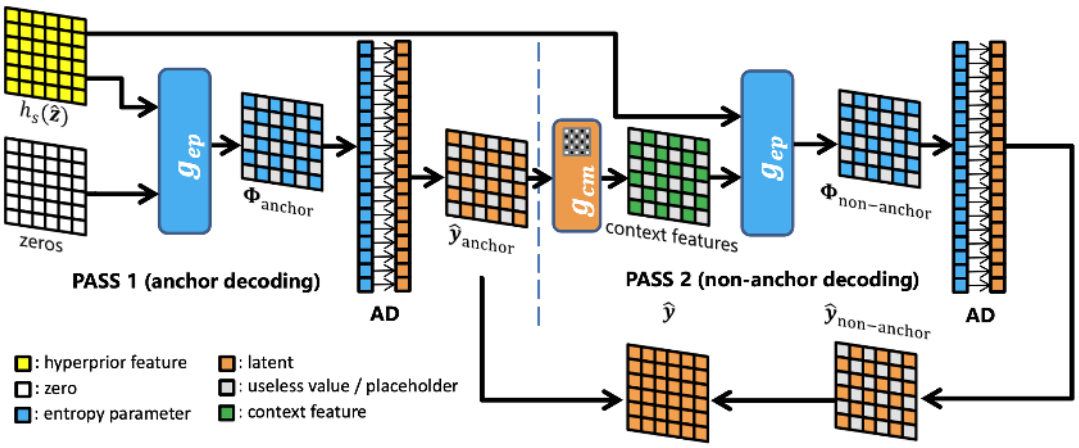


Figure 2.6: Two-pass decoding process [Minnen and Singh, 2020]

2.3.5 Unevenly Grouped Space-Channel Contextual Adaptive Coding

[He et al., 2022] further improves the performance of LIC techniques by combining CCM with channel conditioning. Instead of splitting latent evenly, they introduce an uneven channel-conditioning. This approach is based on the observation of energy compaction in LIC, where the earlier channels contain more significant information and higher entropy compared to the later channels. By unevenly grouping these channels, the method allocates more bits and computational resources to the initial, high-entropy channels while merging the later channels into larger chunks. The pro-

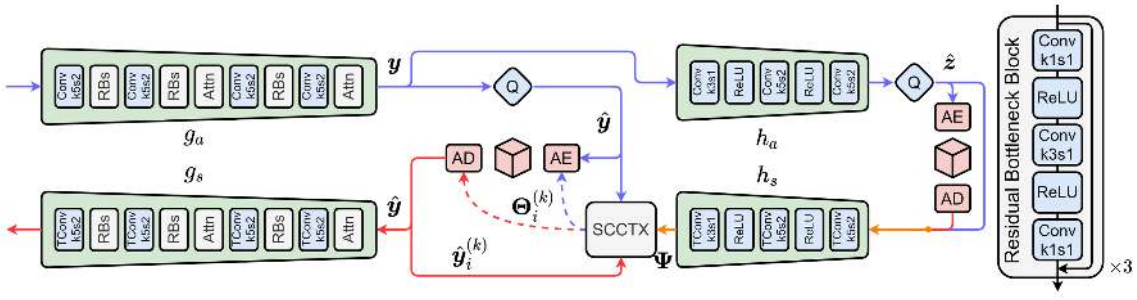


Figure 2.7: Overall architecture of ELIC [He et al., 2022]

posed method combines this uneven grouping with CCM [He et al., 2021], resulting

in a multi-dimension entropy estimation model called the Space-Channel ConTeXt (SCCTX) model. This model effectively reduces spatial and channel-wise redundancies, enhancing coding efficiency while maintaining a fast running speed. By leveraging both spatial and channel contexts, the SCCTX model provides a more comprehensive entropy estimation, which is critical for efficient compression.

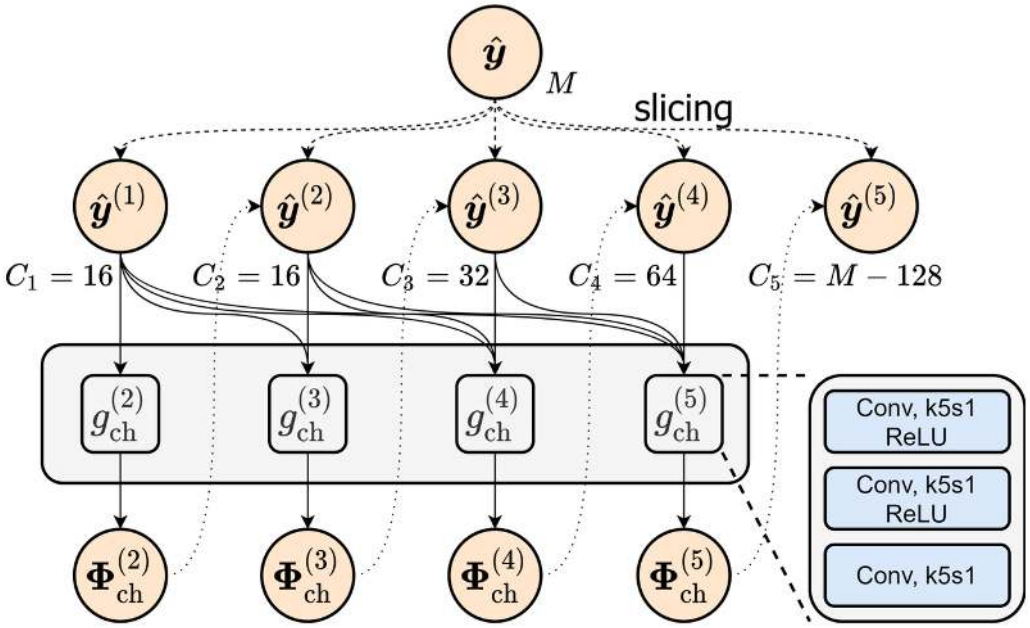


Figure 2.8: Uneven Channel Conditioning [He et al., 2022]

In addition to the SCCTX model, the authors replace the traditional Generalized Divisive Normalization (GDN) layer with stacked residual blocks. This substitution enhances the model's nonlinearity and expressiveness, resulting in improved Rate-Distortion (RD) performance. Moreover, the ELIC model supports extremely fast preview decoding because most of the semantic information is available in the initial decoded slices on the decoder side. To achieve this, after completing the training of the model, they freeze the entire network and train a thumbnail synthesizer using the first few slices. This process enables a low-resolution preview of the image. Additionally, they provide results for the progressive decoding of images, where starting from the first slice, each subsequent slice adds high-frequency information

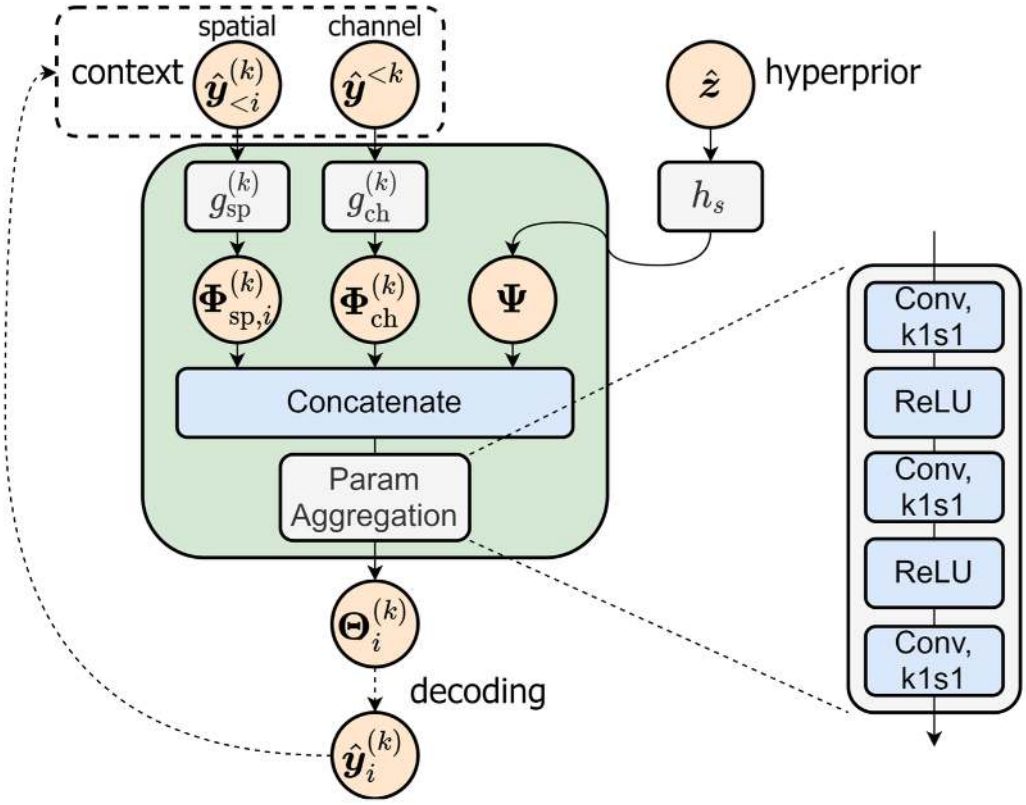


Figure 2.9: Space-Channel context model [He et al., 2022]

to the reconstructed image. These features make the ELIC model highly suitable for practical applications where rapid image retrieval and preview are essential.

Empirical results show that the proposed ELIC model achieves state-of-the-art performance in terms of both compression quality and speed. Specifically, it surpasses the VVC standard in terms of PSNR and MS-SSIM metrics while providing a significantly faster decoding process compared to other learned codecs.

Chapter 3

FLEXIBLE LUMA-CHROMA BIT ALLOCATION IN LEARNED IMAGE COMPRESSION FOR HIGH-FIDELITY SHARPER IMAGES

3.1 Introduction

In recent years, the field of image compression has seen significant advancements with the development of standards-based codecs like HEVC [Sullivan et al., 2012] and VVC [Bross et al., 2020], which primarily operate in the YCrCb 420 space and utilize YCrCb PSNR for evaluation. Contrastingly, most learned image compression models are designed around RGB 444 input, are optimized for RGB MSE, and are assessed through RGB PSNR [Ballé et al., 2017, Ballé et al., 2018, Minnen et al., 2018, Cheng et al., 2020]. This difference in approach has sparked a discourse on the efficacy of RGB PSNR as a measure of visual quality, particularly because image sharpness and detail are more closely tied to the luma component, to which the human visual system is more sensitive.

Despite learned codecs achieving similar RGB PSNRs across different input formats like RGB 444, YCrCb 444, and YCrCb 420, the reliance on RGB MSE optimization poses significant challenges. It limits precise bit allocation across luma (Y) and chroma (Cr, Cb) channels, which is crucial for enhancing image sharpness. Although there are prior works that optimize image/video codecs in YCrCb format [Egilmez et al., 2021, Ma et al., 2021], they focus on challenges posed by handling YCrCb 420 inputs and the alignment of luma and chroma channels. However, these studies typically did not provide the flexibility needed in network architectures to continuously adjust bit allocations between luma and chroma components, which could potentially improve overall visual quality.

Hence, optimizing learned codecs for YCrCb MSE not only facilitates more pre-

cise control over bit distribution between luma and chroma but also significantly impacts the visual quality of compressed images. This chapter of the thesis presents experimental results that underscore the importance of this optimization, showing that proper bit allocation is essential for achieving sharper images with higher Y-PSNR and VMAF scores, thereby challenging the prevailing emphasis on RGB MSE optimization in learned image compression models.

3.2 Flexible Compression in Luma-Chroma

Popular variational AE-based learned image compression methods train a set of models, one for each discrete R-D operating point. A continuous R-D curve is then interpolated from 6-8 such discrete points. Recently, new approaches have been proposed to achieve continuous rate adaptation to obtain a continuous R-D curve using a single model without extra training [Cui et al., 2021, Sun et al., 2021, Song et al., 2021]. In this work, we follow the approach in [Cui et al., 2021], which is shown in 3.1. The gain unit utilizes a matrix $M \in \mathbb{R}^{c \times n}$ comprising vectors $m_s = \{m_{s,0}, m_{s,1}, \ldots, m_{s,c-1}\}$ that scale the latent representation y channel-wise to enhance rate-distortion (R-D) performance and reduce quantization loss. After scaling and quantization, the inverse gain unit reverts the data to its original scale to ensure accurate reconstruction.

These components are integral to the autoencoder, enabling adaptive scaling and rescaling. This setup, known as the discrete variable rate (DVR) method, permits adaptation at specific points on the R-D curve. Additionally, continuous rate adjustment is facilitated through an exponent interpolation formula:

$$mv = (mr)^l \cdot (mt)^{1-l}, \quad m'_v = (m'_r)^l \cdot (m'_t)^{1-l} \tag{3.1}$$

where l is a real-valued coefficient that modulates bit rates between predefined settings. The continuously variable rate (CVR) method allows for smooth transitions along the R-D curve without the need for further training, thus enhancing compression flexibility and maintaining optimal R-D performance. In addition to using it to achieve continuous overall rate adaptation using a single model, we, for the first time, exploit it to achieve continuous bit allocation between luma and chroma

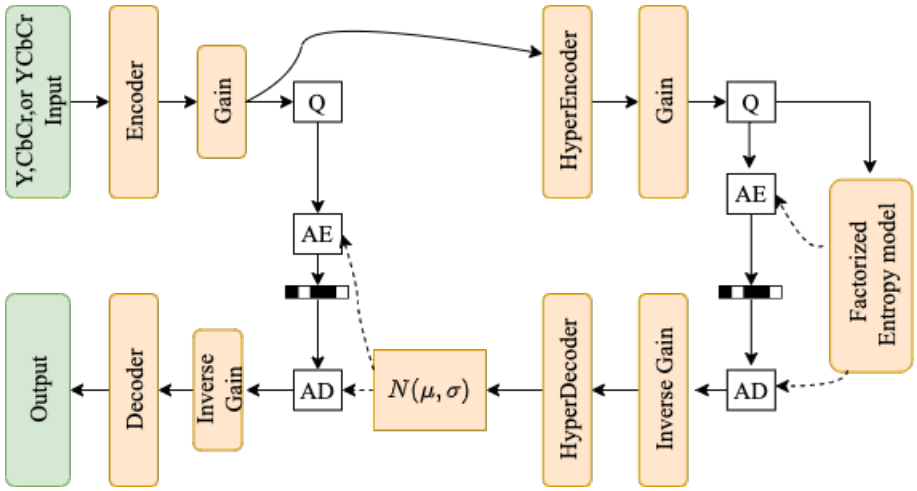


Figure 3.1: The gained VAE architecture, which is used for compressing luma only (one input channel), chroma only (two-channel input), and joint coding of luma and chroma.

components, similar to the effect of varying QP offset in classical standards-based codecs [Sullivan et al., 2012, Bross et al., 2020].

We propose two different solutions for flexible luma-chroma image compression with image-adaptive bit allocation between luma and chroma components to either obtain sharper images at fixed bpp or reduce overall bpp at fixed Y PSNR.

3.2.1 Separate Luminance and Chrominance Coding

In this approach, luma and chroma channels are coded separately using the same network architecture shown in Figure 3.1, which is the mean-scale architecture [Ballé et al., 2018] modified to add gain and inverse gain units to the encoder, decoder, and hyper-encoder/decoder modules. One network takes luma as input, while the other takes concatenation of Cr and Cb channels as input. The gain and inverse gain units enable continuous rate adaptation for both luma and chroma components separately, which allows us to fine-tune bit allocations to both luma and chroma components per image as described in Section 3.2.3.

Even though this method cannot exploit the correlation between luma and chroma components, it provides more flexibility compared to joint coding in Sec-

tion 3.2.2 in that with separate models, we can either allocate more bits to Y channel to achieve sharper images at a fixed bpp, or simply reduce chroma bits to reduce total bpp at a fixed Y channel quality. Since there is no need to align spatial dimension of luma and chroma components, this method can be directly used for input images with YCrCb 444 or YCrCb 420 color format.

3.2.2 Joint Coding using Gained VAE

In this approach, luma and chroma components are coded using a single model, where the input to the network is the concatenation of all three components. Joint coding of luma and chroma channels requires proper bit allocation between channels during training. In the literature, this is achieved by weighting the distortion of each component (e.g., 6:1:1 or 4:1:1) while computing the R-D loss. This means training a new model for each weight combination, which does not offer flexibility for continuous bit allocation between luma and chroma to fine-tune color fidelity at a fixed Y PSNR to optimize bits per pixel per image. Moreover, the best weights for optimal luma-chroma bit allocation may be image-dependent and may even be different for each rate point.

Since we only have one set of gain and inverse gain vectors for both luma and chroma in the joint model, in order to allow for image-adaptive luma-chroma bit allocation, we train 4 models, each tuned to a pre-determined Y channel rate, where the gain and inverse gain units for each model are optimized at 4 chroma quality levels corresponding to different Y, Cb, and Cr distortion weights, i.e., 6:1:1, 4:1:1, 2:1:1, and 1:1:1, for levels 1-4, respectively. Hence, we can choose any chroma rate in between successive chroma quality levels for each of the four fixed Y channel rate points by the virtue of gained VAE architecture.

Unlike separate luma-chroma coding, this method utilizes cross-channel correlations. However, it offers less flexibility, since it only allows fine tuning chroma bit allocations at a fixed Y PSNR during inference. As a result, the joint coding model can save bpp compared to RGB optimized model at a fixed Y PSNR, but unlike separate coding it cannot optimize image sharpness at fixed bpp.

3.2.3 Fine-tuning Image Sharpness and Color Fidelity

We observe that for each luma PSNR/bpp level, there is a range of chroma bpp, where we can either increase luma bpp (hence, image sharpness) at the same total bpp or save chroma bpp without degrading perceptual image quality. For both separate and joint coding, we used a set of 8 levels (interpolated from 4 in the case of joint coding) for fine-tuning chroma quality, where the bpp cost of chroma components approximately doubles between levels L_i and L_{i+2}.

The block diagram for the proposed rate adaptation method is shown in Fig. 3.2. The procedure for fine-tuning luma-chroma bpp allocation can be summarized as:

1) Pick a luma quality level and encode the image with the corresponding chroma quality level.

2) Compute the percentage of bpp allocated to luma and chroma components.

3) If the percentage of bpp for chroma is higher than τ percent, then reduce the bits allocated to chroma components by α percent.

At this point, we have two options: We can either stop after reducing the chroma quality and achieve chroma bpp savings or use those saved bpp to increase the quality level of the Y channel to obtain a sharper image at the same bpp. In our experiments, we empirically set τ and α values to 8 and 20, respectively.

The only difference in fine-tuning of bit allocation in separate vs. joint coding is in joint coding; instead of starting with different chroma quality level for each Y model, we start with a fixed quality level since chroma qualities are learned relative to the Y quality. Moreover, in the joint coding approach, there is no continuous rate adaptation for the Y channel. Therefore, we can only obtain additional bit savings without degrading the perceptual quality. It is also important to note that this procedure can be applied for optimizing YCrCb PSNR for given distortion weights for each component.

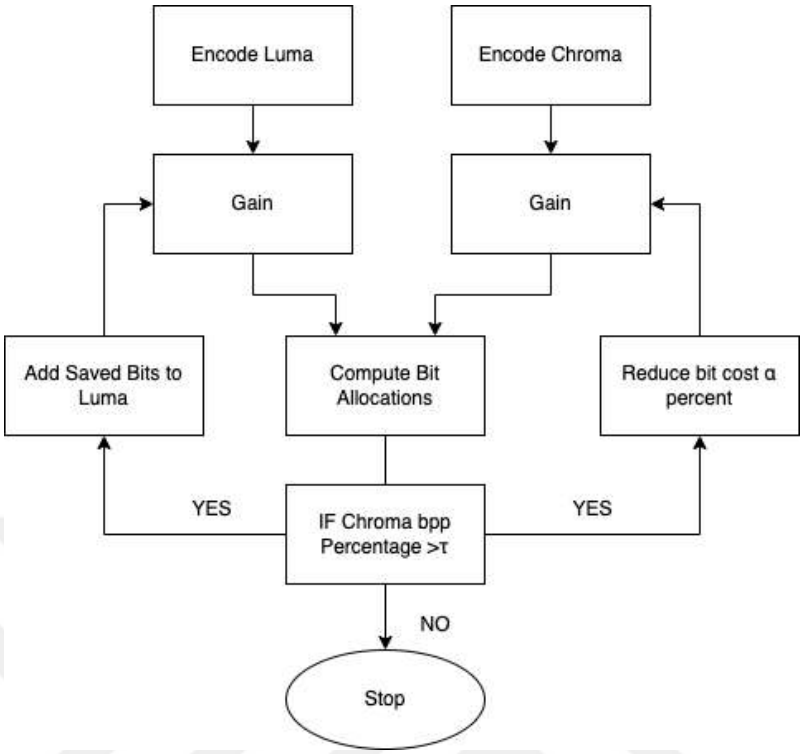


Figure 3.2: Flowchart for inference-time luma-chroma rate adaptation.

3.3 Experimental Results

We evaluate the R-D performance of the proposed separate and joint luma-chroma coding models on the Kodak dataset, which consists of 24 uncompressed 768×512 RGB images. The rate is measured by bits per pixel (bpp), and the distortion is measured by RGB and Y PSNR. We also provide VMAF scores[vma,]. Although, the VMAF has originally been developed as a quality measure for video coding, it employs VIF [Sheikh and Bovik, 2006] and DLM/ADM [Li et al., 2011] image quality metrics and a motion metric (the temporal difference between frames). For still images or static video shots, motion metric becomes zero and VMAF produces scores based on VIF and DLM. Hence, it has also been used as a still image quality metric in the literature.

3.3.1 Training Details

We used the PyTorch framework to implement all models. For all models, we used the modified version (with the gain and inverse gain units) of the mean-scale hyperprior image compression model in the CompressAI library [Bégaint et al., 2020]. We train our models on the Vimeo-90k dataset [Xue et al., 2019].

For separate channel coding, we trained the luma network for four different λ = 0.0018, 0.0035, 0.0067, and 0.0130 to construct the RD curve. For the chroma network, we used eight different λ = 0.00025, 0.00045, 0.0009, 0.0018, 0.0035, 0.0067, 0.0130, and 0.0250 in order to have more flexibility while selecting the chroma rate for a given luma rate. For joint coding of luma and chroma components, we used the same lambda values with the luma model, and we used 4 different distortion weights α = 1, 0.25, 0.1, and 0.025 for each of those models, where the loss function is

$$L = R + \lambda * (MSE_Y + \alpha * MSE_{Cr} + \alpha * MSE_{Cb}).^1$$

For training the networks, we followed the same procedure in [Cui et al., 2021]. In separate coding, we randomly select the index s of gain vectors, which runs 1 to 4 and 1 to 8 for luma and chroma models, respectively. In each iteration, we randomly select s to obtain the gain/inverse vector pairs and corresponding λ value, which optimizes gain/inverse vector pairs with the corresponding λ value in order to adapt to different bit-rates. In joint coding, the same procedure is followed, but instead of selecting the corresponding lambda values, we select the corresponding distortion weights for gain/inverse vector pairs.

During training, we take 256 $\times$ 256 random crops for data augmentation. The mini-batch size is set to 16, and optimization is performed for 2M steps except for the chroma model, which is trained for 1M steps. Adam optimizer [Kingma and Ba, 2015] is used with the initial learning rate set to 10^{-4}, reduced to 10^{-5} at the half of training. We also applied gradient clipping on the norm of the gradients with the maximum norm value set to 1.

3.3.2 Comparison with RGB Optimized Model

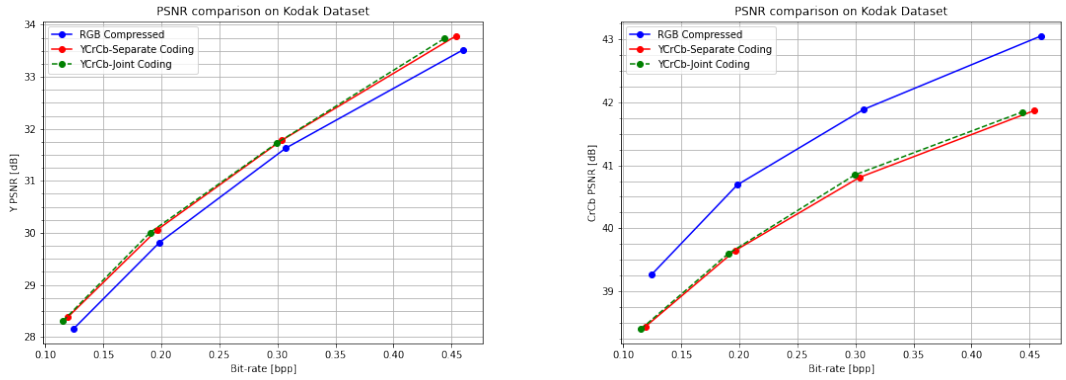
We compare the performances of our separate and joint coding models optimized for YCrCb MSE with that of the mean-scale architecture optimized for RGB MSE [Ballé et al., 2018] which uses exactly the same architecture with our models except for gain and inverse gain units. The Y PSNR improvements for randomly selected images are tabulated in Table 4.4.3.

Table 3.1: RGB and Y PSNR values for RGB-MSE and YCrCb-MSE optimized models for randomly selected different test images at the same bpp using the model for rate point 2. "Before" and "after" refer to luma-chroma rate adaptation.

	RGB-MSE optimized		YCrCb-MSE optimized			
			before		after	
PSNR (dB)	RGB	Y	RGB	Y	RGB	Y
Kodim02	30.37	31.41	30.30	31.50	30.25	31.65
Kodim07	31.09	31.89	30.95	32.24	30.85	32.38
Kodim15	30.58	31.30	30.40	31.41	30.15	31.80
Kodim22	28.91	29.46	28.80	29.60	28.75	29.75

The data in the table shows that YCrCb-MSE optimization improves the PSNR values of the Y channel more than RGB-MSE optimization. This improvement is seen in many images, both before and after adaptation. The analysis highlights that YCrCb MSE effectively maintains and enhances luma quality, which is important for image compression. By optimizing the Y channel, this method efficiently manages chroma bit allocation, which usually consumes too many resources without significant improvements in image quality. These results show the benefits of using YCrCb MSE optimization when chroma components are less important. It helps use resources better and improves overall image quality by avoiding unnecessary allocation of bits to less important chroma components.

The average R-D curves for luma and chroma components over the entire Kodak dataset are shown in Figure 3.3, where the horizontal axis is the total bitrate, i.e.,



(a) (b)

Figure 3.3: Rate-Distortion Curves of RGB (a) vs YCrCb (b) obtimized models on Kodak Dataset.

the sum of R, G, B or Y, Cr, Cb bits in all graphs. The results demonstrate that by sacrificing a slight performance drop on chroma, which has no visible effects, we can gain 0.25 dB PSNR on average on the luma component, which results in sharper and perceptually preferable images. Moreover, Figure 3.4 shows that our YCrCb optimized models outperform RGB optimized models in terms of VMAF scores as expected since VMAF only evaluates the luma component. We note that there is no noticeable difference between the performances of our separate and joint luma-chroma coding models.

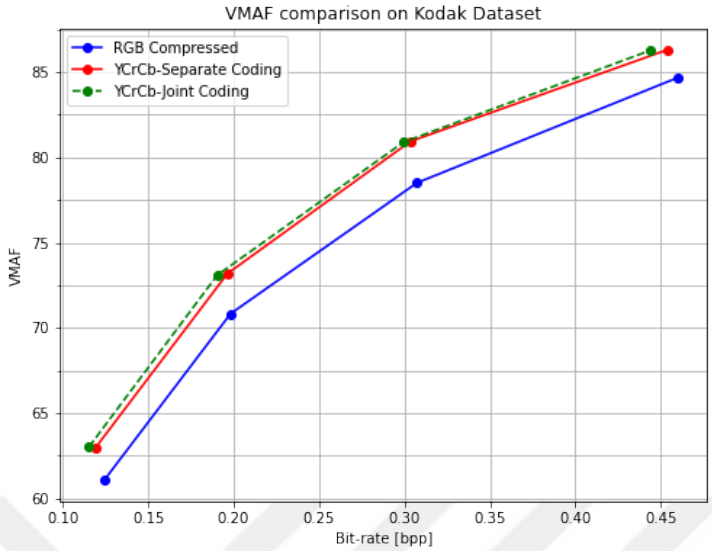
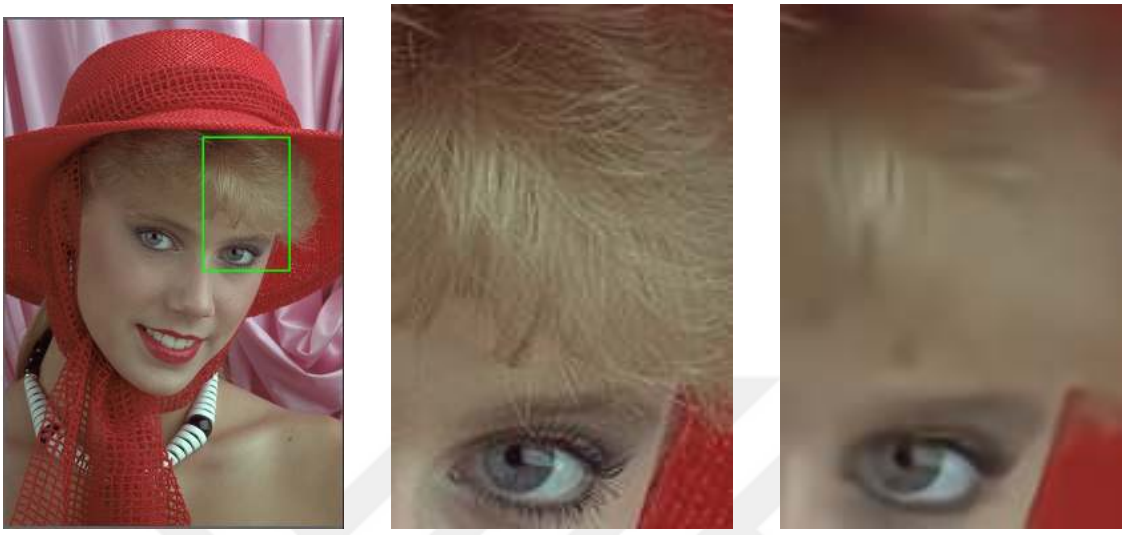


Figure 3.4: Comparison of VMAF scores on Kodak Dataset.

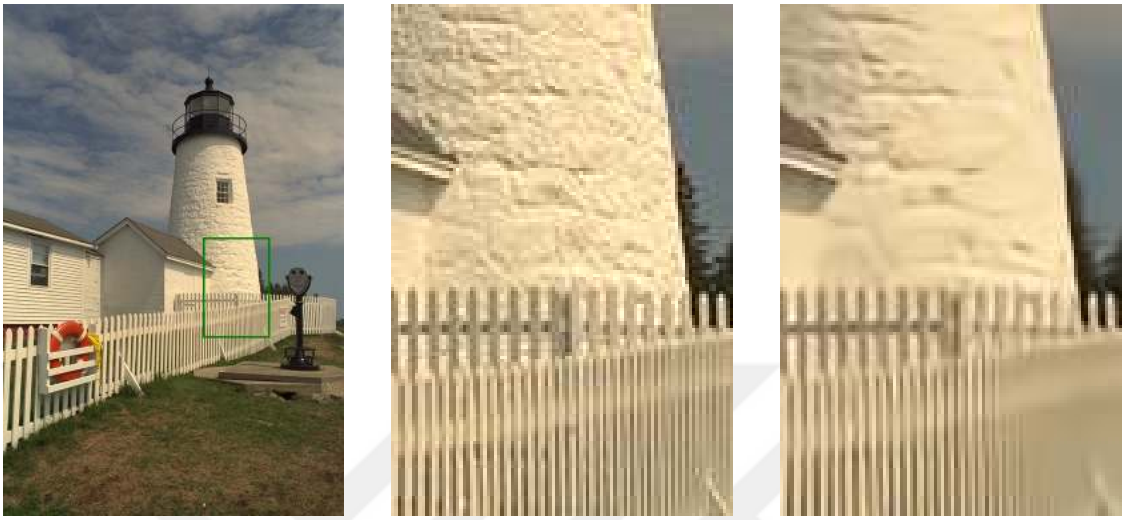
For qualitative visual comparison, Figures 3.5,3.6 shows the visual quality of images obtained by RGB and YCrCb optimized models. RGB optimized model produces noticeably blurry images that lack texture details. On the other hand, YCrCb optimized model preserves more texture and produces perceptually preferable images since Y-PSNR is a better indicator of perceptual quality.



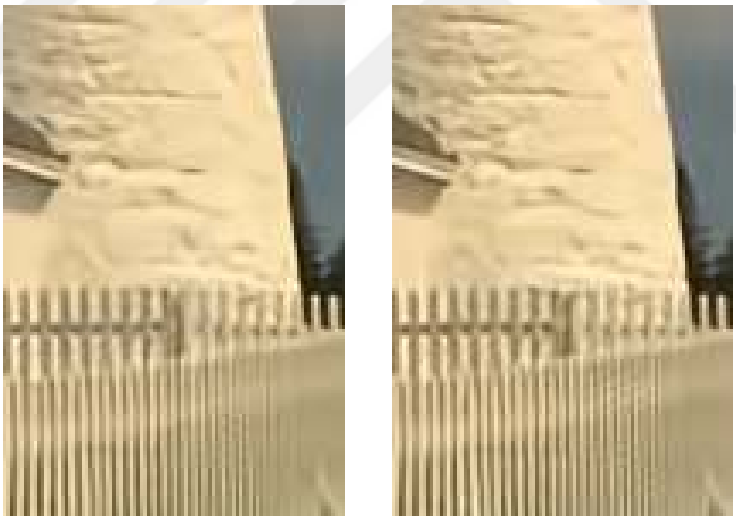
(a) (b) (c)

(d) (e)

Figure 3.5: Visual evaluation on a example from Kodim04 from kodak dataset at ≈0.12 bpp. (a) original image, (b) original crop
(c) RGB optimized: Y-PSNR 32.22dB, RGB-PSNR 31.95dB, VMAF 71.08,
(d) YCrCb optimized (before): Y-PSNR 32.61, RGB-PSNR 31.32, VMAF 71.12,
(e) YCrCb optimized (after): Y-PSNR 32.71, RGB-PSNR 31.28, VMAF 76.79.



(a) (b) (c)



(d) (e)

Figure 3.6: Visual evaluation on a example from Kodim19 from kodak dataset at ≈0.16 bpp. (a) original image, (b) original crop
(c) RGB optimized: Y-PSNR 29.69, RGB-PSNR 29.43, and VMAF 71.87,
(d) YCrCb optimized (before): Y-PSNR 29.72, RGB-PSNR 29.38, and VMAF 72.15,
(e) YCrCb optimized (after): Y-PSNR 29.90, RGB-PSNR 29.33, and VMAF 74.73

Chapter 4

LEARNED IMAGE COMPRESSION WITH DIFFUSION-BASED POST-PROCESSING

4.1 Introduction

It has been demonstrated that diffusion models produce visually pleasing textures. They have also been used as an unsupervised solver for general linear inverse problems such as deblurring, super-resolution, inpainting, and colorization by utilizing a pre-trained denoising diffusion model [Kawar et al., 2022a]. Later [Kawar et al., 2022b] applied a similar method for JPEG artifact correction and image dequantization problems and caught the performance of domain-specific GANs. However, it is not straightforward to apply DDRM to more advanced codecs, such as VVC [Bross et al., 2020], due to complications arising from their features, such as intra-prediction.

Motivated by these advancements, we propose a wavelet-based codec using a fixed linear transform and a learned entropy model combined with diffusion-based post-processing. When using an invertible transform (learned or fixed) instead of variational autoencoders (VAE), the decoder cannot learn to compensate for the quantization error; hence, a separate post-processing stage is required to obtain results that are competitive with the state of the art codecs [Ma et al., 2022, Xie et al., 2021a]. We show that using diffusion-based post-processing, instead of a CNN, we can considerably enhance the perceptual R-D performance without compromising the PSNR R-D performance. Our experimental results show that sharper images with better perceptual quality measures and YCrCb PSNR can be obtained compared to state-of-the-art classic and learned codecs.

4.2 Wavelet-domain image compression with a learned entropy model

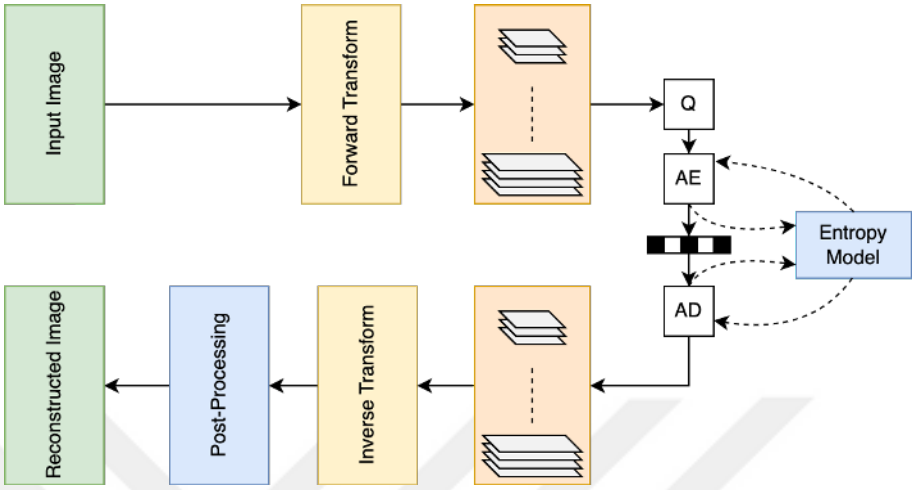


Figure 4.1: Overview of the wavelet based codec. "Q", "AE" and "AD" stand for quantization, arithmetic encoding and arithmetic decoding, respectively. The forward and inverse transforms are CDF 9/7 filters implemented by lifting. Post-processing module is either a CNN or a pre-trained diffusion model.

4.2.1 Subband Decomposition by a Fixed Transform

The Cohen-Daubechies-Feauveau (CDF) 9/7 is a symmetric biorthogonal wavelet transform that is preferred for image coding due to its mathematical properties and has been incorporated into the JPEG2000 standard [Skodras et al., 2001].

The CDF 9/7 transform is implemented by a lifting scheme, which consists of three primary steps: split, prediction, and update. In the case of a 1-D signal, the signal is partitioned into its even x_e and odd x_o indexed samples, which can be used to predict each other. In the case of a 2-D image, the image is initially partitioned into two parts L, H, based on the odd and even rows. The same operation is carried out for both the L and H bands in the column direction, resulting in four subbands: LL, HL, LH, and HH. This same operation is repeatedly applied to the LL band to obtain a multi-scale pyramid decomposition. After undergoing D levels of decomposition, the signal is decomposed into 3D+1 subbands denoted as $S = \{s_1, s_2, .., s_{3d+1}\}$, or alternatively as $S = \{HH_1, LH_1, HL_1, HH_2, ..., HL_d, LL_d\}$.

To reverse the process, the inverse lifting scheme is applied to the subbands, and then the subbands are merged for perfect reconstruction of the original image.

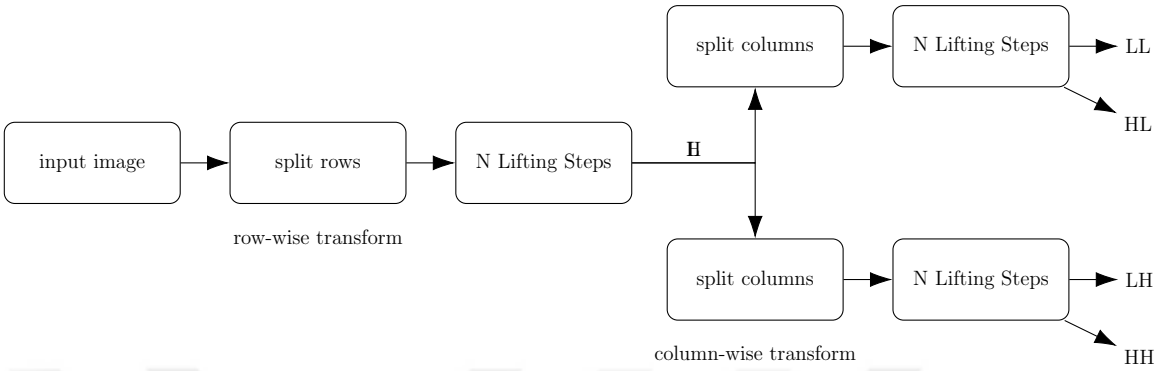


Figure 4.2: 2-D wavelet transform with lifting implementation

4.2.2 Learned Entropy Model in the Wavelet-Domain

The wavelet transform coefficients undergo quantization to obtain discrete indices, which are then arithmetic encoded using an entropy model illustrated in Figure 4.3. We parameterize the distribution of quantized wavelet coefficients by a discretized mixture of Gaussians as proposed by Cheng et al. [Cheng et al., 2020].

While estimating the parameters of the entropy model, previously encoded subbands are used as condition c_t to estimate the entropy model parameters of the current subband s_t in order to exploit the correlation between subbands, which enhances the compression efficiency. For the subband HL_t, c_t is set to the subband LL_t; for the subband LH_t, c_t is the concatenation of the subbands LL_t and HL_t, etc. Encode/Decoder order of the subbands are shown in 4.4, each bend is conditinally encoded and decoded using previous subbands. For the LL band we use hyperprior networks in addition the the context model. We have also tested the usage of hyperprior architecture for other bands but did not see any improvement.

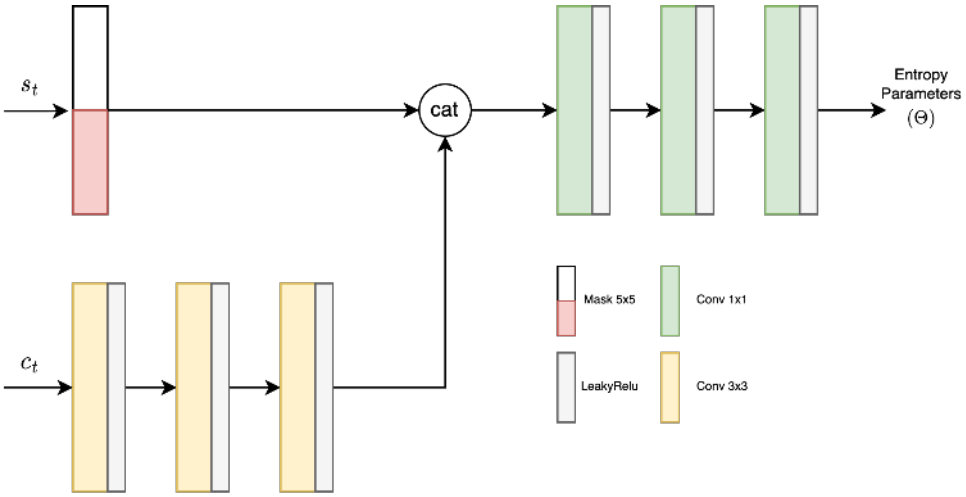


Figure 4.3: The conditional entropy model for coding the wavelet coefficients, where s_t and c_t denote the current and conditioning wavelet bands, respectively. HL_t band is conditioned LL_t band; LH_t band is conditioned LL_t and HL_t bands, and HH_t band is conditioned LL_t, HL_t and LH_t bands.

4.3 Post-Processing via Diffusion Models

Our post-processing step is similar to the procedure in [Kawar et al., 2022b], which applies the general DDRM process [Kawar et al., 2022a] to the JPEG artifact correction task. That is, we employ the iterations

$$\acute{x_t} = f_\theta^{t+1}(x_{t+1}) - IW(Q(FW(f_\theta^{t+1}(x_{t+1})) + IW(y) \tag{4.1}$$

$$x_t = \sqrt{\alpha_t}\left(\eta_b \acute{x_t} + (1-\eta_b) f_\theta^{t+1}(x_{t+1})\right) + \sqrt{1-\alpha_t}\epsilon_t \tag{4.2}$$

where $f_\theta^t(\cdot)$ denotes the diffusion model at step t, FW and IW stand for forward and inverse wavelet transforms, Q is quantization, y is quantized wavelet coefficient, $\epsilon_t \sim N(0, \boldsymbol{I})$ is a random vector, and η_b is a hyperparameter, which is set to 0.4. Eqn. 4.1 generates a corrected y value and Eqn. 4.2 computes the value for the next iteration. Following [Kawar et al., 2022b], we use 20 evenly spaced diffusion steps and start the diffusion process at $t = 300$ by adding noise (see [Kawar et al., 2022b]) to the initial reconstruction. Since the sampling is probabilistic (due to added noise

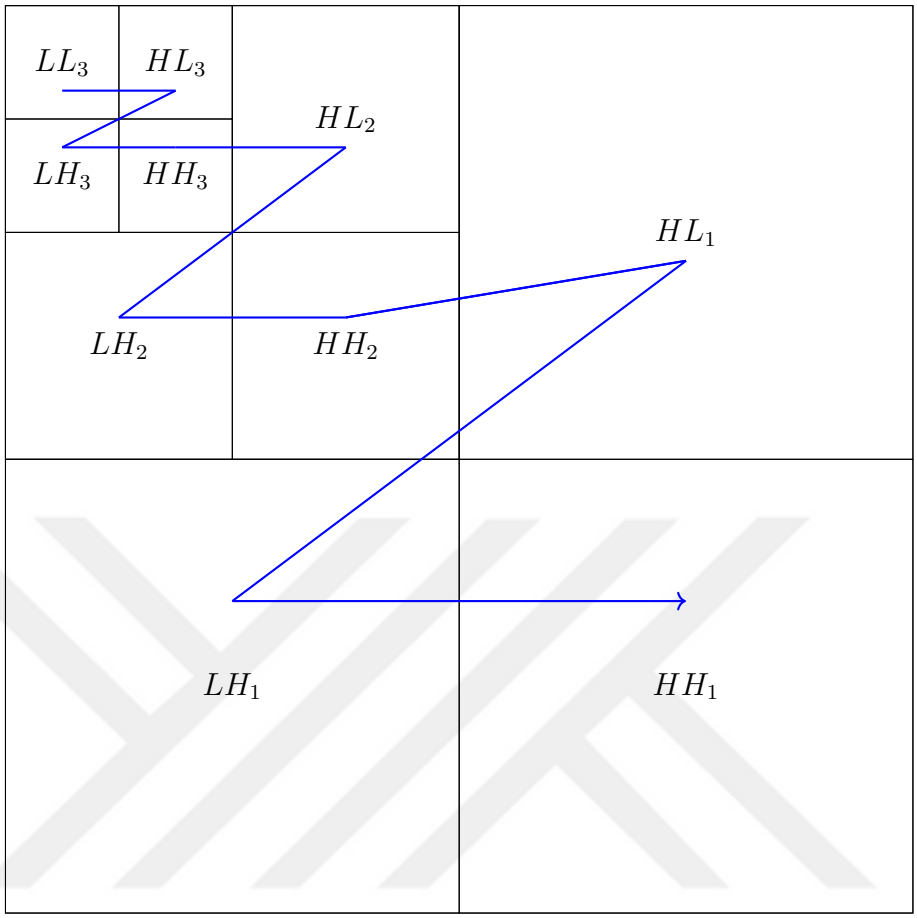


Figure 4.4: Encode/Decode order of subbands. Each subband is conditionally encoded/decoded using previous subbands.After decoding the HH_t, LL_{t-1} is reconstructed and same conditioning order is performed

and ϵ_t), we apply the diffusion restoration process multiple times and then average the results. In the following, we denote a single output sample by Diffusion (S) and average of multiple samples by Diffusion (A).

For the diffusion model $f_\theta^t(\cdot)$, we use pre-trained model, available in the authors' official repository [Dhariwal and Nichol, 2021], which is trained on 256×256-pixel ImageNet training images [Deng et al.,] for a diffusion schedule of 1000 steps.

4.4 Experimental Settings & Results

4.4.1 Training Details

We jointly trained the entropy model and CNN post-processing network on the Vimeo-90k dataset [Xue et al., 2019] for six different rate points. For CNN architecture, we use a basic residual network similar to RCAN proposed in [Lim et al., 2017]. For the experiment with no post-processing, only the entropy model is trained. During training, we apply 256 $\times$ 256 random cropping for data augmentation. The mini-batch size is set to 16 and optimization is performed for 500k steps. Adam optimizer [Kingma and Ba, 2015] is used with the initial learning rate set to 1e-4 and dropped to 1e-5 at 250k steps. We also apply gradient clipping on the norm of the gradients with maximum norm value is set to 1.

4.4.2 Experimental Setup

We evaluate the performance of our wavelet-based codec using the widely adopted Kodak image dataset, which comprises 24 uncompressed images. To evaluate the rate-distortion performance, we employ the peak signal-to-noise ratio (PSNR) as a quality metric, which is computed separately for the luma and chroma channels. The bits-per-pixel (BPP) is used to measure the rate of compression. In addition to PSNR metric, we also provide LPIPS [Zhang et al., 2018] and DISTS [Ding et al., 2022] scores as perceptual metrics.

4.4.3 Comparative Results

We compare the performance of the our wavelet-based codec with diffusion-based post-processing with CNN based post-processing and no post-processing. We provide the results of VVC Intra using version 18.0 [Bross et al., 2020] and deep learning based codecs including Balle'18 [Ballé et al., 2018], Minnen'18 [Minnen et al., 2018], Cheng'20 [Cheng et al., 2020] which are available in CompressAI library [Bégaint et al., 2020] and Xie'21 [Xie et al., 2021b], and Lu'22 [Lu and Ma, 2022] using authors' official implementation.

Figure 4.5 presents a comparison of the rate-distortion performance of our codec with VVC and other deep learning-based codecs. In terms of luma PSNR, the Diffusion (A) method performs comparably to CNN post-processing and the Cheng'20 model, whereas Diffusion (S) exhibits significantly poorer performance. Notably, diffusion sampling excels in correcting compression artifacts in chroma channels, significantly outperforming CNN post-processing, VVC, and other learned codecs. Since luma and chroma components are coded separately in our wavelet-based codec, we can adjust the bit allocation between these channels by using QP offset to further leverage the diffusion process's superior ability to correct chrominance artifacts and enhance the overall performance.

Furthermore, the LPIPS and DISTS curves demonstrate that the diffusion model produces perceptually better images than the RGB optimized learned codecs and CNN post-processing, while exhibiting similar RD performance in terms of PSNR. Moreover, it can be observed that Diffusion (A) provides better RD performance in the PSNR metric compared to Diffusion (S) and does not affect the LPIPS metric while increasing the DISTS scores.

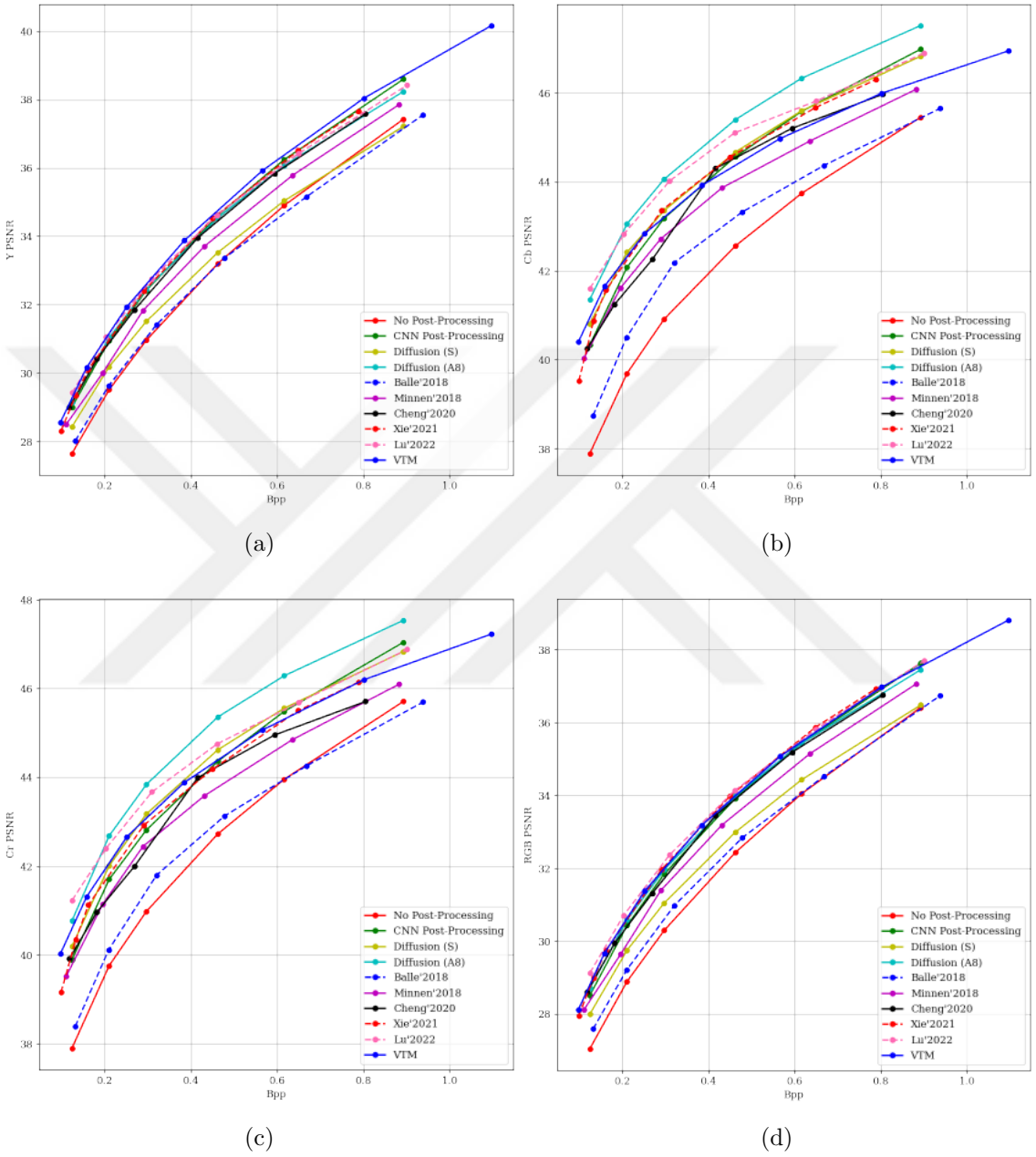
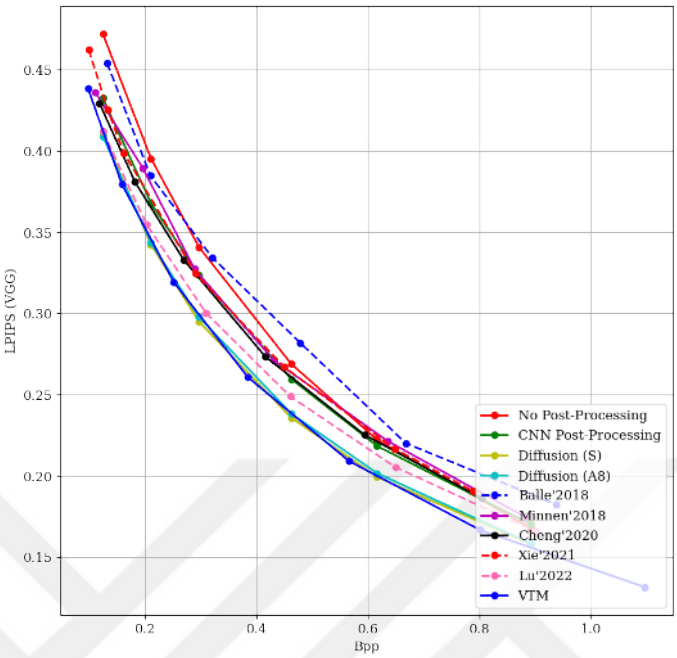
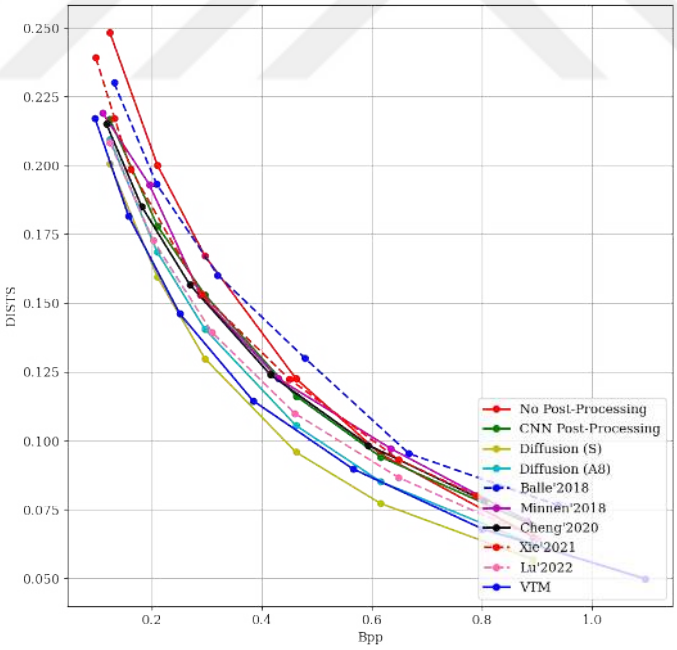


Figure 4.5: Comparison of the rate-distortion performance for various codecs on the Kodak dataset across different color channels. Subfigure (a) shows the Y PSNR versus bits per pixel (bpp), subfigure (b) Cb PSNR versus bpp, subfigure (c) Cr PSNR versus bpp, and subfigure (d) RGB PSNR versus bpp.



(a)

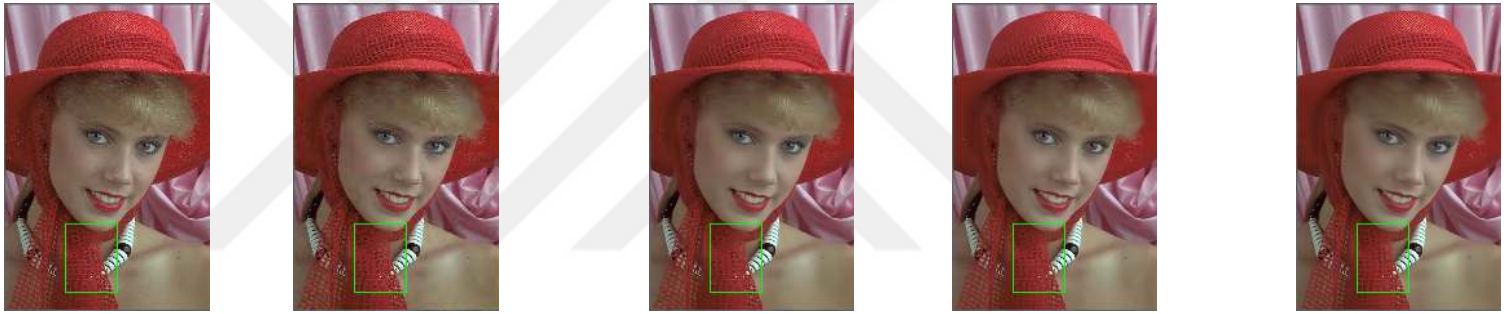


(b)

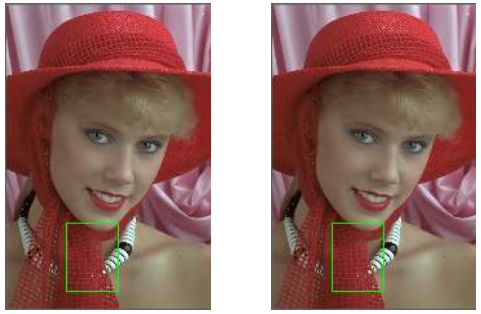
Figure 4.6: Comparison of the perceptual metrics for various codecs on the Kodak dataset. Subfigure (a) displays the LPIPS (VGG) metric versus bpp, while subfigure (b) shows the DISTS metric versus bpp.

For a qualitative visual comparison, Figure 4.7 illustrates the visual quality of images obtained by different algorithms at similar bit rates. Upon careful inspection of the cropped sections, it can be observed that CNN-based post-processing introduces some blur to the inverse transform, resulting in perceptual quality degradation of the images. In contrast, diffusion-based post-processing effectively corrects quantization artifacts, producing visually pleasing images. Moreover, compared to VVC, diffusion-based post-processing preserves more texture details. When comparing Diffusion (S) and Diffusion (A8), even though the outputs look similar, some blur is still noticeable in the high-frequency regions.

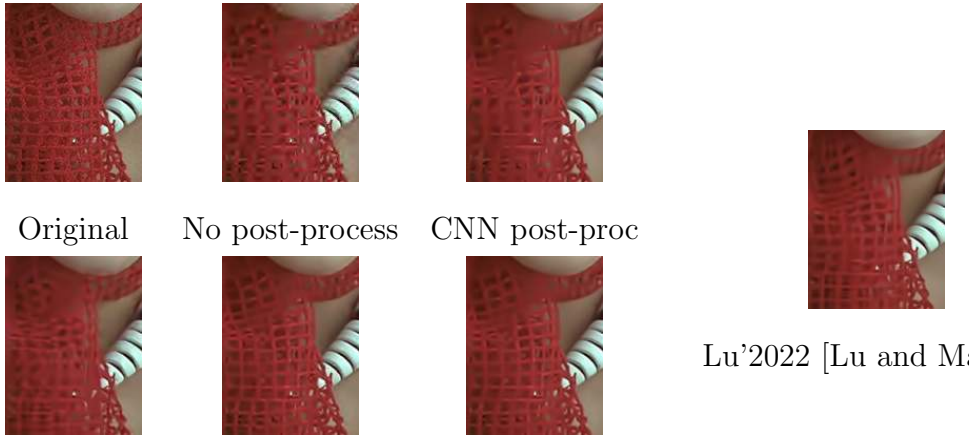
The LPIPS and DISTS scores (lower is better) provided in Table 1 verify that the perceptual quality of images generated by diffusion-based post-processing outperforms that of CNN-based post-processing and VVC. Although Lu'2022 achieves the highest RGB and Y PSNR, its perceptual quality lags behind that of diffusion post-processing, as also observed in Figure 4.7. When comparing the Cr and Cb PSNR of these methods, we observe that even without averaging, the diffusion model performs much better than CNN. While CNN post-processing improves Cb PSNR to 44.48 dB, Diffusion (S) achieves 46.32 dB, and Diffusion (A) achieves 47.11 dB.



(a) Original (b) No post-processing (c) CNN post-processing (d) VVC VTM (e) Lu'2022 [Lu and Ma, 2022]



(f) Diffusion (S) (g) Diffusion (A8)



Original No post-process CNN post-proc

VVC VTM Diffusion (S) Diffusion (A8)

Lu'2022 [Lu and Ma, 2022]

Figure 4.7: Visual results on KODIM 04 (full image and zoom-in green box) from the KODAK dataset at approximately ≈ 0.12 bpp. Inspection of zoom-ins show that Diffusion (S) and Diffusion (A8) recreate the texture details much better than all others. Unlike GAN-based approaches, this is not just hallucination of details because we also maintain fidelity (do not sacrifice PSNR).

Table 4.1: PSNR, LPIPS and DISTS values for KODIM04 from the Kodak dataset at ≈ 0.12 bpp. Best results are shown in bold. Diffusion (S) is a single sample obtained using the pre-trained diffusion model and Diffusion (A8) is the average of 8 samples.

	RGB PSNR	Y PSNR	Cr PSNR	Cb PSNR	LPIPS (Alex)	LPIPS(VGG)	DISTS
VVC VTM	31.384143	32.181229	39.798054	45.432324	0.282612	0.356193	**0.140927**
Lu'2022 [Lu and Ma, 2022]	**31.666540**	**32.306499**	39.913402	47.077681	0.296379	0.359620	0.160952
Wav-No post-process	29.897149	30.842479	37.279679	41.904940	0.309356	0.403134	0.178747
Wav-CNN post-process	31.109262	31.935107	39.342350	44.481274	0.274785	0.374371	0.160410
Wav-Diffusion (S)	30.936644	31.602342	39.955317	46.321542	**0.265502**	0.344016	0.146464
Wav-Diffusion (A8)	31.561743	32.221240	**40.685881**	**47.114457**	0.266754	**0.341852**	0.157266

4.4.4 Ablation Study on Entropy Modelling

Due to the autoregressive nature of the context model and the high spatial resolution of the subbands, the decoding speed of our codec is not comparable to VAE-based codecs. To improve decoding speed, we tested the performance of two different approaches.

First, we replaced the Z-order subband conditioning with tree order subband conditioning, which allows parallel decoding of the HL, LH, and HH bands, resulting in a 3x increase in decoding speed. Figure 4.8 illustrates the tree order conditioning of subbands. In this approach, each subband is conditionally encoded/decoded using the previous LL band. After decoding the HL_t, LH_t, and HH_t bands in parallel, the LL_{t-1} band is reconstructed, and the same conditioning order is applied recursively.

Additionally, we examined the use of a Checkerboard Context Model (CCM) for entropy coding. The CCM allows for parallel decoding and achieves a decoding time comparable to non-autoregressive VAE-based approaches. In the CCM model, all masked convolution operations are replaced with CCM, except for the LL band, as there are no conditioning signals available during the encoding/decoding of the LL band.

Figure 4.9 shows the rate-distortion performance of the wavelet-based codec using different entropy models. The baseline model uses an autoregressive context model with Z-order subband conditioning. Compared to the baseline model, tree order subband conditioning increases the rate by 3.57%, while the CCM increases the rate by 9.55%. This demonstrates the trade-off between decoding speed and rate efficiency. However, the increase in rate can be justified by the significant improvements in decoding speed, particularly in scenarios where real-time processing is critical.

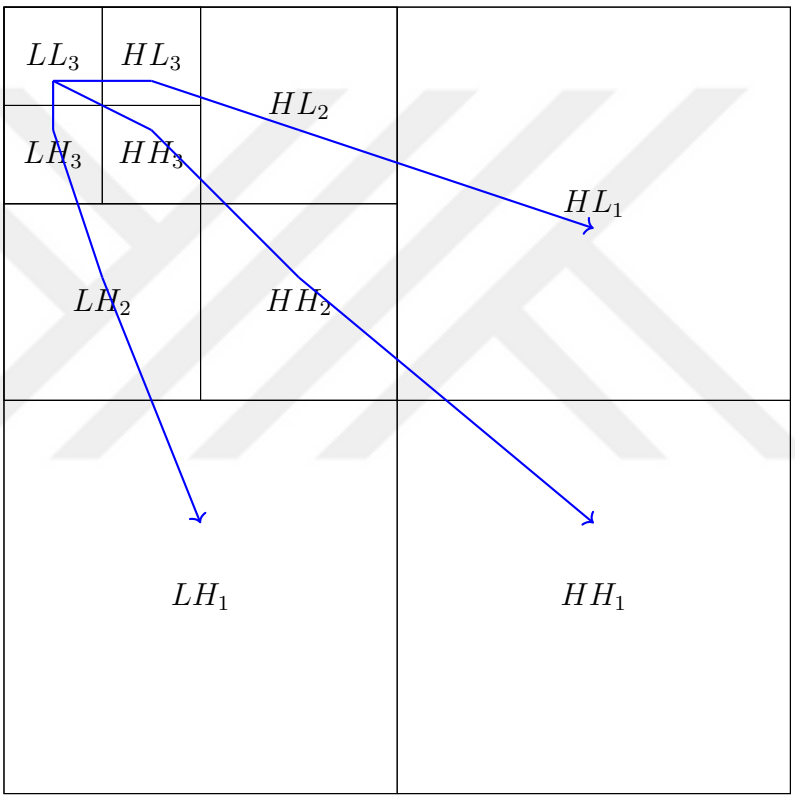


Figure 4.8: Encode/Decode order of subbands in tree order conditioning.

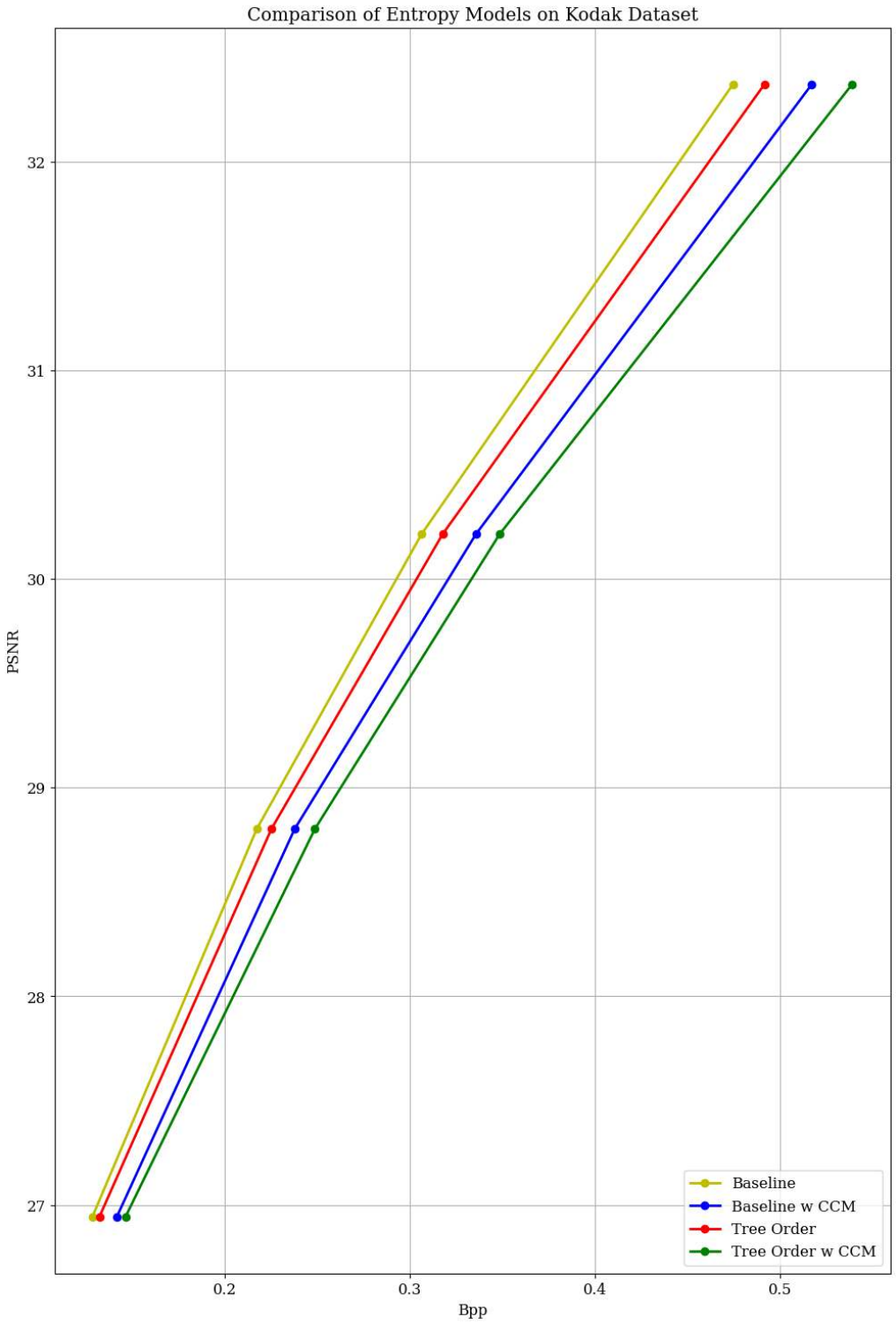


Figure 4.9: Rate Distiortion Performance of wavelet based codec with different entropy models on Kodak dataset

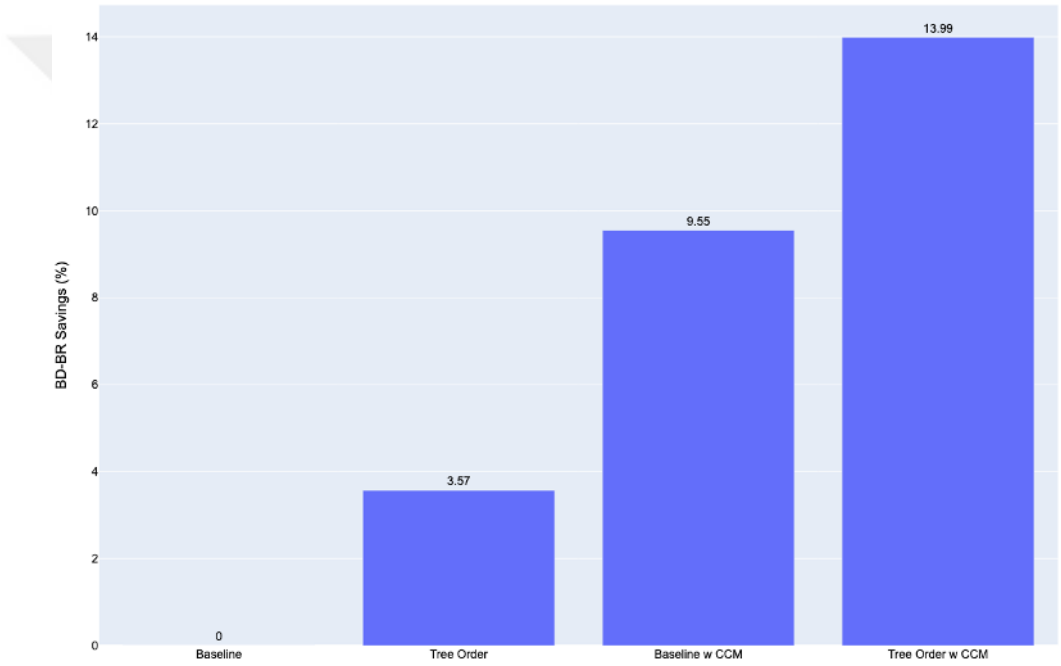


Figure 4.10: Rate savings of wavelet based codec with different entropy models on Kodak dataset compared to baseline

Chapter 5

CONCLUSION

This thesis explores advanced methods in learned image compression, focusing on flexible luma-chroma bit allocation for luminance and chrominance components, and diffusion-based post-processing techniques.

The flexible luma-chroma coding method introduced in this thesis provides two distinct architectures for separate and joint coding of luma and chroma components and shows that end-to-end optimization of learned image compression models for YCbCr MSE is more advantageous compared to optimization for RGB MSE since the former allows for flexible, image-adaptive bit allocation between luma and chroma channels during encoding at inference time to obtain sharper images at the same bpp or alternatively save bits without degrading perceptual image quality.

Additionally, we developed a diffusion-based post-processing method that integrates a fixed, invertible transform with a learned entropy model. The findings show that diffusion-based post-processing is clearly superior to CNN-based post-processing for image compression. Furthermore, diffusion-based post-processing coupled with a wavelet-based codec significantly improves the visual quality of images while maintaining the PSNR-based rate-distortion performance compared to the state of the art classic VVC-VTM and learned codecs. In comparison to the state-of-the-art LIC approaches optimized for MSE, our codec provides better perceptual quality while maintaining the PSNR performance. Unlike GAN-based approaches, this is not just a hallucination of details because our method also maintains fidelity (without sacrificing PSNR).

Despite the clear advantages in visual quality offered by the proposed wavelet-based codec, there are notable limitations. One significant drawback is the decoding speed, particularly when combining the wavelet-based codec with diffusion-based post-processing. The serial nature of autoregressive entropy modeling presents a

challenge for real-world applications. While we explored parallel entropy models to mitigate this issue, they tended to reduce rate-distortion performance. Additionally, the inherently sequential nature of the diffusion process, which requires multiple iterations to refine the image, poses challenges for real-time applications.

To address these limitations, future research could explore several promising directions. For instance, improving entropy modeling for wavelet-based codecs could enable parallel decoding without compromising rate-distortion performance. Additionally, investigating conditional diffusion models may facilitate more targeted post-processing, potentially reducing the number of required diffusion steps. This would not only accelerate the process but could also enhance image quality by conditioning the model on additional context or features. Exploring advanced sampling strategies within the diffusion process could also yield significant improvements. These techniques would allow the model to allocate computational resources more efficiently, reducing the overall computational load and improving the speed of the post-processing stage.

In conclusion, while the methods proposed in this thesis represent a significant advancement in learned image compression, particularly in terms of visual quality and flexible bit allocation, further research is necessary to address their limitations in decoding speed and efficiency. By investigating better entropy modeling for wavelet-based codecs and optimizing the diffusion process, future work could develop models that not only produce superior visual quality but also operate efficiently enough for real-time applications.

BIBLIOGRAPHY

[vma,] VMAF - Video Multi-Method Assessment Fusion. *https://github.com/Netflix/vmaf*.

[Agustsson et al., 2017] Agustsson, E., Mentzer, F., Tschannen, M., Cavigelli, L., Timofte, R., Benini, L., and Van Gool, L. (2017). Soft-to-hard vector quantization for end-to-end learning compressible representations. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1141–1151, Red Hook, NY, USA. Curran Associates Inc.

[Ballé et al., 2017] Ballé, J., Laparra, V., and Simoncelli, E. P. (2017). End-to-end optimized image compression. In *Int. Conf. Learning Representations (ICLR)*.

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

[Ballé et al., 2018] Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N. (2018). Variational image compression with a scale hyperprior. In *Int. Conf. on Learning Representations (ICLR)*.

[Bégaint et al., 2020] Bégaint, J., Racapé, F., Feltman, S., and Pushparaja, A. (2020). Compressai: a pytorch library and evaluation platform for end-to-end compression research. *arXiv preprint arXiv:2011.03029*.

[Bellard, 2015] Bellard, F. (2015). Bpg image format. *https://bellard.org/bpg*.

[Bross et al., 2020] Bross, B., Chen, J., Liu, S., and Wang, Y.-K. (2020). Versatile video coding (draft 10). *Joint Video Experts Team (JVET) of ITU-T SG16 WP3 and ISO/IEC JTC 1/SC29, Output Document JVET-S2001*.

[Cheng et al., 2020] Cheng, Z., Sun, H., Takeuchi, M., and Katto, J. (2020). Learned image compression with discretized Gaussian mixture likelihoods and attention modules. In *IEEE/CVF Conf. on Computer Vision and Patt. Recog. (CVPR).*

[Cover and Thomas, 2006] Cover, T. M. and Thomas, J. A. (2006). *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing).* Wiley-Interscience, USA.

[Cui et al., 2021] Cui, Z., Wang, J., Gao, S., Guo, T., Feng, Y., and Bai, B. (2021). Asymmetric gained deep image compression with continuous rate adaptation. In *IEEE/CVF Comp. Vis. Patt. Recog. (CVPR)*, pages 10527–10536.

[Deng et al.,] Deng, J., Socher, R., Fei-Fei, L., Dong, W., Li, K., and Li, L.-J. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Conference on Computer Vision and Pattern Recognition(CVPR).*

[Dhariwal and Nichol, 2021] Dhariwal, P. and Nichol, A. (2021). Diffusion models beat gans on image synthesis. In *Advances in Neural Information Processing Systems*, volume 34, pages 8780–8794. Curran Associates, Inc.

[Ding et al., 2022] Ding, K., Ma, K., Wang, S., and Simoncelli, E. P. (2022). Image quality assessment: Unifying structure and texture similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(5):2567–2581.

[Egilmez et al., 2021] Egilmez, H., Singh, A. K., Coban, M., Karczewicz, M., Zhu, Y., Yang, Y., Said, A., and Cohen, T. (2021). Transform network architectures for deep learning based end-to-end image/video coding in subsampled color spaces. *IEEE Open Jour. of Signal Proc.*, 2:441–452.

[He et al., 2022] He, D., Yang, Z., Peng, W., Ma, R., Qin, H., and Wang, Y. (2022). Elic: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding. In *IEEE/CVF Conf. on Comp. Vision and Pattern Recognition (CVPR)*, pages 5708–5717.

[He et al., 2021] He, D., Zheng, Y., Sun, B., Wang, Y., and Qin, H. (2021). Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14771–14780.

[Huffman, 1952] Huffman, D. A. (1952). A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101.

[Jiang and Wang, 2023] Jiang, W. and Wang, R. (2023). Mlic++: Linear complexity multi-reference entropy modeling for learned image compression. In *ICML 2023 Workshop Neural Compression: From Information Theory to Applications*.

[Kawar et al., 2022a] Kawar, B., Elad, M., Ermon, S., and Song, J. (2022a). Denoising diffusion restoration models. In *ICLR Workshop on Deep Generative Models for Highly Structured Data*.

[Kawar et al., 2022b] Kawar, B., Song, J., Ermon, S., and Elad, M. (2022b). Jpeg artifact correction using denoising diffusion restoration models. In *Neural Information Processing Systems (NeurIPS) Workshop on Score-Based Methods*.

[Kingma and Ba, 2015] Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Int. Conf. on Learning Representations, (ICLR), San Diego, CA, USA, May 7-9*.

[Koyuncu et al., 2022] Koyuncu, A. B., Gao, H., Boev, A., Gaikov, G., Alshina, E., and Steinbach, E. (2022). Contextformer: A transformer with spatio-channel attention for context modeling in learned image compression. In Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T., editors, *Computer Vision – ECCV 2022*, pages 447–463, Cham. Springer Nature Switzerland.

[Koyuncu et al., 2024] Koyuncu, A. B., Jia, P., Boev, A., Alshina, E., and Steinbach, E. (2024). Efficient contextformer: Spatio-channel window attention for fast context modeling in learned image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, pages 1–1.

[Li et al., 2011] Li, S., Zhang, F., Ma, L., and Ngan, K. N. (2011). Image quality assessment by separately evaluating detail losses and additive impairments. *IEEE Transactions on Multimedia*, 13(5):935–949.

[Lim et al., 2017] Lim, B., Son, S., Kim, H., Nah, S., and Lee, K. M. (2017). Enhanced deep residual networks for single image super-resolution. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1132–1140.

[Liu et al., 2023] Liu, J., Sun, H., and Katto, J. (2023). Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1–10.

[Lu and Ma, 2022] Lu, M. and Ma, Z. (2022). High-efficiency lossy image coding through adaptive neighborhood information aggregation. *arXiv preprint arXiv:2204.11448*.

[Ma et al., 2021] Ma, C., Wang, Z., Liao, R.-L., and Ye, Y. (2021). A study of deep image compression for YUV420 color space. In *SPIE Applications of Digital Image Processing XLIV*, volume 11842, pages 9 – 16.

[Ma et al., 2020] Ma, H., Liu, D., Xiong, R., and Wu, F. (2020). iwave: Cnn-based wavelet-like transform for image compression. *IEEE Transactions on Multimedia*, 22(7):1667–1679.

[Ma et al., 2022] Ma, H., Liu, D., Yan, N., Li, H., and Wu, F. (2022). End-to-end optimized versatile image compression with wavelet-like transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(3):1247–1263.

[Minnen et al., 2018] Minnen, D., Ballé, J., and Toderici, G. D. (2018). Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, volume 31.

[Minnen and Singh, 2020] Minnen, D. C. and Singh, S. (2020). Channel-wise autoregressive entropy models for learned image compression. *IEEE Int. Conf. on Image Processing (ICIP)*, pages 3339–3343.

[Qian et al., 2022] Qian, Y., Lin, M., Sun, X., Tan, Z., and Jin, R. (2022). Entroformer: A transformer-based entropy model for learned image compression. In *International Conference on Learning Representations*.

[Sheikh and Bovik, 2006] Sheikh, H. and Bovik, A. (2006). Image information and visual quality. *IEEE Trans. on Image Processing*, 15(2):430–444.

[Skodras et al., 2001] Skodras, A., Christopoulos, C., and Ebrahimi, T. (2001). The jpeg 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58.

[Song et al., 2021] Song, M., Choi, J., and Han, B. (2021). Variable-rate deep image compression through spatially-adaptive feature transform. In *IEEE/CVF Int. Conf. on Computer Vision (ICCV)*, pages 2380–2389.

[Sullivan et al., 2012] Sullivan, G. J., Ohm, J.-R., Han, W.-J., and Wiegand, T. (2012). Overview of the high efficiency video coding (HEVC) standard. *IEEE Trans. on Circuits and Systems for Video Tech.*, 22(12):1649–1668.

[Sun et al., 2021] Sun, Z., Tan, Z., Sun, X., Zhang, F., Qian, Y., Li, D., and Li, H. (2021). Interpolation variable rate image compression. *ACM Int. Conf. on Multimedia*.

[Ulas and Tekalp, 2022] Ulas, O. U. and Tekalp, A. M. (2022). Flexible luma-chroma bit allocation in learned image compression for high-fidelity sharper images. In *2022 Picture Coding Symposium (PCS)*, pages 31–35.

[van den Oord et al., 2016] van den Oord, A., Kalchbrenner, N., Espeholt, L., kavukcuoglu, k., Vinyals, O., and Graves, A. (2016). Conditional image generation with pixelcnn decoders. In Lee, D., Sugiyama, M., Luxburg, U., Guyon,

I., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc.

[Wang et al., 2004] Wang, Z., Bovik, A., Sheikh, H., and Simoncelli, E. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

[Witten et al., 1987] Witten, I. H., Neal, R. M., and Cleary, J. G. (1987). Arithmetic coding for data compression. *Commun. ACM*, 30(6):520–540.

[Xie et al., 2021a] Xie, Y., Cheng, K. L., and Chen, Q. (2021a). Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, page 162–170, New York, NY, USA. Association for Computing Machinery.

[Xie et al., 2021b] Xie, Y., Cheng, K. L., and Chen, Q. (2021b). Enhanced invertible encoding for learned image compression. In *Proc. of the ACM Int. Conf. on Multimedia*, pages 162–170.

[Xue et al., 2019] Xue, T., Chen, B., Wu, J., Wei, D., and Freeman, W. T. (2019). Video enhancement with task-oriented flow. *Int. Jour. of Computer Vision (IJCV)*, 127(8):1106–1125.

[Yang et al., 2020] Yang, Y., Bamler, R., and Mandt, S. (2020). Improving inference for neural image compression. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

[Zhang et al., 2018] Zhang, R., Isola, P., Efros, A. A., Shechtman, E., and Wang, O. (2018). The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*.

[Zou et al., 2022] Zou, R., Song, C., and Zhang, Z. (2022). The devil is in the details: Window-based attention for image compression. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 17492–17501.