Optimizing Learned Image Compression Models for Complexity and Rate-Distortion-Perception Performance

by

Ogün Kırmemiş

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

Doctor of Philosophy

in

Electrical and Electronics Engineering

KOÇ ÜNİVERSİTESİ

September 7, 2023

Optimizing Learned Image Compression Models for Complexity and Rate-Distortion-Perception Performance

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a doctoral dissertation by

Ogün Kırmemiş

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. Dr. A. Murat Tekalp (Advisor)

_____

Prof. Dr. Yücel Yemez

_____

Asst. Prof. Dr. Zafer Doğan

_____

Assoc. Prof. Dr. Mehmet Erkut Erdem

_____

Prof. Dr. Gözde Bozdağı Akar

Date: _____

Biricik Öykü'ye

ABSTRACT

Optimizing Learned Image Compression Models for Complexity and Rate-Distortion-Perception Performance

Ogün Kırmemiş

Doctor of Philosophy in Electrical and Electronics Engineering

September 7, 2023

Lately, the rate-distortion performance of learned image compression models has surpassed that of traditional codecs by the virtue of recent advancements in learned entropy and context models. However, state-of-the-art learned models currently exhibit higher complexity and slower processing times compared to conventional image codecs. Furthermore, optimization of models for just rate-distortion performance as currently done does not result in the best perceptual image quality. This thesis addresses these issues.

One of the contributions of this thesis is to explore the impact of the activation function on the performance of image compression, considering both objective and subjective evaluation criteria, as well as runtime efficiency. The widely used generalized divisive normalization (GDN) activation function is one of the reasons for its high complexity. Our findings reveal that the latent variables generated by hard shrinkage activation align more closely with a Laplacian distribution. Our method achieves comparable rate-distortion results, along with superior visual performance, at reduced computational complexity.

The second contribution of this thesis lies in the exploration of practical approaches to the optimization of rate-distortion-perception (RDP) performance. To date, the use of mean squared error (MSE) in rate-distortion optimization (RDO) has remained the standard practice in the field of image and video compression. This has been beneficial for gauging codec performance by offering a quantitative measurement of results through peak-signal-to-noise ratio (PSNR). However, it's broadly accepted that PSNR does not accurately reflect the perceptual quality of images, making RDO unsuitable for codec optimization in terms of perceptual quality. Recently, the notion of RDP has been formally defined by Blau and Michaeli

[1]. Yet, there's still a lack of practical methodology for setting the RDP function at a desired level in a feasible way. We propose a practical method to enable perception-distortion analysis by keeping the rate constant. This approach allows for a principled perceptual evaluation of the codec at predetermined bitrates. Additionally, we present a method for compressing a set of images at a desired RDP point by converting the problem to an integer linear programming model. Our experimental results provide essential insights into the practical analysis of RDP in learned image compression.

ÖZETÇE

Karmaşıklık ve Hız-Bozulma-Algı Performansı İçin Öğrenilmiş İmge Sıkıştırma Modellerini Eniyileme
Ogün Kırmemiş
Elektrik ve Elektronik Mühendisliği, Doktora
7 Eylül 2023

Son zamanlarda, öğrenilmiş görüntü sıkıştırma modellerinin hız-bozulma performansı, öğrenilmiş entropi ve bağlam modellerindeki son gelişmeler sayesinde geleneksel kodlayıcı/çözümleyici performansını aşmıştır. Ancak, mevcut son teknolojiye sahip öğrenilmiş modeller, geleneksel görüntü kodlayıcı/çözümleyicilerine kıyasla daha yüksek işlem karmaşıklığı göstermekte ve daha uzun işlem süreleri sunmaktadır. Dahası, modellerin sadece hız-bozulma performansı için optimize edilmesi, en iyi algısal görüntü kalitesini sağlamaz. Bu tez, bu konuları ele almaktadır.

Bu tezin katkılarından biri, hem nesnel hem de öznel değerlendirme kriterlerini, aynı zamanda yürütme zamanı verimliliğini de göz önünde bulundurarak, aktivasyon fonksiyonunun görüntü sıkıştırmanın performansı üzerindeki etkisini incelemektir. Yaygın olarak kullanılan genelleştirilmiş bölücü normalizasyon (GDN) aktivasyon fonksiyonu, yüksek işlem karmaşıklığının nedenlerinden biridir. Bulgularımız, Hard-Shrinkage aktivasyonu ile oluşturulan gizli değişkenlerin, bir Laplace dağılımı ile daha yakından uyum sağladığını ortaya koymaktadır. Yöntemimiz, azaltılmış işlem karmaşıklığı ile birlikte, karşılaştırılabilir hız-bozulma sonuçları ve üstün görsel performans elde etmektedir."

Bu tezin ikinci katkısı, hız-bozulma-algı (RDP) performansının eniyilenmesine yönelik pratik yaklaşımların keşfedilmesi üzerinedir. Bugüne kadar, hız-bozulma optimizasyonu (RDO) konusunda karesel ortalama hatasının (MSE) kullanımı, görüntü ve video sıkıştırma alanında standart uygulama olmuştur. Bu, en yüksek sinyal gürültü oranı (PSNR) aracılığıyla sonuçların nicel bir ölçümünü sunarak kodlayıcı-çözümleyici performansının değerlendirilmesi için faydalı olmuştur. Ancak, PSNR'nin görüntülerin algısal kalitesini doğru bir şekilde yansıtmadığı genel olarak kabul edilmiştir, bu da RDO'nun algısal kalite açısından kodlayıcı/çözümleyici eniyilemesi

için uygun olmadığını göstermektedir. Yakın zamanda, hız-bozulma-algı (RDP) kavramı, Blau ve Michaeli tarafından titizlikle tanımlanmıştır [1]. Ancak, RDP fonksiyonunu uygulanabilir bir şekilde istenen bir seviyede belirlemek için hala pratik bir metodoloji eksikliği bulunmaktadır. Hızı sabit tutarak algı-bozulma analizini mümkün hale getirecek pratik bir yöntem öneriyoruz. Bu yaklaşım, belirlenmiş bit hızlarında kodlayıcı/çözümleyici'nin algısal açıdan değerlendirilmesine izin verir. Ek olarak, bir imge kümesini istenen bir RDP noktasında sıkıştırma problemini tam sayı lineer programlama modeline dönüştüren bir yöntem sunuyoruz. Deneysel sonuçlarımız, öğrenilmiş görüntü sıkıştırmasında RDP'nin pratik analizi hakkında temel içgörüler sağlar.

ACKNOWLEDGMENTS

I am truly fortunate to have Prof. Murat Tekalp as my advisor, and I appreciate his support and belief in my capabilities. Prof. Tekalp provided invaluable insights and constructive feedback whenever I needed it. I am honored to have had the opportunity to learn from him. I am also grateful to members of my thesis committee Prof. Yücel Yemez, Asst. Prof. Zafer Doğan, Assoc. Prof. Erkut Erdem and Prof. Gözde Bozdağı Akar for their invaluable feedback and serving on my defense committee.

I would like to thank my amazing SO, Öykü, for her unwavering love, encouragement, patience, and support throughout the challenging process of completing this thesis. I am not sure if I could finish this thesis without her. I hope we share a long life together so that I can repay you for at least some of the things you did for me.

I would like to thank my fellow graduate students Akın and Onur for our many discussions on deep learning and the fun we had. I thank Buket for all the help she provided during our instructorship and more importantly for helping me win the biggest treasure of my life. Last but not least, I would like to extend my deepest gratitude to my mother and father for believing in me and my brother for all the times he made me laugh, and for being my tech support 24/7.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

BPP	Bit Per Pixel
BPG	Better Portable Graphics
CLIC	Challenge on Learned Image Compression
DP	Distortion-Perception
FID	Frechet Inception Distance
GAN	Generative Adversarial Network
GDN	Generalized Divisive Normalization
GMM	Gaussian Mixture Model
HEVC	High Efficiency Video Coding
ILP	Integer Linear Programming
IS	Inception Score
KID	Kernel Inception Distance
MOS	Mean Opinion Score
MSE	Mean Squared Error
MS-SSIM	Multi-Scale Structural Similarity Index Measure
NIQE	Natural Image Quality Evaluator
NTIRE	New Trends in Image Restoration and Enhancement
PSNR	Peak-Signal-to-Noise Ratio
QP	Quantization Parameter
RDO	Rate-Distortion Optimization
RDP	Rate-Distortion-Perception
RP	Rate-Perception
SISR	Single-Image Super-Resolution
SSIM	Structural Similarity Index Measure
VAE	Variational Autoencoder

Chapter 1

INTRODUCTION

1.1 Motivation

The early 2010s marked the emergence of the third era of deep learning, a significant milestone that has since irrevocably shaped the evolution of artificial intelligence and machine learning. Furthermore, this wave has also affected the fields of computer vision, image and video processing, inducing considerable paradigm shifts. Single-image super-resolution (SISR) stands as one of the initial successful implementations of deep learning in image processing [5], subsequently sparking an emergence of SISR methodologies that are predominantly learning-based. A series of subsequent New Trends in Image Restoration and Enhancement (NTIRE) challenges have clearly illustrated that deep learning-oriented techniques have emerged as the superior approach for addressing ill-posed inverse problems in image and video processing [6, 7, 8]. Before the formal introduction of perception-distortion tradeoff by Blau and Michaeli [9], Ledig et al. [10] discovered that adversarial training along with VGG loss improved mean opinion score (MOS) significantly in the context of SISR. VGG loss is defined as the mean squared distance between learned features of two images (an original and a processed) at a particular layer of the VGG network.

The power of deep learning is also dramatically reflected in the realm of image/video compression, contributing significantly to the end-to-end optimization of image/video codecs. A crucial development in image compression techniques emerged from the Challenge on Learned Image Compression (CLIC) contests. The contest results indicated the superiority of methods utilizing an end-to-end learning framework over conventional image codecs, such as JPEG [11]. In fact, these meth-

ods have been found to match the efficiency of state-of-the-art codecs at the time, such as the Better Portable Graphics (BPG) [12]. Starting from the first contest, methods are evaluated by human critics, making perceptual optimization imperative. At the time, participants employed some combination of multi-scale structural similarity index measure (MS-SSIM) and VGG loss. Initial works on end-to-end optimized learned codecs leaned towards conventional rate-distortion optimization, whereas more recent research efforts have begun probing into the rate-distortion-perception (RDP) optimization.

Although Matsumoto [13, 14] and Blau et al. [1] have introduced the concept of RDP trade-off, they did not propose a practical approach to achieve combined optimization of RDP. Mentzer et al. [15] proposed optimization of a combination of rate, MSE, and perceptual distortion measure, together with an adversarial loss in order to encode images at low bitrates with an acceptable RDP trade-off. However, they employed ad-hoc tricks to keep the rate approximately constant.

Another challenge in front of the widespread adoption of learned image codecs is their computational complexity. Most learned codecs cannot decode images in a reasonable time when working on CPUs. Moreover, there has been a debate about which activation function is more suitable for learned image compression. In order to alleviate this, we suggest a new activation function that works faster than the widely used generalized divisive normalization (GDN) function.

In this thesis, we explore the two different aspects of learned image compression. First, we explore the effects of the activation functions. Following that, our exploration extends to the development of novel practical rate-distortion-perception (RDP) optimization techniques.

1.2 Learned Image Compression Framework

In this section, we present a limited review of the evolution of image compression algorithms from classical to learning-based approaches in order to provide background. Figure 1.1 shows the outline of how images are compressed. Most classical compression methods first divide images into equal-sized non-overlapping blocks or tiles.

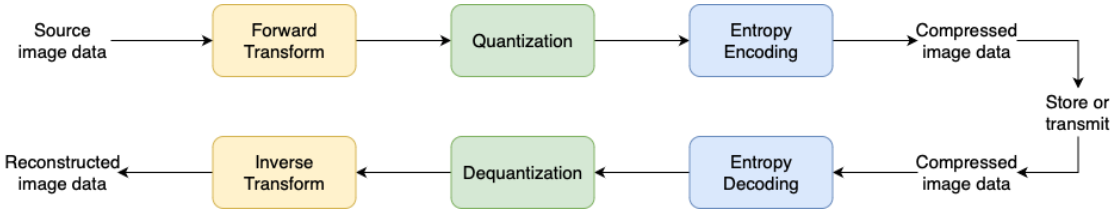


Figure 1.1: Diagram of generic image compression.

They transform pixels within each block/tile into another space, quantize the transform coefficients, and code the output using entropy coding. In the case of JPEG, the Discrete Cosine Transform is selected as the transform and it is applied to every 8×8 block. Entropy coding is realized via Huffman coding. On the other hand, JPEG2000 [16] employs the Discrete Wavelet Transform. Entropy coding is taken care of by the Embedded Block Coding with Optimized Truncation of the Embedded Bitstreams (EBCOT) algorithm. As the resolution of images increases, small block transforms become suboptimal at capturing redundancy. Therefore, more recent codecs enlarge the transform block size. For example, BPG codec, which is based on High Efficiency Video Coding (HEVC), can use transform blocks up to 32×32.

In contrast, CNN-based image compression algorithms process the input as a whole therefore they can capture spatial redundancies better. Moreover, unlike traditional codecs, neural networks are non-linear models and thus they can find more powerful representations. The quantization step is a challenge for neural networks due to its non-differentiable nature. There are different techniques that solve this problem, such as soft quantization [17], stochastic quantization [18], and modeling the quantization via uniform noise [19].

Another advantage of learned image compression is that the entropy model of the bitstream is also learned. In general, the distribution of quantized symbols is assumed to be a well-known distribution like Gaussian or Laplacian. A secondary autoencoder network, called the hyperprior network, predicts the mean and variance of every symbol so that the probability of every symbol can be calculated. The arithmetic encoder encodes the symbols according to this probability. The la-

tent generated by the hyperprior network is also quantized and sent to the receiver as side information. This is not the only way to build an entropy model, other types of hyperprior networks are discussed in Chapter 2.

By imposing a prior distribution on the symbols, the loss function that we use in training becomes the same as that of Variational Autoencoders [20], therefore these kinds of networks are called VAE-based image compression models. Since the VAE loss function is a combination of rate and distortion, training this kind of network leads to joint optimization of both. However, VAE models have an inherent drawback: The transform itself is lossy and some information will always be lost even without quantization. More specifically, VAE models use strided convolutions to make the latent smaller than the input image. Due to such downsampling operations, the transformation is not invertible, unlike traditional compression algorithms.

1.3 Contribution and Organization

The main contributions of this thesis are as follows:

- We introduce the hard shrinkage activation function to be used in learned image compression instead of generalized divisive normalization (GDN). We show that latents generated by a network using hard shrinkage fit a Laplacian distribution better than one using GDN. We also show hard shrinkage has less complexity and the proposed method runs faster than the network with GDN activation both on CPU and GPU. This is an important aspect in deploying end-to-end learned codecs in practical real-life applications. The proposed method performs on par in terms of PSNR and structural similarity index measure (SSIM) but it produces visually superior images. This work was presented in IEEE ICIP 2020 [21].

- We put forth a practical approach to fix the bitrate of the encoded image at a desired value to perform perception-distortion analysis at a fixed bitrate. At the end of the process, a latent that is generated by a single encoder can be

decoded by multiple decoders that will produce images on different points on the perception-distortion plane. This work was presented in PCS 2021 [22].

- We present a formulation to optimize the rate allocation between a set of encoded images even if the encoder cannot provide any form of rate adaptation. This is achieved by formulating the rate-distortion (RD) optimization of a set of codecs over a test set of images as integer linear programming (ILP). By solving the ILP problem, we can encode a set of images in between the fixed RD points of the pre-trained encoders. We also demonstrate a similar ILP formulation applies to RDP optimization for a set of images yielding perceptually better images from a set of pre-trained codecs at a given total rate. The optimization problem we provide can also be applied to RDP optimization of video compression algorithms.

The rest of the thesis is organized as follows:

- Chapter 2 presents related work on VAE-based image compression algorithms. We examine the literature from the point of RD and RDP optimization. We also discuss the evaluation criteria used for comparing compression methods.

- Chapter 3 presents an activation function that is better suited to Laplacian priors. We show that the computational complexity of the proposed activation function is lower than the widely used GDN. This chapter is based on our paper [21].

- Chapter 4 presents practical approaches to RDP optimization. First, we establish a method of distortion-perception (DP) optimization of codecs by fixing the rate. Then we establish the RD optimization of distinct models in the form of integer linear programming. This approach is also extended to RP optimization and generalized to RDP by adding a new constraint. This chapter is based on our papers [22, 23].

- Chapter 5 presents the conclusion.

Chapter 2

RELATED WORK

2.1 Rate-Distortion Optimization

Early work on image compression by deep learning focuses on recurrent neural networks (RNN). Toderici et al. [24] present an RNN-based encoder and decoder, a binarizer, and LSTM modules for entropy coding. Covell et al. [25] propose stop code tolerant RNNs that can decide to stop encoding for certain tiles of image. Johnston et al. [26] propose convolutional recurrent networks that can realize spatially adaptive bit allocation.

In the meantime, it has been discovered that better outcomes can be achieved by end-to-end RD-optimized learned image compression methods. The groundwork for this form of image compression, using an autoencoder, is laid by Ballé et al. [27, 28], in which they introduced a joint optimization strategy for rate, distortion, and a probability model. They estimate the rate by the entropy of a non-parametric prior distribution on the latent representation. During training, they circumvent the non-differentiability of quantization by adding uniformly distributed noise.

In subsequent work, Ballé et al. [29] propose a forward adaptation of the entropy model through a learnable scale hyperprior. In their model, they considered the latent as independent Laplacian random variables and determined the scale of these variables via a hyperprior network. Minnen et al. [30] introduced a backward adaptation stage that involved the addition of a context model. For this purpose, they used a PixelCNN architecture. In addition to the scale, their network also estimated the mean of the latent representation. However, the PixelCNN considerably reduced the speed of the decoding process, as it produced output for a channel one pixel at a time. To address this problem, a channel-wise autoregressive entropy model was

introduced [31]. This model divided the latent representation in the channel dimension and each slice was conditioned on the prior one. Due to the inherent nature of CNNs, this process is easily parallelizable, thus, much quicker than PixelCNN.

Theis et al. [18] proposed a different quantization function in which values are quantized stochastically. Latent code is parametrized as a Gaussian scale mixture. Choi et al. [32] present one of the first rate adaptive models. Instead of training multiple models for different bitrates, they introduce two rate control parameters: a Lagrange multiplier and quantization step size. They train the network for a set of Lagrange multipliers which provides coarse rate adaptation. Finer adaptation can be attained by changing the quantization step. Yang et al. [33] introduce a modulating network that takes the desired Lagrange multiplier as input. The output of the modulating network is multiplied with feature maps resulting in scaled latent code. Likewise, Jia et al. [34] put forward another modulating network and manages to model RD curve of their model.

Ayzik and Avidan [35] propose an algorithm with decoder-only side information. The encoder encodes the image without seeing the side information which is created at the decoder side by correlating a partially decoded image to another set of images. Then the partially decoded image along with the side information is passed to another network generating the fully decoded image.

Cui et al. [36] explore using asymmetric Gaussian distribution for estimation of entropy. Additionally, they propose using gain units that multiply latent code to a desired bitrate. At the decoder side, there is an inverse gain unit that reverts the scaling. The gain and inverse gain coefficients are interpolated exponentially, meaning the model can adapt to any bitrate. Hu et al. [37] posit a coarse-to-fine compression scheme in which an image is compressed at different scales. The side information of every scale depends on the decoded codewords of the smaller scale. Decoded latents are aggregated in an information aggregation network to yield the decoded image.

Chen et al. [38] propose using non-local attention blocks [39] in analysis and synthesis networks. To alleviate the increase in computational complexity due to

the attention block, they suggest downsampling inside the attention block. Cheng et al. [40] offer the use of Gaussian Mixture Model (GMM) in the parametrization of latent code. Additionally, they explore a simpler attention block that works locally and relies on the large receptive field created by the use of multiple residual blocks. Lee et al. [41][42] advances the use of GMM by an elaborate model that incorporates information from hyperprior, adjacent known representations, and global context.

He et al. [43] offer a checkerboard context model that aims to speed up autoregressive context models. Half of the latent code is selected to be anchors in a checkerboard pattern that is decoded independently. The other half, called non-anchors, is decoded with the help of the anchors. Since anchors are decoded independently, they can be decoded in parallel making this method faster than the PixelCNN context model. Efficient Learned Image Compression (ELIC) model [44] incorporates the checkerboard context with an unevenly grouped channel-wise context model. The authors posit unevenly grouping channels results in more energy compaction.

Recently, invertible image compression algorithms that do not rely on VAEs emerged. Helminger et al. [45] leverages normalizing flows to learn a bijective mapping from image space to latent representation, achieving a range of qualities from low bit-rate to near-lossless. Since the decoder is the exact inverse of the encoder, the algorithm has the unique advantage of maintaining constant quality through multiple re-encodings, setting a new precedent in utilizing normalizing flows for lossy image compression. Xie et al. [46] argue that flow networks have 2 main disadvantages: First, the invertibility of the network limits the capacity of nonlinear representation. Therefore they add a feature enhancement module between the encoder and the input at the encoder side and between the decoder and the output at the decoder side. The second downside is the fact that flow networks cannot alter the number of elements in the input tensor and thus they cannot get rid of redundant pixels. The authors solve this problem by creating an attentive channel squeeze operation that decreases the number of channels in the latent.

In addition to flow-based methods, there are wavelet-like neural network models. Ma et al. [47] propose a wavelet-like transform, called iWave that employs CNNs

as prediction filters of the wavelet transform. This model is trained with the reconstruction loss only. There is also an end-to-end optimized version of this method called iWave++ [48].

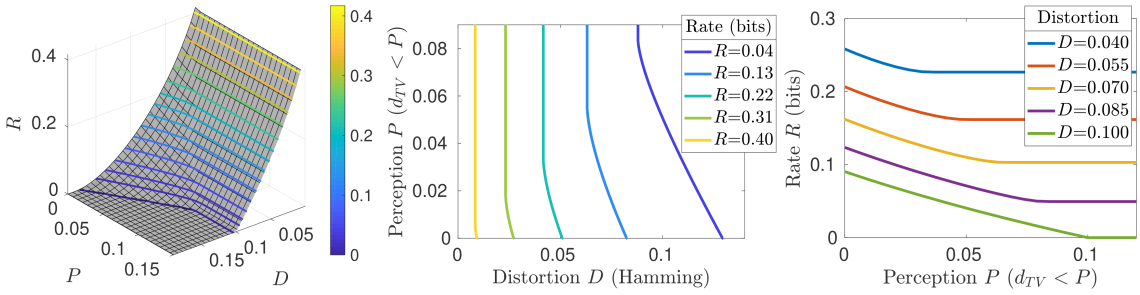
The research efforts in this section predominantly focus on optimization in relation to MSE or SSIM losses and do not specifically target the perception dimension.

2.2 Rate-Distortion-Perception Optimization

In order to produce visually pleasing images, Balle et al.[19] propose that distortion between the reference and its reconstruction should be calculated in a feature space instead of the original signal space. They select the normalized Laplacian pyramid as the feature space. Liu et al. [49] employed adversarial loss along with VGG-loss in addition to MSE.

The idea of the perception-distortion trade-off is first introduced in the context of image restoration algorithms by Blau and Michaeli [9]. The authors assert that, unlike distortion, perceptual quality can only be measured by no-reference measures. Moreover, the authors reveal mathematical proof indicating that distortion measures and perceptual quality in image restoration algorithms are inversely related. Every image restoration algorithm has an unattainable region in the PD plane. Therefore the study suggests that adversarial training can be used to traverse the boundary at the edge of the unattainable region. Matsumoto [13, 14] introduced the perception-distortion trade-off to rate-distortion theory. Blau and Michaeli [1] formalized RDP and calculated the closed-form solution for a Bernoulli source. Figure 2.1 shows the most important takeaways. Figure 2.1a shows at higher bitrates good perceptual quality can be achieved whereas at lower bitrates better perceptual quality can only be attained if the distortion is sacrificed. Figure 2.1b shows the same trade-off at different bitrates. Perception-distortion function starts to bend as the bitrate decreases meaning perception and distortion become inversely correlated. Figure 2.1c demonstrates that at fixed distortion, perception can be improved by increasing the rate.

Chen et al. [50] proved that the RDP function is achievable by deterministic



(a) (b) (c)

Figure 2.1: The rate-distortion-perception function of a Bernoulli source. (a) Equi-rate level sets are illustrated on the RDP function. When reaching perfect perception (P=0), the equi-rate lines curve significantly at low bitrates, highlighting the escalating tradeoff between distortion and perceptual quality. (b) Cross sections of RDP along the perception-distortion planes are shown. Observe the balance between perceptual quality and distortion, which intensifies at lower bitrates. (c) Cross sections of RDP along rate-perception planes. Notice that at a fixed distortion, the perceptual quality can be enhanced by increasing the rate. Adapted from [1]

.

codes. Yet, the authors argued that the concept of perceptual quality needs to be defined more effectively within the rules and concepts used in information theory. Theis and Wagner [51] prove that the RDP function is achievable by stochastic variable length codes. Similar to our approach, Zhang et al. [52] consider fixing an encoder and optimizing the decoder to achieve desired distortion and perception constraints. However, unlike our approach where the codec is deterministic, their method relies on common randomness between the encoder and decoder. They only provide experimental results on MNIST [53] and SVHN [54] datasets.

Chen et al. [55] converts the RDP problem to Wasserstein Barycenter Model which can be solved numerically by alternating Sinkhorn algorithm. They demonstrate the efficacy of the algorithm by solving for the RDP function for the Bernoulli and the Gaussian source. Yan et al. [56] explore how prioritizing high perceptual quality in lossy compression algorithms impacts distortion. Specifically, they find out that the minimal achievable MSE without perceptual constraint is doubled in order to achieve perfect perception. Interestingly, an optimal solution for tradi-

tional rate-distortion issues remains ideal for perceptual compression scenarios. The authors also propose a new training framework, using a Generative Adversarial Network (GAN) conditioned on an MSE-optimized encoder to maintain minimal MSE under a perfect perception constraint. However, it should be noted that the method is only tested on MNIST.

Compressing images at extremely low bitrates (less than 0.1 bpp) is a difficult task because the severe limit on the number of bits used for the compressed data considerably lowers the quality of the reconstructed image. The methods that employ GANs can encode images with extremely low bitrates and decode perceptually meaningful images since GANs are able to learn the image manifold and create real-looking images. Early works on generative image compression include Rippel and Bourdev [57] who utilized standard GAN framework [58]. Santurkar et al. [59] demonstrate that Wasserstein GANs [60] perform better perceptually. Tschannen et al. [61] employs Wasserstein GAN in such a way that the decoder takes the encoded latent and random noise vector together. Since it works with random noise vectors, it can generate images even without the latent like standard GANs. Raman et al. [62] also employ standard GAN formulation and propose adding layer-wise loss to the overall loss function. Agustsson et al. [63] propose using conditional adversarial training which helps the decoder to identify which regions to hallucinate and which regions to decode faithfully. However, these methods did not consider a formal approach for rate-distortion-perception trade-off and decoded images might not be faithful to the original images by some desired amount.

Winner of CLIC 2020 [64] proposes two post-processing networks that work on the output of Versatile Video Coding codec. One of the networks suppresses noise, and the other is for the restoration of textures. The latter is trained with adversarial and VGG loss in addition to L_1 loss. Winner of low bitrate track of CLIC 2021 Gao et al. [65] employ L_1 loss instead of L_2 in addition to GAN and LPIPS losses for perceptual quality. Additionally, they compress images according to regions of interest. This means they can allocate more bits for visually important parts of the image. They also admit that training a GAN network at a very low bitrate is a

challenging task since training becomes unstable. He et al. [66] won every image compression track of CLIC 2022 by perceptually oriented ELIC, dubbed PO-ELIC. This model is not only trained with LPIPS and adversarial loss but also style loss to improve the perceptual quality of the baseline ELIC model. The perceptual performance of the model is on par with HIFIC but at a much lower bitrate.

Agustsson et al. [67] present another generative compression approach that optimizes the RDP trade-off, allowing users to control the level of detail synthesized, thereby addressing concerns about misleading reconstructions. The methodology can either reconstruct a low mean squared error output close to the input or a high perceptual quality from a single encoded representation.

2.3 Optimization and Evaluation Criteria

The classical RD optimization theory traditionally relies on PSNR for measuring the fidelity and visual quality [68]. Perceptual Image Restoration and Manipulation Workshop 2018 [69] is the first super-resolution challenge in which evaluation methodology included a perceptual score in addition to MSE. The perceptual score was defined as a combination of the Natural Image Quality Evaluator (NIQE) [70] and Ma et al.'s measure [71]. NIQE is a no-reference measure that is based on natural scene statistics of features that are derived from natural, undistorted images. Ma et al.'s measure is also a no-reference measure specifically designed for evaluating single-image super-resolution algorithms. Both NIQE and Ma et al.'s measure are non-differentiable functions therefore we cannot train neural networks with them.

Salimans et al. [72] introduce Inception Score (IS) to evaluate GANs. Calculation of IS involves passing the generated images through an Inception Network, computing a conditional class distribution for each image, and comparing these distributions to a marginal class distribution over all images. One disadvantage of IS is it does not compare generated samples to real samples directly. Heusel et al. [73] propose Frechet Inception Distance (FID) to alleviate this problem. FID calculates the distance between the distribution of Inception features of generated examples and real examples. Binkowski et al. [74] argue that FID is a biased estimator and

they put forth kernel inception distance (KID). KID is an unbiased estimator and it estimates the Maximum Mean Discrepancy between Inception features using a radial basis function kernel. Since FID and KID are calculated between sets of images, it is not possible to use them as loss functions during training.

SSIM [75] and its multi-scale version MS-SSIM [76] are used not only to evaluate the performance of compression algorithms but also as a loss function. It has been shown that they correlate more with human opinion than PSNR. MS-SSIM was used heavily in optimization in the early stages of learned image compression algorithms since it remedied the well-known deficiency of MSE loss.

CLIC evaluated PSNR and MS-SSIM but always relied on MOS to determine the ranking. Since this is a time-consuming process, instead of users evaluating all shown images, an ELO system was created in which users evaluate only two methods at a time in 2020. This is the reason the recent winners of the competition use adversarial training and perceptually oriented metrics like MS-SSIM or LPIPS [77]. Video Multimethod Assessment Fusion (VMAF) by Netflix [78] is introduced for the perceptual evaluation of video codecs. It is one of the metrics used by JPEG AI. Yet, it is generally not used in image compression papers.

Chapter 3

SHRINKAGE AS ACTIVATION IN LEARNED IMAGE COMPRESSION

The efficacy of deep networks is significantly influenced by both the activation function employed and the method of normalization utilized. In the sphere of inverse problems, the predominant choice of activation is the rectified linear unit (RELU) while batch normalization is less preferred in EDSR [79]. On the other hand, within the context of image and video compression, the prevailing preference leans towards the adoption of generalized divisive normalization (GDN), a procedure that incorporates an element of normalization, serving as the activation function. GDN has been claimed to yield Gaussian latent variables for natural images [80] and it has an inverse to transform the latent back into the image domain. Winners of CLIC 2018 and 2019 [2, 81] use GDN activation. He et al. [44, 66] who won CLIC 2022 discard GDN in favor of classical residual bottleneck block [82]. The resulting network is deeper than a GDN network but they argue residual network enables better scalability of the model.

In addition to the choice of activation function, entropy modeling is an important aspect of compression. Balle et al. [29] explore an image codec that employs Laplacian hyperprior for entropy estimation while using GDN/IGDN. However, Minnen et al. [30] claimed that Gaussian distribution is a better hyperprior for entropy estimation. Despite Minnen's assertions, Zhou et al. [2][81] used Laplacian hyperpriors claiming Laplacian distribution fits the latent code better.

The primary limitation associated with image or video compression techniques utilizing deep learning is related to their high computational complexity. In general, high encoding complexity is ignored since the contents are encoded only once on the server side. Unfortunately, decoders of deep learning based methods are also

complex and thus they often lack the capability to execute near real-time on CPUs or, occasionally even on GPUs. In response to this challenge, this section of the dissertation introduces the implementation of hard shrinkage as an activation function designed specifically for image compression tasks. The findings demonstrate that latent variables created by the hard shrinkage are more suitably modeled by a Laplacian distribution. Furthermore, the use of hard shrinkage function results in a reduction in computational complexity compared to GDN, all the while preserving a comparable level of performance.

3.1 Proposed Network Architecture

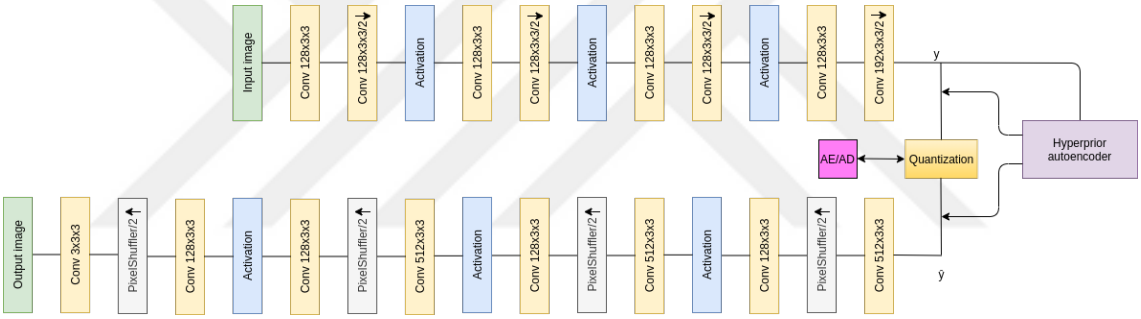


Figure 3.1: Block diagram of encoder/decoder network. Conv $C \times H \times W$ denotes a convolution layer with C channels and spatial kernel dimensions $H \times W$. The right and left arrows indicate resolution decrease and increase by 2, respectively.

Our model's design, as depicted in Figure 3.1, is similar to that described in [2]. Since Pixelshuffler layer [83] eliminates the occurrence of checkerboard artifacts, we utilized it for upsampling rather than transposed convolutions. In the original network [2], the top branch in Fig. 3.1 has GDN activation whereas the bottom branch has inverse GDN activation. Our proposed network employs Hard Thresholding function given in Eqn. 3.1 with channelwise variable thresholds as shown in Fig. 3.3. First, channels of the input tensor are divided by their corresponding threshold. Then the hard thresholding function is applied and every channel is multiplied by the same threshold to transform the tensor to its original scale. In essence, the proposed activation function passes values that are larger than a learned threshold

and rejects smaller values.

First, input images are scaled to a range of [-1, 1] before being fed to the encoder. The encoder then transforms the image to y. Following this, y is put through the hyperencoder, producing another output labeled as z. Subsequently, z is quantized, yielding $\hat{z}$, through the application of a rounding function and is then compressed using an arithmetic encoder. $\hat{z}$ is then relayed to the decoder, serving as side information. Simultaneously, back at the encoder, $\hat{z}$ is decoded to obtain the mean (μ_y) and standard deviation (σ_y) of y. The normalization of y then takes place, resulting in a zero-mean, unit variance Laplacian variable, designated as $y_{norm} = \frac{y-\mu_y}{\sigma_y}$. The next step involves quantizing y_{norm} to generate $\hat{y}_{norm}$, which is then compressed with an arithmetic encoder and forwarded to the decoder. Upon reception, the decoder starts decompressing both $\hat{y}_{norm}$ and $\hat{z}$. Following this, it scales and shifts $\hat{y}norm$ back to retrieve $\hat{y}$ using the formula $\hat{y} = \sigma_y \hat{y}_{norm} + \mu_y$. Finally, the decoder receives $\hat{y}$, which then leads to the creation of the decompressed image.

$$HardThreshold(x) = \begin{cases} 0, & -0.5 < x < 0.5 \\ x, & otherwise \end{cases} \tag{3.1}$$

The pair of networks are trained for 400,000 iterations, utilizing the Adam optimizer described by [84]. The training data is compiled from randomly selected patches of dimensions 256×256, and processed in batches consisting of 16 items each. This procedure is implemented on the training dataset from the 2019 CLIC competition. The loss function is defined as below:

$$L = \lambda D + R_y + R_z \tag{3.2}$$

where D is distortion (in terms of mean squared error), R_y is the mean code length for $\hat{y}$, R_z is the mean code length for $\hat{z}$. In our experiments, we trained models with $\lambda = 144$.

3.2 Algorithmic Complexity Analysis

For input and output tensors v and u, GDN operation is defined as follows:

$$u_i(m,n) = \frac{v_i(m,n)}{(\beta_i + \sum_j \gamma_{i,j}(v_j(m,n))^2)^{0.5}} \tag{3.3}$$

where $v_i(m,n)$ is the ith channel of the input at spatial location (m,n). γ and β are the learnable parameters of GDN. For an input tensor which has C channels, size of γ and β are $C \times C$ and C, respectively. Multiplication of γ with v introduces $\mathcal{O}(C^2)$ multiplications which cause the biggest bottleneck compared to our method. This analysis is also true for the inverse GDN since it is defined in the same way but instead of division, there is multiplication.

In contrast, the proposed activation function is characterized by a weight count equal to the number of input tensor channels C. Unlike GDN, every channel of the input tensor interacts only with its corresponding weight. Therefore, the computational complexity of all operations within this proposed function is on the order of $\mathcal{O}(C)$ since every channel is divided and multiplied by a single corresponding scalar. Consequently, this activation function exhibits a reduction in both the number of parameters and the computational complexity, thereby making it a favorable choice.

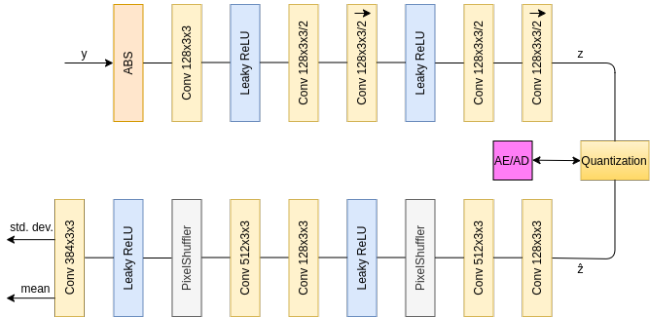


Figure 3.2: Block diagram of the hyper encoder/decoder network that is the same as in [2] except for the upsampling blocks. Scale output passes through the absolute value function.

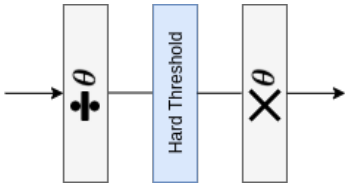


Figure 3.3: Proposed activation function for image compression. The parameter θ is a learnable threshold for every channel similar to [3]. Number of elements of θ is same as the number of channels of the input tensor.

3.3 Experimental Results

3.3.1 Rate-Distortion Evaluation

Table 3.1 shows Kullback-Leibler (KL) divergence between the histogram of latent code and Laplacian distribution with mean 0 and variance 1. Both on CLIC test set and Kodak set, *HardThreshold* yields lower KL divergence than GDN, which means it is more successful at fitting a Laplacian distribution.

RD curves given in Figures 3.4 and 3.5 clearly show that the proposed method performs better at relatively higher bitrates. The curves are drawn by changing the quantization parameter. The middle point in all curves has a unit quantization step. Other quantization steps are found by multiplying the unit quantization step by 2, $\frac{3}{2}$, $\frac{2}{3}$ and $\frac{1}{2}$ (from left to right in figures).

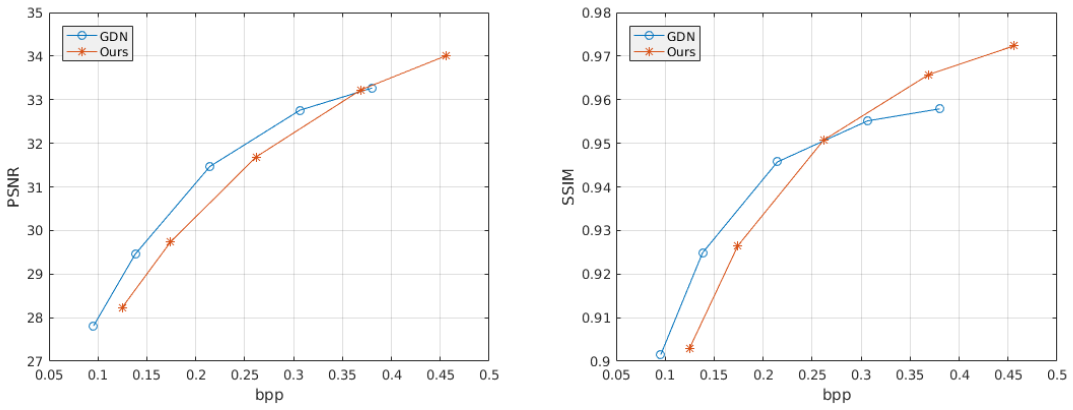


Figure 3.4: RD curves of Kodak dataset.

Figure 3.6 shows that the GDN network yields blurry results whereas the proposed network yields sharper results. Figure 3.7 shows an example of what happens at higher bitrates. Figure 3.8 shows that at low bitrates PSNR and SSIM drop however, the proposed network is able to hallucinate some of the details and because of this it seems sharper.

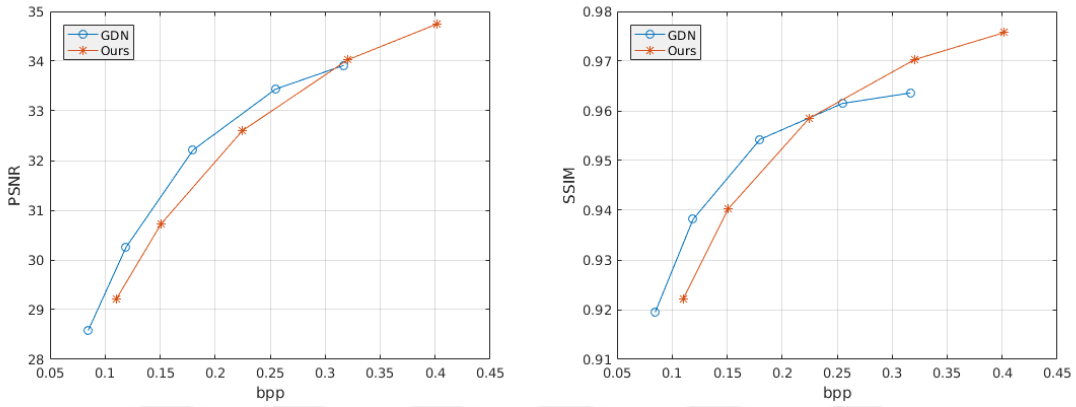


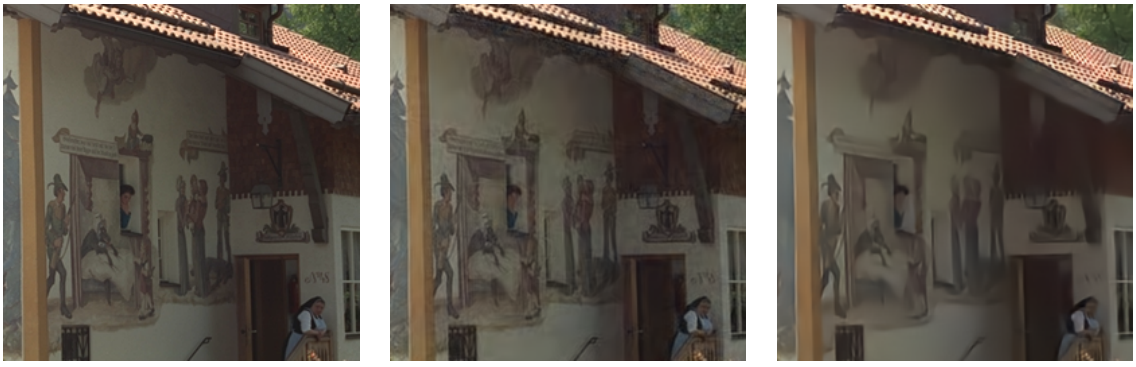
Figure 3.5: RD curves of CLIC 2018 test set.

Table 3.1: KL divergence between the histogram of latent code and Laplacian distribution $\mathcal{L} \sim (0, \sqrt{0.5})$

	GDN	Ours
CLIC Test	0.3452	0.2843
Kodak	0.4277	0.3460

3.3.2 Runtime Evaluation

Our empirical results reveal that the GDN is slower than our newly proposed model by approximately 3.8 seconds when put to the task of encoding and decoding the complete set of 24 images constituting the Kodak dataset, utilizing a dual Intel Xeon Gold 5118 2.30GHz CPU setup on a workstation. Another experiment is also conducted where both networks are tested on Nvidia Geforce GTX 1080 GPU. We



(a) (b) (c)

Figure 3.6: Example crop from Kodak dataset: (a) original crop, (b) compressed-decompressed with the proposed network at 0.3347 bpp with 29.89 dB PSNR and 0.9507 SSIM, (c) compressed-decompressed with GDN network at 0.2730 bpp with 30.0169 dB PSNR and 0.9424 SSIM. Although both images are at similar quality levels, Figure (c) is more blurry than Figure (b) on paintings on the wall. Both images are quantized with the unit quantization step.

observe that the GDN network lags by a mere 0.085 seconds. While this difference may appear negligible on initial observation, repeating the test on the CLIC dataset resulted in the GDN network exhibiting a greater lag of 2.6 seconds on the GPU. A notable difference between the two datasets is the dominant presence of high-resolution images in the CLIC dataset. The complexity analysis of GDN reveals a noteworthy dependence on channel size C, and with larger resolutions, bottlenecks are expected to emerge as a consequence of the restrictions of GPU hardware design. Consequently, maintaining a low level of computational complexity is crucial to ensure scalability. Our proposed methodology accommodates this imperative without an overly detrimental effect on rate-distortion performance.

3.3.3 Empirical Analysis of the Effects of Activation Functions

In order to gain additional insight into how the proposed network produces visually pleasing images, we investigated histograms of features. The first difference we noticed is that the variance of GDN features is much greater than the variance of Shrinkage features (at least an order of magnitude). Because of this reason, we divide the features by their standard deviation before creating their histograms

(a) (b) (c)

Figure 3.7: Example crop from Kodak dataset: (a) original crop, (b) compressed-decompressed with the proposed network. Compressed at 0.3186 bpp with 34.80 dB PSNR and 0.9656 SSIM, (c) compressed-decompressed with the GDN network at 0.3233 bpp with 34.22 dB PSNR and 0.9526 SSIM. Although both images are at similar bitrate and PSNR, Figure (c) is more blurry than Figure (b), especially noticeable in the woman's hair and the right side of her hat. Figure (b) is quantized with $\frac{2}{3}$ step size whereas Figure (c) is quantized with $\frac{1}{2}$ step size.

except for the actual symbols we encode. The symbols themselves are normalized by the hyperprior network therefore they do not need any normalization. Figure 3.10 presents the histograms of features after every activation layer side by side. Since we expect the features to look like Laplace distribution, we also show the discretized Laplace probability mass function (PMF). Even after the first activation layer features are centered around zero and have decreased entropy in both networks as shown in Figures 3.10a and 3.10b. The second and third activation layers depicted in Figures 3.10c, 3.10e, 3.10d and 3.10f do not reveal any valuable information unfortunately except in the case of GDN we observe a slight elevation in the number of zeros. Our network, on the other hand, yields similar distributions. The histogram of symbols to compress given in Figures 3.10g and 3.10h shows that there are more zeros in the latent code generated by the GDN. This helps the GDN network to yield output at a lower bitrate compared to the Shrinkage network. There is a clear discrepancy between the histogram and the desired PMF in both networks but it is more apparent in GDN's case since there are a limited number of nonzero elements. As the number of zeros increases the representation becomes more compressible

(a) (b) (c)

Figure 3.8: Example crop from Kodak dataset: (a) original crop, (b) compressed-decompressed with the proposed network. Compressed at 0.0923 bpp with 30.21 dB PSNR and 0.9329 SSIM, (c) compressed-decompressed with the GDN network at 0.0987 bpp with 31.36 dB PSNR and 0.9472 SSIM. Although both images are at similar bitrate, they have 1 dB difference in PSNR. Figure (c) is more blurry than Figure (b) but high frequency components of Figure (b) are mostly hallucinations since they do not agree with the original, apparent from the PSNR. Figure (b) is quantized with step size 2 whereas Figure (c) is quantized with $\frac{3}{2}$.

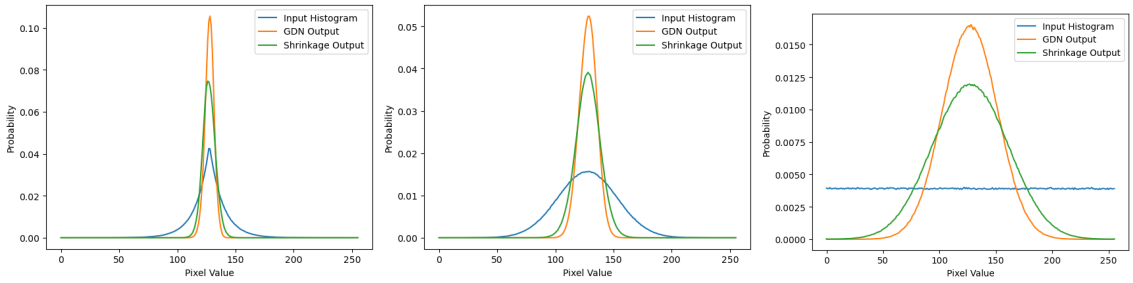
however, it also loses information. Figures 3.10g and 3.10h explain why the KL divergence in Table 3.1 is lower for Shrinkage network and additionally how the GDN network operates at a lower bitrate.

RDP theory defines perception as the divergence between the underlying distributions of the decoded image and natural images. Unfortunately, it is impossible to find the underlying distribution of natural images analytically. Because of this reason, we generate random noise from known distributions and compare the histogram of the decoded image with input to have a sense of perceptual quality in terms of KL divergence.

Table 3.2 presents the average RDP performance for random input. The size of generated random noise is 1024×1024 and the metrics presented are the average of 1000 trials. Noises are generated by the discretized version of distributions since we want the input to be in the standard $[0, 255]$ range. Since we use Laplace distribution to model the entropy, networks are most successful in the case of Laplace distribution. They use the least bitrate and have the least distortion and KL divergence

Table 3.2: Average PSNR, bitrate and perceptual score for random inputs

Distribution	GDN			Shrinkage (Ours)		
	PSNR	Bitrate	KL-div	PSNR	Bitrate	KL-div
Laplace	23.95	0.035	0.286	23.98	0.053	0.173
Gaussian	20.23	0.157	0.331	20.19	0.128	0.204
Uniform	11.25	0.446	0.410	11.63	0.394	0.273



(a) (b) (c)

Figure 3.9: The response of the networks to different random noise types. (a) Laplace noise, (b) Gaussian noise, (c) Uniform noise. Regardless of the input type, the histogram of the Shrinkage network is wider than GDN. Histograms are normalized by the number of pixels to convert into PMF.

though it should be noted that since the random noise does not look like the natural images overall reconstruction quality is low. Gaussian noise is in between Laplace and uniform noise in terms of performance. It is easier to compress than uniform but harder than Laplace distribution. Uniform noise is the hardest distribution since it has the largest entropy. This results in very high distortion and bitrate.

The GDN network performs on par with our network in RD performance in all distributions. Both networks have comparable PSNR and bitrate. On the other hand, the KL divergence between the histograms of inputs and outputs is lower for our network. Overall, the table shows that our network tries to match the distribution of the input which means it is better at learning the image manifold. To further illustrate this phenomenon, we provide Figure 3.9 which shows the histograms of

the random inputs and their corresponding outputs. Noise distributions are dramatically different than the histogram of the outputs. Both networks tend to yield an output around the mean of the input. The variance of the output's histogram is less than the input's. The most important conclusion is that the variance of GDN output is less than Shrinkage output. This explains the blurriness of the outputs of the GDN network since blurry images have low variance whereas sharp images have higher variance in general.

All in all, we can say that the manifold learned by Shrinkage activation is closer to the natural image manifold in high dimensional latent space. On the other hand, there is no clear answer to how you define the natural image manifold and what distance measure to use. Therefore we cannot be sure how and why the Shrinkage network pays more attention to perceptual quality. There is only one thing we know predicted by the RDP theory: the higher the PSNR, the further away we are from the natural image manifold.

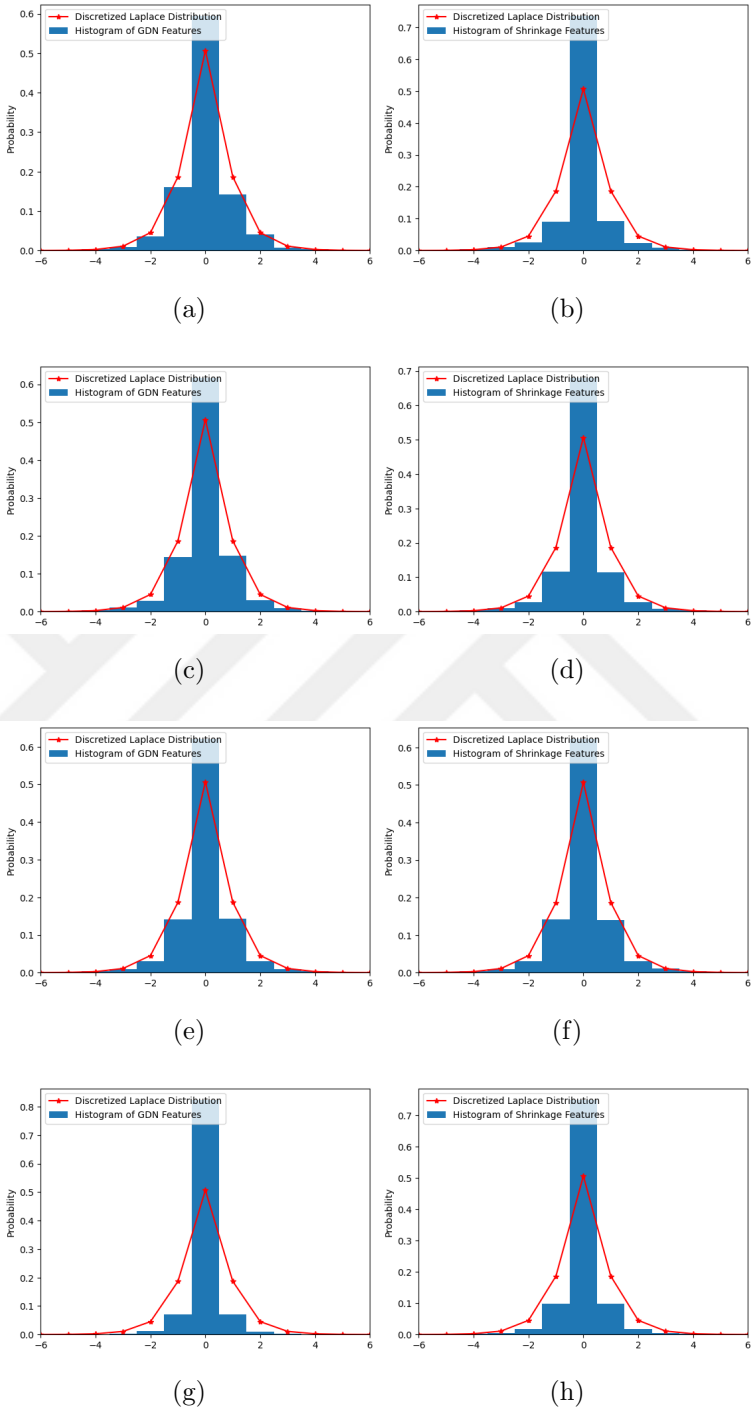


Figure 3.10: Histograms of features after activation layers. Frequencies are normalized so that we can compare with PMF of discretized Laplace distribution. (a) GDN features after 1^{st} activation, (b) Shrinkage features after 1^{st} activation, (c) GDN features after 2^{nd} activation, (d) Shrinkage features after 2^{nd} activation, (e) GDN features after 3^{rd} activation, (f) Shrinkage features after 3^{rd} activation, (g) GDN symbols to be compressed, (h) Shrinkage symbols to be compressed.

Chapter 4

PRACTICAL RATE-DISTORTION-PERCEPTION OPTIMIZATION

The optimization and assessment of compression algorithms have traditionally relied on Shannon's rate-distortion paradigm, a framework aimed at minimizing the distortion, typically quantified by the MSE, at a specified bitrate or vice versa. While this model enables the development of codecs with superior PSNR performance at a given bitrate, the pursuit of optimal PSNR often culminates in visually undesirable blurry images. This discrepancy arises due to the well-known weak correlation between PSNR and the human perception of image quality. In order to gauge perceptual quality, a common approach within the realm of image compression is to undertake subjective tests based on the MOS. This methodology, however, presents its own challenges due to its time-consuming nature, lack of scalability for vast datasets, and the results cannot be optimized for perceptual quality. The recent emergence of learned image compression has introduced the possibility of end-to-end optimization of codecs with perceptual distortion metrics such as SSIM [75] or no-reference perceptual losses, in addition to MSE.

The post-2018 era witnessed rapid advancements in the state-of-the-art in learned image compression, primarily driven by the yearly CLIC. Participants were tasked with compressing images below the threshold of 0.15 bpp while also outperforming BPG in terms of PSNR and SSIM. The final ranking of codecs is decided by human critics. Most participants employed deep learning in the first year of the contest. End-to-end optimized learned image compression presents several key benefits over traditional engineered codecs.

First, it eliminates the necessity for reliance on linear and block-based transforms as well as engineered context models for arithmetic coding. Furthermore,

this method allows for system-wide optimization in accordance with a specified non-convex, but differentiable, loss function. The loss function can be a combination of L^1/L^2 loss, a feature-driven visual distortion loss like SSIM, or a no-reference perceptual loss in addition to rate.

Recent research led by Matsumoto [13, 14] and Blau et al. [1] has revealed the existence of a triple trade-off between rate, distortion (fidelity), and perception (RDP). Perception is defined by the distance between the distributions of the original and reconstructed images. The authors posit minimizing adversarial loss is the only way to optimize for perceptual quality. Nevertheless, the implementation of this innovative rate-distortion-perception optimization approach encounters two significant challenges:

- the absence of a universally agreed-upon objective metric for assessing perceptual quality,
- the lack of a practical approach to independently control rate, distortion, and perception.

Consequently, these challenges impede the current utilization of this framework in learned codecs.

In this chapter, we introduce a practical approach to performing rate-distortion-perception analysis in learned image compression. We perform perception-distortion analysis at various fixed bitrates sequentially. For this purpose, we fix the bitrate by freezing the weights of the encoder and hyperprior networks at successive desired bitrate points. Once the best perception-distortion point is found at a fixed rate, we continue training the decoder of another rate point. We continue this procedure of finding the best perception-distortion points at certain fixed bitrates until the set of desired rate points is covered. Second, we suggest a way to realize rate adaptation for a family of fixed bitrate models. This is done by converting the underlying RD optimization problem to an integer linear programming (ILP) problem. Additionally, the same technique can be used for improving the perceptual performance at a

desired bitrate. Finally, we add a perceptual constraint to the ILP problem in order to produce RDP-optimized images.

4.1 Rate-constrained Optimization of Perception and Distortion in Training

Traditional codecs are conventionally designed to optimize the trade-off between compression rate and distortion by exhaustively searching for optimal compression parameters under strict constraints. However, its neural network counterparts are trained using loss functions that impose soft constraints. As a consequence, the same neural network model may yield varying rate-distortion points when applied to different images. Ballé et al. proposed a method to optimize the rate-distortion trade-off by minimizing the following equation:

$$L = d(x, \hat{x}) + \lambda H(y) \tag{4.1}$$

Here, x represents the original image, $\hat{x}$ denotes the decoded image, $d(x, \hat{x})$ is a distortion function, (e.g., MSE), and $H(y)$ is the entropy (bitrate) of the compressed representation y.

Incorporating a perceptual loss term into the overall loss function further complicates the optimization process, as it requires determining appropriate weights for the three terms. Various approaches have been proposed to optimize these quantities.

The approach taken by Agustsson et al. [63] assumes an upper limit on the entropy of the latent code. In scenarios where the quantization levels are finite and the latent code's dimensionality is fixed, an estimate of the upper bound on entropy can be obtained by assuming that the latent code follows independent uniform distributions. However, it is important to acknowledge that this bound is considerably loose, as the actual latent code does not possess independent or uniform distribution characteristics. To incorporate perceptual loss, an adversarial approach is adopted, wherein a discriminator D is introduced to differentiate between the decoded image and the original image. Consequently, the optimization process involves the simultaneous optimization of two distinct loss functions. These loss functions are expressed

as follows:

$$L = d(x, \hat{x}) + \lambda H(y) - \beta log(D(\hat{x}, y)) \tag{4.2}$$

$$L_D = -log(1 - D(\hat{x}, y)) - log(D(x, y)) \tag{4.3}$$

The additional terms in (4.2) come from the discriminator D, λ, and β controls the trade-off between rate, distortion, and perception. Equation (4.3) is the standard binary cross-entropy loss employed for training the discriminator.

Mentzer et al. [15] also use the loss functions (4.2) and (4.3) in their work, albeit with a notable distinction: Agustsson et al. [63] assigned λ to be 0, indicating that the rate was not directly optimized. Instead, the authors relied on an approximate upper bound, as previously discussed. Since the optimization did not address entropy directly, it was not feasible to enforce a specific distribution on the latent code, as achieved in the RDO framework proposed by [27]. This simplified the optimization process considerably. In contrast, the approach described in [15] utilized both loss functions in (4.2) and (4.3) with non-zero hyperparameters. However, this formulation introduced challenges in optimizing for different bitrates, as modifying one hyperparameter in (4.2) affected the relative weight of the other two terms. To regulate the rate, the authors introduced two hyperparameters, λ^a and λ^b with $\lambda^a \gg \lambda^b$ allowing for adaptive adjustment of the entropy term's weight. If the target rate is exceeded then λ^a was utilized to restore the model to the desired bitrate; otherwise λ^b is used.

Our approach diverges from the aforementioned methodologies as we address the bitrate issue by freezing the encoder. Initially, we train the network utilizing (4.1) to attain a specific target bitrate. As the entropy solely relies on the encoder and the hyperprior network's defined entropy model, fixing these two networks enables us to establish a constant bitrate. Consequently, we can focus on optimizing the trade-off between distortion and perception directly, given that the loss function incorporates a single hyperparameter:

$$L = \gamma d(x, \hat{x}) + L_p(\hat{x}, x) \tag{4.4}$$

where L_p is a generic perceptual loss, γ is a hyperparameter controlling the perception-distortion trade-off.

Algorithm 1 succinctly outlines our proposed methodology for optimizing perception and distortion at a fixed bitrate, employing our practical approach of maintaining a fixed bitrate. This procedure can be iterated to encompass multiple rate points, as desired.

Algorithm 1: Algorithm for PD optimization.

Input: Randomly initialized network

Output: PD optimized network

Train the network to reach the desired rate-distortion point using Eq. 4.1;

Freeze the weights of the encoder and entropy model estimator sub-networks to fix the encoding rate;

Start the search for optimal γ with the loss function in Eq. 4.4;

Find the γ that corresponds to the knee point (see Figures 4.1 and 4.2);

4.1.1 Experimental Results

In this section, we present empirical findings to substantiate the existence of an optimal balance between perception and distortion when encoding images at a fixed bitrate. To this end, we generate multiple decoded images from the same encoded bitstream at a constant rate, employing a procedure that involves optimizing the parameters of the decoder network based on various weighting schemes for the distortion and perceptual loss functions, as detailed above.

We preferred using pretrained models of Bégaint et al. [85] instead of using our proposed network in Chapter 3 because our trained models could not reach the RD performance of Minnen et al.'s [30] models. Their training dataset is not open source, unfortunately. On the other hand, Bégaint et al. have made available pretrained models of [30] for eight distinct bitrates which are close to the RD performance of [30]. We have selected the models corresponding to the four lowest bitrates, as the

trade-off between perception and distortion becomes more evident in the low bitrate range, as suggested by [1]. Prior to training, we freeze the encoder and entropy models. For the perceptual loss, we employed LPIPS v0.1 [77] with VGG features [86]. VGG features have been widely used to enhance perceptual quality, following the findings of Ledig et al. [10] who demonstrated that the utilization of VGG loss substantially improves MOS. However, during the perceptual evaluation, we employ LPIPS with AlexNet features[87], as the authors suggest that this network exhibits a stronger correlation with MOS. Using a different perceptual metric in test time also provides some impartialness in evaluation. The network is trained with Adam optimizer [84] with a learning rate of 10^{-5} and default parameters. We train the network for 400000 iterations on Vimeo-90k dataset [88] with a batch size of 16 on 256×256 patches.

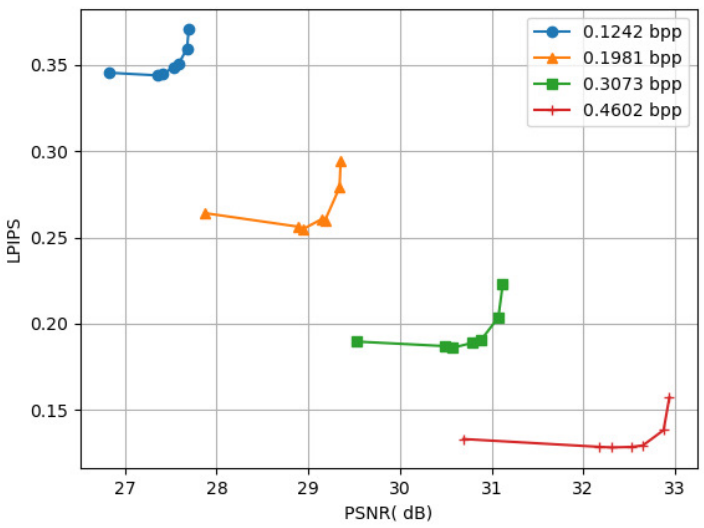


Figure 4.1: LPIPS vs. PSNR plots of models trained at different fixed bitrates on Kodak dataset. The best perception (LPIPS)-distortion (PSNR) trade-off point at each fixed bitrate are determined as the knee-point of the respective curves.

As depicted in Figure 4.1, the perception-distortion trade-off is observed in relation to the LPIPS and the PSNR. Each curve denotes varying bitrates, with the uppermost right-hand point on each curve signifying a pretrained model trained

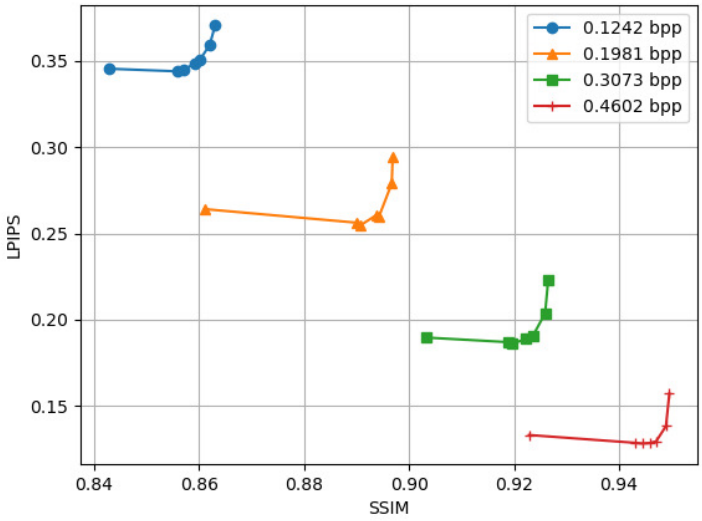


Figure 4.2: LPIPS vs. SSIM plots of models trained at different fixed bitrates on Kodak dataset. The best perception (LPIPS)-distortion (SSIM) trade-off point at each fixed bitrate is determined as the knee-point of the respective curves.

solely using equation (4.1). This model is referred to as MSE only, due to the fact that it is not optimized for perceptual quality. Subsequent to the freezing of the encoder and hyperprior components, the perception-distortion optimization process begins. Subsequent points along each curve represent experimental outcomes using γ values of 10^{-2}, 10^{-3}, 5×10^{-4}, 10^{-4}, 5×10^{-5}, and 0 from right to left respectively. The experiment with $\gamma = 10^{-3}$ improves LPIPS while mildly reducing PSNR. The model trained with $\gamma = 10^{-4}$ manifests some advancement in LPIPS, however, the marginal gain is relatively insignificant, also evident from the slope between the points $\gamma = 10^{-3}$ and $\gamma = 10^{-4}$. The extreme case of $\gamma = 0$ serves as a benchmark, demonstrating the underlying nature of the optimization issue. With the objective of MSE being to minimize pixel-wise discrepancies, the output of the network diverges excessively from the input without MSE loss, this divergence proves detrimental to both PSNR and LPIPS. Figure 4.1 shows the optimal point for terminating hyper-parameter search lies at the knee point of the curves.

In Figure 4.2, we witness the perception-distortion trade-off between LPIPS and

(a) (b) (c)

Figure 4.3: Example crop from Kodak dataset: (a) original image, (b) output of decoder model optimized with respect to MSE only, (c) output of decoder model with $\gamma = 5 \times 10^{-4}$. Both images (b) and (c) are reconstructed from the same encoded bitstream at 0.40 bpp. PSNR of (b) and (c) are 24.58 dB and 24.35 dB, respectively. SSIM scores are also similar: 0.7967 and 0.7920, respectively. Although (b) has slightly less distortion in terms of MSE, (c) looks sharper, especially on the grass and the stones in the water. This is evident from the LPIPS scores: (b) scored 0.41 whereas (c) scored 0.35. (Lower LPIPS is better.)

SSIM. Despite the absence of explicit SSIM optimization, the observations made in reference to Figure 4.1 remain applicable. Upon reaching the knee of the curve, any further decrease in γ is disadvantageous to the network, leading to a rise in distortion devoid of perceptual enhancement. Both figures unequivocally demonstrate the existence of an optimal perception-distortion point for each fixed bitrate.

Figure 4.3 provides an example of an image from the Kodak dataset compressed using the trained networks. The compressed images, while retaining similar distortion levels, are at an identical bitrate. However, Figure 4.3c appears sharper in the depiction of grass and rocks despite higher distortion levels in terms of both PSNR and SSIM. These experimental results underline that our optimization strategy is capable of optimizing rate-distortion-perception. Figure 4.4 displays another example from the Set14 dataset [4]. The image depicts a woman's scarf losing its stripes in both compressed images on the left, however, the MSE-trained network additionally blurs the region beneath the woman's palm, whereas the perceptually

trained network retains sharpness. The evidence presented in Figures 4.3 and 4.4 demonstrate the success of our approach across a spectrum of bitrates.

(a) (b) (c)

Figure 4.4: Example crop from Set14 dataset [4]: (a) original image, (b) output of decoder model optimized with respect to MSE only, (c) output of decoder model with $\gamma = 5 \times 10^{-4}$. Both images (b) and (c) are reconstructed from the same encoded bitstream at 0.16 bpp. PSNR of (b) and (c) are 25.14 dB and 25.11 dB, respectively. SSIM scores are 0.8253 and 0.8243, respectively. Although (b) has slightly less distortion in terms of MSE, (c) looks sharper in the area on the left side of the woman's hand. This is also evident from the LPIPS scores: (b) scored 0.359 whereas (c) scored 0.347. (Lower LPIPS is better.)

4.2 An Integer Linear Programming Approach to Optimize Overall Rate-Distortion-Perception Performance for a Test Set of Images

In our previous work [23], we proposed a post-processing network that improves the quality of BPG compressed images for CLIC 2018. Although the network was trained on images compressed with a single quantization parameter (QP), it could improve the output of different QPs. The competition required a predetermined average bitrate on the test set. A trivial way to conform to this requirement would be encoding all images with a single QP that outputs the desired bitrate. However, this would result in suboptimal use of the allowed bitrate since in a diverse set of images some images would be more complex and, hence, difficult to encode than others. This means we can compress simpler images more and make room for complex

images which in turn increases the fidelity of complex images.

4.2.1 Problem Formulation

In the classical RD optimization framework, the problem is formulated as a constrained optimization problem:

$$\begin{aligned} \min \quad & d(x, \hat{x}) \\ \text{s.t.} \quad & R \leq R_c \end{aligned} \tag{4.5}$$

This problem tells us to minimize the distortion subject to a constraint R_c on the number of bits used R. The problem or its Lagrangian counterpart needs to be solved to choose encoding parameters in a standard encoder.

Suppose there are N images in the test set and we have M encoder-decoder pairs. Assuming the models we train have fixed operational points, i.e. the models do not have continuous rate adaptation, a set of images will be compressed and the results cannot be altered. If achieving a certain overall average bitrate R_{avg} is desired, we can only alter the encoder, hence the bitrate, for individual images. Choosing which image to encode at which bitrate can be formulated as the following problem.

$$\min_{x_i} \quad \sum_{i=1}^{N} d_i^T x_i \tag{4.6a}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} r_i^T x_i \leq N R_{avg}, \tag{4.6b}$$

$$\mathbf{1}_{1\times M} x_i = 1, \ \forall i = 1, 2, \cdots N, \tag{4.6c}$$

$$x_{i_j} \in \{0, 1\}, \ \forall i = 1, 2, \cdots N, \ \forall j = 1, 2, \cdots M \tag{4.6d}$$

where x_i is $M \times 1$ one-hot vector such that the entry which equals 1 indicates the model selected for the i^{th} image, d_i is $M \times 1$ vector whose components are the mean squared error between the raw and encoded images for different models, and r_i is $M \times 1$ vector whose components denote the bitrate in bpp when i^{th} image is encoded with all possible models. Constraint 4.6b enforces that the average bitrate of all images is below the given bitrate constraint R_{avg}. Constraints 4.6c and 4.6d

require that one and only one model is selected for each image. In principle, one can also optimize for perception instead of distortion by changing the optimization goal 4.5 with the perceptual score. However, RP optimization does not make sense in the context of training a codec since without the fidelity criterion decoder cannot produce an output similar to the input.

In the more general RDP optimization problem, we wish to choose the optimal model for each image to maximize the average PSNR subject to a desired bitrate and perceptual quality. In order to realize this, we compress and decompress all of the images and calculate the measures we want to optimize for. Then we formulate this problem as an integer linear programming problem, given by

$$\min_{x_i} \quad \sum_{i=1}^{N} d_i^T x_i \tag{4.7a}$$

$$\text{s.t.} \quad \sum_{i=1}^{N} r_i^T x_i \leq N R_{avg}, \tag{4.7b}$$

$$\sum_{i=1}^{N} p_i^T x_i \geq N P_{avg}, \tag{4.7c}$$

$$\mathbf{1}_{1\times M} x_i = 1, \ \forall i = 1, 2, \cdots N, \tag{4.7d}$$

$$x_{i_j} \in \{0, 1\}, \ \forall i = 1, 2, \cdots N, \ \forall j = 1, 2, \cdots M \tag{4.7e}$$

This problem is the same as the optimization problem 4.6 if constraint 4.7c is excluded. This constraint enforces that the average perceptual score of all encoded images is above the desired perceptual quality P_{avg}. It should be noted that the optimization target can be exchanged with either rate or perception constraint to make traversing RDP space easier. In fact, we optimized for the perceptual score instead of distortion in our experiments.

4.2.2 Experimental Results

Table 4.1 shows the performance of BPG on the test set of CLIC 2018 for QPs between 38 and 42. Due to the requirement of the challenge, images were to be encoded at 0.15 bpp. From Table 4.1, we see that the only QP that yields below 0.15

Table 4.1: Results for the CLIC 2018 test set comparing BPG before and after optimization according to our RD formulation

BPG				RD Optimized BPG			
QP	PSNR	MS-SSIM	Bitrate (bpp)	No.	PSNR	MS-SSIM	Bitrate (bpp)
38	31.414	0.959	0.237	1			
39	30.833	0.954	0.206	0			
40	30.333	0.949	0.179	109	29.692	0.944	0.15
41	29.735	0.943	0.152	120			
42	29.249	0.938	0.132	56			

bpp is 42. However, we can achieve the desired 0.15 bpp if we solve the optimization problem 4.6. The right side of Table 4.1 shows the distribution of chosen QPs under the No. column. Given PSNR, MS-SSIM, and bitrate are the average of the images that are selected according to this distribution. The fidelity we achieve is better than QP of 42 due to the fact that the bulk of images are encoded with QPs of 40 and 41. At this bitrate, our method achieves 29.69 dB which is 0.4 dB better than the QP of 42. Additionally, PSNR and MS-SSIM are on par with the QP of 41 even though the bitrate is slightly less. These results prove that our method is resourceful in choosing images that will use the available bandwidth most efficiently.

Next, we present experiments on pretrained HIFIC models provided by the authors [15] to confirm our RDP formulation. The authors provide 3 models for 3 distinct bitrates and they are all trained adversarially.

Figure 4.5 compares the baseline HIFIC models (HIFIC-lo, HIFIC-mi, HIFIC-hi) with RDP-optimized models. For reference results of RD and RP optimized models called HIFIC-RD and HIFIC-RP are also included. LPIPS (AlexNet) and MSE are chosen for the perceptual and distortion measure in the optimization procedure. Lower LPIPS values mean better perceptual quality. Rate constraints of optimized models are chosen to be around the bitrate of HIFIC-mi to provide enough room for the solution. If the rate constraint is selected too low, the likelihood of infeasibility

increases. If it is selected too high, either the solution becomes the same as picking the HIFIC-hi, or the solution stays the same as one of the lower rate solutions due to the perception constraint. HIFIC-RD has a worse perceptual score than both HIFIC-RP and HIFIC-mi as predicted by the RDP theory. Since it has a lower perceptual quality, it is better in terms of distortion as expected. Similarly, HIFIC-RDP1 and HIFIC-RDP2 achieve the same bitrate, yet both perceptual scores of HIFIC-RDP2 are better than HIFIC-RDP1 despite being worse in terms of PSNR. Of all solutions that attain 0.28 bpp, HIFIC-RD is the best in terms of PSNR whereas HIFIC-RP is the most optimal in the perceptual sense. The performance of HIFIC-RDP3 and HIFIC-RDP4 demonstrates the trade-off between RD and RP optimized models.

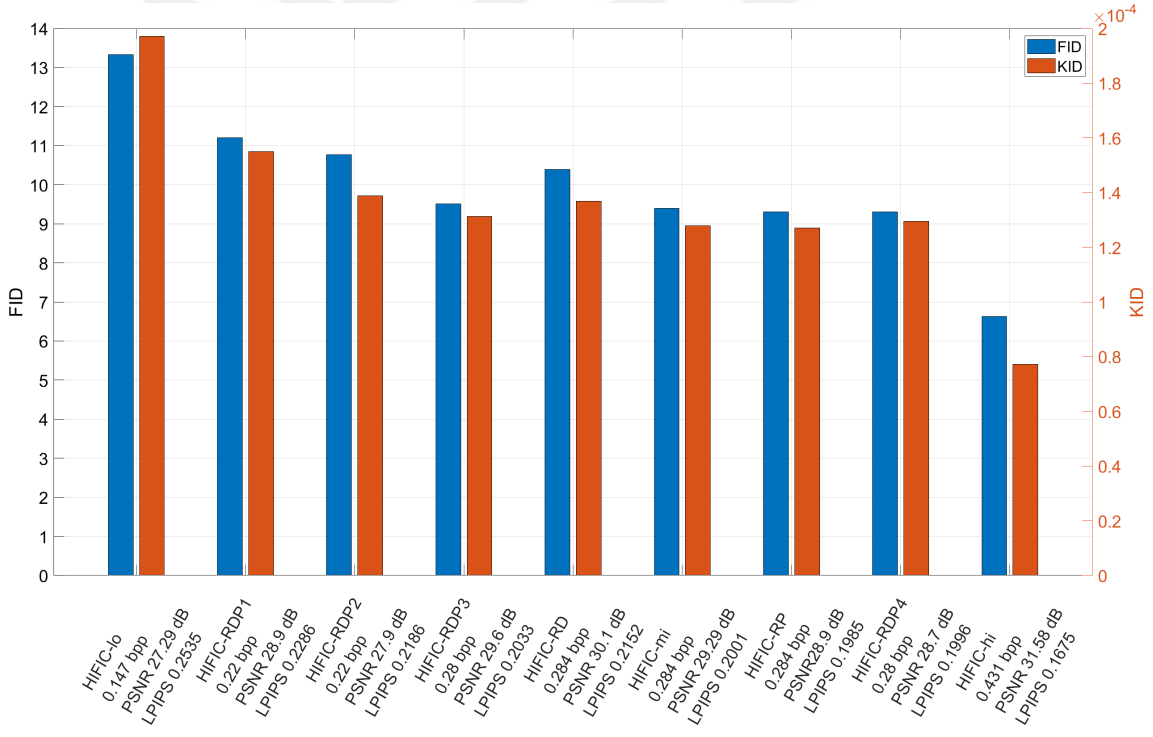


Figure 4.5: Comparison of HIFIC models by perceptual quality. HIFIC-lo, mi and hi are the baseline pretrained models. HIFIC-opt columns are the results of our method of RDP optimization. Lower is better for FID, KID, and LPIPS.

Let us compare our optimized models at 0.28 bpp. Table 4.2 shows the selected encoders for optimized models. Although most of the images are encoded at the lowest bitrate HIFIC-RD is the most successful in terms of PSNR. Some images (e.g.

(a) (b) (c)

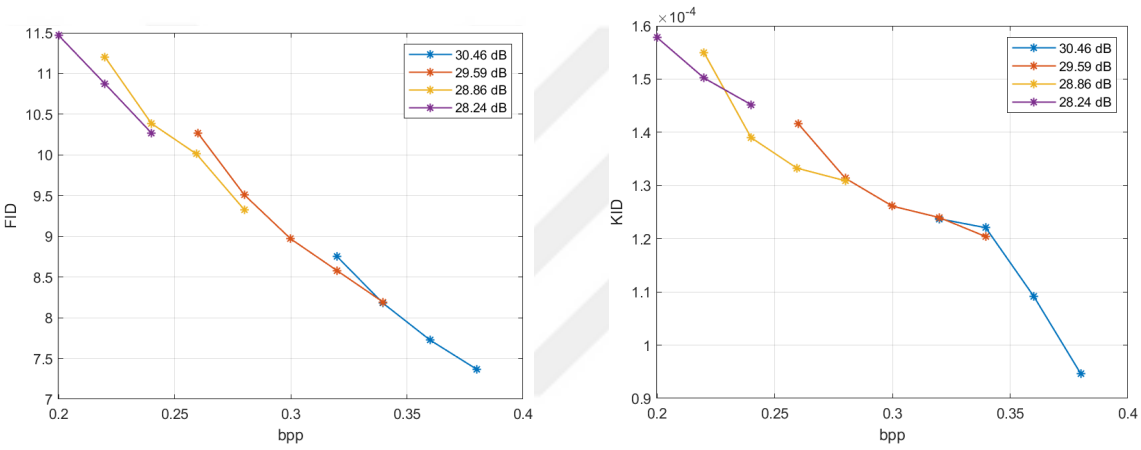
Figure 4.6: Example crop from CLIC 2018 test set: (a) original crop, (b) output of HIFIC-RDP3, (c) output of HIFIC-RDP4. (b) is encoded by HIFIC-lo at 0.1437 bpp, 32 dB PSNR, LPIPS 0.1980. (c) is encoded by HIFIC-mi at 0.2706 bpp, 34.5 dB PSNR, LPIPS 0.1481.

Table 4.2: Selected encoders for RDP optimized output

	HIFIC-RD	HIFIC-RDP3	HIFIC-RDP4	HIFIC-RP
HIFIC-lo	158	55	31	27
HIFIC-mi	65	210	197	198
HIFIC-hi	63	21	58	61

dull images that have little content) cannot be enhanced much even though they are encoded at higher rates. Our algorithm finds these images and encodes them at the lowest possible bitrate so that the bandwidth left is used more efficiently. Yet, this drops perceptual quality a lot. Since distortion is not always correlated with perception, optimizing for perception immediately increases the number of images encoded by HIFIC-mi model. Normally, without distortion constraint, the optimized model would like to choose as many HIFIC-hi encoded images as given that they fit the rate constraint. However, the distortion constraint of HIFIC-RDP3 is so dominant, it has to use the bandwidth for improving distortion. This leads to worse LPIPS than HIFIC-RDP4. The performance of HIFIC-RDP4 and HIFIC-RP is fairly close therefore their choices of encoders are also similar.

Figure 4.6 demonstrate the difference between solution of ILP problems. The text in Figure 4.6b is clearly illegible whereas the text on top of the numbers in Figure 4.6c can be easily read. HIFIC-RDP3 sacrifices 2.5 dB PSNR to improve the fidelity of other images. Note that due to the logarithm in the definition of PSNR, the same amount of PSNR difference means very different at lower and higher fidelity levels. Therefore HIFIC-RDP3 prefers losing 2.5 dB at a low distortion level if it leads to an improvement of a very distorted image.



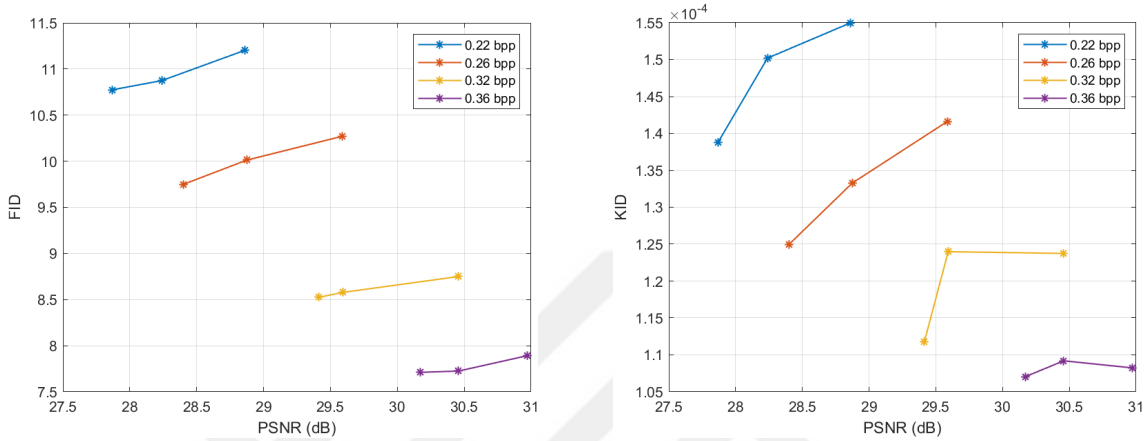
(a) (b)

Figure 4.7: Rate-Perception plane. (a) FID versus rate, (b) KID versus rate. Both (a) and (b) show perceptual quality is proportional to bitrate when distortion is constant.

Figure 4.7 depicts the rate-perception plane. At constant distortion, perceptual quality can be improved by increasing the rate. The figures also show if the rate is fixed, lower distortion can be achieved only if the perceptual score is sacrificed. However, Figure 4.7b shows some of the curves are intersecting contrary to the RDP theory. This might be due to imperfections in the KID metric. In an ideal world, we would measure perception by the divergence between compressed and raw images. Since we cannot realize this some irregularities might occur.

Figure 4.8 demonstrates the negative relationship between distortion and perception. As the theory suggests distortion-perception trade-off is more apparent in lower rates. This is evident by the slope of the curves in Figure 4.8a which are

steeper at lower rates. Overall, Figures 4.7 and 4.8 support the claim that distortion and perception are at odds when the bitrate is constant: as perceptual quality improves, PSNR drops.



(a) (b)

Figure 4.8: Perception-Distortion plane. (a) FID versus PSNR, (b) KID versus PSNR. Both (a) and (b) show perceptual quality and distortion are inversely correlated when the rate is fixed.

Before concluding the chapter, we should also mention the weaknesses of ILP-based RDP optimization. The first disadvantage is that ILP is an NP-complete problem. This means it will be computationally challenging if the number of options grows too much. Not only the finding the solution is difficult but also the preparation it takes might take too long since we need to calculate the output for every setting of the codec before solving the ILP problem. On the other hand, if there are too few settings to choose from, some of the desired constraints might result in infeasible problems. This is the case for the results presented in this chapter. Unfortunately, some of the constructed ILP problems did not have solutions.

Chapter 5

CONCLUSION

In this thesis, we explored learned image compression algorithms from the point of RDP. It has been demonstrated that a network utilizing hard shrinkage activation generates latents that better conform to a Laplacian distribution in comparison to the ones generated by a network with GDN activation. Furthermore, it has been shown that hard shrinkage activation exhibits lower computational complexity, and when implemented, the method outperforms its GDN equivalent in terms of speed on both CPU and GPU. This speed advantage is critical when incorporating these end-to-end learned codecs in real-world scenarios. Even though both types of activations showed similar outcomes in terms of quantitative metrics such as PSNR and SSIM, visual inspections highlight images produced by hard shrinkage are perceptually better.

Second, our work has put forth a practical methodology to set a fixed bitrate for the encoded image, enabling perception-distortion analysis to be performed at a constant rate. We have also suggested a systematic approach to ascertain the optimal perception-distortion trade-off point at a given fixed rate. We used pretrained models of Bégaint et al. [85] in experiments instead of the network we proposed in Chapter 3 in order to show even the most optimized models can benefit from our method. Due to the unavailability of the original dataset and lack of detail on optimization tips, we could not reach RD performance of Minnen et al. [30] with the Shrinkage network. On the other hand, pretrained models of Bégaint et al. were on par with Minnen et al therefore we proceeded with these models. Our experimental results indicated the existence of a prime perception-distortion trade-off point for each fixed bitrate. Furthermore, we have developed a framework and a procedure for carrying out perception-distortion analysis across a range of fixed bitrates, ensuring

optimal rate-distortion-perception optimization for learned image compression. Our method can prove to be useful even with continuous rate adaptive models because our model avoids training on all three loss functions at the same time. For example, conditional generator proposed in Agustsson et al. [67] can be trained by our method simplifying the training process.

Third, we demonstrated that we can optimize for perceptual score by using our previous formulation of RD optimization on fixed models. Moreover, we expand RD optimization on fixed-rate compression models to RDP optimization by adding a perceptual constraint to ILP problem. Our findings revealed the solution of the proposed ILP improves perceptual quality both quantitatively and qualitatively. Additionally, empirical results showed that emerged solutions conform with the theoretical results of previous works on RDP. This method can be used to squeeze the last ounce of performance out of a fixed-rate model. The proposed method can be incorporated with video compression algorithms by solving ILP for every group of pictures. On the other hand, there are certain limitations to this method in terms of computational complexity since ILP is an NP-complete problem. Furthermore, our method needs to know the outcomes of every possible encoding setting which means there is considerable work to be undertaken before starting the optimization process.

In conclusion, the proposed methods are shown to be practical and effective means of RDP optimization of learned image codecs.

BIBLIOGRAPHY

[1] Y. Blau and T. Michaeli, "Rethinking lossy compression: The rate-distortion-perception tradeoff," in *eprint arXiv:abs/1901.07821*, May 2019.

[2] L. Zhou, C. Cai, Y. Gao, S. Su, and J. Wu, "Variational autoencoder for low bit-rate image compression," in *The IEEE Conf. on Comp. Vis. and Patt. Recog. (CVPR) Workshops*, June 2018.

[3] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *2015 IEEE Int. Conf. on Comp. Vis. (ICCV)*, pp. 370–378, Dec 2015.

[4] R. Zeyde, M. Elad, and M. Protter, "Single image scale-up using sparse-representations," in *Curves and Surfaces*, pp. 711–730, Springer, 2012.

[5] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, vol. 38, no. 2, pp. 295–307, 2016 (first published in ECCV 2014).

[6] R. Timofte, E. Agustsson, L. Van Gool, M.-H. Yang, L. Zhang, *et al.*, "NTIRE 2017 challenge on single image super-resolution: Methods and results," in *IEEE Conf. on Comp. Vis. Patt. Recog. (CVPR)*, July 2017.

[7] R. Timofte, S. Gu, J. Wu, L. Van Gool, M.-H. Yang, L. Zhang, *et al.*, "NTIRE 2018 challenge on single image super-resolution: Methods and results," in *IEEE Conf. on Comp. Vis. Patt. Recog. (CVPR)*, July 2018.

[8] J. Cai, S. Gu, L. Zhang, *et al.*, "NTIRE 2019 challenge on single image super-resolution: Methods and results," in *IEEE Conf. on Comp. Vis. and Patt. Recog. (CVPR)*, July 2019.

[9] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018.

[10] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *eprint arXiv:abs/1609.04802*, May 2017.

[11] G. K. Wallace, "The jpeg still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, pp. xviii–xxxiv, Feb 1992.

[12] "Better Portable Graphics encoder/decoder and bitstream specification." https://bellard.org/bpg/.

[13] R. Matsumoto, "Introducing the perception-distortion tradeoff into the rate-distortion theory of general information sources," *IEICE Communications Express*, vol. 7, no. 11, p. 427–431, 2018.

[14] R. Matsumoto, "Rate-distortion-perception tradeoff of variable-length source coding for general information sources," *IEICE Communications Express*, vol. 8, no. 2, p. 38–42, 2019.

[15] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson, "High-fidelity generative image compression," *Advances in Neural Information Processing Systems*, vol. 33, 2020.

[16] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The jpeg2000 still image coding system: an overview," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, pp. 1103–1127, 2000.

[17] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. Van Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," in *Proceedings of the 31st Int. Conf. on Neural Information Processing Systems, NIPS*, (NY, USA), p. 1141–1151, 2017.

[18] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," in *eprint arXiv:abs/1703.00395*, 2017.

[19] 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)*, pp. 1–5, 2016.

[20] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *2nd Int. Conf. on Learning Representations, ICLR 2014, Banff, AB, Canada*, April 2014.

[21] O. Kirmemis and A. M. Tekalp, "Shrinkage as activation for learned image compression," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 1301–1305, 2020.

[22] O. Kirmemis and A. M. Tekalp, "A practical approach for rate-distortion-perception analysis in learned image compression," in *2021 Picture Coding Symposium (PCS)*, pp. 1–5, 2021.

[23] O. Kirmemis, G. Bakar, and A. Murat Tekalp, "Learned compression artifact removal by deep residual networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018.

[24] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *2017 IEEE Conf. on Patt. Recog. Vis. and Patt. Recog. (CVPR)*, pp. 5435–5443, July 2017.

[25] M. Covell, N. Johnston, D. Minnen, S. J. Hwang, J. Shor, S. Singh, D. Vincent, and G. Toderici, "Target-quality image compression with recurrent, convolutional neural networks," in *eprint arXiv:abs/1705.06687*, 2017.

[26] 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 *eprint arXiv:abs/1703.10114*, Mar 2017.

[27] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *5th Int. Conf. on Learning Representations, ICLR 2017, Toulon, France*, April 2017.

[28] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici, "Nonlinear transform coding," *IEEE Jour. Special Topics in Signal Processing*, vol. 15, no. 2, pp. 339–353, 2021.

[29] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *6th Int. Conf. on Learning Representations, ICLR 2018, Vancouver, BC, Canada*, May 2018.

[30] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Advances in Neural Information Processing Systems 31* (S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, eds.), pp. 10771–10780, Curran Associates, Inc., 2018.

[31] D. Minnen and S. Singh, "Channel-wise autoregressive entropy models for learned image compression," in *2020 IEEE International Conference on Image Processing (ICIP)*, pp. 3339–3343, 2020.

[32] Y. Choi, M. El-Khamy, and J. Lee, "Variable rate deep image compression

with a conditional autoencoder," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2019.

[33] F. Yang, L. Herranz, J. v. d. Weijer, J. A. I. Guitián, A. M. López, and M. G. Mozerov, "Variable rate deep image compression with modulated autoencoder," *IEEE Signal Processing Letters*, vol. 27, pp. 331–335, 2020.

[34] C. Jia, Z. Ge, S. Wang, S. Ma, and W. Gao, "Rate distortion characteristic modeling for neural image compression," in *2022 Data Compression Conference (DCC)*, pp. 202–211, 2022.

[35] S. Ayzik and S. Avidan, "Deep image compression using decoder side information," in *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part XVII*, vol. 12362, pp. 699–714, 2020.

[36] Z. Cui, J. Wang, S. Gao, T. Guo, Y. Feng, and B. Bai, "Asymmetric gained deep image compression with continuous rate adaptation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10532–10541, June 2021.

[37] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning end-to-end lossy image compression: A benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 8, pp. 4194–4211, 2022.

[38] 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.

[39] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems* (I. Guyon, U. V. Luxburg, S. Bengio,

H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.

[40] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[41] J. Lee, S. Cho, and M. Kim, "An end-to-end joint learning scheme of image compression and quality enhancement with improved entropy minimization," in *eprint arXiv:abs/1912.12817*, 2019.

[42] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," in *the 7th Int. Conf. on Learning Representations*, May 2019.

[43] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, "Checkerboard context model for efficient learned image compression," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 14766–14775, 2021.

[44] 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 (CVPR)*, pp. 5718–5727, June 2022.

[45] L. Helminger, A. Djelouah, M. Gross, and C. Schroers, "Lossy image compression with normalizing flows," in *Proceedings of the International Conference on Learning Representations Workshop Neural Compression*, 2021.

[46] Y. Xie, K. L. Cheng, and Q. Chen, "Enhanced invertible encoding for learned image compression," in *Proceedings of the 29th ACM International Conference on Multimedia*, MM '21, Association for Computing Machinery, 2021.

[47] H. Ma, D. Liu, R. Xiong, and F. Wu, "iwave: Cnn-based wavelet-like transform for image compression," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1667–1679, 2020.

[48] 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.

[49] 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 (CVPR) Workshops*, June 2018.

[50] J. Chen, L. Yu, J. Wang, W. Shi, Y. Ge, and W. Tong, "On the rate-distortion-perception function," *IEEE Journal on Selected Areas in Information Theory*, vol. 3, no. 4, pp. 664–673, 2022.

[51] L. Theis and A. B. Wagner, "A coding theorem for the rate-distortion-perception function," in *Neural Compression Workshop at ICLR*, 2021.

[52] G. Zhang, J. Qian, J. Chen, and A. J. Khisti, "Universal rate-distortion-perception representations for lossy compression," in *Advances in Neural Information Processing Systems* (A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, eds.), 2021.

[53] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[54] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[55] C. Chen, X. Niu, W. Ye, S. Wu, B. Bai, W. Chen, and S.-J. Lin, "Computation of rate-distortion-perception functions with wasserstein barycenter," *eprint arXiv:abs/2304.14611*, 2023.

[56] Z. Yan, F. Wen, R. Ying, C. Ma, and P. Liu, "On perceptual lossy compression: The cost of perceptual reconstruction and an optimal training framework," in *Proceedings of the 38th International Conference on Machine Learning*, vol. 139, pp. 11682–11692, PMLR, 18–24 Jul 2021.

[57] O. Rippel and L. Bourdev, "Real-time adaptive image compression," in *Proceedings of the 34th Int. Conf. on Machine Learning*, vol. 70 of *Proceedings of Machine Learning Research*, (Sydney, Australia), pp. 2922–2930, 06–11 Aug 2017.

[58] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems* (Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, eds.), vol. 27, Curran Associates, Inc., 2014.

[59] S. Santurkar, D. Budden, and N. Shavit, "Generative compression," in *2018 Picture Coding Symposium (PCS)*, pp. 258–262, 2018.

[60] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, p. 214–223, 2017.

[61] M. Tschannen, E. Agustsson, and M. Lucic, "Deep generative models for distribution-preserving lossy compression," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, (Red Hook, NY, USA), p. 5933–5944, Curran Associates Inc., 2018.

[62] S. Dash, G. Kumaravelu, V. Naganoor, S. K. Raman, A. Ramesh, and H. Lee, "Compressnet: Generative compression at extremely low bitrates," in *2020 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2314–2322, 2020.

[63] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. V. Gool, "Generative adversarial networks for extreme learned image compression," in *eprint arXiv:abs/1804.02958*, 2019.

[64] Y. Kim, S. Cho, J. Lee, S.-Y. Jeong, J. S. Choi, and J. Do, "Towards the perceptual quality enhancement of low bit-rate compressed images," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020.

[65] S. Gao, Y. Shi, T. Guo, Z. Qiu, Y. Ge, Z. Cui, Y. Feng, J. Wang, and B. Bai, "Perceptual learned image compression with continuous rate adaptation," in *4th Challenge on Learned Image Compression*, Jun 2021.

[66] D. He, Z. Yang, H. Yu, T. Xu, J. Luo, Y. Chen, C. Gao, X. Shi, H. Qin, and Y. Wang, "Po-elic: Perception-oriented efficient learned image coding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pp. 1764–1769, June 2022.

[67] E. Agustsson, D. Minnen, G. Toderici, and F. Mentzer, "Multi-realism image compression with a conditional generator," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 22324–22333, June 2023.

[68] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Magazine*, vol. 15, pp. 74–90, Nov 1998.

[69] Y. Blau, R. Mechrez, R. Timofte, T. Michaeli, and L. Zelnik-Manor,

"The 2018 PIRM challenge on perceptual image super-resolution," in *eprint arXiv:abs/1809.07517*, Jan 2019.

[70] A. Mittal, R. Soundararajan, and A. C. Bovik, "Making a "completely blind" image quality analyzer," *IEEE Signal Processing Letters*, vol. 20, no. 3, pp. 209–212, 2013.

[71] C. Ma, C.-Y. Yang, X. Yang, and M.-H. Yang, "Learning a no-reference quality metric for single-image super-resolution," in *eprint arXiv:abs/1612.05890*, 2016.

[72] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, and X. Chen, "Improved techniques for training gans," in *Advances in Neural Information Processing Systems* (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.

[73] M. Heusel, H. Ramsauer, T. Unterthiner, B. Nessler, and S. Hochreiter, "Gans trained by a two time-scale update rule converge to a local nash equilibrium," in *eprint arXiv:abs/1706.08500*, 2018.

[74] M. Bińkowski, D. J. Sutherland, M. Arbel, and A. Gretton, "Demystifying mmd gans," in *the 6th Int. Conf. on Learning Representations (ICLR)*, May 2018.

[75] Zhou Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. on Image Processing*, vol. 13, no. 4, pp. 600–612, 2004.

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

[77] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *IEEE/CVF Conf. Comp. Vision and Patt. Recog. (CVPR)*, June 2018.

[78] Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, and M. Manohara, "Toward a practical perceptual video quality metric." https://netflixtechblog.com/toward-a-practical-perceptual-video-quality-metric-653f208b9652, June 2016.

[79] B. Lim, S. Son, H. Kim, S. Nah, and K. M. Lee, "Enhanced deep residual networks for single image super-resolution," in *IEEE Conf. on Comp. Vis. Patt. Recog. (CVPR) Workshop*, July 2017.

[80] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," in *4th Int. Conf. on Learning Representations, ICLR 2016, San Juan, Puerto Rico*, May 2016.

[81] L. Zhou, Z. Sun, X. Wu, and J. Wu, "End-to-end optimized image compression with attention mechanism," in *The IEEE Conf. on Patt. Recog. Vis. and Patt. Recog. (CVPR) Workshops*, June 2019.

[82] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.

[83] W. Shi, J. Caballero, F. Huszár, J. Totz, A. P. Aitken, R. Bishop, D. Rueckert, and Z. Wang, "Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network," in *2016 IEEE Conf. on Patt. Recog. Vis. and Patt. Recog. (CVPR)*, pp. 1874–1883, June 2016.

[84] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd Int. Conf. on Learning Representations, ICLR*, May 2015.

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

[86] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[87] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), vol. 25, Curran Associates, Inc., 2012.

[88] T. Xue, B. Chen, J. Wu, D. Wei, and W. T. Freeman, "Video enhancement with task-oriented flow," *International Journal of Computer Vision (IJCV)*, vol. 127, no. 8, pp. 1106–1125, 2019.