LEARNING FOR COMPRESSION

Report submitted to the SASTRA Deemed to be University
as the requirement for the course

BCSCCS801: **Project Work**

Submitted by

Venkateshwar Ragavan
(Regno: 121003308, B.Tech Computer Science and Engineering)

June 2021

SCHOOL OF COMPUTING
THANJAVUR, TAMIL NADU, INDIA – 613 401

ACKNOWLEDGEMENT

Firstly, I express my gratitude to **Dr S.Vaidhyasubramaniam**, Vice-Chancellor, SASTRA Deemed to be University who provided all facilities and necessary encouragement during the study.

I extend my sincere thanks to **Dr.R. Chandramouli**, Registrar, SASTRA Deemed To be University for providing an opportunity to pursue this project.

I dedicate my sincere thanks to **Dr Umamakeswari A**, Dean, SoC and **Dr K.Shankar Sriram**, Associate Dean, SoC, SASTRA Deemed to be University, for their support in the accomplishment of this project work.

I render sincere thanks to the project coordinator **Dr Rajendiran P**, Associate Professor, SoC, SASTRA Deemed to be University, for their involvement and encouragement during this mini-project.

I owe my deepest gratitude to mentor **Dr Joan Serra-Sagrista**, Full Professor, Head of Department, Department of Information and Communications Engineering, School of Engineering, Universitat Autònomo de Barcelona, for his continuous support and guidance throughout the process during the pursuit of my project work. His deep insight in the field and valuable suggestions helped me in making progress through my project work.

I thank my parents and friends who stood by our side and offered moral support throughout this period.

LIST OF FIGURES

ABSTRACT

Data Compression is a quintessential part of day-to-day computer applications. It has helped in the efficient storage and transmission of data. The rise of Machine Learning and Deep Learning which has facilitated the processing of data such as text, images to find hidden patterns has led to those techniques being leveraged across a wide spectrum of applications including Data Compression. This is a comprehensive survey on the state of the art Machine Learning algorithms used for Data Compression, most importantly, Image Compression. Besides, this also covers the fundamentals of Data Compression, types of Data Compression algorithms, types of Performance metrics and datasets.

Keywords: Compression, Machine Learning, Deep Learning

TABLE OF CONTENTS

CHAPTER 1
INTRODUCTION

1.1 Artificial Intelligence

Artificial Intelligence (AI) is the study of an artificial entity or agent that responds intelligently to its environment. It is the study of intelligent beings and how their behaviour can be incorporated into an artificial entity such as a computer. The term *Artificial Intelligence* was coined in 1956 by *John McCarthy* [1]. A particular class of very popular AI algorithms called Deep Learning which deals with *Artificial Neural Networks*, has resurged from the *AI Winter* and is now being leveraged ubiquitously in myriad applications. Artificial Intelligence is an amalgamation of a wide variety of disciplines such as Psychology, Neuroscience, Mathematics, Computer Science etc. The field of AI has its major subfields such as Machine Learning, Deep Learning etc.

1.1.1 Machine Learning

It is a subfield of AI which is the study of algorithms that *learn* from previous experience. They create a model or approximate a function based on the given data and use it to make predictions. They are also used to find any hidden patterns in the data. Machine Learning algorithms are mainly categorized based on the availability of labelled data. The class of algorithms that learn from unlabeled data are called the *Unsupervised Learning* algorithms. *PCA (Principal Component Analysis), K-Means* are an example of these algorithms. The class of algorithms that learn from labelled data are called *Supervised Learning* algorithms. *Linear Regression, Logistic Regression* and *Support Vector Machines* are examples of these algorithms. *Semi-Supervised Learning* algorithms are a hybrid of *Unsupervised* and *Supervised Learning*, i.e. it learns from very few labelled data and a significant amount of unlabeled data. The majority of the Machine Learning Algorithms try to optimize an *objective function* (1.1). F Is the total Loss or objective function of the training samples, f Is the loss for individual training samples, x is the model parameters that need to be optimized, $\boldsymbol{\xi}$ is the data sample or data points, $\boldsymbol{\lambda}$ is the regularization parameter that is used to mitigate overfitting of model parameters to the training samples and ||.|| is the Euclidean norm.

$$F(x) = \sum_{i=1}^{n} f_i(x, \xi) \quad + \quad \lambda \, ||x|| \tag{1.1}$$

1.1.2 Deep Learning

It is a branch of Machine Learning that deals with *Deep Neural Networks* or *Artificial Neural Networks*. They were inspired by their biological counterparts which communicate and process information. It has been employed in raft fields such as *Computer Vision, Natural Language Processing, Generative modelling, Speech Recognition, Reinforcement Learning, Data Compression* etc. owing to its State of the Art performance in the respective fields. Deep Neural Networks are *Universal Function Approximators* [2], hence they can approximate non-linear functions to an acceptable error threshold with a high generalization which makes them a preferable candidate for modelling complex problems. A Deep Neural Net comprises the interconnection of nodes or *units* which has *weights* assigned to each of them. They have a layered structure. The Neural Network has an *Input* and an *Output layer*. The layers in between are called *Hidden layers*. At the junction of interlinks of many units lies another unit which has an

activation function, σ, that activates the weighted sum of inputs from other connected units. The activation function is used to induce non-linearity into the Neural Network.

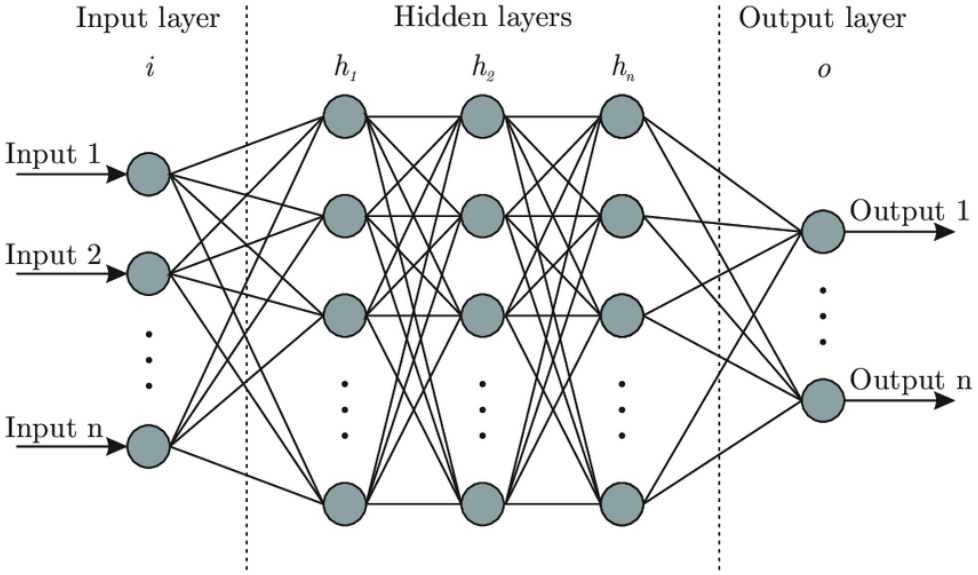

Fig 1.1: A Neural Network with 3 hidden layers.

1.1.4 Shallow Learning

These are the type of learning algorithms that require preemptive feature selection and extraction, unlike Deep Neural Networks that extracts their features. Machine Learning algorithms such as K-Means, Logistic Regression fall under this category. All the Supervised, Unsupervised and Semi-supervised algorithms that use handcrafted features come under this category.

1.2 DATA COMPRESSION

Data Compression is the process of encoding data into a different form to reduce its size. The new data representation, which is of less size, offers a lot of advantages such as efficient use of storage hardware, network bandwidth which results in savings in cost, storage memory and time for data transmission. In Data Compression literature, the data to be compressed is called a *Message.* A compression task has 2 components, *Encoder* and *Decoder*. The *Encoder* module transforms the *Message* into a form with a reduced number of bits. The *Decoder* module is used to reconstruct the original *Message* from its encoded form. There are 2 classes of Data Compression algorithms, Lossy and Lossless compression algorithms. Lossy algorithms tend to reconstruct the Message by approximating its original form. Lossless algorithms reconstruct the exact Message from its compressed form.

Data Compression came into light with the introduction of the *Morse Code*, which uses a combination of dots, dashes and spaces to represent letters, numbers and punctuation marks. It was the first attempt at compressing data. The modern fundamentals of data compression took shape in the 1940s with the inception of *"Information Theory"* by *Claude Shannon*. The *Shannon-Fano coding*, which is a term for 2 separate yet closely related algorithms, *Shannon's method* [3] and *Fano's method*, introduces the usage of prefix codes which is composed of a set of characters and their frequencies. *David Huffman* introduced *Huffman Coding* [4] in 1951 which used variable-length encoding of characters in place of the fixed-length encoding.

In 1977, *Abraham Lempel* and *Jacob Ziv* proposed a pointer-based encoding method called the *LZ78 algorithm* which uses a dictionary for compressing data. In the 1980s, *Terry Welch* modified the LZ78 algorithm and proposed the *LZW (Lempel-Ziv-Welch) algorithm* which became the

standard for general-purpose compression systems. The early 1990s saw the onset of *Lossy* compression algorithms. Many of the current compression standards like *JPEG, MP3, PNG* were developed in the mid-1990s.

1.2.1 Fundamentals: Entropy

The notion of *Entropy (*aka *Shannon Entropy*) was introduced by *Shannon* in 1948 [3]. The *Entropy* of a distribution is the expected amount of information in an event drawn from that distribution [5]. Entropy can be defined for random variables, processes, vectors and dynamical systems. For a discrete random variable X with a set of possible states denoted by x_i and $P(x_i)$ denoting probabilities of each state, the *Entropy* of X is given by (1.2)

$$H(S) = \quad - \sum_{i=1}^{n} P\left(x_i\right) log\left(P\left(x_i\right)\right) \tag{1.2}$$

1.2.2 Lossy and Lossless Algorithms

Data compression falls under 2 categories, *Lossy* and *Lossless* compression. *Lossy compression*, also called Irreversible *compression,* is the mode of compression that gives rise to a loss of information while compressing the data. The compression algorithm eliminates data that it deems to be unnecessary. Lossy compression is mostly employed to compress multimedia data, especially in streaming applications. An ideal lossy compression algorithm will compress the data in such a way that the data degradation will be indiscernible to the user. Some of the widely used Lossy compression algorithms are *Discrete Cosine Transform (DCT)* [6]*, Wavelet compression* and *Transform Coding* [7].

Lossless compression, also known as *Reversible compression,* is the type of compression in which the encoded form of data can be reconstructed precisely to its prior true form. In other words, there is no loss of information during data compression. Lossless compression algorithms have been widely used for the zip file format. Some of the algorithms such as *LZW, Huffman Coding* and *Arithmetic coding* are well-known lossless compression algorithms.

1.2.3 Image Coding Systems

In applications such as remote sensing, lossy compression algorithms are employed to facilitate the efficient use of image storage and bandwidth for transmitting the images. Such algorithms tend to rely on Image coding techniques. An effective image coding technique is to decompose the image signal into a set of components that can be more efficiently quantized than the original signal [8]. JPEG1993 and JPEG2000 are some of the well-known image coding systems. Most Image coding systems follow a two or a three-stage pipeline, which is Decorrelation, Quantization and Entropy Coding.

1.2.1.1 Decorrelation

Decorrelation refers to a technique that is used to lessen the autocorrelation in a signal and cross-correlation among other signals, whilst keeping the features of the signal intact. Decorrelation techniques are predominantly linear.

1.2.1.2 Quantization

Quantization is a technique that is used to map an input signal, which is in the continuous value space, to a set of discrete values. Quantization plays a pivotal role in the Analog-to-Digital signal conversion process. Quantization induces a noninvertible mapping. Hence, it gives rise to information loss and *Quantization error*.

1.2.1.3 Entropy Coding

Entropy coding is a lossless data compression technique that is independent of the idiosyncrasies of a medium. A common practice in Entropy encoding involves assigning a unique prefix code to each unique symbol in the input code. These symbols are compressed by replacing them with variable-length prefix codes. *Huffman Coding* and *Arithmetic Coding* are the two most commonly used entropy coding techniques.

Report Outline

- We revisit how Image compression is modelled as an optimization problem in Chapter 2.
- In Chapter 3, we review the recent progress in using ML or DL techniques in Image Compression.
- We discuss in-depth some promising DeepLearning approaches for Image Compression in Chapter 4
- Chapter 5 covers the Performance metrics and datasets that are often used to benchmark Image Compression methods.
- In Chapter 6, we conduct experiments of the methods discussed in Chapter 4 and analyse the findings
- Chapter 7 covers the conclusion and possible future directions.

CHAPTER 2
PROBLEM FORMULATION

Given the ubiquitous use of multimedia, image compression has become a pivotal part of our lives. JPEG, BMP, PNG are some of the commonly used image compression standards. Lossy compression is employed to minimize the bitrate [9]. Non-Linear Transform Coding (NTC) methods have shown great performance gains and have procured results on par with the best linear transform coding approaches. The optimization problem for end-to-end image compression methods is constructed as follows. The objective function, also called the Rate-Distortion loss, has 2 components, R and D, with λ giving a tradeoff between R and D, D being the Distortion loss. R is the "Rate" term which consists of parameters of Analysis transform, Synthesis transform and the entropy model.

$$R = -log\ p_{\hat{y}}(\hat{y}; \boldsymbol{\theta}), \tag{2.1}$$

$$D = d(x, \hat{x}), \tag{2.2}$$

$$L(\boldsymbol{\theta}, \delta, \phi) = E_{x\sim\chi}[R + \lambda D], \tag{2.3}$$

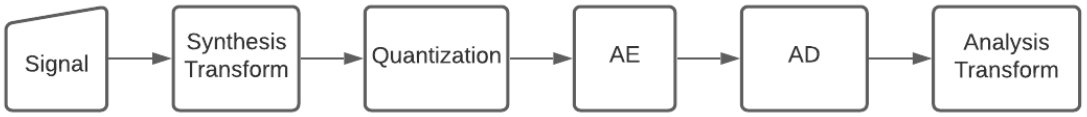

Fig 2.1: A standard End-to-End Image Compression system.

$x$is an image drawn from a distribution χ, $\hat{x}$ is the Synthesis Transform $S(\hat{y}; \delta)$, $\hat{y} = Q(y)$is the Quantization function, $\hat{x} = A(x; \varphi)$is the Analysis transform. $\boldsymbol{\theta}$is the parameters of the entropy model. The Analysis and Synthesis transform can also be a Convolutional Neural Network(in the case of NTC) with a series of linear Convolutions followed by non-linear activations. An End-to-End Image Compression system consists of Synthesis Transform, Quantization, Arithmetic Encoder, Arithmetic Decoder and an Analysis Transform as shown in Fig 2.1.

CHAPTER 3

REVIEW OF RECENT PROGRESS IN IMAGE COMPRESSION

Recent works have leveraged the high approximating power of Deep Neural Networks for Image Compression that has led to high-performance gains over the compression standards from JPEG to BPG. Initial works employed Deep Neural Networks for Transform Coding and explored various Neural Net architectures to propose an end-to-end trainable image compression method. Balle et al [10],[11],[12] presented Generalized Normalization Transform, a non-linear transform that Gaussianizes the input images, and an image compression method with Synthesis, Analysis Transforms and a Quantizer.

Some works focused on circumventing the non-differentiable nature of the Quantization function. Works such as [12],[13] replace the quantizer with additive uniform noise. Meanwhile, other works employed direct rounding during forward propagation and backpropagated the derivative of the identity function ($y = x$).

Some emphasis was on the design of Convolutional Neural Nets(CNN) that could effectively lower the spatial redundancy of the input image signal. State of the Art techniques such as Residual blocks [14], Attention modules[15] were explored to increase the approximating power of the transforms.

Entropy coding techniques compress the latent representations of the input signal that was subject to Transform Coding and Quantization. Prior works tend to employ an Arithmetic encoder to individually encode the elements and use element-wise entropy models to estimate the probability distribution of its representations[12],[14]. Subsequent works estimate entropy using Hyperpriors[13],[16], predictive models [17],[18],[19],[15],[20],[21] and tunable parametric models [21],[22],[23].

CHAPTER 4

PROMISING DEEP LEARNING METHODS FOR IMAGE COMPRESSION

The following table 4.1 summarizes some promising Deep Learning techniques that were used for Image Compression. All the given techniques use Deep Neural Networks for encoding and decoding the input signal. As the decoded signal is an approximation of the ground truth signal, it induces an error that makes the compression lossy

Table 4.1(a): A summary of promising Deep Learning techniques used in Image Compression

Paper Title	**Authors**	**Publishing Venue**	**Summary**	**Limitations or Future work**
Variable Rate Image Compression with Recurrent Neural Networks	George Toderici, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell & Rahul Sukthankar	ICLR'16	Earliest work to employ sequence models such as a Conv LSTM for Variable Bit End-to-end Image compression	Gives promising results for small images but fails to replicate that for high-resolution images.
Density modelling of images using a generalized normalization transformation	Johannes Ballé, Valero Laparra & Eero P. Simoncelli	ICLR'16	Proposes GDN, Generalized Normalization Transform, a non-linear transform that Gaussianizes the input images	Does not investigate the multistage optimization for deeply stacked models
Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations	Eirikur Agustsson, Fabian Mentzer, Michael Tschannen, Lukas Cavigelli, Radu Timofte, Luca Benini, Luc Van Gool	NeurIPS'17	Proposes a method to learn compressible representations which involves employing soft-to-hard annealing techniques for quantization and entropy	Extending the work to other applications that require a compressed representation
Lossy Image Compression with Compressive Autoencoder	Lucas Theis, Wenzhe Shi, Andrew Cunningham& Ferenc Huszar	ICLR'17	First work to employ Residual blocks in the Autoencoder for Image Compression to deal with non-differentiability involved in the Autoencoder training	Investigating more on optimizing Compressive Autoencoders on different metrics
End-to-End Optimized Image Compression	Johannes Ballé, Valero Laparra, Eero P. Simoncelli	ICLR'17	Proposes an end-to-end Image Compression method with a non-linear Analysis and Synthesis transform and a uniform quantizer	Insufficient empirical results to conclude about the effect of ReLU nonlinearities in the model
Variational Image Compression with a Scale Hyperprior	Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, Nick Johnston	ICLR'18	Earliest work to employ Hyperpriors for Image Compression and incorporate side-information as part of the end-to-end system	Surpasses SoTA on MS-SSIM but does not reach such levels on PSNR

Image-Dependent Local Entropy Models for Learned Image Compression	David Minnen, George Toderici, Saurabh Singh, Sung Jin Hwang, Michele Covell	ICIP'18	Introduces a method to enhance the DNNs-based image coders with data-dependent side-information that in turn results in high bit rate reductions	Does not properly leverage promising methods like forward-backward adaptation techniques
Deep Generative Models for Distribution-Preserving Lossy Compression	Michael Tschannen, Eirikur Agustsson, Mario Lucic	NeurIPS'18	Proposes the use of WGANs and Wasserstein Autoencoders to solve the constrained Rate-Distortion optimization problem	Focusing on extending the method to full-resolution images and other types of data
Deep Image Compression with Iterative Non-Uniform Quantization	Jianrui Cai, Lei Zhang	ICIP'18	Proposes a non-uniform quantization for enhancing the capacity and versatility of CNNs to compress certain image structures	N/A
Neural Image Compression via Non-Local Attention Optimization and Improved Context Modeling	Tong Chen, Haojie Liu, Zhan Ma, Qiu Shen, Xun Cao, and Yao Wang	TIP'21	Proposes a method that employs non-local networks, attention mechanism and a 3D-CNN-based context model to obtain SoTA compression performance on MS-SSIM and PSNR metrics.	Extending the method for end-to-end video compression system with more priors procured from spatio-temporal information and simplifying the models for Embedded systems.

Table 4.1(b)

Paper Title	**Public Availability of Code; Code Link**	**Datasets used and link to the dataset**	**Compared with what**
Variable Rate Image Compression with Recurrent Neural Networks	Yes; https://github.com/alexandru-dinu/cae	Uses a 32x32 benchmark dataset with 216M images scraped from the web. Link: Dataset not available	The proposed Conv LSTM Compressor is compared with header and headerless- JPEG, Conv Residual Encoder.
Density modelling of images using a generalized normalization transformation	Yes; https://github.com/jorge-pessoa/pytorch-gdn	Experiments are carried out on 16x16 images drawn from the Kodak set. Link: http://r0k.us/graphics/kodak/	GDN is compared with ICA and Random Gaussian model
Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations	No	Kodak dataset was used in the experiments. Link: http://r0k.us/graphics/kodak/	The proposed method is compared with JPEG, BPG, JPEG2000

Lossy Image Compression with Compressive Autoencoder	Yes; https://github.com/alexandru-dinu/cae	Used 434 high-resolution images scrapped from flickr.com. Link: Dataset not available	The method is compared with JPEG2000, JPEG and Toderici et al
End-to-End Optimized Image Compression	Yes; https://github.com/tensorflow/compression	ImageNet dataset was used. Link: https://www.image-net.org/	The method is compared with JPEG2000 and JPEG
Variational Image Compression with a Scale Hyperprior	Yes; https://github.com/GuoLusjtu/DVC	Kodak dataset was used in the experiments. Link: http://r0k.us/graphics/kodak/	The method is compared to BPG and JPEG
Image-Dependent Local Entropy Models for Learned Image Compression	No	Tecknik dataset was used. Link: https://testimages.org/sampling/	The method is compared to BPG, JPEG2000 and JPEG
Deep Generative Models for Distribution-Preserving Lossy Compression	Yes; https://github.com/mitscha/dplc	LSUN, CelebA datasets are used. Link to CelebA: https://www.kaggle.com/jessicali9530/celeba-dataset and LSUN:https://www.yf.io/p/lsun	It is compared with Compressive Autoencoders
Deep Image Compression with Iterative Non-Uniform Quantization	No	Kodak dataset was used in the experiments. Link: http://r0k.us/graphics/kodak/	It is compared with JPEG and JPEG2000
Neural Image Compression via Non-Local Attention Optimization and Improved Context Modeling	No	Kodak and CLIC dataset was used in the experiment. Link to Kodak: http://r0k.us/graphics/kodak/ and CLIC: http://compression.cc/	The method is compared to BPG, JPEG2000 and JPEG

Table 4.1(c)

Paper Title	**Scores**	**Metrics**
Variable Rate Image Compression with Recurrent Neural Networks	For a patch size of 8x8 and 32x32, the Conv LSTM Compressor outperforms its counterparts and gives an SSIM score of 0.69 and 0.77 respectively	SSIM is used as a measure to measure the compression performance

Density modelling of images using a generalized normalization transformation	GDN has the flexibility to capture a wide array of distributions whilst ICA and RG do not	No metrics are used
Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations	SHA method gives an accuracy of 92.1 and a Compression ratio of 20.15	PSNR, MS-SSIM, SSIM are used as metrics
Lossy Image Compression with Compressive Autoencoder	Provides better performance at lower and higher bitrates than its counterparts on all metrics such as PSNR, SSIM, MS-SSIM	PSNR, MS-SSIM, SSIM are used as metrics
End-to-End Optimized Image Compression	It gives an MS-SSIM of 0.9039 which is higher than its counterparts (JPEG- 0.8079, JPEG2000- 0.8860)	MS-SSIM IS used as the metric
Variational Image Compression with a Scale Hyperprior	Provides better performance at lower and higher bitrates than its counterparts on all metrics such as PSNR, MS-SSIM given the loss function is MS-SSIM	MS-SSIM and PSNR are used as metrics
Image-Dependent Local Entropy Models for Learned Image Compression	It gives a 17.8% rate reduction over SoTA ANN-based on an evaluation set and 70–98% reductions on images with low visual complexity.	MS-SSIM and PSNR are used as metrics
Deep Generative Models for Distribution-Preserving Lossy Compression	The proposed method obtains an rFID,sFID, MSE scores of 0.0277, 10.93 and 23.36 for the CelebA dataset and 0.0321, 27.52 and 60.97 for the LSUN dataset	rFID, sFID and MSE are used to gauge the performance
Deep Image Compression with Iterative Non-Uniform Quantization	The method gives a PSNR value of 27.19dB which is better than its counterparts	PSNR is used as the metric
Neural Image Compression via Non-Local Attention Optimization and Improved Context Modeling	It gives a PSNR of 32.54 and MS-SSIM of 0.9759 which is better than its counterparts (BPG: PSNR-31.97 MS-SSIM- 0.9581 and JPEG: PSNR- 25.17 MS-SSIM- 0.8629)	PSNR, MS-SSIM are used as the metrics

Prior SoTA performances in image compression were achieved by JPEG, JPEG2000. Given the resurgence of Neural Networks and the rise of hardware accelerators such as GPU, TPU, DNNs were employed for image compression as well. Existing Autoencoder-based approaches were not equipped to handle variable bit rate compression. Toderici et al[33] addressed this issue by employing a language model such as a Conv-LSTM as a Compressor. This was the first instance of a sequence model being used for image compression. Balle et al[10] proposed GDN, a non-linear transform that Gaussianizes the input signal. It is more robust than RG and ICA techniques in capturing different distributions.

Augustsson et al[34] proposed an annealing technique for quantization and entropy. It introduces a unified architecture that helps in the joint tuning of the transform, Quantization and Entropy parameters. This approach alleviates the non-differentiability problem of the Quantizer. Theis et al[36] propose an Autoencoder-based approach that circumvents the non-differentiability problem of the Quantizer. This is the first work to employ Residual blocks in Autoencoders.

Lee et al[19] introduced an end-to-end image compression pipeline that comprises a nonlinear Analysis transform, a uniform quantizer and a nonlinear Synthesis transform. This approach paved the way for many modern image codecs to incorporate a nonlinearity in the Analysis and Synthesis transform. Minnen et al[16] introduced a technique that uses local, image-dependent entropy models as side-information to augment CNNs in image codecs. It provides high bitrate reductions on the standard evaluation set.

Given the rapid ascent of GANs[37] to one of the most sought-after Generative modelling techniques, it resulted in GANs being employed in a flurry of applications such as Image denoising, Image Super-Resolution. Image Compression soon joined the long list of such applications. Tschannen et al[35] employed Wasserstein GAN and Auto Encoders for image compression. This was the first work to use GANs for low bitrate compression.

Cai et al[38] presented a non-uniform quantization scheme that improves the versatility and capacity of CNNs that are used in image codecs. The NLAIC(Non-Local Attention optimization and Improved Context modelling-based image compression) method introduced by Chen et al[21], incorporates non-local processing blocks into the architecture of VAE(Variational Autoencoder) that learns global and local dependencies among the pixels. The attention mechanism is also leveraged, with ReLU used for interposing non-linearity between the Convolutional layers. This method can be extended to video compression given the extraction of appropriate spatio-temporal information is used as priors or side-information.

4.1 A Closer Look into the Promising DL Techniques

Previous end-to-end image compression works often resort to the structure mentioned in Chapter 2 (See Fig 2.1). An input signal is transformed into its latent representation using *Analysis transform*. It is Quantized, encoded and decoded. It is then reconstructed using a *Synthesis Transform* to obtain the original signal, albeit having some reconstruction error.

The Hyperpior approach has gained popularity given its ease of integration into the end-to-end image compression system and efficient encoding and decoding. We discuss Minnen et al [25] and Balle et al[13] ‘s proposed approaches, in the following subsections, that solved significant bottleneck issues in the image compression pipeline and vastly improved rate-distortion performance.

Most promising works employ forward and backward adaptation techniques to increase the predictive power of the entropy model. This in turn results in better compression performance without exacerbating the distortion. Forward adaptation leverages side information and backward adaptation are usually context-adaptive models. In such models, encoding can be performed parallelly using a hardware accelerator like a GPU or TPU[25].

Meanwhile, Decoding allows only for serial processing. This slows down the entire pipeline of the end-to-end image compression system. Minnen et al [25]’s proposed techniques, Latent Residual Prediction(LRP) and Channel Conditioning (CC) minimize the serial processing time and improves rate-distortion performance.

Traditional compression approaches leverage *side-information* to improve the compression performance. The form of the side information is usually hand-crafted. Balle et al[13] incorporate the side-information in the end-to-end image compression pipeline in turn making it learnable and adaptive to the input signal. Balle et al were also the first work to leverage Hyperprior for Image Compression and set the path to many more promising works.

4.1.1 Latent Residual Prediction and Channel Conditioning Blocks

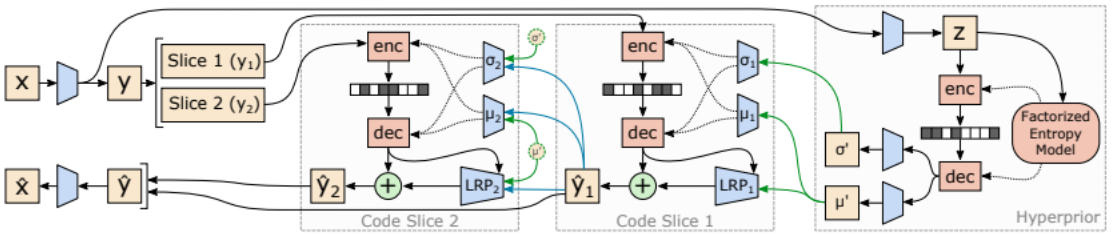

Fig 4.1: An end-to-end image compression architecture with Latent Residual Prediction and Channel Conditioning blocks

The figure depicts the compression architecture proposed by Minnen et al[25]. $x$is the input signal or image. The blue boxes denote a transform that consists of a series of Convolution layers. Dark- red denotes entropy coding(encoder and decoder), green represents arithmetic operations and light-red denotes the data tensor. It consists of 2 types of blocks, Channel Conditioning block and Hyperprior block. Transforms are applied to the input image, x, to obtain its latent representation, y. The tensor $y$has a shape of C x H x W(num_channels x height x width).

It is split along the channel dimensions into N slices(N=2 here) to obtain tensors $y_i, (1 <= i <= N)$ which will have a shape of C/N x H x W. The Hyperprior block produces the Hyperpriors, σ' , μ' ,that are used to condition the slices. Each slice is fed into a Channel Conditioning block where they are conditioned, quantized and encoded. The output of the decoder is fed into the Latent Residual Prediction block.

When the slices are reconstructed using a decoder, it inadvertently causes extra distortion and leads to a residual error which is denoted by $r = Q(y) - y$. Latent Residual Prediction(LRP) is responsible for predicting the residual based on the Hyperprior and the other reconstructed slices. The output of the Latent Residual Prediction block is added to the reconstructed slice (to get $\hat{y}_i$) that in turn reduces the distortion. The outputs of all the Channel Conditioning blocks ($\hat{y}_i$) are concatenated to obtain the final reconstructed image, $\hat{x}$.

This architecture improves rate-distortion performance and minimizes the need for serial processing. They report an average rate savings of 6.7% on the KODAK dataset and 11.4% on the Tecknick dataset.

4.1.2 Scale Hyperprior Method

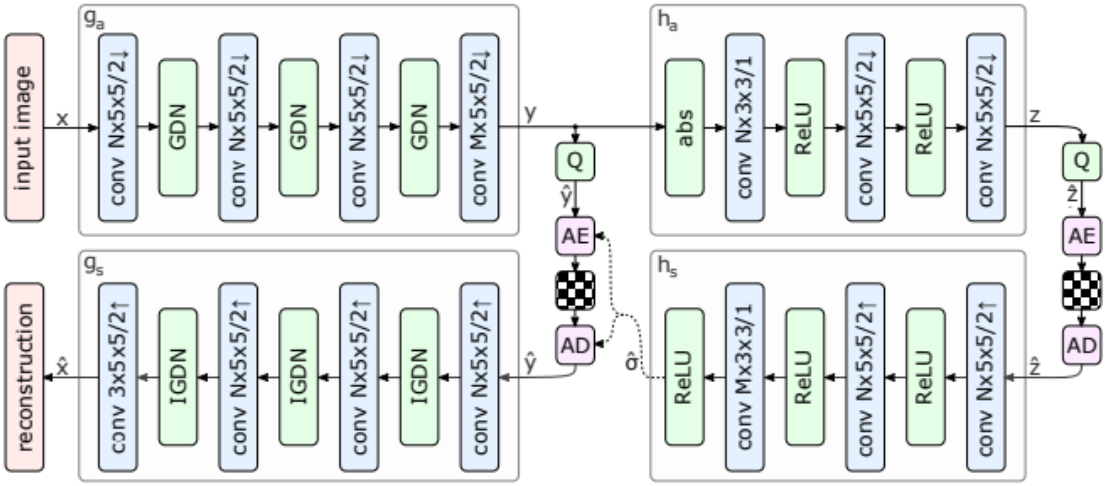

Fig 4.2: Architecture for the Scale Hyperprior Method

This method of using side information for image compression whilst employing Deep Neural Networks (DNNs) for Transform Coding was proposed by Balle et al [13]. There are 2 components to this architecture. Each component has an Autoencoder.

The right-hand side AutoEncoder gets the latent representation of the input image, y, as an input. It is subject to an Analysis Transform, h_a, Quantizer Q, Arithmetic Encoder and Decoder (AE-AD) and a Synthesis Transform h_s. The Synthesis and Analysis Transform differ from that of its left-hand counterpart for the type of activation function used in between Convolution layers. The right hand Auto Encoder produces $\hat{\sigma}$,which gives the correct probability estimates for properly reconstructing$\hat{y}$.

The left-hand side autoencoder is responsible for encoding the input image in the latent space, quantizing and reconstructing it. g_s, $g_a$correspond to the Synthesis and Analysis transforms. The input image $x$is transformed into its latent representation$y$ by g_a. y is fed as an input to the right-hand side Autoencoder to produce $\hat{\sigma}$that helps in proper reconstructing of $\hat{y}$. Meanwhile, $y$is quantized by the Quantizer Q to get $\hat{y}$ which is encoded and decoded with the help of Arithmetic Encoder and Decoder(AE-AD) and the side information, $\hat{\sigma}$. The decoded representation, $\hat{y}$is fed into the Synthesis Transform $g_s$to reconstruct the input image.

CHAPTER 5

EVALUATION

5.1 Metrics

The evaluation of Image Compression methods is done with the help of 2 types of Image Quality Assessment(IQA) techniques, Subjective and Objective. These methods give a score that reflects the effectiveness of the Compression algorithm. Subjective methods follow the recommendations provided by the ITU[27] and give a Mean Opinion Score (MOS) based on the individual scores of human judges that reflects the quality of the image.

Due to the onerous nature of the Subjective techniques, Objective techniques came into the limelight. They can be categorized into 3 types, Full Reference (FR), No Reference (NR) and Reduced Reference(RR) methods. FR methods compare the reconstructed image with the ground truth. NR methods do not involve the ground truth image. RR is a hybrid of NR and FR methods. The most frequently used Objective techniques are PSNR, MSE etc.

Among the long line of Human Visual System (HSV) IQA methods designed to bridge the gap between Subjective and Objective evaluation, Structural Similarity Index (SSIM), which was proposed Wang et al [28] to measure the perceptual quality of the reconstructed image, obtained some success. Despite the emergence of SSIM, no method has been fully able to replicate the Human Visual System precisely.

5.2 Datasets

End-to-end image compression does not require a labelled dataset as it poses a self-supervised learning problem. KODAK[29], Tecknick [30], CLIC[31], are some of the popular datasets used to train end-to-end image compression models. The Kodak dataset consists of 24 images with a resolution of 512x768. The dataset comprises a variety of images ranging from human faces to animals to natural sceneries.

Fig 5.1: Sample images from the KODAK dataset

The Tecknick dataset contains a SAMPLING test set that consists of 40 RGB reference images of 2400x2400 resolution. Additionally, it also has resized versions of the reference images to resolutions such as 1200x1200, 800x800.

Fig 5.2: Sample images from the CLIC 19 dataset

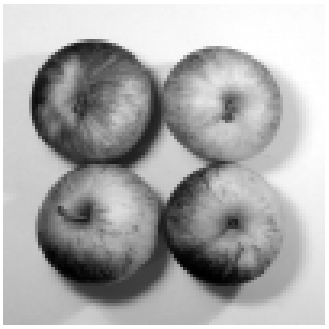

Fig 5.3: Sample images from the Tecknick dataset

The CLIC dataset was created as part of CVPR's Workshop on Challenge on Learned Image Compression(CLIC) that fosters research in the domain of learned image compression. The dataset comprises images taken from mobile as well as a professional camera. The average resolution of the images taken by the former is 1913x1361 and the latter is 1803x1175.

CHAPTER 6

EXPERIMENTS

The following experiments on Latent Residual Prediction and Channel Conditioning were performed by Minnen et al[25] and that of Scale Hyperprior were performed by Balle et al[13].

6.1 Latent Residual Prediction and Channel Conditioning:

This Chapter discusses the empirical results of a learned image codec that is incorporated with Latent Residual Prediction(LRP), Channel Conditioning(CC). Fig 6.1 shows that the RD measure of a codec with Channel Conditioning (10 splits)+Latent Residual Prediction averaged with respect to the Kodak dataset, outperforms the compression standards such as JPEG, JPEG2000, WebP and the learned codec baselines such as the context-adaptive entropy model proposed by Lee et al [19] and Autoregressive-Hierarchical priors proposed by Minnen et al[17].

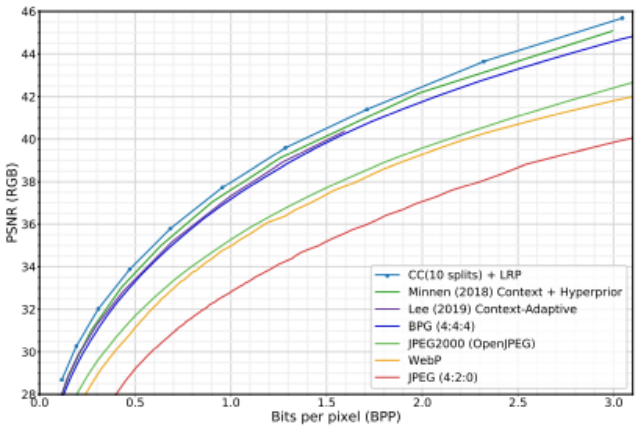

Fig 6.1: RD curve of codecs averaged for the Kodak dataset

Fig 6.2 depicts the relative rate savings on BPG averaged for the Kodak dataset. The Channel Conditioning (10 splits)+Latent Residual Prediction outperforms BPG by 10% at high bit rates. It outperforms the aforementioned learned codec baselines by approximately 16 % at high bit rates and 7% at low bit rates. This combination gives 14 % better average rates savings than BPG and 7% more than Lee et al[19].

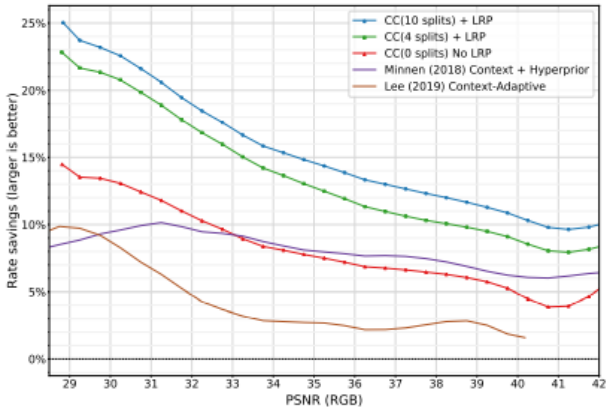

Fig 6.2: Relative rate savings on BPG averaged for Kodak dataset

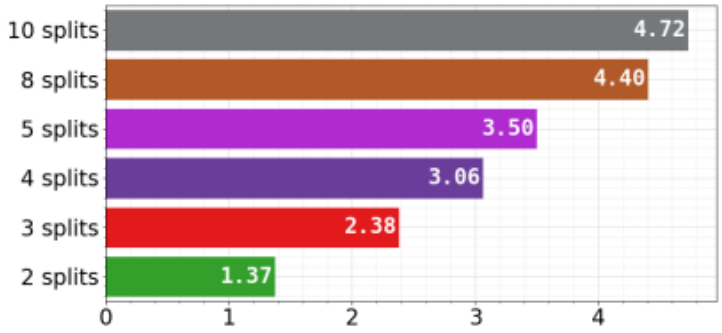

Fig 6.3: Rate Savings vs Baseline

Fig 6.3 portrays how the rate savings vary when the number of splits of Channel Conditioning is increased. This can be attributed to the fact that when there are many splits of the latent representation tensor, it eases modelling complex dependencies among the channels. However, it has a high computational overhead.

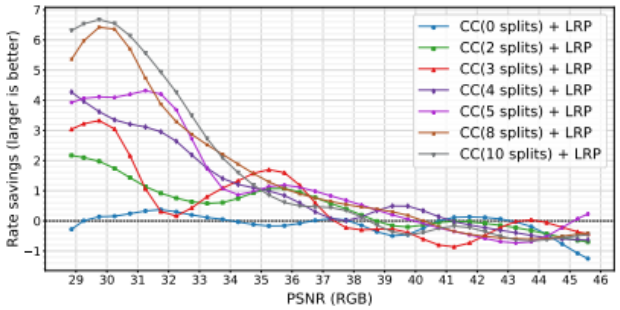

Fig 6.4: Study of the effect of LRP on rate savings on varying the number of CC splits.

Fig 6.4 studies the impact of Latent Residual Prediction on rate savings when the number of splits in Channel Conditioning is increased. Regardless of the number of splits, rate savings flag at high bitrates. Latent Residual Prediction without any splits is ineffective given the rate savings remain in the vicinity of 0. The biggest model(Channel Conditioning (10 splits)+Latent Residual Prediction) gives a rate savings of about 6% at low bitrates.

6.2 Scale Hyperprior

The proposed model is evaluated using an experiment with the Kodak dataset and the compression performance is reported. The performances are measured using the PSNR and MS-SSIM distortion measures. Fig 6.5 depicts the RD curves for all the models for the Kodak dataset. The RD curves are aggregated over multiple λvalues, which gives a trade-off between R and D.(See Chapter 2). There is a substantial deviation in the performance that depends on the distortion metric that is employed as a loss function. MS-SSIM and PSNR are the two choices for loss function in this experiment. When PSNR is used as the distortion measure, both the factorized prior and hyperprior models, when optimized for MS-SSIM, underperform compared to the compression baselines such as BPG. Meanwhile, both the models perform comparatively better than JPEG, when they are optimized for MSE. BPG outperforms Hyperprior and Factorized prior in either scenario.

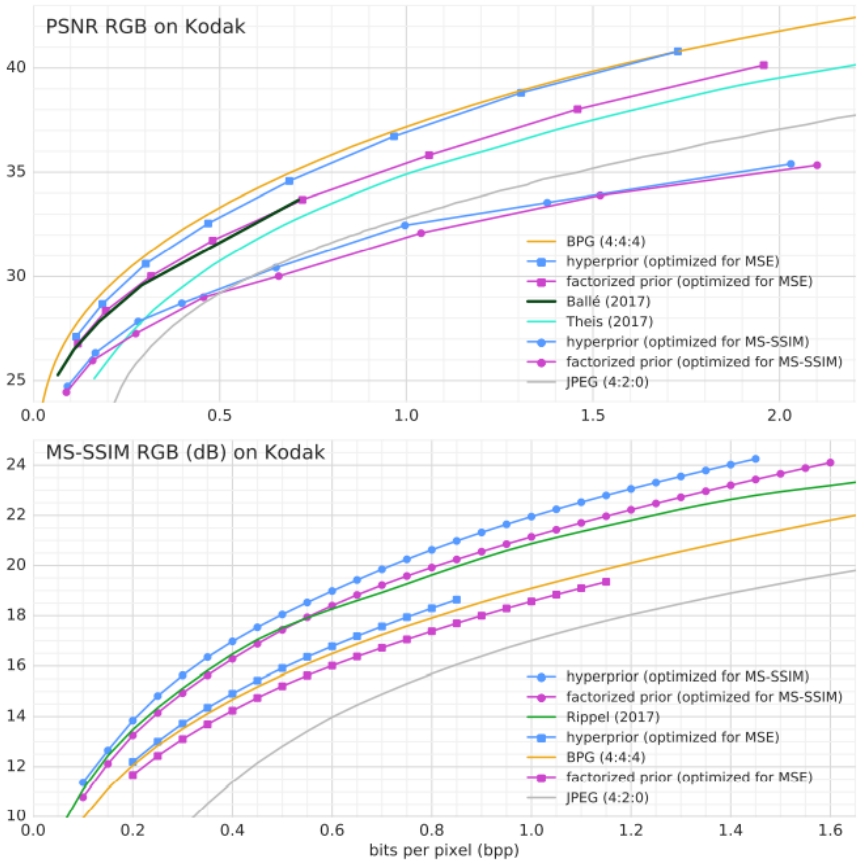

Fig 6.5: Rate-Distortion curves of models trained on the Kodak dataset

The Hyperprior and Factorized prior models, which are intrinsically DNN-based models, perform well when MS-SSIM is used as a distortion measure as the conventional compression standards were not optimized and tuned for MS-SSIM measure during their formative years. Both the models outperform the state of the art for compression performance for MS-SSIM (Rippel et al[32]).

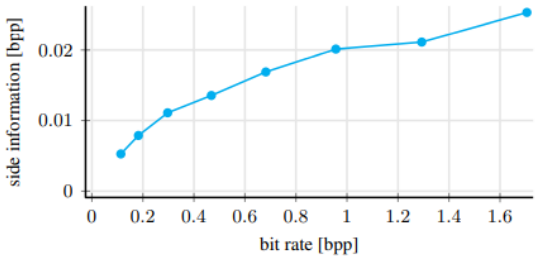

Fig 6.6: Side information vs bits per pixel

Figure 6.6 portrays the fraction of bit rate used by the hyperprior model as side information. It has a positive correlation with the bit rate but it is bounded by 0.1 bpp.

CHAPTER 7
CONCLUSION AND FUTURE WORK

We review the fundamentals of data compression. We see how Image compression is formulated as an optimization problem to facilitate the use of non-linear function approximators such as Deep Neural Networks in Image Compression. We discuss some promising Deep Learning approaches that have been employed in the domain of Image Compression.

We dive deep in particular about 2 works, Latent Residual Prediction-Channel Conditioning[25] and Scale Hyperprior[13]. The usage of Latent Residual Prediction-Channel Conditioning blocks in the end-to-end image compression architecture improves the rate-distortion performance and most importantly, minimizes the serial processing in decoders. Prior works such as Toderici et al[33](employs a Conv-LSTM compressor), Agustsson et al[34](uses a soft to hard annealing technique for Quantization) have been discussed in Table 4.1. The Latent Residual Prediction-Channel Conditioning blocks speed up the entire end-to-end image compression pipeline compared to the other discussed works.

Scale Hyperprior technique was the first instance of using Hyperprior and side-information for Image compression. It was the precursor to a long line of promising work[17][19][35] that have persistently pushed the frontiers of SoTA Rate-Distortion performance.

Meanwhile, we also revisit the evaluation metrics, datasets that have been leveraged to benchmark Deep Learning-based Image Compression approaches.

The advent of high-resolution video cameras and high processing power has made video compression of paramount importance. Future works will heavily focus on developing efficient techniques for video compression.

CHAPTER 8

REFERENCES

[1] McCarthy, John (1988). "Review of The Question of Artificial Intelligence". Annals of the History of Computing. 10 (3): 224–229.

[2] Pinkus, Allan (January 1999). "Approximation theory of the MLP model in neural networks". Acta Numerica. 8: 143–195. doi:10.1017/S0962492900002919.

[3] C. E. Shannon, "A mathematical theory of communication," in The Bell System Technical Journal, vol. 27, no. 3, pp. 379-423, July 1948, doi: 10.1002/j.1538-7305.1948.tb01338.x.

[4] Huffman, D. (1952). "A Method for the Construction of Minimum-Redundancy Codes" (PDF). Proceedings of the IRE. 40 (9): 1098–1101. doi:10.1109/JRPROC.1952.273898.

[5] Goodfellow, Ian, et al. Deep learning. Vol. 1. No. 2. Cambridge: MIT press, 2016.

[6] N. Ahmed, T. Natarajan, and K. R. Rao. 1974. Discrete Cosine Transform. IEEE Trans. Comput. 23, 1 (January 1974), 90–93. DOI:https://doi.org/10.1109/T-C.1974.223784

[7] J. Ballé et al., "Nonlinear Transform Coding," in IEEE Journal of Selected Topics in Signal Processing, vol. 15, no. 2, pp. 339-353, Feb. 2021, doi: 10.1109/JSTSP.2020.3034501.

[8] https://www.sciencedirect.com/book/9780080918549/handbook-of-visual-communications

[9] Hu, Yueyu, et al. "Learning end-to-end lossy image compression: A benchmark." IEEE Transactions on Pattern Analysis and Machine Intelligence (2021).

[10] J. Balle, V. Laparra, and E. P. Simoncelli, “Density modeling of ´ images using a generalized normalization transformation,” Proc. Int. Conf. Learn. Representations, 2016.

[11] J. Balle, V. Laparra, and E. P. Simoncelli, “End-to-end optimization ´ of nonlinear transform codes for perceptual quality,” in Proc. Picture Coding Symp., 2016.

[12] J. Balle, V. Laparra, and E. P. Simoncelli, “End-to-end optimized ´ image compression,” Proc. Int. Conf. Learn. Representations, 2017

[13] J. Balle, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, ´ "Variational image compression with a scale hyperprior," in Proc. Int. Conf. Learn. Representations, 2018.

[14] L. Theis, W. Shi, A. Cunningham, and F. Huszar, "Lossy image ´ compression with compressive autoencoders," Proc. Int. Conf. Learn. Representations, 2017.

[15] "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2020.

[16] D. Minnen, G. Toderici, S. Singh, S. J. Hwang, and M. Covell, "Image-dependent local entropy models for learned image compression," in Proc. Int. Conf. Image Process., 2018.

[17] D. Minnen, J. Balle, and G. D. Toderici, "Joint autoregressive and ´ hierarchical priors for learned image compression," in Proc. Adv. Neural Inform. Process. Syst., 2018.

[18] J. Klopp, Y.-C. F. Wang, S.-Y. Chien, and L.-G. Chen, "Learning a code-space predictor by exploiting intra-image-dependencies." in Proc. Brit. Mach. Vis. Conf., 2018.

[19] J. Lee, S. Cho, and S.-K. Beack, "Context adaptive entropy model for end-to-end optimized image compression," in Proc. Int. Conf. Learn. Representations, 2019.

[20] H. Ma, D. Liu, N. Yan, H. Li, and F. Wu, "End-to-end optimized versatile image compression with wavelet-like transform," IEEE Trans. Pattern Anal. Mach. Intell., 2020.

[21] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, "Neural image compression via non-local attention optimization and improved context modeling," IEEE Trans. on Image Process., 2021.

[22] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018.

[23] N. Johnston, D. Vincent, D. Minnen, M. Covell, S. Singh, T. Chinen, S. Jin Hwang, J. Shor, and G. Toderici, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2018

[24] A. van den Oord, N. Kalchbrenner, L. Espeholt, K. Kavukcuoglu, O. Vinyals, and A. Graves, “Conditional image generation with PixelCNN decoders,” in Advances in Neural Information Processing Systems 29, 2016.

[25] Minnen, David, and Saurabh Singh. "Channel-wise autoregressive entropy models for learned image compression." 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020.

[26] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell. Full resolution image compression with recurrent neural networks, 2016b. arXiv:1608.05148v1.

[27] BT, RECOMMENDATION ITU-R. "Methodology for the subjective assessment of the quality of television pictures." International Telecommunication Union (2002).

[28] Wang, Zhou, et al. "Image quality assessment: from error visibility to structural similarity." IEEE transactions on image processing 13.4 (2004): 600-612.

[29] E. Kodak, “Kodak lossless true color image suite (photocd pcd0992). [online]. available: http://r0k.us/graphics/kodak/.”

[30] N. Asuni and A. Giachetti, “TESTIMAGES: a large-scale archive for testing visual devices and basic image processing algorithms.” in Proc. Eur. Italian Chapter Conf., 2014.

[31] Challenge on learned image compression 2018. [Online]. Available: http://www.compression.cc/challenge/

[32] Rippel, Oren and Lubomir Bourdev (2017). “Real-Time Adaptive Image Compression”. In: Proc. of Machine Learning Research. Vol. 70, pp. 2922–2930.

[33] Toderici, George, et al. "Variable rate image compression with recurrent neural networks." *arXiv preprint arXiv:1511.06085* (2015).

[34] Agustsson, Eirikur, et al. "Soft-to-hard vector quantization for end-to-end learning compressible representations." *arXiv preprint arXiv:1704.00648* (2017).

[35] Tschannen, Michael, Eirikur Agustsson, and Mario Lucic. "Deep generative models for distribution-preserving lossy compression." *arXiv preprint arXiv:1805.11057* (2018).

[36] Theis, Lucas, et al. "Lossy image compression with compressive autoencoders." *arXiv preprint arXiv:1703.00395* (2017).

[37] Goodfellow, Ian, et al. "Generative adversarial nets." *Advances in neural information processing systems* 27 (2014).

[38] Cai, Jianrui, and Lei Zhang. "Deep image compression with iterative non-uniform quantization." *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2018.