

End-to-End Learned Image Compression: Approaches, Deployment, and Standardization

Jasser Kraiem
 Technical University of Munich
 Chair of Media Technology
 Arcisstr. 21, 80333 Munich, Germany
 jasser.kraiem@tum.de

Zongxie Chen
 Technical University of Munich
 Chair of Media Technology
 Arcisstr. 21, 80333 Munich, Germany
 zongxie.chen@tum.de

Abstract

Learned image compression (LIC) has emerged as a promising alternative to conventional image compression (CIC) methods, leveraging deep learning techniques to achieve superior compression efficiency and visual quality. This paper provides a comprehensive review of LIC, covering fundamental concepts, key innovations, and state-of-the-art models. To evaluate practical feasibility, we benchmark representative models on a Tesla T4 GPU and discuss deployment challenges. Additionally, we explore the JPEG AI standardization efforts, highlighting its architecture and features. We conclude that while standardization efforts like JPEG AI are promising, future research must balance performance with practical deployability, for instance through adaptive compression techniques.

Index Terms—Learned Image compression, rate-distortion optimization, variational autoencoder, JPEG AI.

1. Introduction

Visual information plays a fundamental role in modern applications, from multimedia communications and computer vision to emerging technologies such as digital twinning. Image compression is a critical enabling technology that sits at the intersection of information theory, signal processing, probability theory, and deep learning.

Compressing image data is typically formulated as a lossy compression problem. Lossy compression approximates continuous sources to achieve finite representation, which introduces information loss, in contrast to lossless compression that preserves all original information. *Quantization* is the fundamental operation in lossy compression that approximates the continuous source with a finite set of values. It maps a continuous input x to a discrete output

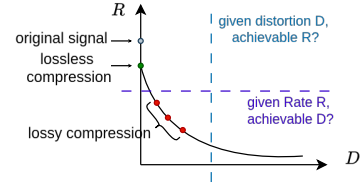


Figure 1: The Rate-Distortion function $R(D)$.

$\hat{x} = Q(x)$. Rate is the number of bits needed to represent the source, e.g. for an image:

$$R = \frac{\text{number of bits}}{\text{number of pixels}} \quad (1)$$

Distortion is the expected loss between original and reconstructed data, using some distortion metric d such as MSE or MS-SSIM:

$$D = E[d(X, \hat{X})] \quad (2)$$

In the context of lossy compression we always have a trade-off between rate and distortion, [10]. This trade-off can be visualized with the rate-distortion curve in Figure 1.

Images, especially high-resolution ones, contain vast amounts of data but also exhibit significant redundancy. Conventional image compression methods exploit this redundancy through *transform coding*, which converts the image into a domain with decorrelated coefficients, yielding a more compact representation. In JPEG, [20], an image x is transformed using the discrete cosine transform (DCT) to obtain the transformed coefficients y , [20]. The transformation can be represented as a linear operation using a transformation matrix A as shown in equation 3. After transformation, quantization is applied to the transformed

coefficients to obtain the quantized values $\hat{y}$.

$$\begin{aligned} y &= Ax \\ \hat{y} &= Q(y) \\ x' &= A^{-1}\hat{y} \end{aligned} \quad (3)$$

Since we obtain a discrete source after quantization, we can apply *entropy coding* to losslessly encode the quantized values to bitstreams. For decoding, we apply the inverse operations to reconstruct the image x' . The compression efficiency is then evaluated by measuring the rate and distortion according to equations 1 and 2.

CIC methods such as JPEG and JPEG2000 have dominated the field for decades. However, these handcrafted techniques are approaching theoretical and practical limits in terms of compression efficiency and visual quality, [23, 14]. Transform coding, quantization, and entropy coding blocks in CIC methods are designed based on expert knowledge and optimized separately, [14]. Additionally, CIC relies on linear transforms which implicitly assume that the image data follows a Gaussian distribution, [3]. In this context, LIC aims to address two issues: 1) Use non-linear transformations to better capture the complex distribution of natural images and 2) Optimize the compression pipeline in an end-to-end (E2E) manner.

This paper provides a comprehensive review of LIC. We begin by introducing the VAE-based baseline framework proposed by [4], which serves as the foundation for most modern LIC approaches. Building on this foundation, we survey key innovations in transforms, quantization, and entropy modeling, and discuss representative state-of-the-art models. Subsequently, we explore perceptual compression methods that extend beyond traditional rate-distortion optimization. We also address practical deployment challenges and conclude with an overview of the recent JPEG AI standardization efforts.

2. VAE: Baseline Framework for LIC

2.1 VAE-based Architecture

Most of the LIC methods build upon the VAE based architecture proposed by [4], which has a similar structure as JPEG. The encoder g_a and decoder g_s correspond to the analysis and synthesis transforms. [4] implements them with CNNs. For quantization, the researchers use simple rounding with equal bins of size 1. The assumption is that the transformations are powerful and flexible enough such that the combination of the transform and quantization can approximate a Vector-Quantization. The entropy coding block uses arithmetic coding with an entropy model P_q to estimate the PMF of the quantized latents. The overall architecture is shown in Fig. 2.

2.2 E2E training with rate-distortion loss

For training the model, a representative loss function L of the image compression task is needed. [4] use the loss function in equation 4 to jointly optimize rate and distortion:

$$L = R + \lambda D \quad (4)$$

For training, we can not evaluate the rate using 1, since counting bits is not differentiable. Instead, we estimate it with cross entropy which is valid as long as the entropy coding technique is efficient, for example using arithmetic coding. For the distortion, we can use any differentiable metric $d(x, x')$, for example MSE or MS-SSIM, [21]. We obtain this objective:

$$L(\theta, \phi, P_q) = -\mathbb{E}[\log_2 P_q] + \lambda \mathbb{E}[d(x, x')] \quad (5)$$

The expectations are approximated with averaging over the training dataset. End-to-end training requires all the operations to be differentiable, to update the weights of the NNs using backpropagation. However, quantization is not differentiable. From 5, it is also clear that the Probability mass function (PMF) of the quantized latents P_q required to estimate the rate is discrete and thus not differentiable as well. To address the non differentiability of the quantization, [4] proposed to replace quantization with additive uniform noise during training to approximate the quantization operation.

$$\tilde{y} = y + \epsilon, \quad \epsilon \sim U(-\frac{1}{2}, \frac{1}{2}) \quad (6)$$

Since $\tilde{y}$ is continuous, we can learn its probability density function (PDF) $p_{\tilde{y}}$ as a proxy for the PMF of the quantized latents P_q to estimate the rate:

$$L(\theta, \phi, \psi) = -\mathbb{E}[\log_2 p_{\tilde{y}}] + \lambda \mathbb{E}[d(x, x')] \quad (7)$$

where ψ are the parameters of the entropy model. The architecture during training is illustrated in Fig. 3.

2.3. Improvements and SOTA

2.3.1 Transformations

Several improvements have been proposed for the design of the encoder and decoder networks. [4] proposed to use *generalized divisive normalization (GDN)* layers instead of standard activation functions such as ReLU. [19] proposed to use residual blocks in the architecture. [8] were the first to introduce channel attention mechanisms. This allows the network to learn the feature maps based on their importance. Later work proposed to use transformer architectures, for example a *transformer-CNN Mixture (TCM)* was proposed in [17]. The design is motivated by the ability of transformers to capture global context, which is complementary to the local feature extraction of CNNs.

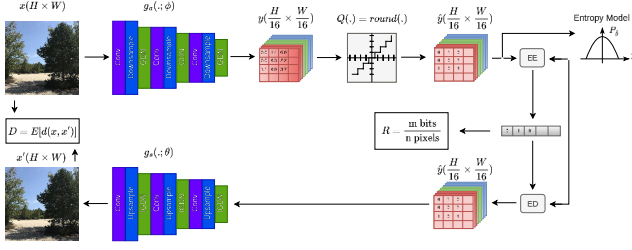


Figure 2: Inference workflow of the VAE architecture proposed in [4]. The encoder and decoder utilize CNNs. Latents are quantized via nearest-neighbor rounding and compressed using arithmetic coding based on the learned PMF.

2.3.2 Quantization

One of the first improvements concerning quantization was to use *vector quantization (VQ)* instead of simple rounding, [1]. Additionally, [1] propose to parametrize the quantization to train it jointly with the other NNs. Although additive uniform noise proposed by [4] helps with differentiability during training, it results in a rate mismatch between train and test. [1] proposed to use *soft-to-hard VQ* to address this issue. The training is divided into two stages. In the first, additive uniform noise is used for end-to-end training and in the second stage, rounding is used, but the encoder is fixed, and only the decoder is fine-tuned to learn to reconstruct from discrete values.

2.3.3 Entropy Models

[5] proposed to using a *hyperprior* to better estimate the PMF of the latents. The hyperprior is a second latent representation transmitted to the decoder as side information along the bitstream. This concept is used to capture the local scale (σ) of the latent distribution. To further improve entropy modeling, [19] introduced a *context model* (autoregressive prior). While the hyperprior captures local spatial variation, the context model captures spatial correlations between latents by using previously decoded neighbors to predict the distribution parameters of the current latent. The previous methods model the latents with a single Gaussian distribution. [8] proposed to use a *Gaussian Mixture Model (GMM)* to better model complex source distributions.

2.3.4 Representative SOTA Models

We consider two representative SOTA models for the VAE-based approach. The first model, Spatial to Channel Former (S2CFormer), achieves best performance by focusing on the transform design [7]. It argues that channel aggregation is more important than spatial aggregation for LIC. Thus, it simplifies spatial operations and uses a sophisticated FeedForward Network (FFN) to better aggregate channel information. The second model, Hierarchical Progressive Context Model (HPCM), focuses on improving

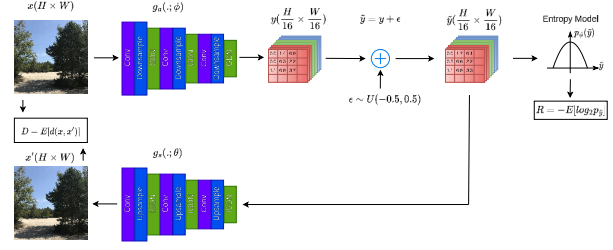


Figure 3: Training workflow of the VAE architecture [4]. To preserve differentiability, quantization is approximated by additive uniform noise. The rate is estimated via the cross-entropy of the continuous noisy latents, bypassing discrete entropy coding.

the entropy model, [16]. HPCM introduces a novel context model that splits the latent representation into multiple scales from coarse to fine details. Then, it uses cross-attention fusion to predict the distribution parameters of each scale based on the previous scales.

3. Beyond Rate-Distortion: Perceptual Models

Models discussed so far optimize for rate-distortion trade-off using traditional metrics such as MSE or MS-SSIM. Those metrics operate pixel-wise and do not correlate well with human perception. Moreover, in low bitrates, optimizing for those metrics often leads to blurry results. Perceptual image compression aims to solve this issue by adding a perception or realism metric to the objective

$$L = L_{Rate} + \lambda_D L_{Distortion} + \lambda_P L_{Perceptual} \quad (8)$$

One of the first models in this direction is High-Fidelity Generative Image Compression (HiFiC) [18], where they proposed to add a *conditional discriminator (D)* to the VAE-based architecture. D distinguishes between real and reconstructed images, pushing the generator/decoder (G) to produce more realistic images. The overall architecture is shown in Fig. 4. The model is trained in an alternating manner typical of GANs, where the Encoder-Decoder minimize the rate-distortion objective along with the output of D for the reconstructed images, while D is trained with the standard adversarial loss.

Perceptual Models generate high-quality images even at low bitrates, however they often hallucinate details that were not present in the original image, [14]. To address this issue, the discriminator in [18] is conditioned on the latent representation, which helps to preserve the content of the original image. Other models include information in the prior, such as a description of the image, to guide the compression process, [14].

4. Deployment Limitations

Although LIC achieved better RD performance compared to conventional methods, there are still several lim-

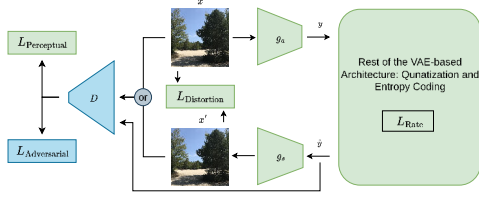


Figure 4: Overall Architecture (adapted from [18]). The green elements indicate the actual image compression pipeline, while D and its adversarial loss are in blue.

itations that hinder their deployment in real-world applications. LIC models are orders of magnitude more computationally expensive than conventional methods and typically require specialized hardware such as GPUs or TPUs to achieve acceptable inference times, [22]. Computational complexity of NNs is commonly measured by the number of multiply-accumulate operations (MACs). For hardware-independent comparison across different input resolutions, complexity is typically normalized and reported as kilo-MACs per pixel (kMACs/pixel). The number of operations directly affects the latency and inference time of the model, which is a critical factor for real-time applications. The size of models indicated by the number of parameters also affects the deployment feasibility.

Many models try to reduce computational complexity by using efficient layers, [15, 12]. Since image compression is an *asymmetric task*, with one-time encoding and multiple-time decoding, [22] proposed to use a shallow decoder with a more complex encoder to reduce the decoding complexity. A big limitation of LIC models is the need to store different models for different rate-distortion trade-offs. [9] proposed a model that can adapt to different rates by conditioning on the Lagrange multiplier λ from equation 4. Although context modeling is essential for entropy coding, some models use autoregressive models that are inherently sequential. To address this issue, [13] proposed a checkerboard context model that allows for parallel processing. Table 1 shows the efficiency benchmark results for various models.

Table 1: Efficiency benchmark results on 512×768 resolution. Experiments were performed using CompressAI [6] on an NVIDIA Tesla T4 GPU. For autoregressive models, entropy coding and spatial context modeling are executed sequentially on an Intel Xeon CPU. Enc/Dec KMACs/px denote the computational complexity of the neural transforms (analysis/synthesis and hyperprior). Encoding and decoding latency report the end-to-end system runtime measured using `compress()` and `decompress()`, including CPU-bound entropy coding.

Model	Enc KMACs/px	Dec KMACs/px	Enc. Latency (ms)	Dec. Latency (ms)	Params (M)
Factorized [4]	42.2	42.2	31.1	82.5	3.0
Hyperprior [5]	43.5	43.5	43.2	94.0	5.1
MBT (Joint) [19]	93.7	96.8	4356.6	8039.4	14.1
Cheng2020 [8]	166.3	255.3	3823.2	8915.1	11.8
ELIC [12]	113.2	113.2	310.2	182.1	15.5

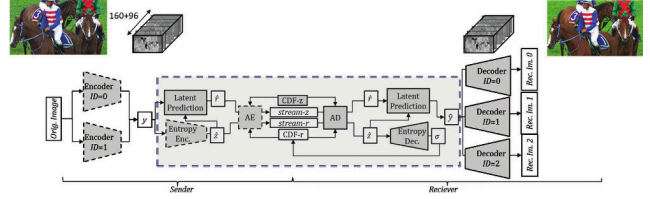


Figure 5: JPEG AI overall architecture (reproduced from [2]). The framework employs a multi-branch design to enable varying complexity-performance trade-offs.

5. JPEG AI Standardization

JPEG AI is the first image coding standard based on deep learning, [2]. The standardization Efforts started in around 2019 and officially became a standard in early 2025. The standard is based on the VAE framework proposed by [4]. *JPEG AI* aims to provide a practical LIC solution that is high performant While providing computational efficiency and practical deployability. The model employs a multi-branch architecture offering different complexity-performance trade-offs. *JPEG AI* uses two encoders: a lightweight version for mobile devices and a more complex high-performance variant. Three decoder configurations are available with increasing complexity: 8, 23, and 214 kMAC/pixel, defining low operating point (LOP) and high operating point (HOP) modes. The LOP encoder/decoder uses simple convolutions with residual connections, while HOP incorporates attention mechanisms [11]. For Quantization, *JPEG AI* uses the additive uniform noise during training and rounding during inference, as proposed by [4]. For entropy coding, it uses a learned entropy model similar to [5], with context modeling based on generalized version of the checkerboard context model [13] called Multi-stage Context Model (MCM). Similar to [9], *JPEG AI* supports variable rate compression. However, the architecture is implemented twice, because it uses YUV 4:2:0 color space to allow for more compression. Thus, Y channel and UV channels are handled separately with different models.

6. Conclusion and Future Work

The paper reviewed the rapid evolution of learned image compression, starting from the VAE-based baseline framework to recent SOTA models. We discussed key innovations in transformations, quantization, and entropy modeling, along with the perceptual compression approach. We also addressed practical deployment challenges and provided an overview of the *JPEG AI* standardization efforts. Future research directions include exploring more efficient architectures, better perceptual models and metrics, and adaptive compression techniques that can dynamically adjust to different content and application scenarios to balance performance and applicability of LIC.

References

- [1] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool. Soft-to-hard vector quantization for end-to-end learning compressible representations. *Advances in neural information processing systems*, 30, 2017.
- [2] E. Alshina, J. Ascenso, and T. Ebrahimi. Jpeg ai: The first international standard for image coding based on an end-to-end learning-based approach. *IEEE MultiMedia*, 31(4):60–69, 2024.
- [3] J. Ballé, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici. Nonlinear transform coding. *IEEE Journal of Selected Topics in Signal Processing*, 15(2):339–353, 2020.
- [4] J. Ballé, V. Laparra, and E. P. Simoncelli. End-to-end optimized image compression. In *International Conference on Learning Representations (ICLR)*, 2017.
- [5] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston. Variational image compression with a scale hyperprior. In *International Conference on Learning Representations (ICLR)*, 2018.
- [6] J. Bégin, F. Racapé, S. Feltman, and A. Pushparaja. Compressai: A pytorch library and evaluation platform for end-to-end compression research. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3363–3367. IEEE, 2021.
- [7] Y. Chen, Q. Li, B. He, D. Feng, R. Wu, Q. Wang, L. Song, G. Lu, and W. Zhang. S2cformer: Reorienting learned image compression from spatial interaction to channel aggregation. *arXiv e-prints*, pages arXiv–2502, 2025.
- [8] 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*, pages 7939–7948, 2020.
- [9] 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*, pages 3146–3154, 2019.
- [10] T. M. Cover. *Elements of information theory*. John Wiley & Sons, 1999.
- [11] S. Esenlik, Y. Wu, Z. Zhang, Y.-K. Wang, K. Zhang, L. Zhang, J. Ascenso, and S. Liu. An overview of the jpeg ai learning-based image coding standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.
- [12] 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*, pages 5718–5727, 2022.
- [13] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin. Checkerboard context model for efficient learned image compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14771–14780, 2021.
- [14] C.-H. Huang and J.-L. Wu. Unveiling the future of human and machine coding: A survey of end-to-end learned image compression. *Entropy*, 26(5):357, 2024.
- [15] N. Johnston, E. Eban, A. Gordon, and J. Ballé. Computationally efficient neural image compression. *arXiv preprint arXiv:1912.08771*, 2019.
- [16] Y. Li, H. Zhang, L. Li, and D. Liu. Learned image compression with hierarchical progressive context modeling. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 18834–18843, 2025.
- [17] J. Liu, H. Sun, and J. Katto. Learned image compression with mixed transformer-cnn architectures. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14388–14397, 2023.
- [18] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson. High-fidelity generative image compression. *Advances in neural information processing systems*, 33:11913–11924, 2020.
- [19] D. Minnen, J. Ballé, and G. D. Toderici. Joint autoregressive and hierarchical priors for learned image compression. *Advances in neural information processing systems*, 31, 2018.
- [20] G. K. Wallace. The jpeg still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.
- [21] Z. Wang, E. P. Simoncelli, and A. C. Bovik. Multiscale structural similarity for image quality assessment. In *The thirty-seventh asilomar conference on signals, systems & computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.
- [22] Y. Yang and S. Mandt. Computationally-efficient neural image compression with shallow decoders. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 530–540, 2023.
- [23] Y. Yang, S. Mandt, and L. Theis. An introduction to neural data compression. *Foundations and Trends in Computer Graphics and Vision*, 15(2):113–200, 2023.