LOW COMPLEXITY LEARNED IMAGE CODING USING HIERARCHICAL FEATURE TRANSFORMS (LOC-LIC)

Ayman A. Ameen[⋆]*, Thomas Richter*[‡]*, and André Kaup*[⋆]

[⋆] Multimedia Communications and Signal Processing,
Friedrich-Alexander University Erlangen-Nürnberg, Germany
[‡] Fraunhofer Institute for Integrated Circuits IIS, Erlangen, Germany

ABSTRACT

Learned image compression has demonstrated superior rate-distortion performance compared to traditional standards for over the past years. However, despite these advancements, legacy formats like JPEG remain dominant. The primary barrier to adoption is not quality, but computational complexity. Real-time applications demand low decoding latencies for smooth user experiences, whereas current learned models often require significantly longer processing times, rendering them impractical for standard hardware. We identify that the majority of this computational burden lies in the initial high-resolution convolutional layers that transform pixels into feature maps. To address this, we propose an innovative hierarchical feature extraction transform. By utilizing fewer channels for high spatial resolution inputs and increasing channel depth only as spatial dimensions are reduced in the latent space, we significantly cut computational load without sacrificing bit rate reduction efficiency. This strategy reduces forward pass complexity from 1256 kMAC/Pixel to just 270 kMAC/Pixel. This architectural shift offers an immediate solution for deploying efficient learned compression on existing devices without relying on future hardware acceleration.

1. INTRODUCTION

It has been almost a decade since the introduction of end-to-end learned image compression [1]. Since then, the field has witnessed substantial improvements in coding efficiency, consistently outperforming traditional methods [2]. Yet, despite being over 25 years old, JPEG, and its successors like JPEG 2000, remains the dominant standard in the industry. This raises a critical question: if learned image compression offers significantly better improvement in visual quality and file size, why has there been almost no widespread adaptation?

Acknowledgments The authors gratefully acknowledge the scientific support and HPC resources provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project b290dc. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) - 440719683.

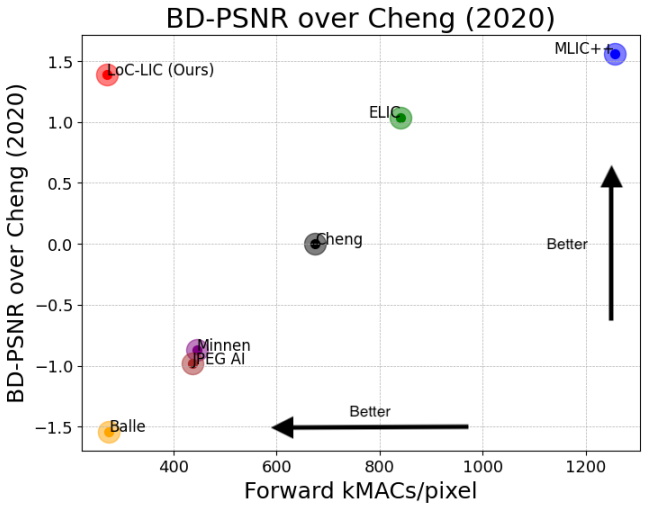


Fig. 1: The compression efficiency vs complexity of different learned image compression models. The complexity is measured in terms of (kMAC/Pixel) using BD-PSNR.

The answer is simple: complexity. While the rate-distortion performance of learned models is impressive, the computational cost required to achieve it is often prohibitive. In practical applications, user experience is paramount; for example, when scrolling through a gallery app, an image must be decoded and displayed in less than a second. If a model takes seconds or even minutes to open an image, it is unusable regardless of the compression quality.

Furthermore, we cannot simply rely on the potential arrival of specialized hardware accelerators. To ensure learned compression is viable today, even on older devices, we must address the architecture itself. We must ask: where exactly does the complexity lie?

In a standard Convolutional Neural Network (CNN) based codec, the architecture typically consists of an encoder, quantization, entropy coding, and a decoder. An analysis of the computational flow reveals that the complexity is not evenly distributed. The entropy coding and operations within the latent space (the "hybrid" encoder/decoder stages) are relatively efficient because they operate on compressed, low-resolution

data. The true bottleneck lies in the first few layers of the encoder and the final layers of the decoder. These layers are responsible for converting high-resolution raw images into sophisticated feature maps. Standard architectures often maintain a fixed number of channels throughout these layers, resulting in massive computational overhead at high spatial resolutions.

To solve this, we propose a hierarchical feature extraction approach designed to optimize efficiency where it matters most. By restructuring the network to use fewer channels for high-resolution input layers and reserving higher channel counts for the lower-resolution latent space, we drastically reduce the operations required. Our contributions are as follows:

- **Efficient Hierarchical Architecture:** We introduce a novel hierarchical feature extraction method that maps images from the pixel domain to the latent domain. This approach assigns fewer feature maps to larger spatial sizes and deeper feature representations to smaller sizes.
- **Drastic Complexity Reduction:** By targeting the heavy initial layers, we reduce the forward pass complexity from 1256 kMAC/Pixel to only 270 kMAC/Pixel, making the model viable for devices without specialized hardware accelerators.
- **Competitive Performance:** We utilize a hyper-autoencoder with a multi-reference entropy model, ensuring that despite the reduction in complexity, the model maintains state-of-the-art compression performance. The model is trained on a large dataset spanning a significant portion of the image space manifold to ensure generalization.
- **Open Source Release:** We release our model with open source weights, enabling widespread adoption and reproducibility. Future quantized versions and practical applications for desktop and mobile devices are planned, making learned image compression accessible for real-world deployment across diverse hardware platforms. (available at: loc-lic Webpage)

2. RELATED WORKS

Traditional image compression algorithms, such as JPEG 2000, often lack the flexibility of non-linear mapping to input data through a learning process. To address these limitations, a novel learned image compression method using autoencoders has been developed. This approach involves training an autoencoder on large datasets of images or videos [2, 3, 4, 5].

The key advantage of the learned image codec lies in the autoencoder's capacity to map images into a low-entropy, high-dimensional latent space, and subsequently reconstruct them back into the image space in a lossy manner [1]. The

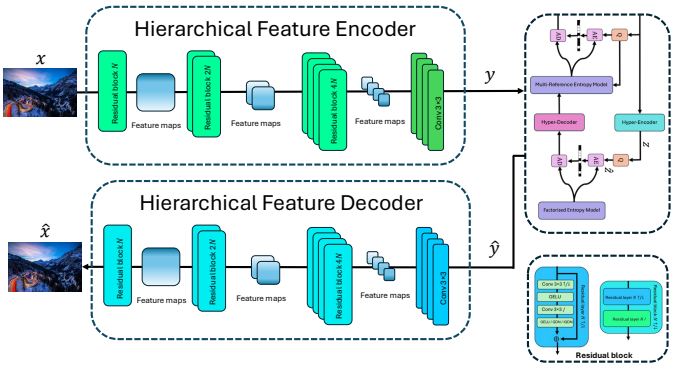


Fig. 2: Overall view of the proposed architecture with hierarchical feature encoder and decoder.

learned image codec network architecture is typically composed of analysis and synthesis transform functions, which can be implemented using pure convolutional layers with a fixed number of channels N [6]. Some architectures incorporate residual connections with convolutional layers as the base model [7], while others utilize transformer-based layers [8] or combine attention mechanisms with convolutional layers for enhanced performance [9]. The goal is for the reconstructed image $\hat{x}$ to closely resemble the original image x while minimizing the bit-rate R used for the latent representation. Generally, higher entropy results in lower distortion and vice versa. Therefore, optimization aims to balance distortion $D(x, \hat{x})$ with entropy measured in bitrate R (bits per pixel). A Lagrangian multiplier is employed to manage this trade-off between distortion and target bit-rate [9]. Quantization plays a critical role in bitrate reduction but introduces non-differentiability, hindering its direct integration with gradient-based optimization. To address this, continuous relaxations of the quantization operator, such as the straight-through estimator (STE), are widely adopted to enable differentiable approximations during training. An alternative strategy involves replacing deterministic quantization with stochastic rounding, which introduces non-biased gradients and has proven effective in recent compression frameworks [10, 1]. Both approaches circumvent the discontinuity in backpropagation while preserving quantization benefits. Stochastic rounding can be easily implemented by adding uniform noise to the unquantized values. A hybrid approach that combines both straight-through estimation and stochastic rounding also exists [2]. Utilizing context information allows for more efficient data compression by reducing the bit-rate necessary for encoding. Context-based entropy models exploit surrounding or neighboring information to better predict and compress the current data. This strategy is particularly important in neural image compression, as it enables accurate bit-rate estimation while minimizing redundancy.

To enhance compression efficiency, various context-based entropy models have been proposed. An autoregressive model

was introduced to condition each pixel on previously decoded pixels for more effective context modeling [10]. Another approach is the checkerboard convolution, which divides the latent representation into anchor and non-anchor parts, using the anchor part to extract context for the non-anchor part [9]. Furthermore, channel-wise context models [11], and channel-wise models with unevenly grouped contexts [4], have been developed to exploit redundancy between channels. Recently, an attention-based architecture has been proposed to capture a diverse range of correlations within the latent representation [3, 2].

Another promising approach for learned image compression involves using an overfitted neural network to represent image data as a continuous neural function instead of discrete pixel values. This neural function can be evaluated to reconstruct the RGB values of image pixels. Various efforts have been made to represent entire datasets, such as MNIST, using neural functions for resolution-agnostic representations [12]. A significant advantage of modeling images as neural functions is their resolution agnosticism: images are represented continuously and can be evaluated at any desired resolution. This approach assumes that image signals are inherently continuous.

Another approach for learned image compression involves using an overfitted neural network to represent image data as a continuous neural function instead of discrete pixel values. This neural function can be evaluated to reconstruct the RGB values of image pixels. COOL-CHIC [13] introduced an advanced overfitted learned image codec with reduced decoding complexity and improved compression efficiency. Recently, CLRIC [14] introduced a hybrid approach that utilizes an overfitted learnable function to compress the latent representation from image autoencoders, showing very promising results.

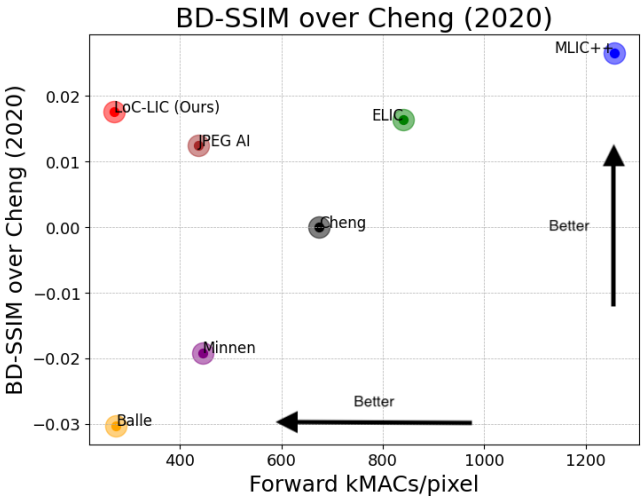


Fig. 3: The compression efficiency vs complexity of different learned image compression models. The complexity is measured in terms of (kMAC/Pixel) using BD-SSIM.

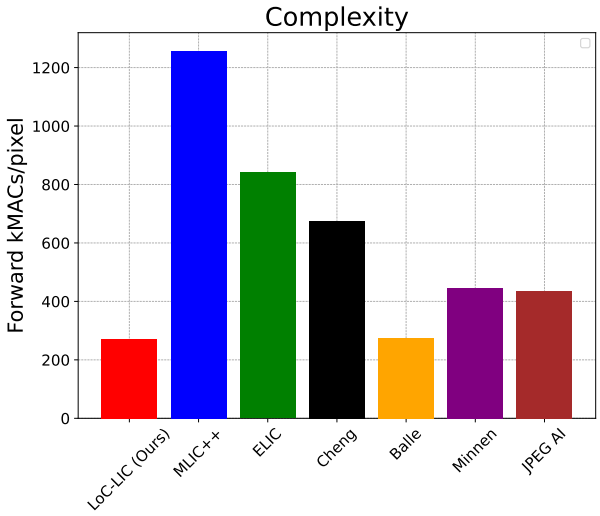


Fig. 4: Comparison of forward complexity between our approach and various learned image compression models

3. METHOD

Recent advancements in learned image compression have successfully minimized the rate-distortion Lagrangian $\mathcal{L} = \lambda D + R$. However, the practical deployment of these models is often hindered by the computational cost of the analysis and synthesis transforms. We formulate the deployment challenge as a constrained optimization problem where we seek to minimize $\mathcal{L}$ subject to a complexity budget $\mathcal{K}$:

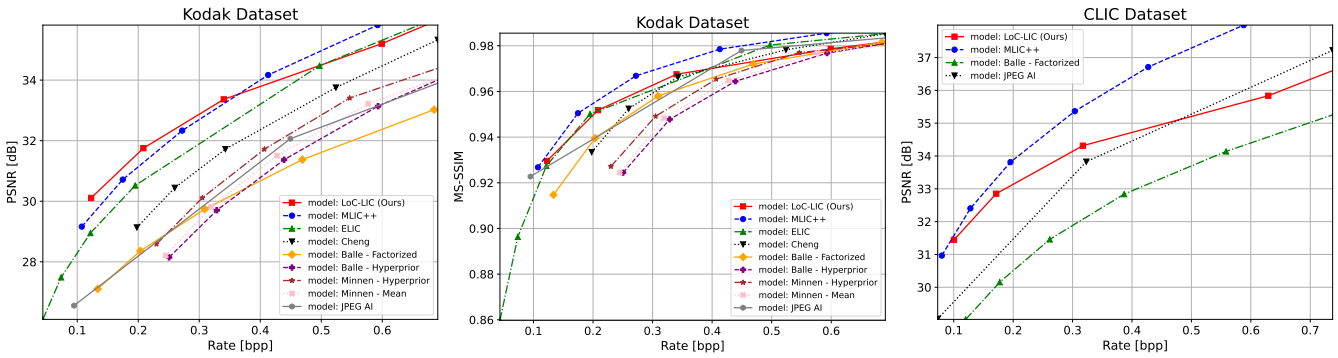
$$\min_{\theta,\phi} \mathbb{E}_{x\sim p_x}\left[-\log_2 p_{\hat{y}}(\hat{y}) + \lambda d(x,\hat{x})\right] \quad \text{s.t.} \quad \mathcal{C}(g_{a_{hf}}, g_{s_{hf}}) \leq \mathcal{K} \tag{1}$$

where $\mathcal{C}(\cdot)$ denotes the operational complexity (e.g., multiply-accumulate operations per pixel). Standard architectures often incur high $\mathcal{C}$ by maintaining high-dimensional feature maps throughout the network. In this paper, we propose a hierarchical feature extraction model designed to satisfy strict complexity constraints. While hierarchical representations have been utilized in generative synthesis and segmentation [15, 16], our approach optimizes the trade-off between the channel dimension C and spatial resolution $H \times W$ to minimize MACs without sacrificing the expressiveness of the latent space.

3.1. Architecture Overview

The proposed architecture, illustrated in Figure 2, aligns with the variational autoencoder framework used in mainstream learned image codecs [2, 9]. We define the input image space as $\mathcal{X} \subset \mathbb{R}^{3\times H\times W}$ and the latent space as $\mathcal{Y}$. The system comprises a Hierarchical Feature Encoder acting as a non-linear analysis transform $g_{a_{hf}}(\cdot;\theta)\colon \mathcal{X} \to \mathcal{Y}$, parameterized by θ.

To enable entropy coding, the continuous latent vector y is discretized via a quantization function $Q\colon \mathbb{R} \to \mathbb{Z}$. To



(a) (b) (c)

Fig. 5: Assessments and comparisons of image compression models using different metrics and datasets. (a) PSNR scores on the Kodak dataset, (b) MS-SSIM scores on the Kodak dataset, and (c) PSNR scores on the CLIC Professional Valid 2020 dataset.

Fig. 6: Our novel approach performance compared to MLIC++ and LIC-TCM on image num. 3 from the CLIC Professional Valid 2020 dataset.

allow for end-to-end differentiability during training, we approximate quantization by adding uniform noise $\mathcal{U}(-\frac{1}{2}, \frac{1}{2})$ or using straight-through estimation. The entropy model utilizes a hyper-prior structure derived from MLIC++ [2], estimating the distribution $p_{\hat{y}}(\hat{y}|\hat{z})$ where $\hat{z}$ is side information. This allows for the estimation of the bit-rate $R(\hat{y}) = -\sum_i \log_2 p_{\hat{y}_i}(\hat{y}_i|\hat{z})$.

The reconstruction is generated by the synthesis transform $g_{s_{hf}}(\cdot;\phi)\colon \mathcal{Y} \to \mathcal{X}$, parameterized by ϕ. The complete forward pass is mathematically formulated as:

$$y = g_{a_{hf}}(x;\theta), \quad \hat{y} = Q(y), \quad \hat{x} = g_{s_{hf}}(\hat{y};\phi) \tag{2}$$

where $x \in \mathcal{X}$ is the input image, y is the unquantized latent representation, and $\hat{x}$ denotes the reconstructed image. The parameters θ and ϕ are optimized jointly to minimize the rate-distortion objective.

3.2. Hierarchical Feature Transform

The hierarchical feature architecture [17, 18] is defined as a composition of layer-wise transformations that map the input x of dimensions $H \times W$ into a sequence of feature tensors with progressively increasing channel depth and decreasing spatial resolution. Let F_i denote the feature representation at layer i, where $F_0 = x$. We define the transformation function for the i-th layer as $\mathcal{T}_i\colon \mathbb{R}^{C_i \times H_i \times W_i} \to \mathbb{R}^{C_{i+1} \times H_{i+1} \times W_{i+1}}$.

The encoder $g_{a_{hf}}$ is constructed such that for the initial layer, the mapping produces basic features with N channels while applying a spatial stride of 2. For subsequent layers $i > 0$, the transformation adheres to an inverse proportionality between spatial resolution and channel capacity. Specifically, the mapping $\mathcal{T}_i$ enforces:

$$C_{i+1} = 2C_i, \quad H_{i+1} = \frac{H_i}{2}, \quad W_{i+1} = \frac{W_i}{2} \tag{3}$$

Consequently, the output of layer $i+1$ is a function of the input from layer i, expressed as:

$$F_{i+1} = \mathcal{T}_i(F_i) \quad \text{s.t.} \quad F_{i+1} \in \mathbb{R}^{2C_i \times \frac{H_i}{2} \times \frac{W_i}{2}} \tag{4}$$

This design explicitly constrains the computational complexity. Assuming a convolutional operation at layer i with kernel size $K \times K$, the computational cost Ω_i in MACs is proportional to:

$$\Omega_i \propto H_{i+1} W_{i+1} \cdot C_i \cdot C_{i+1} \cdot K^2 \tag{5}$$

By reducing the spatial dimensions H_i, W_i geometrically while increasing channels C_i arithmetically, we ensure that the total complexity $\sum_i \Omega_i$ remains bounded, effectively shifting the computational load from high-resolution spatial

Fig. 7: Comparison between our approach and different models on image num. 7 from the Kodak dataset.

processing to high-dimensional semantic feature processing.

4. EXPERIMENTS

We train our models on 256×256 randomly cropped images from a custom dataset containing around 10^6. Our custom dataset images is selected from ImageNet [19] COCO 2017 [20] Vimeo90K [21], and DIV2K [22]. Our objective function consists of two terms. The first one is the mean square error between the original image and the model's output. The second term is the bitrate with a Lagrange multiplier to control the trade-off between the two terms and achieve the target bitrate.

To assess and compare the performance and generalization capability of our model, we conducted validation experiments on two datasets and evaluated its performance against various models. The first dataset utilized is the Kodak dataset, a widely adopted benchmark for validating image compression models comprising 24 images. Additionally, we selected the CLIC Professional Valid 2020 dataset, which contains 41 high-resolution images, making it well-suited for evaluating compression in the current era of digital high-resolution imagery. We compared our approach against several learned image compression models, including MLIC++ [2], LIC-TCM [7], ELIC [4], JPEG AI [23] (JPEG-AI-high variant) and two variations of Balle's model, Factorized and Hyperprior [1], two variations of Minnen's model, Mean, and Hyperprior [10], as well as Cheng's Anchor model [9], were included in the comparison.

Quantitative analysis We evaluated the complexity of our model in terms of forward operations measured as kMAC/Pixel, comparing this against other models to assess its efficiency. Using Cheng [9] as a baseline, illustrated in Figure 1, our model exhibited a significantly reduced complexity of approximately 270 kMAC/Pixel while maintaining superior performance over the Cheng model, which has a complexity of 933 kMAC/Pixel. Moreover, our model outperformed those by Balle and Minnen at both lower and higher bit-rates; these models utilize two different approaches corresponding to varying levels of complexity. An average model complexity was considered for comparison purposes. On the other hand, MLIC++ proved more efficient with a complexity of around 1256 kMAC/Pixel that we could not surpass. Additionally, our model achieves competitive results in terms of the Structural Similarity Index Measure (SSIM) metric, indicating higher values that align with reduced complexity, as shown in Figure 3. We conducted a comprehensive evaluation of our approach, specifically focusing on a forward path that is responsible for the image encoding and decoding phases. Our analysis, depicted in Figure 4, compares our method to other advanced learned image compression models. Notably, our model demonstrated the lowest complexity, outperforming leading models such as MILC++.

We plotted rate-distortion curves for both Peak Signal-to-Noise Ratio (PSNR) and Multi-Scale Structural Similarity Index Measure (MS-SSIM) using the Kodak and CLIC Professional Validation 2020 datasets, as illustrated in Figure 5. Our model exhibits behavior comparable to that of high-complexity models such as MLIC++ [2] and ELIC, while maintaining significantly lower complexity. In terms of PSNR, our model's performance is shown in Figure 5(a), and for MS-SSIM, the performance is depicted in Figure 5(b).

Qualitative analysis In our study, we evaluated the visual performance of our model using two distinct datasets, Kodak and CLIC, as illustrated in Figures 6 and 7. Our findings indicate that our model achieves a performance comparable to the state-of-the-art learned image compression model, like MLIC++ with a marginal increase in the bit rate while maintaining reduced complexity and preserves competitive performance compared to existing models.

5. CONCLUSION

In this study, we introduce an innovative image compression model with reduced computational complexity, achieving performance comparable to state-of-the-art models. Our method leverages hierarchical feature extraction transforms to significantly lower complexity while effectively maintaining bit rate reduction. We conducted various comparisons with existing learned image compression models, focusing on computational complexity and performance metrics such as PSNR, and MS-SSIM. Our model demonstrated performance on par with state of the art models while retaining minimal complexity.

6. REFERENCES

[1] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston, "Variational image compression with a scale hyperprior," 2018.

[2] Wei Jiang, Jiayu Yang, Yongqi Zhai, Feng Gao, and Ronggang Wang, "MLIC++: Linear Complexity Multi-Reference Entropy Modeling for Learned Image Compression," 2024.

[3] Wei Jiang, Jiayu Yang, Yongqi Zhai, Peirong Ning, Feng Gao, and Ronggang Wang, "MLIC: Multi-Reference Entropy Model for Learned Image Compression," in *Proceedings of the 31st ACM International Conference on Multimedia*. 2023-10-27, MM '23, pp. 7618–7627, Association for Computing Machinery.

[4] Dailan He, Ziming Yang, Weikun Peng, Rui Ma, Hongwei Qin, and Yan Wang, "ELIC: Efficient Learned Image Compression With Unevenly Grouped Space-Channel Contextual Adaptive Coding," 2022, pp. 5718–5727.

[5] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár, "Lossy Image Compression with Compressive Autoencoders," 2017.

[6] Jiaying Liu, Dong Liu, Wenhan Yang, Sifeng Xia, Xiaoshuai Zhang, and Yuanying Dai, "A Comprehensive Benchmark for Single Image Compression Artifact Reduction," 2020, vol. 29, pp. 7845–7860.

[7] Jinming Liu, Heming Sun, and Jiro Katto, "Learned Image Compression With Mixed Transformer-CNN Architectures," 2023, pp. 14388–14397.

[8] Ming Lu, Peiyao Guo, Huiqing Shi, Chuntong Cao, and Zhan Ma, "Transformer-based Image Compression," in *2022 Data Compression Conference (DCC)*, 2022-03, pp. 469–469.

[9] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto, "Learned Image Compression With Discretized Gaussian Mixture Likelihoods and Attention Modules," 2020, pp. 7939–7948.

[10] David Minnen, Johannes Ballé, and George D Toderici, "Joint Autoregressive and Hierarchical Priors for Learned Image Compression," in *Advances in Neural Information Processing Systems*. 2018, vol. 31, Curran Associates, Inc.

[11] Jiaheng Liu, Guo Lu, Zhihao Hu, and Dong Xu, "A Unified End-to-End Framework for Efficient Deep Image Compression," 2020-05-23.

[12] Jonathan Gordon, Wessel P. Bruinsma, Andrew Y. K. Foong, James Requeima, Yann Dubois, and Richard E. Turner, "Convolutional Conditional Neural Processes," 2020-04.

[13] Théophile Blard, Théo Ladune, Pierrick Philippe, Gordon Clare, Xiaoran Jiang, and Olivier Déforges, "Overfitted Image Coding at Reduced Complexity," in *2024 32nd European Signal Processing Conference (EUSIPCO)*, 2024-08, pp. 927–931.

[14] Ayman A. Ameen, Thomas Richter, and André Kaup, "Compact Latent Representation for Image Compression (CLRIC)," in *Proceedings IEEE International Conference on Image Processing (ICIP)*, 2025.

[15] Yinghao Xu, Yujun Shen, Jiapeng Zhu, Ceyuan Yang, and Bolei Zhou, "Generative Hierarchical Features From Synthesizing Images," 2021, pp. 4432–4442.

[16] Samia Benyahia, Boudjelal Meftah, and Olivier Lézoray, "Multi-features extraction based on deep learning for skin lesion classification," vol. 74, pp. 101701, 2022-02-01.

[17] Anna Meyer, Fabian Brand, and André Kaup, "Learned wavelet video coding using motion compensated temporal filtering," vol. 11, pp. 113390–113401, 2023.

[18] Anna Meyer, Srivatsa Prativadibhayankaram, and André Kaup, "Variable rate learned wavelet video coding with temporal layer adaptivity," in *Proceedings IEEE International Conference on Image Processing (ICIP)*, 2025.

[19] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009-06, pp. 248–255.

[20] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár, "Microsoft COCO: Common Objects in Context," 2015-02-21.

[21] Tianfan Xue, Baian Chen, Jiajun Wu, Donglai Wei, and William T. Freeman, "Video Enhancement with Task-Oriented Flow," vol. 127, no. 8, pp. 1106–1125, 2019-08.

[22] Eirikur Agustsson and Radu Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2017-07, pp. 1122–1131.

[23] "JPEG AI," https://jpeg.org/jpegai/index.html, 2025.