CapsuleCoder: A Lesion-Aware Compression Framework for Capsule Endoscopy

Hai-Ning Zhao[1], Tian-Cheng Cao[1,2], Chen Shen[1], and Hen-Wei Huang[1,3]
[1]*School of Electrical and Electronic Engineering, Nanyang Technological University,* Singapore
[2]*Department of Emergency Medicine, Brigham and Women's Hospital,* Boston, USA
[3]*Lee Kong Chian School of Medicine, Nanyang Technological University,* Singapore

***Abstract*—Efficient and lightweight image compression is essential for capsule endoscopy systems that are subject to extremely strict constraints on wireless bandwidth, energy consumption, and on-chip memory. Conventional methods such as JPEG are content-agnostic, while most learned compression models remain too computationally expensive for on-device deployment. This paper proposes CapsuleCoder, a perception-driven and resource-efficient compression framework for capsule endoscopy. In this framework, a lightweight encoder firstly generates an importance-ordered latent representation, enabling coarse-to-fine reconstruction from partial channel transmission. Then, an auxiliary classifier predicts lesion probability from compact feature statistics and dynamically adjusts the number of transmitted channels, allocating more bits to lesion frames and fewer bits to non-lesion frames. The on-device pipeline is carefully designed to operate in a fully streaming manner, avoiding full feature-map buffering and significantly reducing on-chip memory requirements. Experiments on the capsule endoscopy dataset show that CapsuleCoder consistently outperforms the JPEG standard in terms of compression efficiency. The auxiliary classifier achieves a recall of 0.94, a precision of 0.62, and an F1-score of 0.75, ensuring reliable preservation of important frames and demonstrating the clinical feasibility of the proposed framework. These results indicate that CapsuleCoder can serve as a practical compression solution for real-world capsule endoscopy systems, enabling energy-efficient transmission while maintaining diagnostic safety.**

***Index Terms*—capsule endoscopy, adaptive image compression, lightweight image compression.**

I. Introduction

Capsule endoscopy has emerged as an effective and non-invasive technique for gastrointestinal tract examination, enabling long-duration and wide-coverage imaging without the discomfort associated with conventional wired endoscopes [1]–[3]. However, capsule systems are subject to extremely strict hardware constraints, including limited wireless bandwidth, finite battery capacity, and severely restricted on-chip computational and memory resources [3], [4]. Among all system components, wireless transmission is one of the dominant sources of power consumption, and its energy cost increases with the amount of transmitted data. These limitations in capsule endoscopes attributes to the poor spatial and temporal resolution compared to its tethered counterparts. Consequently, efficient image compression is a key technology for prolonging system lifetime and improving diagnostic reliability in capsule endoscopy.

Corresponding authors: Hai-Ning Zhao (HAINING001@e.ntu.edu.sg) and Tian-Cheng Cao (tiancheng.cao@ntu.edu.sg).
This work was supported by the Nanyang Assistant Professorship, MOE Tier 1 Grant (RG71/24), and the A*STAR Programmatic Seed Fund (M24N9b0125).

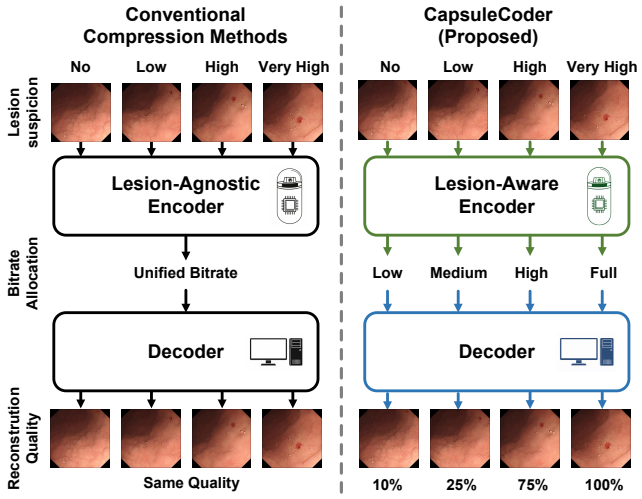


Fig. 1: Comparison between conventional compression methods and the proposed CapsuleCoder.

Early studies on capsule endoscopy compression mainly focused on lossless or near-lossless schemes to ensure diagnostic safety, such as adaptive prediction and entropy coding based methods [5], [6]. Although these approaches preserve visual fidelity, their compression ratios are inherently limited, which restricts their effectiveness in reducing wireless transmission load. In practical capsule systems, lossy transform-based methods, most notably JPEG, have therefore been widely adopted due to their simplicity, controllable compression ratio, and mature hardware support [7], [8]. However, JPEG and similar conventional schemes are fundamentally content-agnostic: the same compression parameters are applied to all frames regardless of their diagnostic importance. In real clinical scenarios, only a small fraction of frames contain lesions, while the majority correspond to normal tissues. Uniform compression thus leads to inefficient bandwidth utilization, either wasting transmission resources on non-informative frames or degrading the quality of clinically critical images.

In recent years, neural-network-based image compression has demonstrated remarkable improvements in rate–distortion performance by learning compact latent representations and

optimizing compression in an end-to-end manner [9], [10]. Subsequent works have further enhanced compression efficiency through hierarchical priors [11], context-based entropy models [12], [13], and architectural innovations such as attention mechanisms and transformer-based structures [14], [15]. Although these models achieve state-of-the-art reconstruction quality, they rely on deep feature transforms and sophisticated entropy models, resulting in heavy computational workloads and large intermediate feature buffering, which are far beyond the capabilities of capsule endoscopy platforms with extremely stringent power and memory budgets. To improve deployability, several lightweight learned compression models have been proposed by simplifying network architectures or coding strategies [16]–[18]. Nevertheless, their inference complexity and feature buffering remain challenging for ultra-low-power capsule platforms, and most existing methods still focus solely on reconstruction fidelity without explicitly considering clinical relevance.

These observations highlight two fundamental challenges for capsule endoscopy compression: (i) the compression strategy should be adaptive to the diagnostic importance of each frame, and (ii) the compression model must be extremely lightweight and hardware-friendly to enable true on-device deployment.

To address these challenges, we propose a perception-driven and resource-efficient compression framework named CapsuleCoder, which is specifically tailored for capsule endoscopy systems. The proposed method tightly couples lightweight neural compression with lesion-aware adaptive transmission. A compact encoder generates an importance-ordered latent representation, while an auxiliary lightweight classifier provides semantic guidance to dynamically adjust the number of transmitted latent channels. Frames predicted as lesion-positive are allocated more bits to preserve diagnostically critical details, whereas non-lesion frames are encoded with higher compression ratios to minimize bandwidth usage and transmission power. Compared with existing lesion-agnostic compression schemes that adopt a unified bitrate for all frames, CapsuleCoder enables a perception-driven adaptive bitrate allocation mechanism, as conceptually illustrated in Fig. 1.

II. Method

A. Overall Framework

As illustrated in Fig. 2 (a), the lightweight encoder and classifier are deployed on the capsule side under strict computational and memory constraints, while the computationally intensive decoder is placed on an external workstation. Each captured frame $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$ is first mapped by the encoder into a compressed latent representation $\mathbf{Y} \in \mathbb{R}^{12\times \frac{H}{8}\times \frac{W}{8}}$. A channel-wise dropout strategy is adopted during training to enforce an implicit importance ordering among latent channels, yielding a coarse-to-fine representation in which earlier channels carry more informative content [18]. This property forms the basis of the proposed adaptive transmission mechanism.

To enable semantic-aware compression, an auxiliary lightweight classifier predicts the lesion probability using compact statistical descriptors of $\mathbf{Y}$, specifically the channel-wise mean and variance, instead of directly processing high-dimensional feature maps. This design substantially reduces computational overhead while preserving sufficient discriminative capability. Based on the predicted lesion probability, the transmitter dynamically selects the number of latent channels to be sent. More channels (e.g., 9–12) are transmitted for lesion-positive frames to preserve diagnostic details, whereas only a small subset of the most important channels is transmitted for non-lesion frames to reduce bandwidth usage and energy consumption. The transmission rate can be flexibly adjusted according to bandwidth availability and clinical requirements.

The selected latent channels are received by the external device and decoded to reconstruct the image. Although only partial latent information is transmitted, the importance-ordered structure of the latent space ensures that high perceptual and diagnostic quality can still be achieved.

Overall, the proposed framework couples semantic understanding, compression, and transmission into a unified perception-driven pipeline, achieving an effective trade-off between reconstruction quality and communication efficiency, and making it well suited for the highly constrained environment of capsule endoscopy.

B. On-Device Module Design

The on-device module is designed under the strict computational and memory constraints of capsule endoscopy, and thus prioritizes lightweight convolutional operations and compact decision signals. It consists of two components: (i) a lightweight encoder that produces an importance-ordered latent representation, and (ii) an auxiliary classifier that predicts lesion probability from low-dimensional latent statistics.

1) Lightweight encoder: Given an input frame $\mathbf{I} \in \mathbb{R}^{3\times H\times W}$, the encoder produces a compact latent tensor $\mathbf{Y} \in \mathbb{R}^{12\times \frac{H}{8}\times \frac{W}{8}}$ using a shallow three-layer convolutional stack, as illustrated in Fig. 2 (b). The first two layers adopt deep separable convolutions for efficiency: a 7×7 layer with stride 2 followed by a 5×5 layer with stride 4, leading to an overall spatial downsampling factor of 8. The third layer is a 3×3 standard convolution that directly outputs the 12-channel latent representation $\mathbf{Y}$.

This hybrid design balances computational efficiency and representational capacity. Deep separable convolution reduces computational burden by decoupling spatial filtering and channel mixing. However, such factorization also weakens cross-channel feature interaction and may limit the expressive power of the network if applied excessively. To avoid this over-simplification, the final layer is intentionally implemented as a standard convolution rather than a deep separable one. This choice is justified from both computational and representational perspectives. On one hand, since this layer uses a small 3×3 kernel, the additional computational overhead is limited, while the benefit in cross-channel feature fusion is more

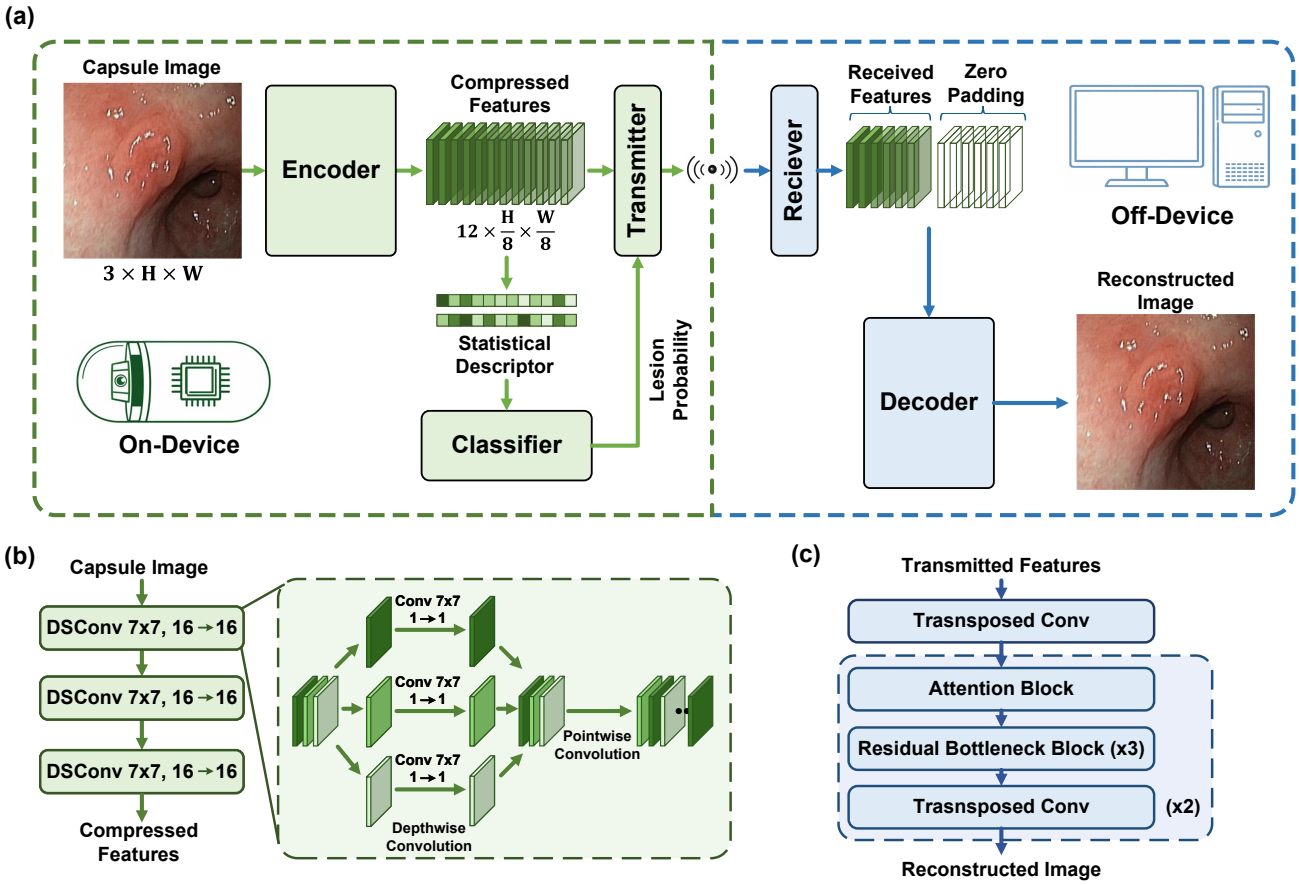


Fig. 2: Overview of the proposed CapsuleCoder framework: (a) system workflow, (b) lightweight encoder architecture, and (c) decoder network for image reconstruction. In this framework, the on-device module performs image compression and lesion-aware channel selection, while the off-device module reconstructs the image from the received features.

pronounced. On the other hand, this layer directly determines the transmitted latent representation, and its expressiveness has a stronger impact on reconstruction fidelity than intermediate features, especially under partial-channel transmission. Preserving a full convolution at this stage enables richer channel-wise feature fusion and stabilizes the expressiveness of the latent space.

2) Compact semantic classifier: To enable semantic-aware transmission without introducing heavy on-device feature processing, the classifier operates on compact descriptors derived from $\mathbf{Y}$ rather than the full feature maps. Specifically, we compute the channel-wise mean and variance over spatial dimensions. The two 12-dimensional vectors are concatenated into a 24-dimensional descriptor and fed into a lightweight two-layer multilayer perceptron (24→64→1) with a sigmoid output, producing a lesion probability $p \in [0, 1]$.

3) Streaming computation: As illustrated in Fig. 3, the on-device pipeline is implemented in a streaming manner to avoid full-frame buffering. The input pixels are processed line-by-line, and each convolution layer maintains only a small line buffer whose depth is determined by the kernel size. Meanwhile, the semantic classifier is computed from streaming statistics of the latent tensor, requiring only a few scalar accumulators and a tiny multilayer perceptron.

For a streaming $K \times K$ convolution, the minimal buffering requirement is approximately K rows of the input feature map. Let b denote the number of bytes per stored element (e.g., b=1 for 8-bit, b=2 for 16-bit), and let the input width be W. Then the encoder line-buffer memory can be estimated as:

$$M_{\text{enc}} \approx b\Big(3 \cdot 7 \cdot W + 16 \cdot 5 \cdot \frac{W}{2} + 16 \cdot 3 \cdot \frac{W}{8} + 12 \cdot \frac{H}{8} \cdot \frac{W}{8}\Big) = 67\, b\, W + \frac{3}{16}\, b\, HW \ \text{ bytes.} \tag{1}$$

Here, the three terms correspond to the line buffers for (i) the 7×7 depthwise convolution on the RGB input, (ii) the 5×5 depthwise convolution on the $16 \times \frac{H}{2} \times \frac{W}{2}$ feature map, (iii) the 3×3 convolution on the $16 \times \frac{H}{8} \times \frac{W}{8}$ feature map, and (iv) the whole compressed latent feature, respectively. Pointwise (1×1) convolutions in deep separable blocks do not require additional line buffering beyond the current pixel stream.

The classifier operates on channel-wise mean and variance of the latent tensor $\mathbf{Y} \in \mathbb{R}^{12 \times \frac{H}{8} \times \frac{W}{8}}$. In streaming mode, $\boldsymbol{\mu}(\mathbf{Y})$ and $\boldsymbol{\sigma}^2(\mathbf{Y})$ are computed using per-channel running sums and running squared sums, which require 2×12 scalar accumulators. In addition, the intermediate activations of the two

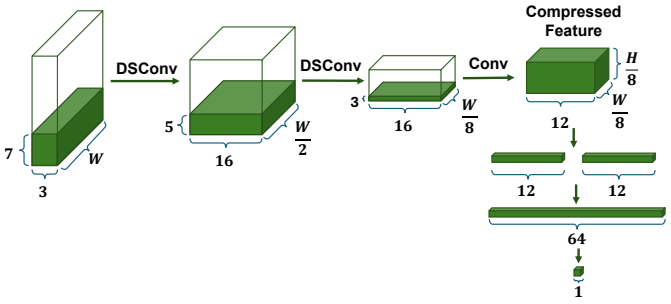


Fig. 3: On-device streaming computation of the lightweight encoder and classifier.

fully connected layers contain 64 and 1 scalars, respectively. Therefore, the total memory consumption of the classifier is

$$M_{\text{mlp}} \approx (24 + 64 + 1)\, b = 89\, b \ \text{ bytes}, \tag{2}$$

Combining the streaming encoder buffers, the classifier statistics, and the classifier activations, the total on-chip RAM requirement is

$$\begin{aligned} M_{\text{RAM}} &\approx M_{\text{enc}} + M_{\text{mlp}} \\ &\approx 67\, b\, W + \frac{3}{16}\, b\, HW + 89\, b \ \text{ bytes} \end{aligned} \tag{3}$$

For a representative input resolution of 320×320 and an 8-bit fixed-point implementation ($b = 1$ byte), this yields

$$\begin{aligned} M_{\text{RAM}} &\approx 67 \times 320 + \frac{3}{16} \times 320 \times 320 + 89 \\ &= 40729 \text{ bytes} \approx 39.8 \text{ KiB}. \end{aligned} \tag{4}$$

Among this, about 21.0 KiB is consumed by the computational buffer of encoder and classifier, while the compressed latent feature requires about 18.8 KiB.

This memory footprint is much smaller than that required for storing a full 320×320 RGB frame (approximately 300 KiB at 8-bit precision). As a result, the proposed design eliminates the need for full-frame buffering and can be implemented as a compact streaming pipeline. With an overall on-chip memory requirement of only tens of kilobytes, the encoder-classifier pair is well suited for deployment on capsule endoscopic platforms equipped with limited on-chip memory and strict power budgets, making true on-device inference practically feasible.

C. Off-Device Decoder Design

The off-device decoder is deployed on an external workstation and is allowed to use a higher-capacity architecture to maximize reconstruction quality. It takes the received partial latent tensor $\tilde{\mathbf{Y}} \in \mathbb{R}^{12 \times \frac{H}{8} \times \frac{W}{8}}$, where the untransmitted channels are filled with zeros, and reconstructs the corresponding RGB frame. Our decoder follows the design principles of learned image compression decoders that combine progressive upsampling with residual refinement and attention mechanisms [14]. As shown in Fig. 2(c), the decoder adopts a progressive upsampling architecture that restores the spatial resolution from $\frac{H}{8} \times \frac{W}{8}$ to $H \times W$ through three transposed-convolution layers with stride 2. Between successive upsampling stages, residual bottleneck blocks are employed to refine local structures and stabilize optimization, while attention blocks are inserted at early and intermediate stages to enhance global feature aggregation, which is particularly important when the latent representation is incomplete.

D. Training Strategy

The model is trained end-to-end to jointly optimize image reconstruction and lesion-aware decision making. To make the learned latent space compatible with adaptive channel transmission, we employ three key training strategies: (i) latent-channel dropout to enforce an importance-ordered representation, (ii) additive noise to emulate quantization and transmission perturbations, and (iii) a multi-task objective that combines reconstruction and classification losses.

a) Latent-channel dropout.: To induce an implicit importance ordering among latent channels and support coarse-to-fine reconstruction from partial latents, we apply a channel-wise dropout strategy during training [18]. During each training iteration, only the first k channels are retained, where k is randomly sampled from $\{1, \ldots, 12\}$. Formally, we construct a transmitted latent $\tilde{\mathbf{Y}}$ by

$$\tilde{\mathbf{Y}}_{1:k,:,:} = \mathbf{Y}_{1:k,:,:}, \quad \tilde{\mathbf{Y}}_{k+1:12,:,:} = \mathbf{0}, \tag{5}$$

This structured masking encourages the network to place the most essential information into earlier channels, so that using fewer channels yields a coarse yet recognizable reconstruction, and additional channels progressively refine details.

b) Noise injection to mimic quantization and transmission perturbations.: To improve robustness to quantization effects and non-ideal wireless links, we further perturb the latent tensor with small additive noise. Given the dropped latent $\tilde{\mathbf{Y}}$, we add uniform noise in a narrow range:

$$\tilde{\mathbf{Y}} \leftarrow \tilde{\mathbf{Y}} + \boldsymbol{\epsilon}, \qquad \boldsymbol{\epsilon} \sim \mathcal{U}(-\delta, \delta), \tag{6}$$

where δ is set to 0.01 in our implementation. This noise injection provides a simple yet effective approximation to quantization and mild channel perturbations, and helps the decoder remain stable when reconstructing from incomplete and slightly corrupted latents.

c) Multi-task objective.: The training objective consists of a reconstruction loss and a lesion classification loss. For reconstruction, we adopt the multi-scale structural similarity (MS-SSIM) loss [19], which better correlates with perceptual image quality than pixel-wise metrics:

$$\mathcal{L}_{\text{rec}} = 1 - \text{MS-SSIM}(\hat{\mathbf{I}}, \mathbf{I}), \tag{7}$$

where $\hat{\mathbf{I}}$ and $\mathbf{I}$ denote the reconstructed and original images, respectively. For lesion prediction, we use a focal loss [20] to address the class imbalance between lesion and non-lesion samples and to emphasize hard-to-classify examples:

$$\mathcal{L}_{\text{cls}} = \text{FocalLoss}(p, y), \tag{8}$$

where $y \in \{0, 1\}$ is the ground-truth lesion label and $p \in [0, 1]$ is the predicted lesion probability. The overall training objective is defined as the sum of the reconstruction and classification losses:

$$\mathcal{L} = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{cls}}. \tag{9}$$

This joint optimization encourages the encoder to learn a latent representation that simultaneously preserves perceptual reconstruction quality and retains discriminative semantic information, thereby supporting reliable lesion-aware adaptive transmission.

E. Dataset

All experiments in this work are conducted on the Kvasir-Capsule dataset [21], which is a large publicly available capsule endoscopy dataset. The dataset is originally divided into three parts: labeled image data, labeled video data, and unlabeled video data. In this work, we only use the labeled image data for both the compression task and the auxiliary lesion-aware classification task. The labeled image set consists of 47,238 images belonging to one of 14 different classes of findings.

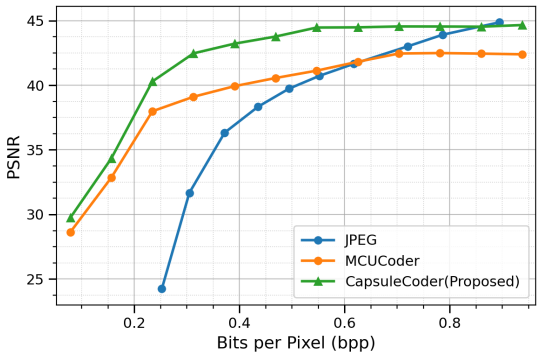
In order to enable lesion-aware dynamic compression, we reformulate the original 14-class classification task into a binary classification problem by grouping the data into lesion and non-lesion categories. Specifically, the dataset contains 4,266 lesion frames and 42,972 non-lesion frames, resulting in a highly imbalanced class distribution. The dataset is further randomly split into training, validation, and test sets with a ratio of 6:2:2.

F. Evaluation Metrics

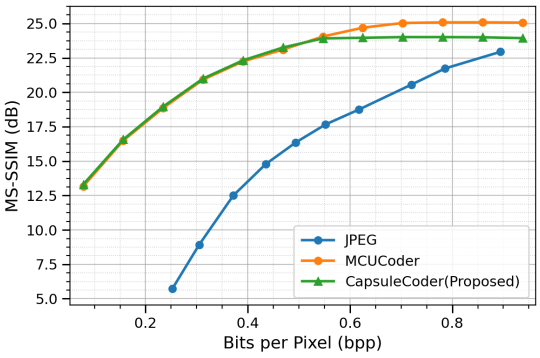
The performance of the proposed framework is evaluated from two aspects: image compression quality and lesion-aware classification accuracy.

For the compression task, we adopt Peak Signal-to-Noise Ratio (PSNR) [22] and Multi-Scale Structural Similarity Index Measure (MS-SSIM) [19] as evaluation metrics. PSNR measures the pixel-level reconstruction fidelity, while MS-SSIM focuses on perceptual and structural similarity, which is more consistent with human visual perception and clinical diagnostic requirements.

For the lesion classification task, Recall, Precision, and F1-score are used as evaluation metrics. In our framework, recall is of primary importance since a false negative result would cause pathological frames to be highly compressed, potentially losing critical diagnostic information. Precision reflects the efficiency of the dynamic compression strategy, as false positives unnecessarily trigger low-compression modes and increase transmission cost. The F1-score, as the harmonic mean of precision and recall, provides a balanced measure that reflects the overall trade-off between diagnostic safety and transmission efficiency. Therefore, these metrics jointly characterize the reliability and effectiveness of the proposed lesion-aware compression system.



(a) PSNR versus bits per pixel (bpp).



(b) MS-SSIM versus bits per pixel (bpp).

Fig. 4: Rate–distortion performance comparison of JPEG, MCUCoder, and the proposed CapsuleCoder on the Kvasir-Capsule dataset.

III. Results

A. Compression Performance

1) Quantitative Analyses: Fig. 4 presents the quantitative rate–distortion performance of JPEG, MCUCoder, and the proposed CapsuleCoder on the Kvasir-Capsule dataset in terms of PSNR and MS-SSIM. As shown in Fig. 4(a) and Fig. 4(b), CapsuleCoder clearly outperforms the conventional JPEG standard in terms of both PSNR and MS-SSIM over almost the entire bitrate range, with particularly significant gains in the low-bitrate regime. This demonstrates that the proposed network is able to preserve both pixel-level reconstruction fidelity and perceptual structural information much more effectively than traditional transform-based compression, which is crucial for maintaining diagnostic reliability under strict bandwidth constraints.

When compared with MCUCoder, CapsuleCoder exhibits even stronger performance in terms of PSNR, as illustrated in Fig. 4(a), where it surpasses MCUCoder over most bitrate intervals. This indicates that the lightweight design not only reduces computational complexity and memory consumption, but can also improve reconstruction accuracy. In terms of MS-SSIM shown in Fig. 4(b), CapsuleCoder achieves highly competitive performance with MCUCoder. Although it is slightly lower in the range of 0.6–1.0 bpp, the gap remains marginal.

Fig. 5: Qualitative comparison on an angiectasia case between JPEG, MCUCoder, and the proposed CapsuleCoder under different bitrates. For each method, reconstruction results are presented from low to high bitrates, together with the corresponding BPP, MS-SSIM (dB), and PSNR values.

Considering that CapsuleCoder is designed with substantially lower computational complexity and significantly reduced on-chip memory requirements, this small difference represents a favorable trade-off between efficiency and perceptual quality.

Notably, despite its more lightweight encoder, CapsuleCoder achieves even better performance in the low-bitrate regime, which is mainly attributed to the auxiliary lesion classifier acting as side supervision during training. This semantic guidance encourages the encoder to learn a more discriminative and compact latent representation, making the leading channels more informative. As a result, CapsuleCoder can preserve more perceptually and diagnostically important content when only a small number of channels are transmitted, thereby outperforming MCUCoder under strict bitrate constraints.

2) Qualitative Analyses: Fig. 5 shows a qualitative comparison among JPEG, MCUCoder, and the proposed CapsuleCoder on a representative angiectasia example. At low bitrates, JPEG suffers from severe compression artifacts and color degradation, leading to obvious blocking effects and loss of structural consistency in the lesion region. The overall visual quality is significantly degraded, and fine appearance cues are hardly preserved. MCUCoder produces visually smoother reconstructions than JPEG, and most blocking artifacts are effectively suppressed. However, its results still exhibit evident blurring and color distortion. For instance, at around 0.16 bpp, MCUCoder shows visible color shifts and over-smoothed textures in the lesion area. In contrast, CapsuleCoder achieves superior reconstruction quality in the low-bitrate regime while relying on a substantially more lightweight encoder. At the same bitrate (e.g., 0.16 bpp), CapsuleCoder preserves more accurate color appearance and introduces less blurring compared with MCUCoder.

B. Classification Performance

Fig. 6 shows the confusion matrix of the auxiliary lesion classifier on the test set. Based on this result, the classifier achieves a recall of 0.94, a precision of 0.62, and an F1-score of 0.75. The high recall indicates that most lesion frames can be successfully identified, which is crucial for avoiding over-compression of diagnostically important images. Although the precision is relatively lower, this behavior is acceptable in our application scenario, since false positives only lead to a slightly higher transmission cost, while false negatives may cause insufficient preservation of lesion-related details. The achieved F1-score further demonstrates a good balance between sensitivity and specificity under an extremely

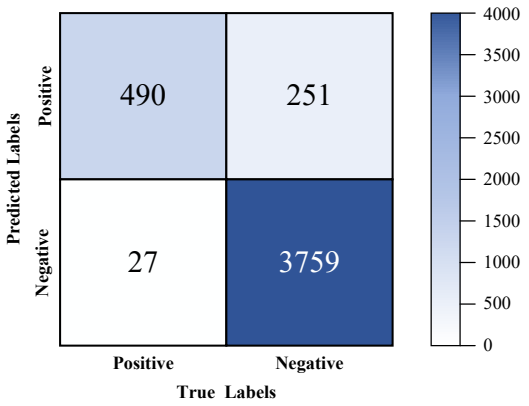


Fig. 6: Confusion matrix of the auxiliary lesion classifier on the test dataset.

lightweight model design, confirming that the classifier provides sufficiently reliable semantic guidance for the proposed adaptive transmission strategy.

IV. Conclusion

This work presented CapsuleCoder, a perception-driven and resource-efficient compression framework tailored for capsule endoscopy. By combining importance-ordered latent representations with lesion-aware adaptive transmission, the proposed method effectively balances diagnostic reliability and communication efficiency. The lightweight encoder, streaming computation scheme, and compact auxiliary classifier enable on-device deployment under severe memory and power constraints. Experimental results demonstrate that CapsuleCoder outperforms JPEG and achieves performance comparable to the neural compression method with substantially reduced resource requirements. These results indicate that CapsuleCoder provides a practical and effective solution for next-generation intelligent capsule endoscopy systems.

References

[1] P. Oka, M. McAlindon, and R. Sidhu, "Capsule endoscopy-a non-invasive modality to investigate the gi tract: out with the old and in with the new?" *Expert Review of Gastroenterology & Hepatology*, vol. 16, no. 7, pp. 591–599, 2022.

[2] C.-C. Su, C.-K. Chou, A. Mukundan, R. Karmakar, B. F. Sanbatcha, C.-W. Huang, W.-C. Weng, and H.-C. Wang, "Capsule endoscopy: Current trends, technological advancements, and future perspectives in gastrointestinal diagnostics," *Bioengineering*, vol. 12, no. 6, p. 613, 2025.

[3] Q. Cao, R. Deng, Y. Pan, R. Liu, Y. Chen, G. Gong, J. Zou, H. Yang, and D. Han, "Robotic wireless capsule endoscopy: recent advances and upcoming technologies," *Nature Communications*, vol. 15, no. 1, p. 4597, 2024.

[4] C. Babu and D. A. Chandy, "A review on lossless compression techniques for wireless capsule endoscopic data," *Current Medical Imaging*, vol. 17, no. 1, pp. 27–38, 2021.

[5] T. H. Khan and K. A. Wahid, "Design of a lossless image compression system for video capsule endoscopy and its performance in in-vivo trials," *Sensors*, vol. 14, no. 11, pp. 20 779–20 799, 2014.

[6] Q. Al-Shebani, P. Premaratne, P. J. Vial, and D. J. McAndrew, "The development of a clinically tested visually lossless image compression system for capsule endoscopy," *Signal Processing: Image Communication*, vol. 76, pp. 135–150, 2019.

[7] G. K. Wallace, "The jpeg still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[8] A. Barabi, D. Sason, and R. Cohen, "Low complexity image compression of capsule endoscopy images," in *2014 IEEE 28th Convention of Electrical & Electronics Engineers in Israel (IEEEI)*. IEEE, 2014, pp. 1–5.

[9] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," *arXiv preprint arXiv:1611.01704*, 2016.

[10] J. Ballé, "Efficient nonlinear transforms for lossy image compression," in *2018 Picture Coding Symposium (PCS)*. IEEE, 2018, pp. 248–252.

[11] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436*, 2018.

[12] M. Li, K. Zhang, J. Li, W. Zuo, R. Timofte, and D. Zhang, "Learning context-based nonlocal entropy modeling for image compression," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 3, pp. 1132–1145, 2021.

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

[14] 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*, 2020, pp. 7939–7948.

[15] M. Lu, P. Guo, H. Shi, C. Cao, and Z. Ma, "Transformer-based image compression," *arXiv preprint arXiv:2111.06707*, 2021.

[16] M. Li, Z. Wang, L. Shen, Q. Ding, L. Yu, and X. Jiang, "Lightweight image compression based on deep learning," in *CAAI International Conference on Artificial Intelligence*. Springer, 2022, pp. 106–116.

[17] Y. Bao, W. Tan, C. Jia, M. Li, Y. Liang, and Y. Tian, "Shiftlic: Lightweight learned image compression with spatial-channel shift operations," *IEEE Transactions on Circuits and Systems for Video Technology*, 2025.

[18] A. Hojjat, J. Haberer, and O. Landsiedel, "Mcucoder: Adaptive bitrate learned video compression for iot devices," in *DAGM German Conference on Pattern Recognition*. Springer, 2025, pp. 123–138.

[19] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," in *The thrity-seventh asilomar conference on signals, systems & computers, 2003*, vol. 2. Ieee, 2003, pp. 1398–1402.

[20] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[21] P. H. Smedsrud, V. Thambawita, S. A. Hicks *et al.*, "Kvasir-capsule, a video capsule endoscopy dataset," *Scientific Data*, vol. 8, no. 1, p. 142, 2021.

[22] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of psnr in image/video quality assessment," *Electronics Letters*, vol. 44, no. 13, pp. 800–801, 2008.