Semantic Fidelity Meets Efficiency: Reviewing Vision-Language Models (VLMs) and Compression Approaches

□ **Mubashir Hussain Shah, Kiho Choi** / Kyung Hee University

Abstract

Vision Language Models (VLMs) enable powerful multimodal reasoning but scale poorly: large parameter counts, long visual token sequences, and high-bandwidth inputs make training and deployment expensive. Meanwhile, modern compression techniques-from learned neural codecs to semantic-aware codecs-offer new avenues to reduce computation, memory, and communication costs. This review synthesizes recent research at the intersection of VLMs and compression, organizing work into six categories: visual token compression, compressed-latent inputs, compressed-domain inference, semantic compression, edge server split computing with feature compression, and VLM-guided image compression. We analyze representative methods in each category, quantify typical trade-offs between efficiency and task performance, and identify key challenges for robustness, generalization, and privacy. Framing the field as a co-design problem, we argue that future progress requires jointly optimizing compression pipelines and multimodal architectures to preserve semantic fidelity under resource constraints.

I. Introduction

The rapid progress of artificial intelligence has been marked by a shift toward multimodal learning, where models integrate information from vision and language to perform tasks such as captioning,

※ This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government(MSIT) (RS-2025-16069081).

retrieval, and question answering. At the center of this shift are VLMs, which align images and text in shared embedding spaces and enable powerful zero -shot and few-shot reasoning capabilities [14, 9, 1]. From assistive technologies and content moderation to healthcare and education, VLMs are increasingly deployed across domains where semantic understanding of visual and textual content is essential [10, 11].

Despite these advances, current VLMs are often prohibitively resource-intensive. Training and inference typically involve hundreds of millions or even billions of parameters, high-resolution inputs decomposed into thousands of tokens, and substantial GPU compute, making large-scale deployment expensive and impractical on mobile, embedded, or bandwidth-limited platforms [16]. Addressing these efficiency bottlenecks is critical for scaling multimodal intelligence beyond research prototypes to real-world systems.

In parallel, compression has long served as a foundational technology for efficient storage and transmission. Traditional codecs such as JPEG and AVC/H.264 exploit redundancy to enable scalable sharing of images and video, while neural network compression techniques, such as pruning, quantization, and knowledge distillation, have reduced the complexity of deep models [3, 4]. More recently, learned neural codecs and semantic-aware compression schemes have shifted focus from pixel fidelity toward perceptual and task-relevant preservation, opening new opportunities to align efficiency with reasoning objectives [8, 13].

These developments highlight a growing convergence: VLMs and compression, historically distinct research threads, are becoming deeply interdependent. Compression enhances VLMs by reducing computational cost through token pruning [15, 17, 18, 19], compressed-latent inputs [7, 20], and compressed-domain inference [12, 2]. At the same time, VLMs are not only consumers of compressed data but also contributors to the design of compression methods. By signaling which regions or features are most relevant for downstream reasoning, they enable codecs to prioritize semantically important content and discard less critical details, resulting in compressed outputs that remain highly informative for multimodal tasks [6, 8, 13]. This two-way interaction shifts the role of efficiency: rather than being treated purely as a cost-saving optimization, it becomes a key mechanism for preserving semantic fidelity within multimodal intelligence. Although earlier surveys have examined VLM architectures [10] or efficiency-oriented strategies [16] separately, a unified account of their intersection has not yet been developed.

This review seeks to close that gap by offering a structured synthesis of research at the interface of VLMs and compression. Specifically, we: (i) present a taxonomy of approaches spanning token pruning, compressed-latent inputs, compressed-domain inference, semantic compression, edge-server feature compression, and VLM-guided codecs; (ii) analyze trade-offs in accuracy, efficiency, and robustness across categories; and (iii) identify open challenges and opportunities for co-designing VLMs and compression frameworks.

Accordingly, this review aims to provide both a synthesis of existing work and a roadmap for future research at the intersection of VLMs and compression, where efficiency and semantics are jointly optimized to enable practical, scalable, and

intelligent multimodal systems.

II. Background

VLMs are designed to integrate visual and textual information into a shared representational space, supporting tasks such as zero-shot recognition, captioning, visual question answering, and multimodal retrieval. Early progress came from large-scale contrastive training on paired image-text datasets, which enabled strong generalization without heavy reliance on task-specific labels [14, 5]. Building on this foundation, more advanced designs combined frozen or pretrained vision encoders with large language models and cross-attention modules, giving rise to architectures capable of richer multimodal reasoning, few-shot adaptation (e.g., Flamingo, BLIP-2), and unified generative systems that extend reasoning to both modalities (e.g., GPT-4V) [1, 9, 11]. Surveys of this field emphasize both the speed of progress and the wide application space, from search and retrieval to clinical imaging, that makes scalability and efficiency increasingly important considerations for real-world deployment [10].

However, despite these successes, VLMs face three interconnected problems when moving from lab settings to practical systems. The first is model size and computation: state-of-the-art VLMs typically rely on hundreds of millions to billions of parameters and demand large amounts of FLOPs during inference, creating significant energy, latency, and financial costs [16]. The second relates to input representation: transformer-based vision modules generate many tokens for each image and exponentially more for video, which inflates both memory usage and computational requirements. To mitigate the token explosion, methods such as dynamic sparsification and visual-token compression have been proposed, though they often introduce trade-offs in accuracy and generalization [15, 17, 18, 19]. Third, many systems split work between edge devices and servers, making transmission of raw or fully decoded inputs a serious bandwidth and latency concern; this motivates compressed-domain inference and feature-level transmission as practical alternatives to raw data transfer [12], [2], [24].

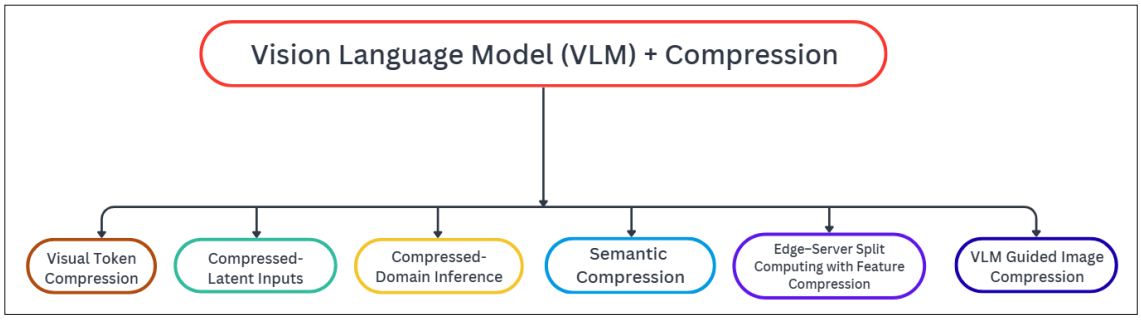
Because these bottlenecks are both computational and semantic, compression for multimodal systems must do more than reduce bits: it must preserve the information needed for downstream reasoning. Promising directions include learned neural codecs and compact latents that retain perceptual and semantic content at far lower bitrate than pixels [7], [20]; compressed-domain methods that operate on transform or bitstream features to avoid decoding overhead [12], [2]; and semantic-aware or model-guided codecs that optimize preservation of task-relevant cues using textual or VLM feedback [6], [8], [13]. Feature compression and lightweight bottlenecks enable practical edge-server pipelines by transmitting compact, task-oriented representations rather than raw images [24], [25], [21]. Overall, the literature points to a bidirectional convergence: on one side, compression strategies are being redesigned to support the computational demands of VLMs, and on the other, VLMs themselves are beginning to inform how compression should prioritize

semantic content. This mutual influence highlights the need for a co-design approach in which codecs, compact representations, and multimodal architectures are optimized together to balance semantic preservation, generalization, and efficiency in settings with limited resources [6, 8, 16].

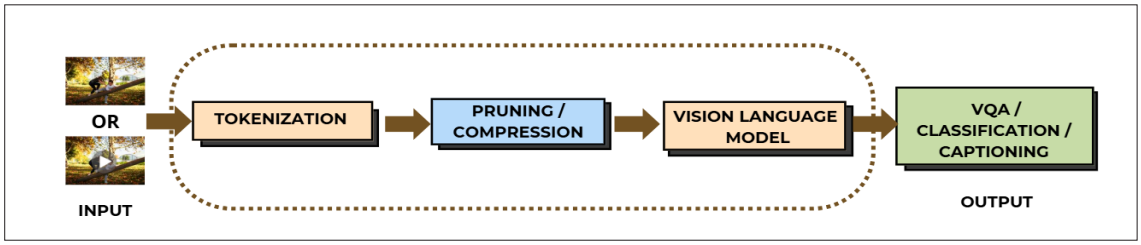
III. Review of Recent VLMs with Compression Approaches

The convergence of VLMs and compression has become a key research focus, engaging both multimodal learning and efficient computing communities. Compression reduces computational, memory, and bandwidth demands by eliminating redundancy in tokens, latents, or intermediate features. In turn, VLMs guide codecs with semantic cues, ensuring task-relevant details are preserved even under heavy compression. This two-way relationship reframes efficiency not just as resource minimization but as a means to sustain semantic accuracy in practical settings. Existing surveys help frame this discussion, for example, Nguyen et al. review VLM progress, outlining applications and ongoing challenges [10]. Complementarily, Shinde et al. emphasize efficiency, identifying model compression and runtime optimization as central to deploying VLMs in practical environments [16]. Building on this foundation, our review turns specifically to works that lie at the intersection of these two domains. To this end, we organize the literature into six categories as shown in Figure 1.

The figure provides a structured view of how compression techniques and VLMs complement each other across different stages of the multimod-



<Figure 1> Categorization of Vision-Language Models with Compression Techniques



<Figure 2> General Pipeline of Visual Token Compression

al pipeline. Each category addresses efficiency from a distinct perspective.

1. Visual Token Compression

Visual token compression is one of the most direct strategies for reducing the computational burden of VLMs. In transformer-based architectures, visual inputs such as images and videos are divided into patches or frame tokens, each processed individually in a long input sequence. While this representation allows fine-grained multimodal reasoning, it also drastically increases sequence length, driving up FLOPs, memory usage, latency, and energy costs. Token compression addresses this issue by pruning or merging redundant tokens before they enter the multimodal pipeline, thereby reducing computational overhead while striving to preserve the semantic fidelity needed for downstream reasoning tasks.

Figure 2 presents a general pipeline: raw inputs are divided into patches or frames, after which a pruning mechanism selects or merges a subset of tokens. The resulting compressed sequence is then passed to the VLM for tasks such as captioning, retrieval, or VQA. The motivation behind this approach is the uneven semantic value of tokens. Background regions or static areas often add little to reasoning, and successive video frames typically repeat information. Processing these tokens consumes resources without boosting performance, which drives interest in pruning strategies that retain only the most informative content. The main challenge, however, is balance such that pruning too much risks losing subtle but important cues needed for grounding and interpretability.

Research in this area has progressed through several influential contributions [21]. DynamicViT introduced adaptive sparsification that removes tokens according to importance scores, demonstrating notable computational savings while preserving accuracy [15]. Building on this, LLaVA-Zip proposed pruning guided by intrinsic image features, enabling the number of tokens to adjust dynamically at inference without retraining [17]. For video inputs, PruneVid addressed temporal redundancy by merging static tokens across frames, maintaining motion cues while reducing sequence length [18]. More recently, DivPrune employed diversity-aware learning, ensuring that pruning preserves semantically distinct tokens rather than collapsing them into oversimplified representations [19].

Empirical studies confirm the benefits of token compression across multimodal benchmarks. Reported gains typically include notable reductions in computation with accuracy levels close to pixel-based baselines [15, 17, 18, 19]. Crucially, most methods integrate smoothly with pretrained VLMs, requiring minimal fine-tuning and making them feasible for existing pipelines.

However, performance is sensitive to pruning rates: overly aggressive compression can degrade semantic grounding, harm fine-grained reasoning, and even induce hallucinations in generative outputs. As such, adaptive and task-aware pruning strategies represent the current trend, emphasizing robustness by modulating pruning intensity based on input complexity and task demands.

Despite promising advances, token compres-

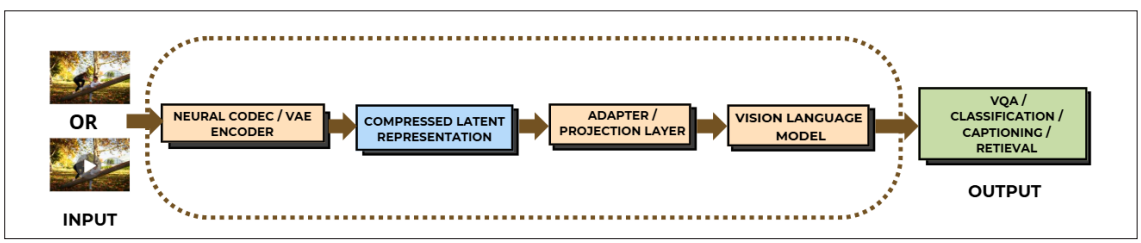
sion faces persistent challenges. Over-pruning rare but important cues, such as small objects or subtle text, remains a significant risk, especially in tasks that depend on fine detail [15, 19]. Generalization across tasks and domains is another limitation, since pruning policies tuned for classification may underperform on VQA or dense-captioning tasks [17]. For video, maintaining a balance between discarding redundancy and preserving motion cues is non-trivial [18]. Finally, evaluation practices remain inconsistent: many works report FLOPs or accuracy but omit metrics such as latency, memory usage, or robustness under pruning variation, making it difficult to compare methods fairly across studies [21].

Addressing these limitations requires methodological and evaluation improvements. On the methodological side, pruning strategies should increasingly become adaptive, multimodal, and task-aware, integrating textual prompts or task objectives when available to avoid discarding relevant visual tokens. Diversity- and uncertainty -aware pruning objectives can further safeguard against losing rare but critical cues [19]. For evaluation, researchers should adopt standardized benchmarking practices, reporting FLOPs, end-to-end latency, memory footprint, and downstream task metrics across multiple pruning rates, while also releasing code and reproducible configurations [21]. Expanding evaluations to domain-shift and multi-task scenarios will also help assess generalization.

In summary, visual token compression provides a foundational pathway for scaling VLMs to resource-constrained environments. By reducing token-level redundancy while retaining semantic relevance, these methods improve computational efficiency without substantially sacrificing accuracy. The field is now moving toward adaptive frameworks that allocate attention and token budgets dynamically according to both input characteristics and task requirements. This evolution positions token compression not just as a cost-saving mechanism but also as an essential enabler of practical, robust, and semantically grounded multimodal intelligence [15, 17, 18, 19].

2. Compressed-Latent Inputs

Compressed-latent inputs offer a promising route to improve VLM efficiency by replacing raw-pixel inputs with compact intermediate representations produced by neural image or video codecs. Instead of decoding full images and then extracting



<Figure 3> General Pipeline of Compressed-Latent Inputs

features for a vision encoder, the pipeline encodes inputs into latent vectors that are both smaller and often rich in perceptual structure. Operating on these latents avoids the repeated decode-encode overhead, reduces storage and transmission costs, and can substantially lower preprocessing FLOPs and memory usage during multimodal inference.

Figure 3 illustrates that an image or video is first passed through a neural codec that produces compact latents; a lightweight adapter or mapper then converts those latents into an embedding space compatible with the VLM; finally, the VLM performs downstream reasoning such as retrieval, captioning, or VQA directly on the adapted latents. The central motivation for this approach stems from advances in learned codecs that produce highly compact representations optimized for perceptual quality. These latents frequently encode edges, textures, and higher-level patterns useful for recognition, making them attractive as substitutes for pixel-based inputs when an appropriate alignment to VLM embeddings can be achieved.

Work in this area has proposed several effective strategies for bridging the representation gap between codec latents and VLM input spaces. Kao et al. developed adapter layers that map compressed image latents directly into the input embedding distribution of large multimodal models, allowing integration without retraining the entire VLM [7]. TiTok showed that images can be distilled into as few as 32 tokens for both reconstruction and downstream generation/reasoning, demonstrating that extremely compact latent encodings can preserve semantics sufficient for multimodal tasks [20]. Other efforts, such as text-guided neural compression (TACO), condition the codec on auxiliary signals so that latents better reflect task-relevant semantics, improving downstream performance under tight bitrate constraints [8].

Empirical results indicate that compressed-latent pipelines can deliver marked savings in bandwidth and memory while maintaining competitive accuracy on many benchmarks, provided the adapter and latent design are well chosen. Typical gains include reduced transmitted bits for edge-cloud workflows and lower preprocessing FLOPs by avoiding full pixel decoding. However, naive use of codec latents may produce a representational mismatch: latents optimized purely for visual fidelity do not necessarily emphasize the semantic features a VLM requires, and this mismatch can degrade reasoning performance. Consequently, adapter quality, joint optimization of codec and adapter, or task-aware latent conditioning often determines whether compressed-latent methods match pixel-based baselines.

Several challenges constrain current progress. First, the representation mismatch between perceptual latents and semantic embeddings remains a core hurdle; resolving it often requires additional data, compute, or co-training that may be impractical for many teams. Second, co-designing codecs and VLM adapters increases engineering complexity and reduces plug-and-play flexibility. Third, latents tuned for one downstream task may not generalize across tasks (e.g., detection versus captioning), which complicates adoption in multi-task settings. Finally, the field lacks standardized evaluation protocols that jointly report bandwidth, FLOPs, latency, and task metrics, making cross-

paper comparisons difficult.

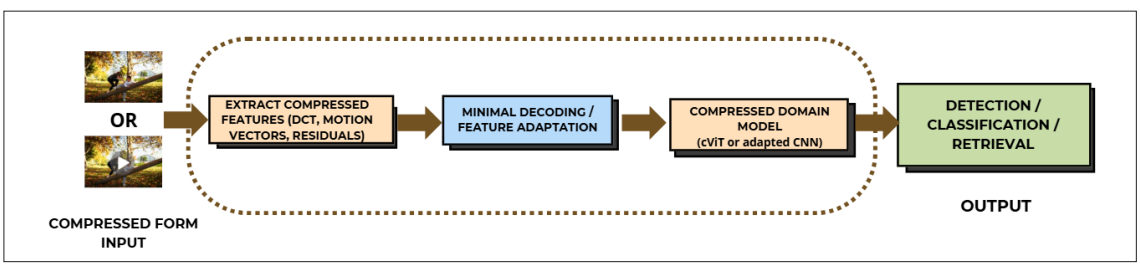
Looking forward, practical recommendations include developing lightweight adapters that align latents to VLM embeddings with minimal fine-tuning, pursuing co-design where feasible to jointly optimize latent objectives and downstream losses, and incorporating task-aware conditioning (e.g., textual prompts or labels) during encoding to bias latents toward preserving relevant semantics [7, 8]. Researchers should also evaluate compressed-latent pipelines across diverse multimodal benchmarks and bitrates while reporting a consistent set of metrics like bandwidth, latency, FLOPs, memory, and downstream accuracy to clarify real-world trade-offs.

In summary, compressed-latent inputs integrate modern neural codec advances into the first stage of the VLM pipeline, offering a compact and bandwidth-efficient alternative to pixel-based processing. When adapters or jointly optimized codecs align latents with multimodal semantics, these pipelines can yield substantial practical gains in memory, bandwidth, and compute without large sacrifices in accuracy. Continued progress will depend on improved co-design practices, standardized evaluation, and methods that explicitly prioritize semantic alignment between codec latents and VLM embeddings [7, 20, 8, 21].

3. Compressed-Domain Inference

Compressed-domain inference represents a com-plementary approach to efficiency in VLM pipelines, wherein models operate directly on compressed representations such as JPEG frequency coefficients or video bitstreams rather than fully reconstructed pixel images. By bypassing pixel decoding, these methods exploit codec-native structures that already capture essential perceptual information, thereby reducing preprocessing cost, memory usage, and latency during multimodal inference.

Figure 4 illustrates that the system extracts compressed-domain features from encoded visual data (e.g., JPEG DCT coefficients or video motion vectors) and adapts them into a representation compatible with the VLM. These adapted features are then fed directly into the multimodal model for tasks such as captioning, retrieval, or visual question answering. The motivation for compressed-domain inference is clear: decoding inputs at the pixel level introduces both computational overhead and redundancy, whereas compressed representations already encode frequency and structural



<Figure 4> General Pipeline of Compressed-Domain Inference

cues sufficient for higher-level reasoning.

Several studies have demonstrated the practicality of this idea. Park and Johnson proposed *RGB No More*, a vision transformer that bypasses pixel reconstruction by consuming JPEG DCT coefficients directly [12]. Their result shows that models operating in the compressed domain can match pixel-based accuracy while reducing data loading and preprocessing cost. Extending this direction, Deng and Karam presented compressed-domain recognition pipelines that employ feature adaptation layers, achieving performance close to pixel-domain baselines at significantly lower computational expense [2]. Together, these works establish compressed-domain inference as a viable and effective route toward efficient multimodal processing.

Empirical findings reinforce these advantages. By eliminating full pixel decoding, compressed-domain methods lower data transfer, memory usage, and preprocessing latency which are key factors for large-scale and real-time VLM applications. Importantly, compressed features such as DCT coefficients or motion vectors highlight structural and perceptual patterns that align with the semantics required for recognition and reasoning [12, 2].

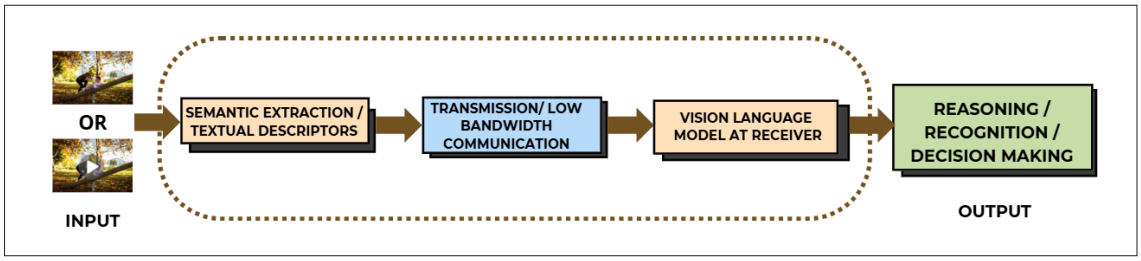
At the same time, this approach faces notable challenges. First, compressed features are tied to specific codecs: models trained on JPEG latents may not generalize well to HEVC/H.265 or VVC/H.266 without adaptation. Second, there is a mismatch between codec-derived features and the embedding spaces expected by VLMs, making adapter layers or joint training necessary. Third, although compressed inputs reduce cost, they may omit subtle semantic details essential for fine-grained reasoning, creating trade-offs for tasks demanding precise grounding. Finally, the absence of standardized benchmarks and consistent reporting complicates fair evaluation across codecs and tasks.

Looking forward, promising directions may include developing codec-agnostic adapters that enable cross-format generalization, co-optimizing compressed features with multimodal objectives to close representation gaps, and adopting evaluation protocols that jointly report efficiency metrics (e.g., FLOPs, bandwidth, latency) alongside accuracy. Embedding compressed-domain inference into larger semantic or adaptive compression pipelines also appears valuable, allowing systems to flexibly switch between pixel, latent, and compressed-domain processing depending on the task at hand.

In short, compressed-domain inference rethinks the preprocessing stage of VLMs by operating directly on frequency- or block-level features inherent in visual codecs. By bypassing pixel decoding, these methods achieve meaningful savings in computation and latency while preserving sufficient semantic structure for downstream reasoning. Future work lies in improving cross-codec generalization, aligning compressed features with multimodal embeddings, and establishing standardized evaluations to fully realize the potential of this efficiency-oriented paradigm [12, 2].

4. Semantic Compression

Semantic compression advances the efficiency of VLM pipelines by focusing on transmitting or stor-



<Figure 5> General Pipeline of Semantic Compression

ing only the task-relevant information contained in an image, rather than the full pixel-level detail. Unlike traditional codecs, which aim to minimize pixel distortion (e.g., PSNR or SSIM), semantic compression prioritizes preserving meaning-rich features such as object attributes, relationships, or multimodal descriptors. This shift reframes compression not as a low-level preprocessing step, but as an active enabler of multimodal reasoning under bandwidth- or resource-constrained environments.

Figure 5 illustrates that the visual inputs are first processed by a semantic-aware encoder that identifies the most critical information for downstream tasks. These compact semantic representations, ranging from symbolic descriptors to cross-modal embeddings are then transmitted or stored. On the receiver or server side, a decoder-or the VLM itself-directly leverages these representations to perform tasks including question answering, retrieval, and caption generation. The guiding motivation is that reasoning tasks rarely require the full pixel grid; instead, maintaining high-level semantics suffices for accurate performance.

Recent work is shifting compression from pixel reproduction toward preserving meaning. Jiang et al. introduce VLM-CSC, a cross-modal semantic communication system where a vision-language model extracts compact semantic descriptors for transmission instead of raw or densely processed images [6]. By sending only semantically aligned information drawn from a vision-language knowledge base, their pipeline achieves large bitrate reductions while keeping downstream reasoning performance stable. Qin et al. take a similar direction with perceptual image compression that is guided by cross-modal side information: brief textual cues steer the codec to preserve features most relevant for interpretation, so compressed outputs retain the original content's semantic intent even at aggressive bitrates [13].

Results from these studies indicate real practical gains: semantic-aware methods often cut bitrate substantially compared with pixel-centric codecs while maintaining task accuracy. For example, transmitting semantic descriptors alone can markedly lower bandwidth requirements with little loss in reasoning quality, and text-guided compression has been shown to boost perceptual quality at fixed bitrates. These findings suggest that, for multimodal reasoning, keeping the meaning matters more than reproducing every pixel.

However, several challenges remain. A major issue is task specificity: a codec tailored for cap-

tioning might drop cues that a detector or retriever needs. Defining “relevant semantics” is therefore nontrivial, since what matters depends on the task, dataset, and user objective. Deployment commonly requires VLMs in the training or inference loop, which raises system complexity and cost. Evaluation practices also lag: most papers report bitrate and task accuracy but rarely measure latency, energy consumption, or cross-task robustness.

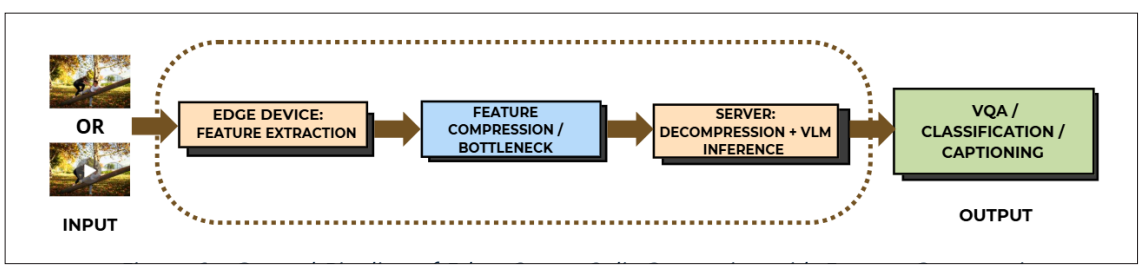
Future progress will likely depend on adaptive systems that can identify task-relevant features at runtime and codecs co-designed with multimodal objectives. Unified benchmarks that jointly evaluate bitrate, compute cost, latency, and downstream performance would enable fair comparisons. Deeper integration of VLM signals, for example attention maps or language cues, into the compression pipeline is another promising path for allocating bits according to semantic priority.

In short, semantic compression reframes the goal from pixel fidelity to preserving the information that matters for downstream reasoning and human understanding. By transmitting or storing only the semantics needed for reasoning, these methods redefine compression as a core component of multimodal intelligence. Looking forward, semantic compression is poised to play a pivotal role in scaling VLMs to real-world, bandwidth-constrained applications while ensuring that efficiency does not come at the cost of semantic grounding [6, 13].

5. Edge-Server Split Computing with Feature Compression

This technique has emerged as a practical paradigm for scaling VLMs to real-world environments where edge devices, such as smartphones, drones, or IoT sensors, operate under tight resource constraints. Instead of transmitting raw images or videos to the cloud, edge devices extract compact feature embeddings, compress them, and send only the reduced representations to powerful server-side models. The cloud server then reconstructs or adapts these features into the VLM’s embedding space, enabling multimodal reasoning with lower bandwidth, reduced latency, and improved privacy.

Figure 6 illustrates that the edge device performs lightweight encoding to generate intermediate features, applies compression to reduce communication load, and transmits these features over the network. The server, equipped with the full VLM, processes the received features for tasks such as captioning, question answering, or retrieval. This



<Figure 6> General Pipeline of Edge-Server Split Computing with Feature Compression

design balances the computational load between edge and cloud, minimizing communication overhead while still leveraging the full reasoning capacity of large-scale models.

Recent research underscores the effectiveness of this strategy. Shinde et al. emphasize that feature compression and runtime optimization are central to extending VLM deployment into mobile and IoT contexts, where bandwidth and compute are scarce [16]. Matsubara et al. demonstrate that learned compression modules for edge features can outperform conventional image transmission, achieving better rate-distortion trade-offs and end-to-end latency by co-designing encoders with task objectives [24]. Complementary approaches, such as FrankenSplit, propose shallow bottleneck modules for compressing neural features at the split point, showing that compact embeddings can substantially reduce communication without heavily degrading downstream accuracy [25].

Empirical evaluations validate the advantages of split computing pipelines. Compared to transmitting raw pixel data, compressed feature transmission reduces bandwidth consumption by orders of magnitude while maintaining accuracy across visual recognition and multimodal reasoning tasks [24, 25]. Furthermore, carefully designed feature encoders trained with distillation or task-specific objectives can achieve semantic fidelity sufficient for robust reasoning, even at very low bitrates. In latency-sensitive applications, such as robotics or autonomous driving, edge-server split frameworks also reduce end-to-end delays by limiting the size of transmitted data.

Nonetheless, challenges remain for widespread adoption. First, designing general-purpose compressed features that perform well across diverse multimodal tasks is nontrivial, as features tuned for one task (e.g., classification) may lack information needed for another (e.g., dense captioning). Second, variable network conditions can sharply reduce effectiveness when a compression method cannot adapt to changing bandwidth. Third, privacy remains a concern: compressed embeddings may still reveal sensitive visual content unless they are properly protected. Finally, tightly co-designing edge encoders with server-side VLMs raises system complexity and undermines modularity and plug-and-play deployment.

Future research should therefore emphasize adaptive feature-compression schemes that reallocate bitrate on the fly according to task priorities and network fluctuations. Equally critical are privacy-preserving solutions-for example, feature obfuscation, differential privacy, or encrypted transmission-that retain utility while limiting information leakage. Developing codec-agnostic intermediate representations that generalize across tasks and heterogeneous server setups would also improve flexibility and interoperability. To evaluate these trade-offs in realistic settings, we need standardized benchmarks that report not only accuracy but also bitrate, latency, and energy use.

In short, edge-server split computing with feature compression offers a practical, scalable route for running VLMs under tight bandwidth and latency constraints: sending compact, semantically rich representations reduces communication demands while preserving multimodal reasoning capabilities [16, 24, 25].

6. VLM-Guided Image Compression

VLM-guided image compression treats a vision-language model as an active participant in the encoding pipeline. Rather than compressing every pixel uniformly, these methods steer bit allocation toward regions that are semantically important that are identified by language prompts, attention maps, or task objectives while compressing less relevant areas more aggressively. In this way, low-level encoding choices are aligned with high-level reasoning priorities.

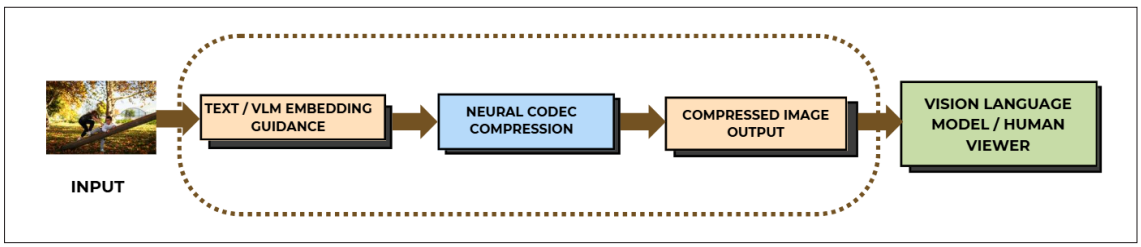
Figure 7 illustrates that the visual data is encoded by a neural codec that incorporates signals from a VLM, such as textual prompts, attention maps, or task objectives, to guide bit allocation. Regions critical for downstream reasoning (e.g., objects mentioned in a question or caption) are preserved with higher fidelity, while background or less relevant areas are compressed at lower quality. The resulting compressed bitstream is then transmitted or stored, with the assurance that task-relevant semantics remain intact.

Recent works illustrate the promise of this approach. Lee et al. propose TACO, a text-guided neural codec in which short textual instructions encourage the encoder to preserve semantically important regions, improving both pixel fidelity and downstream robustness [8]. Qin et al. pursue a related idea by using cross-modal side information, textual cues that bias the codec toward features important for human or machine interpretation, showing that semantics can be preserved even at low bitrates [13].

Empirical results support VLM guidance: compared with conventional codecs optimized only for distortion metrics (e.g., PSNR, SSIM), VLM-guided codecs tend to achieve better task performance at the same or lower bitrates. Tasks such as visual question answering and captioning particularly benefit when fine details in task-relevant regions are retained despite overall bitrate reductions. This evidence suggests that semantic priority, not raw pixel reproduction, often determines success in multimodal reasoning.

Challenges remain. Policies tuned to a specific prompt or task may not generalize to new tasks. Moreover, integrating VLMs into the codec introduces additional computational overhead, raising concerns about deployment in latency-sensitive or resource-constrained environments.

Future research may include developing lightweight VLM-guidance mechanisms that minimize overhead while providing strong semantic align-



<Figure 7> General Pipeline of VLM-Guided Image Compression

ment, as well as exploring adaptive guidance that tailors compression policies dynamically to different tasks or prompts. Research into joint optimization frameworks, where codecs and VLMs are co-trained to balance efficiency and semantics, could further enhance performance. Additionally, expanding evaluation metrics to account for semantic fidelity will enable more meaningful comparisons across approaches.

In summary, VLM-guided image compression exemplifies the bidirectional synergy between multimodal reasoning and efficient encoding. By allowing VLMs to guide bit allocation, these methods ensure that compression prioritizes semantics over pixel fidelity, advancing toward intelligent, task-aware pipelines. While challenges of generalization, overhead, and evaluation remain, VLM-guided compression is poised to become a cornerstone of future multimodal systems where perception, communication, and reasoning are tightly integrated [8, 13].

IV. Conclusion

The intersection of VLMs and compression reframes efficiency as the preservation of semantic fidelity under constrained resources. Our survey organized the literature into six complementary categories-token compression, compressed-latent inputs, compressed-domain inference, semantic compression, edge-server split computing, and VLM-guided image compression-and highlighted representative advances and tradeoffs. While many methods achieve impressive reductions in FLOPs, bandwidth, or storage, open challenges remain: ensuring robustness to over-pruning, maintaining cross-task generalization for semantic codecs, defending privacy when compressing features, and designing benchmarks that measure semantic preservation beyond pixel fidelity. We believe the next wave of progress will come from co-design: jointly optimizing codec objectives, feature representations, and VLM architectures so that compression is aware of, and tailored to, downstream reasoning goals. To enable reproducible progress, the field needs clearer evaluation protocols such as task-aware metrics and cross-task tests, larger public baselines for VLM with compression, and stronger attention to privacy and secure feature transmission. Together these directions will help produce semantically faithful, efficient multimodal systems ready for real-world use.

References

[1] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds, R. Ring, E. Rutherford, S. Cabi, T. Han, Z. Gong, S. Samangooei, M. Monteiro, J. Menick, S. Borgeaud, A. Brock, A. Nematzadeh, S. Sharifzadeh, M. Binkowski, R. Barreira, O. Vinyals, K. Simonyan, and A. Zisserman, "Flamingo: A visual language model for few-shot learning," NeurIPS, 2022. arXiv:2204.14198.

[2] Y. Deng and L. J. Karam, "DNN-compressed domain visual recognition with feature adaptation," arXiv preprint arXiv:2305.08000, 2023.

References

[3] S. Han, H. Mao, and W. J. Dally, “Deep compression: Compressing deep neural networks with pruning, trained quantization and Huffman coding,” presented at ICLR, 2016. arXiv:1510.00149.

[4] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. G. Howard, and H. Adam, “Quantization and training of neural networks for efficient integer-arithmetic-only inference,” in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 2704-2713, doi: https://doi.org/10.1109/CVPR.2018.00286.

[5] C. Jia, Y. Yang, Y. Xia, Y.-T. Chen, Z. Parekh, H. Pham, Q. V. Le, Y. Sung, Z. Li, and T. Duerig, “Scaling up visual and vision-language representation learning with noisy text supervision,” in Proc. Int. Conf. on Machine Learning (ICML), 2021. arXiv:2102.05918.

[6] F. Jiang, C. Tang, L. Dong, K. Wang, K. Yang, and C. Pan, “Visual language model-based cross-modal semantic communication systems (VLM-CSC),” arXiv preprint arXiv:2407.00020, 2024.

[7] C.-H. Kao, C. Chien, Y.-J. Tseng, Y.-H. Chen, A. Gnutti, S.-Y. Lo, W.-H. Peng, and R. Leonardi, “Bridging compressed image latents and multimodal large language models,” arXiv preprint arXiv:2407.19651, 2024.

[8] H. Lee, M. Kim, J.-H. Kim, S. Kim, D. Oh, and J. Lee, “Neural image compression with text-guided encoding for both pixel-level and perceptual fidelity (TACO),” ICML / arXiv:2403.02944, 2024.

[9] J. Li, D. Li, S. Savarese, and S. C. H. Hoi, “BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in Proc. Int. Conf. on Machine Learning (ICML), 2023. arXiv:2301.12597.

[10] H.-T. Ho, L. V. Nguyen, M.-T. Pham, Q.-H. Pham, Q.-D. Tran, N. M. H. Duong, and T.-H. Nguyen, “A review on vision-language-based approaches: Challenges and applications,” Computers, Materials & Continua (CMC), vol. 82, no. 2, pp. 1733-1756, 2025, doi: https://doi.org/10.32604/cmc.2025.060363.

[11] OpenAI, “GPT-4V(ision) system card,” OpenAI Technical Report, 25 Sep. 2023. [Online]. Available: https://cdn.openai.com/papers/GPTV_System_Card.pdf.

[12] J. Park and J. Johnson, “RGB no more: Minimally-decoded JPEG Vision Transformers,” in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2023.

[13] S. Qin, B. Chen, Y. Huang, B. An, T. Dai, and S.-T. Xia, “Perceptual image compression with cooperative cross-modal side information,” arXiv preprint arXiv:2311.13847, 2023.

[14] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, and I. Sutskever, “Learning transferable visual models from natural language supervision (CLIP),” in Proc. Int. Conf. on Machine Learning (ICML), 2021. arXiv:2103.00020.

[15] Y. Rao, W. Zhao, B. Liu, J. Lu, J. Zhou, and C.-J. Hsieh, “DynamicViT: Efficient vision transformers with dynamic token sparsification,” NeurIPS, 2021. arXiv:2106.02034.

[16] G. Shinde, A. Ravi, E. Dey, S. Sakib, M. Rampure, and N. Roy, “A survey on efficient vision-language models,” arXiv preprint arXiv:2504.09724, 2025.

[17] K. Wang and H. Xuan, “LLaVA-Zip: Adaptive visual token compression with intrinsic image information,” arXiv preprint arXiv:2412.08771, 2024.

[18] X. Huang, H. Zhou, and K. Han, “PruneVid: Visual token pruning for efficient video large language models,” Findings of the Association for Computational Linguistics (ACL), pp. 19959-19973, 2025.

[19] S. R. Alvar, G. Singh, M. Akbari, and Y. Zhang, “DivPrune: Diversity-based visual token pruning for large multimodal models,” in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2025. arXiv:2503.02175.

[20] Q. Yu, M. Weber, X. Deng, X. Shen, D. Cremers, and L.-C. Chen, “An image is worth 32 tokens for reconstruction and generation (TiTok),” NeurIPS, 2024. arXiv:2406.07550.

[21] Daixiangzi, “Awesome-Token-Compress,” GitHub repository, 2024. [Online]. Available: https://github.com/daixiangzi/Awesome-Token-Compress.

[22] S. D. Liang et al., “Vision-Language Models for Massive MIMO Semantic Communication,” OpenReview, 2024. [Online]. Available: https://openreview.net/forum?id=IvZrYKLBzH.

[23] J. Xu, S. Wang, J. Chen, Z. Li, P. Jia, F. Zhao, G. Xiang, Z. Hao, S. Zhang, and X. Xie, “Decouple distortion from perception: Region adaptive diffusion for extreme-low bitrate perception image compression (MRIDC),” in Proc. IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR), 2025.

[24] Y. Matsubara, R. Yang, M. Levorato, and S. Mandt, “Supervised compression for resource-constrained edge computing systems,” in Proc. IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 2022.

[25] A. Furutanpey, P. Raith, and S. Dustdar, “FrankenSplit: Efficient neural feature compression with shallow variational bottleneck injection for mobile edge computing,” arXiv preprint arXiv:2302.10681, 2023.

Authors

Mubashir Hussain Shah

- 2020년 ~ 2024년 : National University of Sciences and Technology Pakistan, 소프트웨어공학 학사
- 2025년 ~ 현재 : 경희대학교 전자정보융합공학과 석사과정
- 주관심분야 : 영상처리, 비디오압축, 딥러닝

Kiho Choi

- 2008년 : 한양대학교 정보통신대학 미디어공학과 학사
- 2012년 : 한양대학교 전자컴퓨터통신공학과 박사
- 2012년 ~ 2014년 : 한양대학교 부설연구소 Post Doc.
- 2014년 ~ 2021년 : 삼성전자 삼성리서치 책임연구원
- 2021년 ~ 2023년 : 가천대학교 AI·소프트웨어학부 조교수
- 2023년 ~ 현재 : 경희대학교 전자공학과 조교수
- 주관심분야 : 영상처리, 비디오압축, 딥러닝