Neural Implicit Representations for Object-centric Machine Vision Tasks

Yeoneui Kim and Je-Won Kang
Department of Electronic and Electrical Engineering, Ewha Womans University, South Korea
E-mail: {yeoneui4477, jewonk}@ewha.ac.kr

***Abstract*—Video Coding for Machines (VCM) has become increasingly important with the rapid growth of video data and the rising demand for machine vision systems that require high accuracy under low-bitrate constraints. Implicit Neural Representations (INRs), known for their compactness and strong compression ability, offer a promising direction. In this paper, we propose a novel INR framework tailored for machine vision. Unlike prior INR methods focused on human perception, our model embeds object-aware information into the neural network. Specifically, decoder weights are dynamically generated using a dynamic network conditioned on object-centric latent codes, enabling object-aware reconstruction. Experiments show consistent improvements across various INR baselines in both video compression and machine vision tasks. Our method achieves up to 6% higher accuracy and over 60% bitrate reduction in object detection compared to baselines, even outperforming recent standard codecs in the low-bitrate regime. For semantic segmentation, it also yields over 6% accuracy gains and notable bitrate savings, demonstrating its effectiveness for VCM.**

I. Introduction

As machine vision systems, such as surveillance camera systems and autonomous agents, require large-scale video data, video coding for machines (VCM) has emerged to optimize storage and computational efficiency. For this, conventional video codecs have been adapted for machine vision-centric tasks such as object detection and tracking, to reduce bit-rates while maintaining task performance [1]. Recently, implicit neural representation (INR) has attracted significant research attention as an alternative video representation technique, offering a compact and flexible proxy for the original video signal [2], [3]. In this paper, we introduce a novel VCM framework that newly applies INR models tailored to machine vision tasks.

In INR-based video modeling, a video signal is represented as a continuous function parameterized by a deep neural network [4]. Unlike conventional video representation, which directly manages discrete samples, the INR framework learns an encoder and decoder to predict RGB values from input coordinates. Existing video INR models have demonstrated competitive performance compared to conventional video processing techniques [2], [3], [5]. However, they have focused on pixel-level reconstruction and enhancement for the human visual system (HVS) [6]. Fig. 1 (a) presents existing INR approaches, in which a video signal v_t is modeled using a decoder f_θ. The parameter set θ is trained to minimize the HVS-related metrics such as the structural similarity index measure (SSIM) in addition to mean squared error (MSE) for the pixel-level reconstruction.

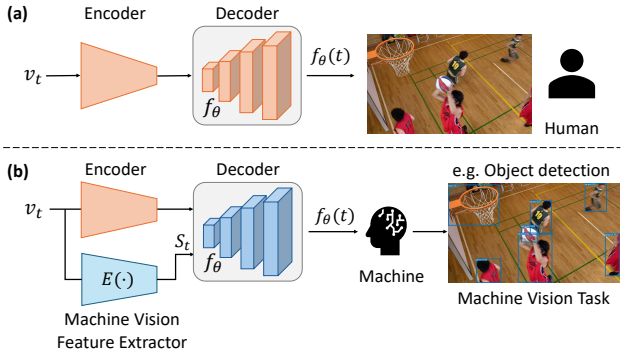


Fig. 1: Video INR models employed in (a) video coding for human perception and (b) video coding for machine analysis.

While conventional INR-based methods focused on pixel-level fidelity and perceptual quality models, they have neglected the semantic or structural information required by tasks such as object detection or segmentation. As a result, HVS-oriented INR models often underperform in VCM settings, highlighting the need for new designs tailored to machine vision objectives. Yet, despite this growing demand, INR-based approaches for machine vision remain relatively limited. SA-NeRV [7] introduced a cost function adapted for object tracking and segmentation. In another line of work, INR features were aligned with multi-modal semantics for tasks such as video retrieval and video-language understanding [8], although their approach have not directly addressed the core challenges of VCM.

In this paper, we propose a novel INR approach designed for machine vision systems for VCM. INRs for machine vision should capture semantic and structural features, prioritizing key object shapes and details for accuracy [9]. For this, as illustrated in Fig. 1 (b), video INR for machine vision requires a different network architecture since its objective differs significantly from that of HVS. To meet these objectives, we propose a novel INR framework for machine vision, consisting of two key components: an object-centric latent extractor

*This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program (No. IITP-2025-RS-2020-II201460) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation).

and an object-adaptive weight generator. The latent extractor generates an object-centric latent embedding per frame using an off-the-shelf object detection model [10]. Conditioned by an object-centric latent embedding, the weight generator utilizes a dynamic neural network to generate temporally modulated object-adaptive weights for the decoder. In our implementation, we employed two recent video INR models as baselines to evaluate the generalizability of the proposed approach. Our experiments demonstrate that our approach improves both compression and downstream vision task performance, highlighting its potential for machine-oriented video representation.

II. Related Works

A. Implicit Neural Representation

The INR has demonstrated compact and powerful representations for various video processing tasks [2], [5]. While early INRs used coordinate-based methods, a neural representation for video (NeRV) model [4] introduced an image-wise representation, modeling a video as a function of time. The NeRV decoder comprises a sequence of NeRV blocks, each consisting of a convolutional layer, a pixel shuffle layer, and an activation layer, delivering higher quality and faster decoding compared to MLP-based networks.

HNeRV[3] employed a content-adaptive frame embedding to preserve spatial contexts of input frames in addition to a learnable decoder. Several studies [2], [5], [11] have attempted to enhance the speed, quality of regression tasks, and compression performance. We integrate our proposed modules into the NeRV-series framework, extending its capability toward object-centric representation and machine vision applications.

B. Video Coding for Machine

The VCM aims to maximize performance on vision tasks while minimizing bitrates [9]. VCM can be addressed through three main approaches [1], [9]. The first approach involves efficient video compression followed by decoding analytics-friendly videos. In this approach, a task-specific network inputs the decoded video and predicts results. Alternatively, extracted features can be compressed, transmitted, and decoded for machine analysis. Finally, a hybrid method that supports human and machine vision employs decoded videos for human vision and can use the decoded video or features for machine analysis. We focus on machine vision-targeted videos that machine vision systems will consume.

A few early efforts have explored INR for machine vision analysis. SA-NeRV [7] reconstructed machine vision-targeted videos using edge masks [12] in the loss function to embed the position and shape of objects. However, relying on an external edge detector and segmentation model introduces significant computational overhead. Furthermore, the edge mask is highly dependent on the reconstruction quality, which can result in unstable performance. Another study [8] aimed to embed semantic meaning in video INR by aligning video features with embeddings from large multi-modal foundation models. The encoded representation could be used for multi-modal downstream tasks such as video retrieval and video chat without video decoding.

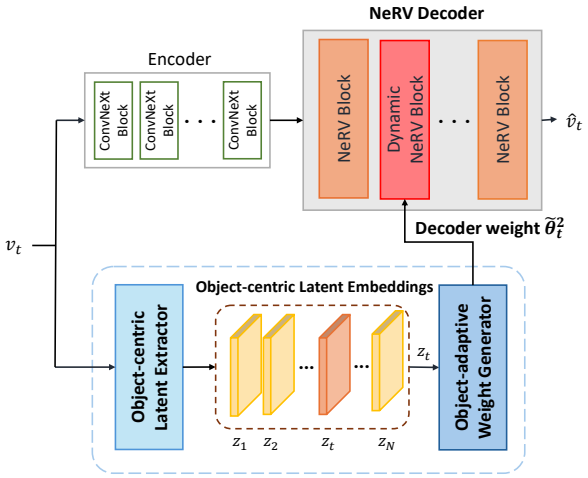


Fig. 2: Overall architecture of the proposed INR for machine vision, including object-centric latent extractor and object-adaptive weight generator. To verify that our method is broadly applicable, we integrate the key modules into different INR models [2], [3] and evaluate the improved performance.

C. Dynamic Neural Networks

Dynamic neural networks can adaptively modulate the network structures or parameters with varying inputs [13]. Dynamic architectures achieve computational efficiency by selectively activating model components based on the input, without compromising representation capability. To achieve this, the modulation parameters or feature-conditioned inputs are kept compact and computationally efficient as side information. In previous studies, dynamic parameters can be directly generated with an independent model with learnable embeddings [14] or intermediate features [15] as input. By employing task-specific information, dynamic neural networks can generate task-aware dynamic weights [16]. Such task-aware weight prediction has demonstrated both data efficiency and effective performance. Building on this foundation, we utilize dynamic neural networks to predict object-adaptive weights, enabling our model to focus on object-aware reconstruction.

III. Method

A. Model Overview

Our framework introduces two key components designed to support object-aware video reconstruction: an object-centric latent extractor (OLE) and an object-adaptive weight generator (OWG). These modules are modular by design and compatible with NeRV-series INR models. We implement the proposed method using two recent video INR models [2], [3] as baselines and evaluate the performance on various vision tasks in comparison with the baseline models and conventional VCM methods.

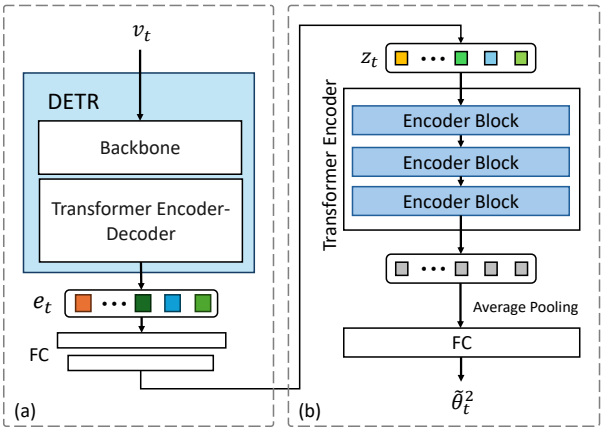


Fig. 3: (a) Object-centric latent generator (b) Object-adaptive weight generator.

We extract content-adaptive frame embeddings from a convolutional neural network (CNN) and feed them into a NeRV decoder as in [2], [3]. However, unlike the baseline models, to incorporate object-level information, we then introduce the OLE, which generates an object-centric latent embedding z_t for each frame. This embedding captures object-level semantics in the video. Conditioned on z_t, the OWG dynamically produces a set of decoder weights θ_t^l, enabling the decoder to adapt its parameters over time t in an object-aware manner. An overview of our model is shown in Fig. 2.

B. Object-centric Latent Extractor

In the OLE, z_t, which is an object-centric latent embedding, is generated using a DETR model [10], conducting object detection based on a transformer. In a DETR decoder, the embedding derived from object queries and self-attention captures object relationships and global context. Later, this embedding is used to predict the class and bounding box in the detection head. Therefore, as presented in Fig. 3, the DETR embedding e_t can provide useful information about objects and regions to the baseline network, enabling the INR to reconstruct the target video with a greater focus on the object parts.

Rather than directly employing the decoder embeddings, we use a fully connected (FC) layer to reduce their dimensions and decrease the computational cost of weight generation. As illustrated in Fig. 3 (a), e_t is projected to z_t via the FC layer and delivered to the OWG.

C. Object-adaptive Weight Generator

The OWG generates per-frame object-adaptive weights from z_t by using a dynamic neural network and applies the changes to the convolution layers in the INR decoder. $\theta_t^i \in \theta_t$ denotes the weight parameters of the convolution layers in the i-th NeRV block of the decoder. Then, the weight parameters are updated as follows:

$$\tilde{\theta}_t^2 = g(z_t), \tag{1}$$

where $g(\cdot)$ denotes a dynamic neural network. We apply weight modulation only to the second NeRV block, referred to as the dynamic NeRV block in Fig. 2, to reduce computational complexity and avoid unstable training.

Injecting object-centric modulation weights into the INR decoder enables frame-wise enhancement of object-specific features during the decoding process. The decoder produces a machine vision-targeted video frame $\hat{v}_t$ as follows:

$$\hat{v}_t = f_\theta(t; \tilde{\theta}_t), \tag{2}$$

where $f_\theta(\cdot)$ is the neural decoder, and $\tilde{\theta}_t = \{\theta_t^1, \tilde{\theta}_t^2, \cdots, \theta_t^M\}$. M is the total number of NeRV blocks.

Specifically, the generator consists of a transformer encoder and an FC layer. The transformer encoder first aggregates object information from z_t using multi-head attention [17]. We then apply average pooling over the output tokens to obtain a global representation, which is subsequently processed by an FC layer to generate θ_t.

D. Optimization

Our objective is to reconstruct videos while maintaining useful features for objects. We use a reconstruction loss L_r for training. L_r is used to minimize the distortion and ensure the preservation of visual features, given as

$$L_r = \frac{1}{N}\sum_{t=1}^{N} \parallel v_t - \hat{v}_t \parallel_2^2, \tag{3}$$

where v_t and $\hat{v}_t$ are the ground truth (GT) and the reconstructed frames, respectively. N is the number of frames.

E. Model Architecture and Compression

For the HNeRV baseline, we use a CNN encoder to extract tiny frame embeddings. For a video resolution of 480×800, we set the stride list to (5,4,2,2,2), resulting in a small spatial size (e.g., 3×5). The neural decoder consists of five NeRV blocks, each progressively reducing the channel width by a factor of 1.2.

We leverage model compression and embedding quantization for video compression. The model weights and embeddings are each quantized by a quantization factor. The weight assigned to the dynamic NeRV block is also quantized in this process. After quantization, we apply entropy encoding to reduce the size further. Specifically, we leverage Huffman coding [18] for lossless compression.

IV. Experimental Results

A. Datasets and Settings

We evaluate the effectiveness of our framework on machine vision tasks, specifically object detection and semantic segmentation, using decoded video sequences.

Evaluation Metrics For object detection, we use the SFU-HW-Objects-v1 dataset [19], which provides bounding boxes and object class labels for High Efficiency Video Coding (HEVC) [20] v1 Common Test Conditions (CTC) video sequences. Experiments are conducted on four Class C videos (832×480). Detection accuracy is evaluated using a pre-trained

YOLOv7 model [21], reporting mAP@50 (mean AP at IoU 50%) and mAP@50:95 (mean AP across IoUs from 50% to 95%) across different bitrates.

For segmentation, we use a subset of 10 VSPW videos [22], remapping labels to align with Pascal VOC classes. A DeepLabv3 model (ResNet-50 backbone) [23], pre-trained on Pascal VOC, is used for evaluation. We report the mean pixel accuracy (Acc.) and the mean IoU (mIoU) across varying bitrates for evaluation.

Furthermore, the Bjøntegaard Delta (BD) [24] rate and performance are measured relative to each baseline, providing a quantitative measure of compression efficiency.

Training details. All models are trained for 200 epochs using the Adam optimizer, with beta as (0.9, 0.999), weight decay as 0, and with an initial learning rate of 0.001 and a cosine decay schedule.

Tested methods and configuration. We conduct experiments on two baseline models: HNeRV [3] and SNeRV [2]. To ensure fair evaluation, we compare the baseline models, their variants integrated with our proposed method, and a recent INR-based approach [7] with equivalent model sizes. Here, model size is defined as the total number of parameters in the frame embedding and decoder.

For compression, we apply 8-bit quantization to model weights and 6-bit quantization to embeddings. The total bitrate is computed based on quantized parameters and expressed in bits per pixel (bpp).

B. Performance Evaluation

1) Video Object Detection: Table I presents the quantitative results of INR-based methods with a model size of 1.0 M. When applied to both HNeRV [3] and SNeRV [2], our method consistently improves performance, surpassing the original baselines across all sequences. These results confirm the effectiveness of our approach and its generalization capability across different baseline models. Furthermore, our method outperforms a recent INR-based method, SA-NeRV [7], demonstrating greater robustness. As illustrated in Fig. 5, our approach reconstructs finer details and preserves object boundaries more clearly than each baseline. In contrast, SA-NeRV [7] exhibits unstable reconstructions, particularly in scenes with dynamic motion, likely due to inaccurate edge mask predictions.

As shown in Table II, our method achieves a 6.44% improvement in BD-mAP@50:95 (BD_{mAP}) and a 63.19% reduction in BDR-mAP@50:95 (BDR_{mAP}), on average over two baselines, validating its effectiveness in enhancing both task performance and compression efficiency. The RD curves in Fig. 4 further support this result, showing significantly improved gains across bitrates, particularly in low bit-rates.

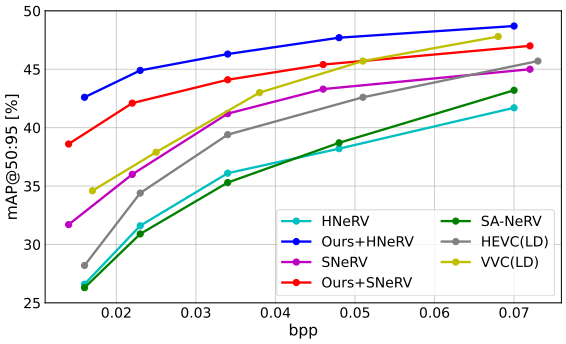
Moreover, we conducted experiments on standard codec methods under the Low-Delay (LD) configuration, as machine-vision tasks are often conducted in low-latency video streaming scenarios. INR-based approaches, including SNeRV [2] and our methods, outperform HEVC [20] and Versatile Video Coding (VVC) [25]. SNeRV was originally designed to preserve high-frequency components in video, thus improving the quality of the reconstructed frames and outperforming HNeRV in HVS. However, when our method is incorporated into both models, we observe different phenomenons. Ours+HNeRV achieves the highest overall performance across a wide range of bit-rates. SNeRV framework would be better aligned with the characteristics of the HVS.

TABLE I: Object detection(mAP@50:95) results.

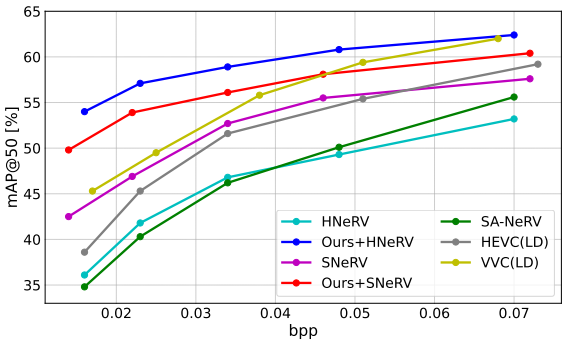
Method	BasketballDrill	BQMall	PartyScene	RaceHorses	Avg.
SA-NeRV [7]	21.72	20.65	48.05	64.52	38.74
HNeRV [3]	23.36	20.57	52.93	56.09	38.24
Ours+HNeRV	**30.61**	**32.59**	**56.90**	**70.57**	**46.30**
SNeRV [2]	28.30	26.26	54.83	63.71	43.28
Our+SNeRV	29.60	28.97	56.06	67.01	45.41

TABLE II: BD metrics for object detection across different model sizes.

Method	BD_{mAP}	BDR_{mAP}
HNeRV [3]	0.0	0.0
Ours+HNeRV	9.94	−87.23
SNeRV [2]	0.0	0.0
Our+SNeRV	2.93	−39.16
Average	**6.44**	**−63.19**



(a) Bitrate vs. mAP@50:95



(b) Bitrate vs. mAP@50

Fig. 4: RD performance curve for object detection.

Fig. 5: Qualitative comparison on BasketballDrill. Each subfigure shows the outputs from (1) Ground Truth, (2) SA-NeRV[7], (3) HNeRV[3], (4) Ours+HNeRV, (5) SNeRV[2], and (6) Ours+SNeRV.

2) Video Semantic Segmentation: Table III reports the average segmentation performance (Acc. and mIoU) of INR-based models with a model size of 0.75 M. When applied to both HNeRV [3] and SNeRV [2], our method consistently improves segmentation quality and significantly outperforms SA-NeRV [7] in both pixel accuracy (Acc.) and mIoU.

As shown in Table IV, our method achieves an average gain of 6.38% in BD-Acc. ($BD_{Acc.}$) and 7.43% in BD-mIoU (BD_{mIoU}), while reducing BDR-Acc. ($BDR_{Acc.}$) by 24.47% and BDR-mIoU (BDR_{mIoU}) by 52.28%.

We further compare our method with standard codecs under the LD configuration. While HEVC and VVC still outperform INR-based methods overall, our approach significantly narrows the performance gap. In particular, the HNeRV-based variant achieves competitive segmentation accuracy.

TABLE III: Semantic segmentation results.

Method	Acc.	mIoU
SA-NeRV [7]	66.29	60.02
HNeRV [3]	64.90	58.04
Ours+HNeRV	**72.72**	**67.87**
SNeRV [2]	68.15	60.07
Our+SNeRV	69.66	62.75

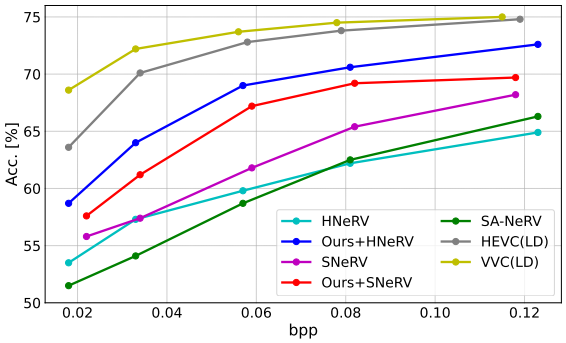
TABLE IV: BD metrics for semantic segmentation.

Method	$BD_{Acc.}$	$BDR_{Acc.}$	BD_{mIoU}	BDR_{mIoU}
HNeRV [3]	0.0	0.0	0.0	0.0
Ours+HNeRV	8.43	−71.47	9.73	−70.84
SNeRV [2]	0.0	0.0	0.0	0.0
Our+SNeRV	4.33	−22.52	5.13	−33.72
Average	**6.38**	**−24.47**	**7.43**	**−52.28**

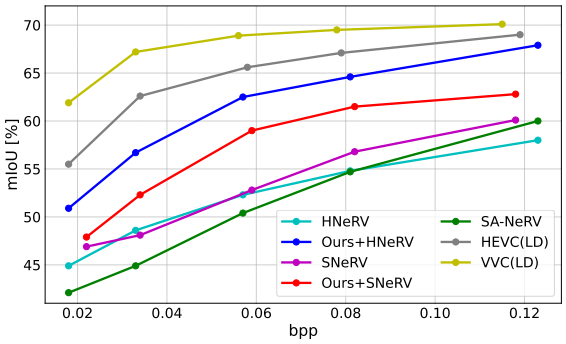
C. Ablation Study

We ablate the position of object-adaptive weight injection by modulating a single NeRV block at different stages, while keeping the total model size fixed at 1.0M under the same object-detection setup described in Section IV-B1. As shown in Table V, we denote the un-modulated baseline as OFF, and the models with our Dynamic NeRV Block (DNB) as ON, which includes early (first block), mid (second block), and late (last block) injections. All adaptive weights are generated using our proposed method and trained jointly with the decoder.

All modulation points outperform the baseline (OFF), confirming the effectiveness of object-adaptive weights. Among them, mid-block modulation yields the highest gains for both baselines [2], [3], suggesting that mid-level features offer semantically rich information within the neural decoder for machine-oriented video decoding. These results support our design choice to modulate the second NeRV block.



(a) Bitrate vs. Acc.



(b) Bitrate vs. mIoU

Fig. 6: RD performance curve for semantic segmentation.

TABLE V: Ablation study of Dynamic NeRV Block (DNB) modulation.

Method	DNB OFF	DNB ON	mAP@50:95	mAP@50
HNeRV [3]	✓	–	38.24	49.28
	–	early	43.74	55.69
	–	mid	**47.67**	**60.79**
	–	late	38.78	49.85
SNeRV [2]	✓	–	43.28	55.51
	–	early	44.65	57.28
	–	mid	**45.41**	**58.14**
	–	late	45.03	57.71

V. Conclusion

In this paper, we introduce a VCM framework based on video INRs. We transform object detection embeddings into frame-wise object-centric latents, which are used to dynamically modulate a convolutional layer of the neural decoder via a

dynamic network that generates object-adaptive weights. This dynamic modulation enables the decoder to capture object-aware information, thereby enhancing the reconstruction of semantically meaningful structures for machine vision tasks. Our framework is compatible with other NeRV series representation models, as it modulates only the convolutional weights without altering the decoder architecture. Experimental results show that our model consistently improves the efficiency and quality of video reconstruction for machine tasks across different baselines, demonstrating its flexibility and general applicability. Furthermore, it achieves superior or competitive performance compared to recent standard codecs, highlighting the potential of INR-based approaches for machine-centric video processing.

REFERENCES

[1] W. Gao, S. Liu, X. Xu, M. Rafie, Y. Zhang, and I. D. D. Curcio, "Recent standard development activities on video coding for machines," *arXiv preprint arXiv:2105.12653*, 2021.

[2] J. Kim, J. Lee, and J.-W. Kang, "Snerv: Spectra-preserving neural representation for video," in *European Conference on Computer Vision*, Springer, 2024, pp. 332–348.

[3] H. Chen, M. Gwilliam, S.-N. Lim, and A. Shrivastava, "Hnerv: A hybrid neural representation for videos," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 10 270–10 279.

[4] H. Chen, B. He, H. Wang, Y. Ren, S. N. Lim, and A. Shrivastava, "Nerv: Neural representations for videos," in *Advances in Neural Information Processing Systems*, vol. 34, 2021, pp. 21 557–21 568.

[5] H. M. Kwan, G. Gao, F. Zhang, A. Gower, and D. Bull, "Hinerv: Video compression with hierarchical encoding-based neural representation," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[6] M. Lee, H. Song, J. Park, *et al.*, "Overview of versatile video coding (h. 266/vvc) and its coding performance analysis," *IEIE Transactions on Smart Processing & Computing*, vol. 12, no. 2, pp. 122–154, 2023.

[7] T. Shindo, K. Yamada, T. Watanabe, and H. Watanabe, "Image coding for machines with edge information learning using segment anything," in *2024 IEEE International Conference on Image Processing (ICIP)*, 2024, pp. 3702–3708.

[8] S. R. Maiya, A. Gupta, M. Gwilliam, M. Ehrlich, and A. Shrivastava, "Latent-inr: A flexible framework for implicit representations of videos with discriminative semantics," in *European Conference on Computer Vision*, 2024, pp. 285–302.

[9] W. Yang, H. Huang, Y. Hu, L.-Y. Duan, and J. Liu, "Video coding for machines: Compact visual representation compression for intelligent collaborative analytics," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[10] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," in *European conference on computer vision*, Springer, 2020, pp. 213–229.

[11] Z. Li, M. Wang, H. Pi, K. Xu, J. Mei, and Y. Liu, "E-nerv: Expedite neural video representation with disentangled spatial-temporal context," in *European Conference on Computer Vision*, Springer, 2022, pp. 267–284.

[12] A. Kirillov, E. Mintun, N. Ravi, *et al.*, "Segment anything," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023, pp. 4015–4026.

[13] Y. Han, G. Huang, S. Song, L. Yang, H. Wang, and Y. Wang, "Dynamic neural networks: A survey," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 11, pp. 7436–7456, 2021.

[14] D. Ha, A. Dai, and Q. V. Le, "Hypernetworks," *arXiv preprint arXiv:1609.09106*, 2016.

[15] X. Jia, B. De Brabandere, T. Tuytelaars, and L. V. Gool, "Dynamic filter networks," *Advances in neural information processing systems*, vol. 29, 2016.

[16] D. Kang, D. Dhar, and A. Chan, "Incorporating side information by adaptive convolution," *Advances in neural information processing systems*, vol. 30, 2017.

[17] A. Vaswani, "Attention is all you need," *Advances in Neural Information Processing Systems*, 2017.

[18] D. A. Huffman, "A method for the construction of minimum-redundancy codes," *Proceedings of the IRE*, vol. 40, no. 9, pp. 1098–1101, 1952.

[19] H. Choi, E. Hosseini, S. Ranjbar Alvar, R. Cohen, and I. Bajić, *Sfu-hw-objects-v1: Object labelled dataset on raw video sequences*, https://doi.org/10.25314/7d8efc0a-3943-4738-b7a5-72badb04d765, 2020.

[20] V. Sze, M. Budagavi, and G. J. Sullivan, "High efficiency video coding (hevc)," in *Integrated circuit and systems, algorithms and architectures*, vol. 39, Springer, 2014, p. 40.

[21] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023.

[22] J. Miao, Y. Wei, Y. Wu, C. Liang, G. Li, and Y. Yang, "Vspw: A large-scale dataset for video scene parsing in the wild," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2021.

[23] L.-C. Chen, "Rethinking atrous convolution for semantic image segmentation," *arXiv preprint arXiv:1706.05587*, 2017.

[24] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," *ITU SG16 Doc. VCEG-M33*, 2001.

[25] B. Bross, Y.-K. Wang, Y. Ye, *et al.*, "Overview of the versatile video coding (vvc) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.