Video Coding for Machines: Compact Visual Representation Compression for Intelligent Collaborative Analytics

Wenhan Yang, *Member, IEEE*, Haofeng Huang, Yueyu Hu, *Student Member, IEEE*, Ling-Yu Duan, *Member, IEEE*, and Jiaying Liu, *Senior Member, IEEE*

***Abstract*—As an emerging research practice leveraging recent advanced AI techniques, e.g. deep models based prediction and generation, Video Coding for Machines (VCM) is committed to bridging to an extent separate research tracks of video/image compression and feature compression, and attempts to optimize compactness and efficiency jointly from a unified perspective of high accuracy machine vision and full fidelity human vision. With the rapid advances of deep feature representation and visual data compression in mind, in this paper, we summarize VCM methodology and philosophy based on existing academia and industrial efforts. The development of VCM follows a general rate-distortion optimization, and the categorization of key modules or techniques is established including feature-assisted coding, scalable coding, intermediate feature compression/optimization, and machine vision targeted codec, from broader perspectives of vision tasks, analytics resources, etc. From previous works, it is demonstrated that, although existing works attempt to reveal the nature of scalable representation in bits when dealing with machine and human vision tasks, there remains a rare study in the generality of low bit rate representation, and accordingly how to support a variety of visual analytic tasks. Therefore, we investigate a novel *visual information compression for the analytics taxonomy* problem to strengthen the capability of compact visual representations extracted from multiple tasks for visual analytics. A new perspective of task relationships versus compression is revisited. By keeping in mind the transferability among different machine vision tasks (e.g. high-level semantic and mid-level geometry-related), we aim to support multiple tasks jointly at low bit rates. In particular, to narrow the dimensionality gap between neural network generated features extracted from pixels and a variety of machine vision features/labels (e.g. scene class, segmentation labels), a codebook hyperprior is designed to compress the neural network-generated features. As demonstrated in our experiments, this new hyperprior model is expected to improve feature compression efficiency by estimating the signal entropy more accurately, which enables further investigation of the granularity of abstracting compact features among different tasks.**

***Index Terms*—Video coding for machines, analytics taxonomy, compact visual representation, multiple tasks, codebook-hyperprior.**

Manuscript received 19 October 2021; revised 8 September 2023; accepted 31 January 2024. Date of publication 20 February 2024; date of current version 5 June 2024. This work was supported in part by the National Natural Science Foundation of China under Grant 62332010 and Grant 62088102, in part by the PKU-NTU Joint Research Institute (JRI) sponsored by a donation from the Ng Teng Fong Charitable Foundation, and in part by AI Joint Lab of Future Urban Infrastructure sponsored by Fuzhou Chengtou New Infrastructure Group and Boyun Vision Company Ltd. Recommended for acceptance by W. Zuo. *(Wenhan Yang, Haofeng Huang, and Yueyu Hu are contributed equally to this work.) (Corresponding author: Jiaying Liu.)*

Wenhan Yang is with Peking University, Beijing 100871, China, and also with Peng Cheng Laboratory, Shenzhen, Guangdong 518000, China (e-mail: yangwenhan@pku.edu.cn).

Haofeng Huang, Yueyu Hu, and Jiaying Liu are with Peking University, Beijing 100871, China (e-mail: huang6013@pku.edu.cn; huyy@pku.edu.cn; liujiaying@pku.edu.cn).

Ling-Yu Duan is with the National Engineering Research Center of Visual Technology, School of Computer Science, Peking University, Beijing 100871, China, and also with Peng Cheng Laboratory, Shenzhen, Guangdong 518000, China (e-mail: lingyu@pcl.ac.cn).

Digital Object Identifier 10.1109/TPAMI.2024.3367293

I. Introduction

THE fields of computer vision and image/video compression have gained great progress in recent decades. While the former aims at crossing the semantic gap and translating image/video pixel signals into high-level semantic understanding information, such as recognition [27], [36], [52] or detection tasks [24], [50], the latter pursues a compact representation of pixel signals to improve storage and transmission efficiency. Driven by different targets, these two domains are developing separately to a large extent, and are rarely put together in discussion in the earlier researches.

In recent years, at the application end, the rapid emergence and prosperity of smart cities [23] and the Internet of Things (IoT) [74] raise the challenges to the original development route of the two domains, but also bring opportunities for their joint exploration and optimization. In the face of Big Data and massive applications, the original paradigm based on pixel signal compression [45], [55], [64] can no longer meet the requirements of efficient analysis. At the theoretical end, the fast development of deep generative and analytics models [25], [34], [41], [73], [76] has broken the bi-directional connection barrier between pixel signals and features. At the same time, the continuous development of multi-task learning [59], [72], disentangled representation learning [43], unsupervised/self-learning [40], [67] and other techniques have greatly expanded the depth and breadth of feature representation learning mechanisms. Researchers are increasingly concerned about and pursuing comprehensive performance of features in open scenarios.

Therefore, in this context, coding compression and analysis techniques for machine vision, called *video coding for machines (VCM)* [21], [70], have emerged to build an efficient joint compression and analytics framework upon the combination

of deep generative models, analytics models and coding techniques. The framework is capable of obtaining accurate, compact and generalized feature representations learned from multiple tasks in an end-to-end manner to effectively support Big Data intelligent analytics for massive diverse applications. In general, there are three paths that new VCM approaches in the recent two years have developed along. The first branch stands on the basis of image/video coding and rebuilds the codecs towards machine vision, called *Machine Vision Targeted Codec* [29], [56], [71]. For these methods, they offer analytics-friendly images/videos, which can achieve better analytics performance with low bit-rates. The second branch extends the route of dedicated feature compression to compressing deep intermediate features, including *Intermediate Feature Compression* [12], [13], [57] and *Optimization* [2], [54]. The former aims to reconstruct the pretrained deep features according to the feature fidelity constraint while the latter directly optimizes the deep intermediate feature and compression models jointly based on task-driven analytics losses. The third branch explores more collaborative operations between video and feature streams, optimizing the image/video coding efficiency towards human vision (categorized as *Feature Assisted Coding* [11], [39]) or improving the performance of both coding performance and intelligent analytics towards both human and machine visions (categorized as *Scalable Coding* [28], [60], [69]).

Besides the academic papers, efforts are made at the standardization end. In July 2019, MPEG Video Coding for Machines Ad-Hoc group began to develop the related video coding standards of "highly-efficient video compression and representation for intelligent machine-vision or hybrid machine/human-vision applications [26]. The proposals cover aspects of use cases, requirements, processing pipelines, plan for potential VCM standards, and the evaluation framework including machine-vision tasks, dataset, evaluation metrics, and anchor generation, *etc*. A proportion of proposals targets provide potential solutions of video coding for machines with compression and analytics gains. JPEG AI[1] also called for proposals of learning-based coding standards in April 2021. A single-stream compressed representation is adopted to improve both subjective quality from the human perspective, and effective performance from the perspective of machines to support a wide range of applications, including visual surveillance, autonomous vehicles and devices.

Some endeavors are put into adopting the idea of VCM into industrial practice and systems. NVidia announced a new video conferencing platform for developers called Nvidia Maxine,[2] which is claimed to solve some of the most common problems in video calls. Maxine uses NVidia's GPUs to process calls in the cloud and enhance them with the help of artificial intelligence techniques. According to the related research paper [63], a novel neural model is developed to synthesize a video of a talking head from a key frame and the motion keypoints. Facebook released their low bandwidth video-chat compression method [46] to reconstruct faces on the decoder side authentically with facial landmarks extracted at the encoder side. The method can run on iPhone 8 in real-time and allow video calling at a few kilobits per second. Aliyun's Video Cloud standards and implementation team[3] also launched an AI-aided conference video compression system that saves 40–65% bitrate compared to the latest Versatile Video Coding (VVC) standard for the same human-eye viewing quality in a lab testing scenario. In their work, the key frame is compressed by VVC while the Jacobian matrix is encoded for motion modeling. The advantages over VVC in terms of high definition and subjective quality are even more pronounced, providing more lifelike facial expressions at lower bit-rates.

Although previously mentioned works improve the intelligent analytics performance via optimizing image/video streams, visual feature representations, or both of them jointly, there is still a blank of research in the design of the low-bit-rate representations for diverse or even unseen visual analytics. Through reviewing existing VCM works, we present the necessity of modeling the novel information compression for analytics taxonomy. The formulations are provided to model the transferability among different machine vision tasks under compression conditions. The exploration of the new problem naturally leads to a novel hyperprior model that estimates the entropy of the neural network (NN)-generated features more accurately. Under the framework, we investigate a series of problems related to multi-task feature representation compression and obtain abundant insights that inspire the community and future works.

In summary, our contributions are as follows.

- We review the state-of-the-art approaches of video coding for machines with a unified generalized rate-distortion (R-D) optimization formulation. We illustrate all methods in five categories (feature assisted coding, scalable coding, intermediate feature compression/optimization, and machine vision targeted codec) and study the impact of analytics resources, approach output, supported analytics tasks, etc., on the potential VCM related techniques.
- The survey naturally reveals the research blank of supporting diverse visual analytics tasks with low-bit-rate representations. To fill in this blank, we propose to investigate the novel *information compression for analytics taxonomy* problem with an adjusted formulation considering the transferability among different machine vision tasks under the compression condition.
- The investigation aims to deduce a unified compressed feature for both high-level semantic-related tasks and mid-level geometry analytic tasks. Different from the traditional hyperprior model that captures pixel-wise dependency, we propose a novel codebook-based hyperprior model by integrating the codebook reconstruction process into the hyperprior model. Codebook is capable of bridging the dependencies between features of varying spatial sizes, and even capturing the correspondence between the feature vector (no spatial dimension) and the feature tensor. In this way, it successfully narrows down the intrinsic dimensionality gap between the NN-generated features from pixels and machine vision features/labels.

[1][Online]. Available: https://jpeg.org/jpegai/index.html

[2][Online]. Available: https://developer.nvidia.com/maxine

[3][Online]. Available: https://segmentfault.com/a/1190000039858782

Thus, the new model can estimate the entropy of NN-generated features more accurately, which helps minimize the bit-rates but still efficiently support different machine vision tasks.

- Under the proposed compression architecture, we further conduct a comprehensive discussion on the joint compression of visual data for a series of tasks. The empirical results demonstrate the feasibility that a series of tasks could be supported by a unified compressed representation.
  We also explore more potentials of the compressed representation, e.g. supporting unseen tasks, and plateau bit-rate in different tasks.

The rest of our paper is organized as follows. Section II conducts a comprehensive literature survey for the development of VCM in recent years, presenting how these works optimize the defined VCM R-D cost in different ways. Section III performs a comprehensive benchmark analysis of existing methods in rich measures, and demonstrates a competitive approach in achieving superior analytics performance in the context of multi-task compression. Section IV shows our exploration in modeling and optimizing the multiple tasks' R-D costs. In Section V, experimental configurations and results are presented. In Section VI, we make tentative discussions on several open issues based on our proposed framework, which provides rich insights for future research. The concluding remarks and potential future directions are given in Section VII.

II. Progress Survey of Video Coding for Machines

A. Formulation of VCM

The $L+1$ tasks are bundled with the labels $\mathbf{Y} = \{Y_0, Y_1, \ldots, Y_L\}$, whose features are denoted by $\mathbf{F} = \{F_0, F_1, \ldots, F_L\}$ extracted from the image I:

$$\{F_i\}_{i=0,1,\ldots,L} = E\left(I|\theta_E\right), \tag{1}$$

$$\widehat{Y}_i = A\left(F_i|\theta_A\right), \tag{2}$$

where $E(\cdot|\theta_E)$ is the feature extractor, and $A(\cdot|\theta_A)$ is the analytics model that maps the feature into the end task representation. $\widehat{Y}_i$ is the label prediction that targets to regress Y_i. We define $l_i(\widehat{Y}_i, Y_i)$ (short as l_i) as the performance measure of the task i regarding the reconstructed feature $\widehat{F}_i$ and label $\widehat{Y}_i$ after lossy encoding and decoding. The *VCM problem* is formulated as *an objective function to maximize the multi-task performance while minimizing the bit-rate cost:*

$$\underset{\Theta}{\text{argmax}} \sum_{0\le i\le L} \omega_i l_i, \text{s.t.} \sum_{0\le i\le L} \omega_i = 1,$$
$$\widetilde{B}\left(\{R_{F_i}\}_{0\le i\le L}\right) \le S_T, \tag{3}$$

where ω_i is the weighting parameter to balance the importance among different tasks. S_T is the total bit-rate cost constraint. R_{F_i} measures the bit-rate of the feature F_i. $\widetilde{B}(\{R_{F_i}\}_{0\le i\le L})$ measures the minimal bit-rate after fully considering the feature dependency among $\{F_i\}_{0\le i\le L}$. Besides $E(\cdot|\theta_E)$ and $A(\cdot|\theta_A)$, a VCM system still includes:

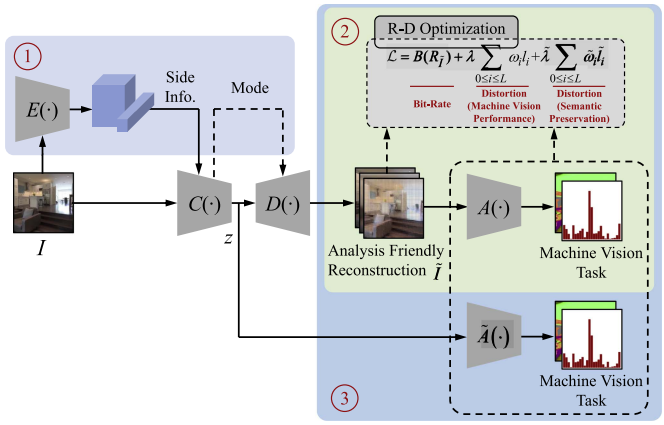


Fig. 1. Framework of machine vision targeted codec, including: 1) side information guidance; 2) machine vision constraint; and 3) semantic information preservation.

- Compression model $C(\cdot|\theta_C)$ that maps the feature into bit-stream;
- Decompression model $D(\cdot|\theta_D)$ that projects the bit-stream back to the feature;
- Feature predictor $G(\cdot|\theta_G)$ that targets predicting the feature of a task based on the reconstructed features of other more abstracted tasks to squeeze out the feature redundancy among different tasks,

which help form up $\widetilde{B}(\cdot)$. $\Theta = \{\theta_E, \theta_C, \theta_D, \theta_G, \theta_A\}$ are the parameters of all modules.

More details about the form of $\widetilde{B}(\cdot)$ and how to optimize (3) in terms of Θ will be briefly discussed in the following subsections.

B. Progress Survey

Based on the model's input, output, supporting tasks, and optimized terms, we categorize existing VCM methods into five classes: machine vision targeted codec, intermediate feature compression, intermediate feature optimization, feature assisted coding, and scalable coding. We will discuss the existing methods of the five classes in detail in the subsequent sections. A comprehensive summary of previous works is given in Table I.

1) Machine Vision Targeted Codec: The first category of methods is a natural evolution of existing image/video codecs. As shown in Fig. 1, the final outputs of the models are still images/videos. Differently, the compressed images/videos do not serve humans but are fed into machines to support machine vision tasks. These methods do not get involved with the optimization of feature extractor $E(\cdot|\theta_E)$, analytics model $A(\cdot|\theta_A)$, and feature predictor $G(\cdot\theta_G)$. In (3), only one bit-stream is extracted from the image, and task-driven constraints are enforced on the reconstructed images or directly on the bit-stream.

There are three directions in this category based on different ways to make the reconstructed images/videos include the information that benefits machine vision tasks, including *side information guidance*, *machine vision constraint*, and *semantic information preservation*. The first direction – *side information*

TABLE I
OVERVIEW OF VIDEO CODING FOR MACHINES METHODS IN LITERATURE

Publication	Category	Analytic Resource	Output	Presented Task	Optimized Terms E ‖ C ‖ D ‖ G ‖ A	Scalable
Suzuki et al. 2019 [56]	Machine Vision Targeted Codec	Image	Image	Classification	○ ‖ ● ‖ ○ ‖ ○ ‖ ○	✗
Yang et al. 2020 [71]		Image	Image	Classification, Object Detection	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Hou et al. 2020 [29]		Color quantized images	Color quantized images	Classification	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Choi et al. 2020 [19]		Image	Image	Classification, Image Caption	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Patwa et al. 2020 [48]		Image	Image	Classification	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Chamain et al. 2021 [8]		Image	Image	Detection	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Le et al. 2021 [37]		Image	Image	Object Detection, Instance Segmentation	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Le et al. 2021 [38]		Image	Image	Instance Segmentation	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Huang et al. 2021 [32]		Image	Image	Classification, Detection, Segmentation	○ ‖ ● ‖ ○ ‖ ○ ‖ ○	✗
Chen et al. 2019 [13]	Intermediate Feature Compression	Feature	Feature	-	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Chen et al. 2020 [14]		Feature	Feature	-	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Chen et al. 2020 [12]		Feature	Feature	-	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Suzuki et al. 2020 [57]		Feature	Feature	Classification	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Xing et al. 2020 [68]		Feature	Feature	Action Recognition	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Choi et al. 2020 [18]		Feature	Feature	Object Detection	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Hu et al. 2020 [31]		Image	Image	Classification, Image Caption	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Ulhaq and Bajić 2021 [58]		-	Feature	-	-	✗
Ikusan and Daiy 2021 [33]		Feature	Feature	Classification	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Alvar and Bajić 2019 [2]	Intermediate Feature Optimization	Feature	Semantic Map, Disparity Map, Image	Semantic Segmentation, Disparity Estimation	○ ‖ ● ‖ ● ‖ ○ ‖ ●	✗
Singh et al. 2020 [54]		Feature	Feature	Classification	● ‖ ● ‖ ● ‖ ○ ‖ ●	✗
Shah and Raj 2020 [51]		Feature	Feature	Classification	● ‖ ● ‖ ● ‖ ○ ‖ ●	✗
Alvar and Bajić 2020 [3]		Feature	Semantic Map, Disparity Map, Image	Semantic Segmentation, Disparity Estimation	○ ‖ ● ‖ ● ‖ ○ ‖ ●	✗
Alvar and Bajić 2021 [4]		Feature	Semantic Map, Disparity Map, Image	Semantic Segmentation, Disparity Estimation	○ ‖ ● ‖ ● ‖ ○ ‖ ●	✗
Zhang et al. 2021 [75]		Feature	Feature	Object Detection Instance Segmentation	● ‖ ● ‖ ● ‖ ○ ‖ ●	✗
Chen et al. 2019 [11]	Feature Assisted Coding	Texture Mask	Semantic Map, Texture Mask, Videos	Texture Region	● ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Li et al. 2019 [39]		-	Image	-	○ ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Huang et al. 2019 [16]		-	Image, Color Hint	-	● ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Chang et al. 2019 [9]		-	Edge, Image	-	● ‖ ● ‖ ● ‖ ○ ‖ ○	✓
Akbar et al. 2019 [1]		-	Semantic Map, Image, Compact Image	-	● ‖ ○ ‖ ○ ‖ ● ‖ ○	✓
Xia et al. 2020 [65]		-	Object Mask, Image	-	● ‖ ● ‖ ● ‖ ○ ‖ ○	✗
Kim et al. 2020 [35]		-	Soft Edge, Video	-	○ ‖ ○ ‖ ● ‖ ○ ‖ ○	✗
Prabhakar et al. 2021 [49]		-	Pose, Face Mesh, Video	-	● ‖ ● ‖ ● ‖ ○ ‖ ○	✓
Wang et al. 2019 [60]	Scalable Coding	Feature	Feature, Image	Face Recognition	○ ‖ ● ‖ ● ‖ ● ‖ ○	✓
Hu et al. 2019 [30]		Image	Quantized Edge, Color Hint, Image	Facial Landmark Detection	● ‖ ● ‖ ● ‖ ● ‖ ○	✓
Xia et al. 2019 [66]		Sparse Points and Motion	Sparse Points and Motion, Video	Action Recognition	● ‖ ● ‖ ● ‖ ● ‖ ●	✓
Hoang et al. 2020 [28]		Semantic Map	Semantic Map, Low-res Image, Image	Semantic Segmentation	● ‖ ● ‖ ● ‖ ● ‖ ○	✓
Yan et al. 2020 [69]		Feature	Feature	Classification	● ‖ ● ‖ ● ‖ ● ‖ ●	✓
Yang et al. 2021 [70]		Image	Quantized Edge, Color Hint, Image	Facial Landmark Detection	● ‖ ● ‖ ● ‖ ● ‖ ○	✓
Wang et al. 2021 [61]		Feature	Feature, Image	Face Recognition	● ‖ ● ‖ ● ‖ ● ‖ ○	✓
Wang et al. 2021 [62]		Feature	Feature, Image	Face Recognition	● ‖ ● ‖ ● ‖ ● ‖ ○	✓
Choi and Bajić 2022 [17]		Feature	Semantic Map, Image	Object Detection, Segmentation, Reconstruction	● ‖ ● ‖ ● ‖ ● ‖ ○	✓
Liu et al. 2021 [42]		Feature	Feature, Image	Classification	○ ‖ ● ‖ ● ‖ ● ‖ ●	✓
Chang et al. 2021 [10]		Image	Semantic Map, Image	Facial Landmark Detection	● ‖ ● ‖ ● ‖ ● ‖ ○	✓
Bajić et al. 2021 [5]	Summary	–	–	–	–	–
Gao et al. 2021 [22]		–	–	–	–	–

✓and ✗denote that the method owns the corresponding feature or not, respectively.

guidance – is to detect the side information related to the analytics performance at first and then utilize the side information to adjust the coding configurations at the encoder and decoder sides. For example, in [32], the region of interest for machines is detected based on the degree of importance for each coding tree unit in visual analysis, which is injected into a novel CTU-level bit allocation model. In [19], the task-specific quantization tables are learned via learning a differentiable loss function to approximate bit-rates.

The second direction, i.e. *machine vision constraint*, adopts the loss functions that target machine vision optimization to train end-to-end learned codecs. In [8], [29], [37], [38], [56],

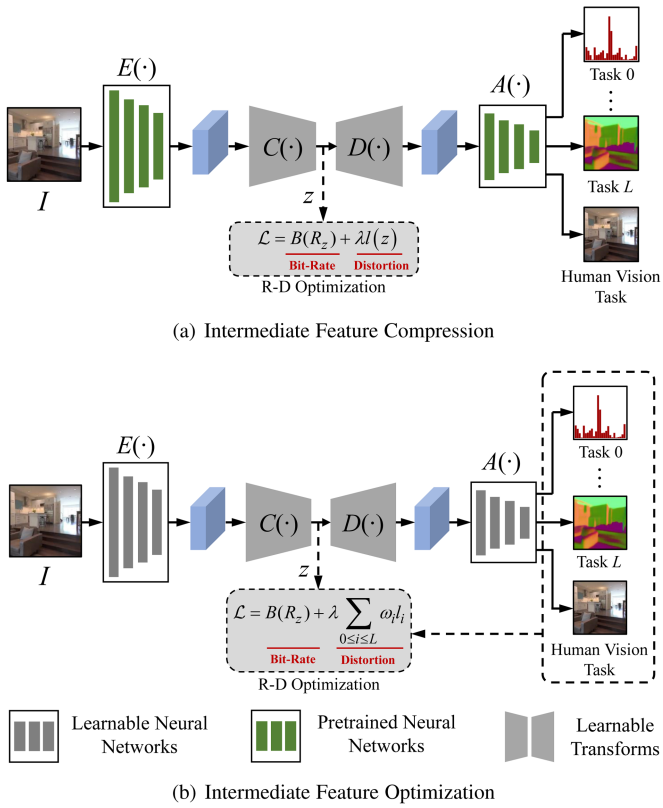


Fig. 2. Frameworks of intermediate feature compression and optimization.

the task-driven losses are adopted. Namely, in these works, the downstream deep networks for machine vision are connected to the outputs of the codecs, and these two parts are trained jointly. Besides, in [38], the perceptual loss is additionally utilized in the R-D optimization function of an inference-time content-adaptive finetuning scheme leading to higher compression efficiency for machine consumption. In [71], the maximum mean discrepancy is adopted to align feature distributions, which results in preserving more consistent perception in the feature domain and better recognition of pre-trained machine vision models.

The third direction [47], i.e. *semantic information preservation*, constrains the encoded/decoded bit-streams to have the capacity of semantic preservation via connecting a classifier to the decoded bit-stream that predicts the semantic labels for machine vision tasks.

To summarize, machine vision targeted codec still outputs images/videos naturally perceived by humans, which are more friendly to the successive machine vision tasks. However, they still need a whole analytics model in the cloud to perform the machine vision tasks, which in fact increases the whole burden at the front-end and cloud side.

2) Intermediate Feature Compression: The second category targets compressing features. Instead of compressing dedicated features for given tasks, as shown in Fig. 2(a), the new methods aim to compress the deep intermediate features, which are expected to be more expressive and compact for successive machine vision applications. This feature compression paradigm also allows the "front-cloud" collaborative processing. Namely, a part of the inference calculation can be placed at the front end. From the perspective of (3), one layer of intermediate feature is compressed and the task performance is defined as the fidelity between the decoded feature and original one, i.e. without a direct connection to the end-task performance.

Chen et al. [13], [14] made the first attempt and presented lossy/lossless compression frameworks based on High Efficiency Video Coding (HEVC) as well as evaluation metrics for intermediate deep feature compression. The follow-up works further improve the coding efficiency via optimizing sub-modules in the framework, most of which focus on removing inter-channel redundancy. In [57], Suzuki et al. proposed a new feature arrangement method that regards the deep intermediate feature as videos and compressed the videos to make full use of spatio-temporal correlation. In [33], Ikusan et al. also compressed deep features from the perspective of video compression, especially distinguishing and making use of key frames. A selection strategy is developed to reduce the feature redundancy and the selected features are then compressed via video encoder. Then, an R-D optimization targeted for computer vision tasks is integrated into the codecs. In [18], Choi et al. also selected a subset of channels of the feature tensor to be compressed and introduced a novel back-and-forth prediction method to infer the original features of shallow layers based on compressed deep layers.

In [12], Chen et al. proposed three new modes to repack features and explored two modes in Pre-Quantization modules to further improve the fidelity maintenance capacity. In [68], Xing et al. adopted logarithmic quantization and HEVC inter encoding to compress 3D CNN features for action recognition. In [31], the different channels' contributions to the inference result are studied and a channel-wise bit allocation is developed. The model takes a two-pass step. In the first step, the channel sensitivity is estimated while in the second step, bits are allocated based on the estimated sensitivity. Ulhaq and Bajić [58] explored the motion relationship between the corresponding feature tensors and concluded that the feature's motion is approximately equivalent to the scaled version of the input motion.

To summarize, this category adopts existing codecs to compress deep intermediate features. However, these codecs are originally designed to deal with the image/video signals, which might be non-optimal to estimate the bit-rate of features and turn the features into compact representations. Furthermore, as the feature fidelity is very hard to accurately define and the model at the decoder side is not jointly optimized, the compression efficiency is not fully optimized.

3) Intermediate Feature Optimization: Beyond intermediate feature compression, this category also aims to compress deep intermediate features, but chooses to optimize the whole compression framework jointly with the successive machine vision tasks instead of feature fidelity, as shown in Fig. 2(b). Alvar and Bajić [3] made the first effort to develop a loss function to measure a feature's compressibility that constrains the training process of multi-task models. Similarly, Singh et al. [53] developed a penalty to train the network in an end-to-end manner for

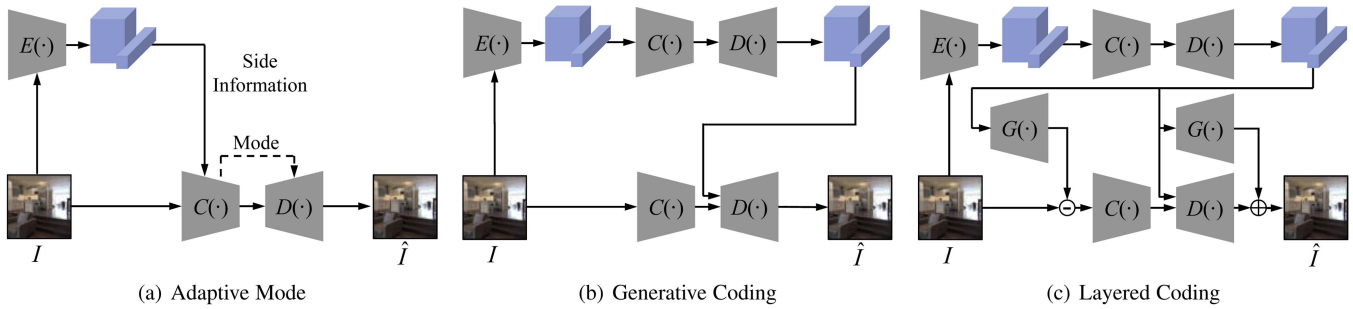


Fig. 3. Three frameworks of feature assisted coding. This category only considers the need for human vision.

balancing the expressiveness and compressibility of deep features. Shah and Raj [51] proposed an Annealed Representation Contraction method that obtains small-scale networks (features) via iteratively tuning the shrunk models with network layer contraction and annealed labels. Zhang et al. [75] developed a multi-scale feature compression method that is jointly optimized in a Mask RCNN trained with mask, box, and class-related losses. In [3], [4], the task distortion is approximated as convex surfaces, which helps derive a closed-form bit allocation solution for both single-task and multi-task systems, and analytical characterization of the full Pareto set.

In summary, it is beneficial to collaborative intelligence with the help of both intermediate feature compression/optimization. However, the methods in this category cannot provide full pixel reconstruction, i.e. reconstructed images/videos. Besides, the compressed and transmitted features still rely on joint optimization with the successive machine vision tasks, which sets barriers to the generalization capacity of the features.

4) Feature Assisted Coding: The fourth category explores collaborative operations between video and feature streams to optimize the coding efficiency from the perspective of human vision, called feature assisted coding. As shown in Fig. 3, the side information or semantic features are first extracted, and these features are used to facilitate the full-pixel image/video reconstruction. In formulation, we have

$$\begin{aligned}\widetilde{B}\left(\{R_{F_i}\}_{0\le i\le L}\right) &= B\left(R_{F_0}\right)+\sum_{1\le i\le L}\min_{0\le j<i}\left\{B\left(R_{F_{i\to j}}\right)\right\},\\ R_{F_0} &= C\left(F_0|\theta_C\right),\\ R_{F_{i\to j}} &= C\left(F_i-\mathrm{G}\left(\widehat{F}_j,i|\theta_G\right)\middle|\,\theta_C\right),\ \text{for}\, i\neq 0,\end{aligned} \tag{4}$$

where $B(\cdot)$ measures the bit-rate. Namely that, the most abstract feature (F_0) is first extracted while the task dependency is fully squeezed out via feature prediction $G(\cdot|\theta)$.

There are three directions in this category: *adaptive mode*, *generative coding*, and *layered coding*. The first direction – *adaptive mode* – is shown in Fig. 3(a). The side information is adopted to control the operations of the encoder and decoder. In [65], an object segmentation network is utilized to separate the (non-)object masks, and two compression networks are used to compress object regions and background ones, respectively. In [11], the pixel-level texture regions are inferred by semantic segmentation with the help of motion cues injected into codecs. In [39], the coding process-related configuration and parameters, i.e. the JPEG configuration, are generated with an agent to infer the compression level adaptively based on extracted features and deep network backbones.

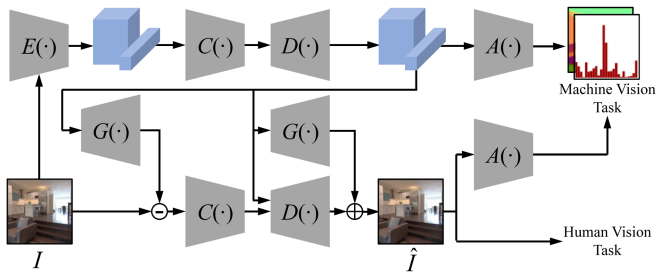
The second direction *generative coding*, as shown in Fig. 3(b), compresses and transmits the extracted features to form a single feature stream. On the decoder side, the transmitted feature will help the decoding of the image/video stream in a generative way. In [16], gray-scale and color hints are extracted and compressed via BPG with an adaptive quantization parameter (QP). At the decoder side, these two parts are processed by the artifact reduction network and colorization network. In [49], the body pose and face mesh are detected at the encoder side and reconstructed into animated puppets at the decoder side to support video reconstruction. In [35], a generative decoder is adopted to map key frames as well as soft edges of non-key frames into the whole reconstructed frames. In [9], the edges are extracted to form the feature stream, which facilitates the image reconstruction at the decoder side.

The third direction *layered coding* [1], as shown in Fig. 3(c), further introduces the prediction mechanism to remove the redundancy between feature and image streams. The segmentation map plays the role of the base layer. A compact image then acts as the first enhancement layer. These two layers are used to form a coarse reconstruction of the image. The residue layer, namely, the difference between the input and the coarse reconstruction, acts as another enhancement layer.

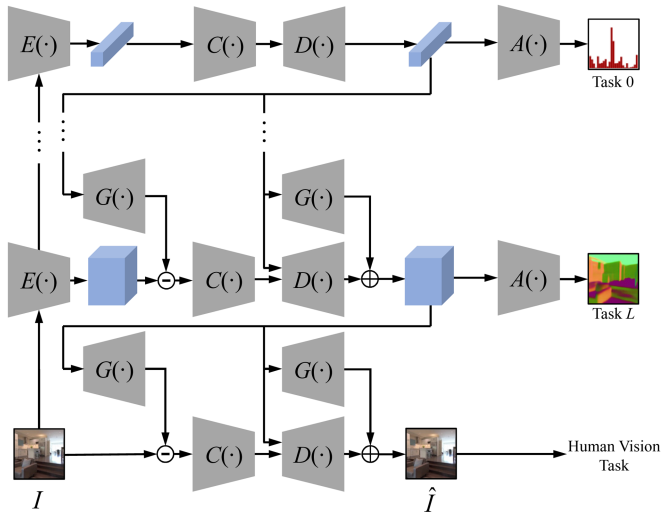
In summary, feature assisted coding introduces a scalable mechanism to improve coding efficiency. However, these methods do not consider supporting machine vision tasks, which will be investigated in the next sub-section.

5) Scalable Coding: This category also compresses and transmits both feature and image/video streams to serve both human and machine visions via the same route of (4). There are two directions in this category.

The first direction *two-stream scalable coding*, as shown in Fig. 4(a), takes the same architecture as *layered coding* in Fig. 3(c). However, differently, besides supporting the image/video reconstruction, the feature stream transmitted by the methods here needs to support the machine vision tasks. The works in [30], [61], [62], [70] focus on the analysis and



(a) Two-Stream Scalable Coding



(b) Multi-Stream Scalable Coding

Fig. 4. Frameworks of scalable coding. Beyond feature assisted coding, this category tries to fulfill the needs of both human and machine vision.

reconstruction of face images via the two-stream structures. In [30], [70], quantized edges and color clues are used to generate reconstruction images via the generative model at different bit-rates that support facial landmark detection and image reconstruction. In [60], [61], [62], in a scalable coding framework, the deep learning feature acts as the base layer to support face recognition, and the enhancement layer learns to reconstruct the full-pixel images. In [10], the semantic prior is modeled and transmitted to facilitate the conceptual coding scheme towards extremely low bit-rate image compression, where the reconstructed images serve facial landmark detection. In [28], the semantic segmentation map and reconstructed image compensate for each other to improve the semantic and visual quality. In [66], the learned motion pattern, i.e. key points as well as the related trajectories, is transmitted for action recognition and video frame reconstruction.

The second direction *multi-stream scalable coding,* as shown in Fig. 4(b), adopts more than one stream to transmit features with a feature salable prediction mechanism. In [17], side, base, and enhancement streams are transmitted together, where the side stream is used for entropy estimation and control, the base stream serves machine vision tasks, while the enhancement stream helps compensate for full-pixel image reconstruction. In [69], a feature compression approach is conducted on the intermediate representations where different layers serve the tasks requiring different grained semantic information. The work in [42] develops a lifting structure based on a trainable and revertible transform that decomposes the image into different bands to support different machine vision tasks.

In summary, with the scalable mechanism and feature stream, these methods can support both human and machine vision in an efficient and flexible way. However, some critical issues in intelligent collaborative analytics of massive data and diverse tasks/applications, e.g. the transferability among different tasks, the generality of the extracted features, etc., are still not explored.

III. Benchmark Existing Methods

In this section, we evaluate representative state-of-the-art methods with diverse kinds of metrics, from which several interesting insights are obtained.

A. Evaluation Protocols

1) Dataset: We conduct the experiments on the Taskonomy dataset [73], which contains approximately 4.5 million images, all labeled by 25 attributes, to support various machine vision tasks. The following experiments are conducted on a subset. The subset is selected at random, while we control the numbers of images in the splits, i.e. 51,316 images for training, 945 for validation, and 1,024 for testing.

Images in different splits of the data are captured in different buildings. Thus, the splits are diverse in content.

We select a set of *real-world* tasks for evaluation, i.e. scene classification, object classification, semantic segmentation, surface normal estimation, reshading, and principle curvature estimation. The selected tasks include diverse categories, with which we evaluate both high-level semantics driven analytics and mid-level geometry related estimation.

We follow the setting in [73] to compare different methods on 256×256 images, and calculate bits-per-pixel on that resolution.

2) Evaluation Task and Measure: The performance of six tasks is measured in the evaluation tasks: scene classification, semantic segmentation, and Object Classification, normal, reshading, and curvature. For semantic segmentation, we adopt mean pixel-level accuracy (Acc.), the accuracy of pixels in the non-background regions (Non-BG Acc.), and mean IoU (mIoU) by averaging the result among all 17 classes to provide a comprehensive performance evaluation. Besides, we measure the performance of the task scene classification in accuracy, surface normal estimation of indoor scenes in L_1 distance, and reshading of an indoor image in L_1 distance. Bit-per-pixel (bpp) is used to calculate the bit-rate usage for measuring the compactness in the compression.

3) Baseline Methods: We select several representative VCM techniques from each category in our benchmark, including:

- *Conventional Image Codec:* BPG (Better Portable Graphics), the version of HEVC compression method using the still picture profiles.
- *Machine Vision Targeted Codec:* Le2021-1 [37] and Le2021-2 [38].

TABLE II
EXPERIMENTAL RESULTS FOR THE SEMANTIC SEGMENTATION TASK IN COMPARING FEATURE COMPRESSION TO MACHINE VISION TARGETED CODEC METHODS

Method	Bit-Rate (bpp)↓	Cross Entropy↓	Acc.↑
Original	/	0.74	91.64%
Control Group	/	0.61	92.31%
Hyperprior [8]	0.025	**0.80**	91.95%
Cheng2020 [15]	0.077	2.72	**96.16%**
BPG-51	**0.024**	1.91	90.18%
BPG-47	0.039	1.30	90.73%
BPG-43	0.063	1.03	90.93%
Le2021-1	0.117	1.02	92.54%
Le2021-2	0.088	1.12	92.83%
Method	Non-BG Acc.↑	mIoU↑	-
Original	86.28%	27.65%	-
Control Group	82.67%	27.07%	-
Hyperprior [8]	79.64%	25.42%	-
Cheng2020 [15]	63.30%	**31.26%**	-
BPG-51	65.97%	19.84%	-
BPG-47	76.46%	24.36%	-
BPG-43	**81.15%**	25.37%	-
Le2021-1 [37]	79.01%	26.96%	-
Le2021-2 [38]	76.91%	28.22%	-

↑ means higher performance, better result, and ↓ vice versa. The best results are denoted in bold.

TABLE III
EXPERIMENTAL RESULTS FOR THE SEMANTIC SEGMENTATION TASK USING VARIOUS FEATURE COMPRESSION SCHEMES

Method	Bit-Rate (bpp)↓	Cross Entropy↓	Acc.↑
Original	/	0.74	91.64%
Control Group	/	0.61	92.31%
IDFC (qp=51) [14]	0.020	6.22	92.74%
IDFC (qp=43) [14]	0.026	1.38	93.94%
Chen2020 (qp=51) [12]	**0.013**	3.39	93.35%
Chen2020 (qp=43) [12]	0.022	0.95	93.41%
Hyperprior [8]	0.025	**0.80**	91.95%
Cheng2020 [15]	0.077	2.72	**96.16%**
Method	Non-BG Acc.↑	mIoU↑	-
Original	86.28%	27.65%	-
Control Group	82.67%	27.07%	-
IDFC (qp=51) [14]	15.62%	11.03%	-
IDFC (qp=43) [14]	72.24%	28.41%	-
Chen2020 (qp=51) [12]	42.02%	18.61%	-
Chen2020 (qp=43) [12]	**80.51%**	30.11%	-
Hyperprior [8]	79.64%	25.42%	-
Cheng2020 [15]	63.30%	**31.26%**	-

Method IDFC is evaluated with different QPs, marked as IDFC (QP) in the table. ↑ means higher performance, better result, and ↓ vice versa. The best results are denoted in bold.

TABLE IV
ANALYTICS PERFORMANCE OF MULTI-TASK COMPRESSION WITH/WITHOUT IMAGE RECONSTRUCTION

Task	Metric	Hyperprior	Cheng-2020	Le2021-1	Le2021-2
Scene Class	Acc.↑	50.59%	55.08%	61.23%	62.30%
Semantic Seg.	mIoU↑	31.88%	34.36%	25.78%	26.72%
Object Class	Acc.↑	45.31%	48.54%	50.59%	51.46%
Recon.	PSNR↑	-	-	23.80	25.80
Bit-Rate	Bpp↓	0.018	0.082	0.155	0.202
Normal	L_1↓	0.131	0.134	0.134	0.132
Reshading	L_1↓	0.230	0.241	0.256	0.254
Curvature	L_1↓	0.389	0.392	0.389	0.387
Recon.	PSNR↑	-	-	23.68	25.16
Bit-Rate	Bpp↓	0.033	0.106	0.154	0.132
Task	Metric	BPG-51	BPG-47	BPG-43	BPG-39
Scene Class	Acc.↑	48.73%	56.64%	61.91%	65.62%
Semantic Seg.	mIoU↑	19.84%	24.36%	25.37%	26.20%
Object Class	Acc.↑	46.39%	50.10%	51.07%	52.64%
Recon.	PSNR↑	27.80	29.95	31.97	34.02
Bit-Rate	Bpp↓	0.024	0.039	0.063	0.099
Normal	L_1 ↓	0.229	0.176	0.147	0.132
Reshading	L_1 ↓	0.410	0.334	0.305	0.294
Curvature	L_1 ↓	0.418	0.402	0.394	0.389
Recon.	PSNR↑	27.80	29.95	31.97	34.02
Bit-Rate	Bpp↓	0.024	0.039	0.063	0.099

- *Feature Compression and Optimization:* Intermediate deep feature compression [14] and Chen2020 [12].
- *Scalable Coding:* Liu2021 [42] and Choi2022 [17].
- *End-To-End Compression:* Hyperprior model [6], [8], Cheng2020 [15] for compressing features.
- *Uncompressed Baseline:* Original and control group. The Original model refers to the results given by the originally provided hourglass-like networks in [73]. To avoid the potential bias due to the training procedure, we set up the Control Group experiments, where a transform network with an identical structure to the compression model is trained, but no bit-rate constraint is applied.

All the methods that are related to the training are retrained on our training data with their original training settings and policies reported in their papers. The feature extraction and analytics models are initialized with the original weights released by [73]. We select the model checkpoint with the lowest R-D cost and compare on the testing set. All learned models are trained with the loss function $\mathcal{L} = R + \lambda \mathcal{L}_{CE}$.

4) Pretrained Feature Extractor and Analytics Models: The abundance of tasks linked to one image provides the desired environment for our study. We utilize the pre-trained models (including feature extractor and analytics models) on the dataset, provided by the authors under the MIT License. All these pre-trained models are hourglass encoder-decoder neural networks, as described in [73].

B. Benchmark Results and Insight

1) Comparisons to Image Codecs: We compare a baseline learned feature compression method [8] and an end-to-end compression method, i.e. Cheng2020 [15], for feature compression with conventional and end-to-end learned codecs that compress images first and then perform analytics.

Insight: It is clearly demonstrated from Table II that, in comparison to codecs that compress images, the feature compression method leads to higher compression efficiency for intelligent analytics.

2) Comparisons to Feature Compression Methods. We compare the results of applying different feature compression methods (learned or handcrafted) for semantic segmentation.

Insight: From Table III, it is observed that, for the feature compression method, the learnable optimization methods achieve superior performance to feature compression methods, which demonstrate the advantages of using the learnable paradigm.

3) Comparisons in Multi-Task Task Scenarios: We compare the results of applying different compression methods for multi-task analytics. The end-to-end learned methods are also adopted for compressing features. The image codecs and scalable coding approach are compared here, as only these methods can support multi-task modeling.

Insight: From Table IV, the experimental results demonstrate that, the scalable coding method might fail to make a good trade-off between the performance of different tasks. After introducing the visual reconstruction task, the overall performance of the visual analytics tasks is degraded.

C. Summary

The benchmark experiments fully demonstrate that, for various analytics tasks, compressing features is more efficient than directly compressing images, and learning-based compression is more efficient than both feature compression and scalable coding (including image reconstruction). However, there is a huge gap between the existing compression paradigms and the intuitively ideal compression route especially for analytics tasks. Existing compression methods basically make an effort to model pixels or smaller size features still have a spatial dimension, while many analytics tasks, e.g. classification or detection, analytics generally only need to estimate/model the distribution of the entire image. Therefore, in the next section, we will first formulate the VCM problem from the perspective of compressive taxonomy. Then, along the paradigm of learning-based feature compression paradigm without introducing image reconstruction, we further propose a new hyperprior model method to make up for the above-mentioned gap.

IV. Revisit Visual Representation Compression in Analytics Taxonomy

A. Formulation of Compressive Analytics Taxonomy

As discussed above, the previous paradigms still face issues when dealing with massive data and diverse kinds of tasks, e.g. Taskonomy [73].

- Transmitting the multi-task features one-by-one leads to additional redundancies as different tasks inevitably have semantic gaps.
- Each feature is deeply coupled with the corresponding task, therefore is hard to be generalized to handle unseen tasks.
- Compressing and transmitting multi-task features one-by-one in a scalable way results in greater latency and complexity.

Therefore, we propose to investigate the relationship among different feature representations from a compression perspective, and seek to construct a combined compact feature serving a bundle of tasks, which can also be generalized to deal with unseen tasks. More specifically, we have a reformulated VCM problem:

$$\underset{\widehat{\Theta}}{\operatorname{argmax}} \sum_{0\le i\le L} \omega_i l_i, \text{s.t.} \sum_{0\le i\le L} \omega_i = 1, \quad B\left(R_z\right) \le S_T, \tag{5}$$

where a transfer function $\Phi(\cdot|\theta_\Phi)$ and an inverse-transfer function $\Psi(\cdot|\theta_\Psi)$ with θ_Φ and θ_Ψ as their parameters, are adopted to map multi-task features into and from a combined feature z, respectively. The related important elements are defined as follows,

$$R_z = C\left(\Phi\left(\{F_i\}_{0\le i\le L}|\theta_\Phi\right)|\theta_C\right), \tag{6}$$

$$\widehat{F}_i = \left[\Psi\left(D\left(R_z|\theta_D\right)|\theta_\Psi\right)\right]_i, \tag{7}$$

$$\widehat{\Theta} = \{\theta_E, \theta_C, \theta_D, \theta_G, \theta_A, \theta_\Phi, \theta_\Psi\}, \tag{8}$$

where $[\cdot]_i$ denote to select the i-th element from the set. Equation (6)–(8) show the R-D optimization route for multiple tasks where a combined compact feature z is first derived via $\Phi(\cdot|\theta_\Phi)$ and different tasks are jointly supported by this unified representation via $\Psi(\cdot|\theta_\Psi)$.

Equation (5) provides a specific formulation of (3) within the context of multi-task analytics. Equation (3) serves as a general model framework for defining VCM.

On the other hand, (5) narrows its focus to modeling multi-task relationships while filtering out descriptions of resources that fall outside the scope of our interest. This allows us to concentrate more on the aspects of multi-task analytics and comprehension within the framework of compact feature representation. In this paradigm, when handling multiple tasks, the scalable feature prediction is merged into decoding and transfer mapping, which further improves coding efficiency and reduces the complexity as well as system delay. Furthermore, as the new feature z contains information of multiple tasks rather than bundled with the given task, the framework has the potential to be well generalized to handle unseen tasks.

B. Codebook-Hyperprior Model for Deep Feature Compression

In order to solve the optimization function in (5), we first design a trainable compression framework targeting to compress deep features, illustrated in Fig. 5, to estimate and reduce the information entropy of each deep feature representation F_i. Then, in the next subsection, we consider handling deep features extracted from multiple tasks.

1) Motivation: As evident from the benchmark results, it consistently demonstrates superior performance when applying the learned compression method for feature compression. However, despite their impressive performance, certain issues arise with space for further enhancements. There exist disparities between the deep features and the information necessary for downstream tasks, both in terms of content and dimensionality. This mismatch poses challenges to the efficiency of the vanilla hyperprior model (Fig. 6(a)) in modeling the probability distribution of these features.

To tackle this, we propose a novel codebook-based hyperprior model (Fig. 6(b)). We integrate the codebook reconstruction process into the hyperprior model, which bridges the dependencies between features for different tasks, and even captures the correspondence between the feature vector (no spatial dimension) and the feature tensor. As a result, it successfully narrows down the intrinsic dimensionality gap between the NN-generated features from pixels and machine vision features/labels.

2) Entropy Modeling and Minimization With Hyperprior: We transform F_i into a compact z, whose probability distribution is tractable and can be compactly compressed into bit-stream.

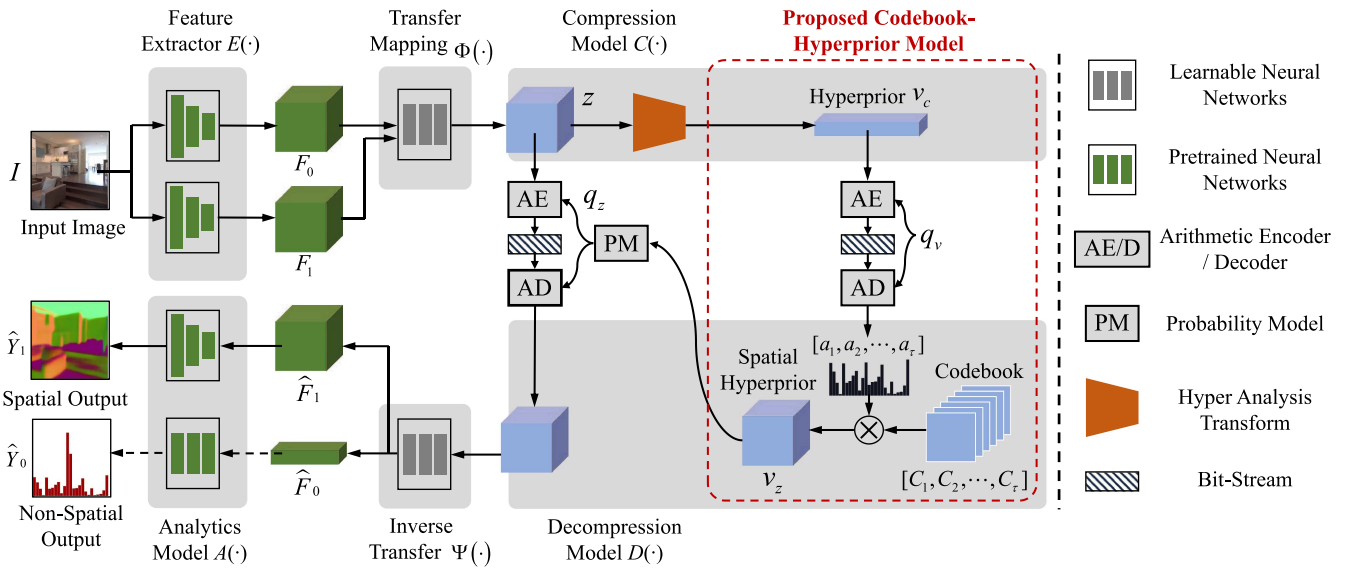


Fig. 5. Visual representation compression in analytics taxonomy for deep features of multiple tasks. With a pre-trained neural network (in **green**), a feature tensor is extracted and compressed by the proposed model (in **blue**). The reconstructed feature tensor is processed by the rest layers of the pre-trained network to produce analytics results. Dashed lines illustrate the processing of the feature vectors without the spatial dimension. Our proposed codebook hyperprior model is capable of bridging the dependencies between features of varying spatial sizes and successfully narrows down the intrinsic dimensionality gap between the NN-generated features from pixels and machine vision features/labels.

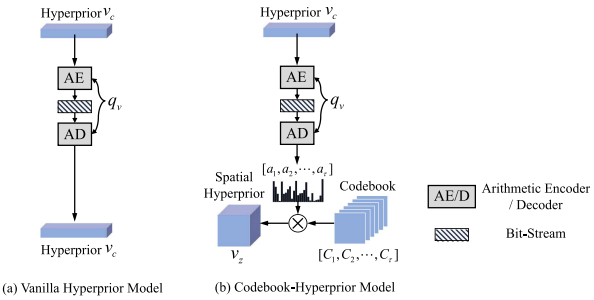


Fig. 6. Architecture of different hyperprior models. (a) Vanilla hyperprior model. (b) Codebook-hyperprior model. Empowered by the codebook, the gaps between features of varying spatial sizes are bridged, which narrows down the intrinsic dimensionality gap between the NN-generated features from pixels and machine vision features/labels.

As the mainstream neural networks do not apply any constraint on its generated F_i, the probability distribution of F_i is usually unknown and it is intractable to estimate the entropy of F_i. Therefore, following [6] we apply a transform to F_i and obtain a functionally equivalent representation z. Hence, we can estimate the entropy of F_i by calculating the entropy of the tractable structured representation z.

The actual bit-rate to encode z with the probability p_z under an estimated entropy model q_z equals to the cross-entropy [20] of p and q, as,

$$H(p,q) = \mathbb{E}_p[-\log q] = H(p) + D_{KL}(p||q). \tag{9}$$

Following Ballé et al. [6], we extract and encode a hyperprior from an image representation for more accurate entropy estimation. For image representation, the hyperprior is often of a lower resolution and used to estimate the probability distribution of the corresponding image representation.

However, feature representations F_i and z are not image-level dense pixel signals. They serve machine vision tasks and do not include information on image appearances. Although their extracted features might take the form of tensor (not vector) and have spatial dimensions, these features in fact can be embedded into very low-dimensional space, which does not have the spatial signal structure. As the image compression-oriented hyperprior model relies on the hierarchical spatial structure of images, it becomes less effective to model the signal structure of F_i and z, resulting in a gap between p and q and making the entropy estimation less effective.

3) Codebook-Hyperprior Model: To reduce the gap, we assume that the extracted feature representations from the neural network can be embedded into a very low-dimensional manifold. Each observed instance can be regarded as a point sampled from the low-dimensional subspace, and the perturbation is independently distributed and conditioned on the coordinates that expand the space. This assumption naturally leads to the proposed low-dimensional hyperprior model.

The main idea is that, we adopt the hyperprior vector without the spatial dimensions in the encoding process to capture the intrinsic signal structure of F_i and z, but transform the hyperprior vector into the hyperprior tensor with the spatial dimensions in the decoding process to augment the hyperprior's modeling capacity. To estimate the probability distribution of z, a hyperprior v_c is extracted from z via a hyper analysis transform $f_{Ha}(\cdot)$ as, namely, $v_c = f_{Ha}(z)$. The estimation of probability $p(z)$ can be divided into $p(z) = p(z, v_c) = p(v_c)p(z|v_c)$. Then, we apply a global pooling operation to reduce the spatial dimensions of z, producing v_c in the vector form. Note that v_c is also quantized

to integers. We further assume that each element in v_c follows a zero-mean Gaussian distribution $\mathcal{N}(0, \sigma_j)$. Conditioned on v_c, each element z_k in z is conditionally independently distributed.

The entropy of v_c is estimated by tuning the parameter σ_j. We model $q_{z_k|v_c}$ with a Gaussian distribution $q_{z_k|v_c} \sim \mathcal{N}(\mu_k = f(v_c; \theta_f), \sigma_k = g(v_c; \theta_g))$, where the mean and scale are generated via a function taking v_c as the input. To achieve this, we decode n sequences of coefficients from v_c. Each sequence $\mathbf{A}^s = (a_1^s, a_2^s, \ldots, a_\tau^s), s \in [1, n]$ indicates a linear combination of the spatial bases, defined by a codebook, in the form of $\{C_1, C_2, \ldots, C_\tau\}$. With the codebook and the sequences of coefficients $\{\mathbf{A}\}_n$, we generate the spatial hyperprior $\hat{\mathbf{Z}}$ as,

$$\hat{Z}_l = a_1^s C_1 + a_2^s C_2 + \cdots + a_\tau^s C_\tau, \text{ for } s = 1, 2, \ldots, n,$$
$$v_z = (\hat{Z}_1, \hat{Z}_2, \ldots \hat{Z}_n). \tag{10}$$

We employ a prediction sub-network to estimate $\mu_k = f(v_c; \theta_f), \sigma_k = g(v_c; \theta_g)$ from v_z. By learning the parameters of the sub-network, θ_f and θ_g are estimated to provide an accurate estimation $q(z|v_c)$ for $p(z|v_c)$. The spatial dimensions of the codebook $\{C_1, C_2, \ldots, C_\tau\}$ are fixed, and therefore it requires a re-sampling to deal with the inputs of different resolutions.

The proposed model is also general and flexible to support the deep features without spatial dimensions, i.e. feature vectors. This can be achieved by directly producing the vector-form probability parameters $\mu_k = f(v_c; \theta_f), \sigma_k = g(v_c; \theta_g)$ with v_c, via multi-layer perceptions.

C. Transfer Mapping: Multi-Task Compression

1) (Inverse-)Transfer Mapping: It has been shown in [73] that, there exist connections among feature representations of different tasks. Thus, if multiple tasks are supported as we mentioned in the problem formulation, the separate compression for each task may be less efficient due to the cross-task redundancy. Therefore, we propose the aggregation transformed compression scheme to generate the compressed representation for different tasks jointly.

An example of the proposed aggregation transformed compression scheme is shown in Fig. 5. The illustrated structure compresses and aggregates the feature representations of two tasks into one bit-stream. Each representation is transformed with a sub-network. The transformed features are concatenated and compressed via a single compression model. The decompressed representation is then split via another set of convolutional layers, serving as the input of the rest of the pre-trained analytics network.

2) Rate-Distortion Loss: The aggregation transformed compression model is trained in two stages, corresponding to the two application scenarios, including: 1) analytics oriented compression in a known set of tasks; and 2) out-of-set analytics, i.e. handling unseen tasks. During the first training phase, the parameters of the compression model and the multi-layer peripheral convolutions before the compression model for each task are tuned. Parameters of the pre-trained analytics models are fixed. The compression model learns to compress different forms of feature representations jointly. The parameters are trained with the joint R-D loss function as,

$$\mathcal{L} = B(R_z) + \lambda \sum_{0 \le i \le L} \omega_i l_i, \tag{11}$$

where λ is the Lagrange multiplier to indicate the relative importance of bit-rate and distortion.

V. Experimental Results

In this section, all experiments take the same configuration as the ones used in Section III. More implementation details of our method are presented in the https://huangerbai.github.io/CompTasko/supple.html.

A. Evaluation of Compression for Semantic Segmentation

We first evaluate the efficacy of the proposed codebook-hyperprior model for compressing deep features. The range of bit-rates we show in Table V is regarded as the *plateau* bit-rate, where the compressed feature representation provides enough information to make the prediction accuracy comparable to the models without the bit-rate control. We also show the radar charts of our method compared to *IDFC* and the primitive *Hyperprior* model in Fig. 7, which can better reflect the performance comparisons in all metrics. It is observed that the proposed method owns a larger area than *IDFC* and *Hyperprior:* The results in Fig. 7 and Table V show that our model can better compress the deep features than existing methods [8], [14], as it consumes fewer bit-rates to reach a higher analytics performance in multiple metrics.

B. Evaluation of Compression for Multiple Tasks

We compare the analytics performance of different methods that can support multiple tasks in Table VI. It is observed that, our method achieves superior analytics performance to BPG with a similar bit-rate. Le2021-1 [37] and Le2021-2 [38] offer very impressive results but they consume an order of magnitude higher bit-rate overhead. Comparatively, our method also provides very impressive analytics performance with a very compact bitstream, i.e. about only 1/3 b-rate of Cheng2020 [15]'s.

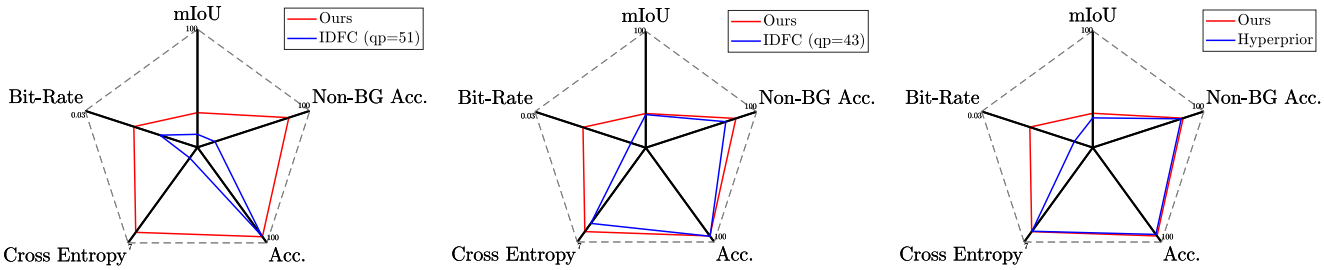
C. Complexity of Different Methods

We compare the complexity of different methods in Tables VII and VIII, i.e. FLOPs, parameter numbers, and encoding/decoding time. Table VII provides the complexity comparison of feature compression without taking the analytics models into account. Meanwhile, Table VIII illustrates the complexity comparison of various image compression methods, which subsequently undergo processing by analytics models to yield specific task results. In this context, our approach is integrated with the analytics models to calculate complexity. The results demonstrate that our method exhibits comparable complexities to the hyperprior model and operates with greater speed than other machine learning-based approaches. Furthermore, with the added power of GPUs, both our method and the hyperprior model achieve nearly the highest efficiency among all the methods under comparison.

TABLE V
EXPERIMENTAL RESULTS FOR THE SEMANTIC SEGMENTATION TASK WITH VARIOUS COMPRESSION SCHEMES

Category	Method	Bit-Rate (bpp)↓	Cross Entropy↓	Acc.↑	Non-BG Acc.↑	mIoU↑
Baseline	Original	/	0.74	91.64%	86.28%	27.65%
	Control Group	/	0.61	92.31%	82.67%	27.07%
Feature Compression	IDFC (qp=51) [14]	0.020	6.22	92.74%	15.62%	11.03%
	IDFC (qp=43) [14]	0.026	1.38	93.94%	72.24%	28.41%
	Chen2020 (qp=0) [12]	**0.013**	3.39	93.35%	42.02%	18.61%
	Chen2020 (qp=43) [12]	0.022	0.95	93.41%	80.51%	30.11%
Codecs	BPG	0.024	1.91	90.18%	65.97%	19.84%
	Le2021-1 [37]	0.117	1.02	92.54%	70.01%	26.96%
	Le2021-2 [38]	0.088	1.12	92.83%	76.91%	28.22%
Learned Compression	Hyperprior [8]	0.025	0.80	91.95%	79.64%	25.42%
	Cheng2020 [15]	0.077	2.72	**96.16%**	63.30%	**31.26%**
Scalable Coding	Liu2021 [42]	0.631	3.82	92.64%	23.93%	14.79%
	Choi2022[17]	0.163	7.05	92.92%	7.6%	11.08%
Compressive Taskonomy	Ours	**0.013**	**0.77**	93.58%	**81.35%**	29.35%

Method IDFC is evaluated with different QPs, marked as IDFC (QP) in the table. ↑ means higher performance, better result, and ↓ vice versa. The best results are denoted in bold.



(a) IDFC (qp=51) vs. Ours (b) IDFC (qp=43) vs. Ours (c) Hyperprior vs. Ours

Fig. 7. Radar chart results on the semantic segmentation task with various compression schemes. The values of Bit-Rate (x_1) and Cross Entropy (x_2) are adjusted via $0.03 - x_1$ and $7 - x_2$ for better visibility, respectively.

TABLE VI
ANALYTICS PERFORMANCE OF MULTI-TASK JOINT COMPRESSION

Task	Metric	Ours	Cheng2020 [15]	Le2021-1 [37]	Le2021-2 [38]	BPG-51	BPG-47	BPG-43
Scene Class	Accuracy↑	55.76%	55.08%	61.23%	62.30%	48.73%	56.64%	61.91%
Semantic Seg.	mIoU↑	28.83%	34.36%	25.78%	26.72%	19.84%	24.36%	25.37%
Object Class	Accuracy↑	49.02%	48.54%	50.59%	51.46%	46.39%	50.10%	51.07%
Reconstruction	PSNR↑	-	-	23.80	25.80	27.80	29.95	31.97
Bit-Rate	Bpp↓	0.010	0.082	0.155	0.202	0.024	0.039	0.063
Normal	L_1 Distance↓	0.133	0.134	0.134	0.132	0.229	0.176	0.147
Reshading	L_1 Distance↓	0.241	0.241	0.256	0.254	0.410	0.334	0.305
Curvature	L_1 Distance↓	0.389	0.392	0.389	0.397	0.418	0.402	0.394
Reconstruction	PSNR↑	-	-	23.68	25.16	27.80	29.95	31.97
Bit-Rate	Bpp↓	0.033	0.106	0.154	0.132	0.024	0.039	0.063

TABLE VII
COMPLEXITY COMPARISONS OF DIFFERENT METHODS FOR COMPRESSING FEATURES WITHOUT CONSIDERING THE ANALYTICS MODELS

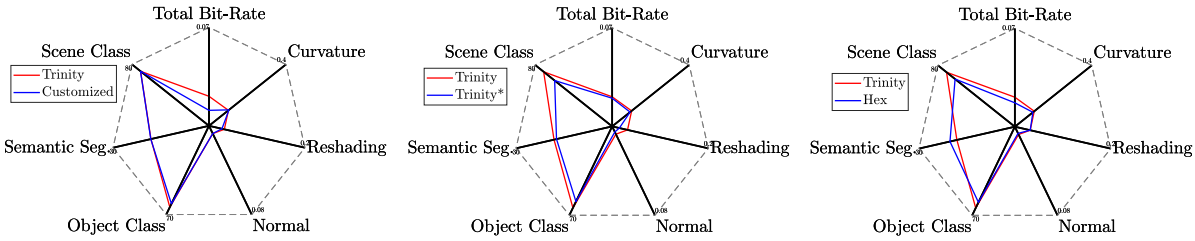
Metrics	IDFC	Chen2020	Hyperprior	Cheng2020
FLOPs	-	-	8.44G	20.41G
Param	-	-	32.97M	79.88M
Time (C)	0.041s	0.038s	0.099s	0.234s
Time (C+G)	-	-	4.32ms	17.05ms
Metrics	**-**	**Liu2021**	**Choi2021**	**Ours**
FLOPs	-	81.39G	19.30G	8.54G
Param	-	23.99M	50.14M	67.85M
Time (C)	-	0.649s	0.809s	0.107s
Time (C+G)	-	78.81ms	23.70ms	4.41ms

C denotes CPU and G denotes GPU.

D. Ablation Study

1) Task Group in Compressive Analytics Taxonomy: In this experiment, we compare the task group scheme in (inverse-)transfer mapping scheme with the customized group setting among multiple tasks. Several baselines are compared:

- *Customized:* Compressing feature maps for each task independently.
- *Hex:* Jointly compressing all six kinds of representations with one model;
- *Trinity:* An intuitively ideal compression setting that separates the six tasks into two groups for compression, i.e. **A**:



(a) Trinity vs. Customized (b) Trinity vs. Trinity* (c) Trinity vs. Hex

Fig. 8. Radar chart results on the multiple tasks in different evaluation measures. The values of the performance of normal (x_1), reshading (x_2), curvature (x_3), and total bit-rate (x_4) are adjusted via $0.08 - x_1$, $0.2 - x_2$, $0.4 - x_3$, and $0.07 - x_4$ for better visibility.

TABLE VIII
COMPLEXITY COMPARISONS OF DIFFERENT METHODS FOR IMAGE COMPRESSION, WHOSE RESULTS ARE FURTHER PROCESSED BY THE ANALYTICS MODELS TO OBTAIN THE RESULTS OF CERTAIN TASKS

Metrics	Le2021-1	Le2021-2	Hyperprior
FLOPs	46.73G	48.18G	20.71G
Param	42.86M	42.79M	73.08M
Time (C)	0.409s	0.435s	0.276s
Time (C+G)	28.56ms	49.21ms	15.80ms
Metrics	Cheng2020	BPG	Ours
FLOPs	37.31G	-	25.44G
Param	119.97G	-	107.96M
Time (C)	0.396s	0.396s	0.277s
Time (C+G)	29.49ms	284.16ms	15.81ms

In this scenario, our method is coupled with the analytics models to calculate the complexity. C denotes CPU and G denotes GPU.

Scene Class, Semantic Seg. and Object Class; **B**: Surface Normal, Reshading and Curvature;

- *Trinity^s:* A compression setting with a similar group partition to *Trinity*, but only changing the feature extractor/analytics model part and keeping the compression/decompression models unchanged;
- *Trinity*:* An intuitively non-optimal compression setting different from *Trinity*, which separates the six tasks into two groups for compression, i.e. **A**: Scene Class, Semantic Seg. and Object Class; **B**: Surface Normal, Reshading and Curvature.

The results are shown in Table IX and Fig. 8. The joint compression of multiple representations saves more bit-rate. When all tasks reach the plateau performance, the *Trinity* setting saves about 16.9% and 19.6% bit-rate than *Customized* and *Trinity** (the last row in Table IX), which are also reflected in an observation from Fig. 8 that *Trinity* owns a larger area than *Customized* and *Trinity**. However, a larger aggregation group affects the analytics performance. This may be because the information from the external tasks tends to act as additional noise for the focused task. By grouping similar tasks in one aggregation, higher analytics performance and lower bit-rate can be achieved. It is noted that, *Customized* does not necessarily lead to higher performance because the features of different tasks might include inter-task redundancy. Therefore, when the features of different tasks are optimized jointly, higher performance might be achieved by *Trinity* and *Hex*.

2) Comparison of supporting image reconstruction or not. We also look into the analytics performance of multi-task compression with/without image reconstruction in Table X. The results clearly demonstrate that, the version without including the visual reconstruction task achieves much better results in multi-task analytics than the one with. This observation is consistent with our benchmark result and confirms the rationality of our method design.

VI. DISCUSSION

A. Plateau Bit-Rate for Different Tasks

In this experiment, we train compression models for each task, respectively, and measure the bit-rate of the compressed feature representations. We search for the minimal bit-rate needed to support a task to its maximally achievable performance by the provided feature, i.e. to make the performance comparable to non-rate-control settings. The experiments involve the tasks of scene classification (Scene Class), semantic segmentation (Semantic Seg.), surface normal estimation of indoor scenes (Surface Normal), and reshading of an indoor image (Reshading).

The results are shown in Table XI. As shown, the performances of different tasks reach their plateau at different bit-rates, indicating that the information entropy to support a machine vision task varies among different tasks. Image-level analytics, e.g., classification, requires less bit-rate to support, while pixel-level analytics require more. There are also differences among pixel-level analytics.

We also show that *IDFC* consumes significantly more bit-rates. Besides, as *IDFC* involves a quantization based transform coding process, the quantization noise can result in unpredictable interference on the analytics performance. The results suggest that such quantization noise degrades the analytics performance more significantly on the geometry related tasks. Meanwhile, the proposed scheme provides better support for different kinds of tasks.

It is noted that, as *Original* is trained on the training set of the whole taskonomy dataset while other compared methods (*Control Group*, *Hyperprior*, and *Ours*) are trained/finetuned on the training set of a subset of taskonomy dataset, the proposed method might achieve better results than *Original*, which also demonstrates that our method has achieved an overall competitive performance.

B. Non-Plateau Bit-Rate for Different Tasks

Furthermore, we adjust the hyper-parameter λ to explore non-plateau bit-rate with reduced performance. The experiments involve the same four tasks as Section VI-A. The results are shown in Fig. 9. The curves demonstrate that there exist various

TABLE IX
ANALYTICS PERFORMANCE AND THE JOINT BIT-RATE *W.R.T* DIFFERENT AGGREGATION SCHEMES

Task	Metric	Original	Control Group	Customized	Trinity	Trinity*	Hex
Scene Class	Accuracy↑	70.02%	75.74%	**71.19%**	71.08%	59.64%	62.18%
Semantic Seg.	mIoU↑	18.37%	18.85%	18.19%	18.14%	17.36%	**20.30%**
Object Class	Accuracy↑	60.17%	60.02%	61.55%	**64.19%**	59.22%	59.75%
Normal	L_1 ↓	0.074	0.071	**0.073**	**0.073**	0.075	0.074
Reshading	L_1 ↓	0.221	0.172	0.173	**0.168**	0.185	**0.168**
Curvature	L_1 ↓	0.300	0.296	**0.296**	0.299	0.307	0.306
Total Bit-Rate	Bpp Sum ↓	/	/	0.059	**0.049**	0.050	0.053

The Customized, Trinity, Trinity* and Hex settings are as described in the main text. The best results are denoted in bold.

TABLE X
ANALYTICS PERFORMANCE OF MULTI-TASK COMPRESSION WITH/WITHOUT IMAGE RECONSTRUCTION

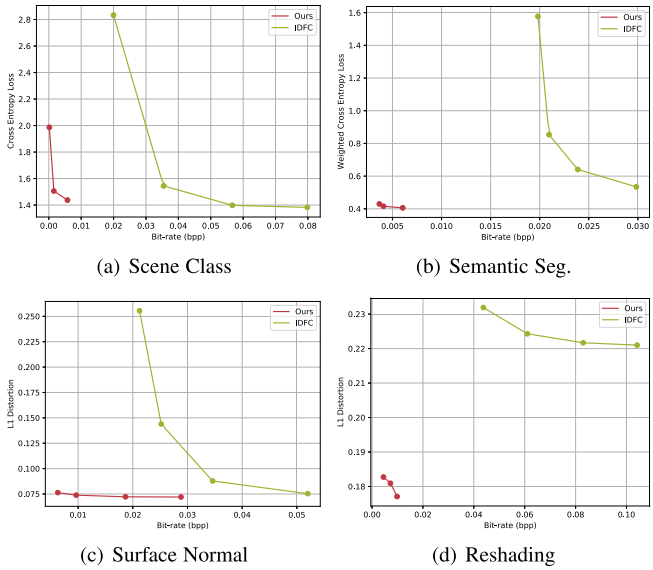
Task	Metric	Trinity	Trinity+
Scene Class	Accuracy↑	**71.08%**	64.62%
Semantic Seg.	mIoU↑	**18.14%**	11.49%
Object Class	Accuracy↑	**64.19%**	60.70%
Reconstruction	PSNR↑	/	31.05
Bit-Rate	Bpp↓	**0.016**	0.052
Normal	L_1 Distance↓	**0.073**	0.081
Reshading	L_1 Distance↓	**0.168**	0.185
Curvature	L_1 Distance↓	**0.299**	0.306
Reconstruction	PSNR↑	/	28.79
Bit-Rate	Bpp↓	**0.033**	0.054

The best results are denoted in bold.

TABLE XI
EVALUATION OF THE PLATEAU BIT-RATE FOR DIFFERENT TASKS WITH THE PROPOSED METHOD, IDFC AND HYPERPRIOR

Task	Method	Val. Perf.	Val. bpp	Test Perf.	Test bpp
Scene Class↑	Original	70.02%	/	67.48%	/
	Control Group	75.66%	/	62.70%	/
	IDFC	61.16%	0.0403	65.43%	0.0408
	Hyperprior	67.48%	0.0088	59.67%	0.0102
	Ours	71.11%	**0.0068**	59.47%	**0.0069**
Semantic Seg.↑	Original	18.37%	/	27.65%	/
	Control Group	18.85%	/	27.07%	/
	IDFC	17.20%	0.0210	28.41%	0.0261
	Hyperprior	19.02%	0.0104	25.42%	0.0250
	Ours	18.19%	**0.0072**	29.35%	**0.0131**
Surface Normal↓	Original	0.0741	/	0.1211	/
	Control Group	0.0700	/	0.1252	/
	IDFC	0.0753	0.0520	0.1281	0.0588
	Hyperprior	0.0718	0.0402	0.1288	0.0454
	Ours	0.0721	**0.0187**	0.1299	**0.0197**
Reshading↓	Original	0.2209	/	0.2836	/
	Control Group	0.1687	/	0.2343	/
	IDFC	0.2217	0.0830	0.2844	0.0959
	Hyperprior	0.1844	0.0134	0.2382	0.0138
	Ours	0.1713	**0.0130**	0.2411	**0.0134**

We present the validation set performance (Val. Perf.) and the test set performance (Test Perf.) along with the related bit-rate. Performances of different tasks are evaluated in different metrics. ↑ means higher performance metric, better result, and ↓ vice versa. The best results are denoted in bold.



(a) Scene Class (b) Semantic Seg.

(c) Surface Normal (d) Reshading

Fig. 9. Evaluation of the non-plateau bit-rate for different tasks with the proposed method and IDFC.

TABLE XII
EVALUATION OF FEATURE TRANSFERABILITY AMONG TASKS

Source Feature	Scene Class↑	Semantic Seg.↑	Surface Normal↓	Reshading↓
Scene Class	71.11%	11.45%	0.1413	0.258
	0.0068	0.0051	0.0077	0.0080
Semantic Seg.	51.27%	18.19%	0.1470	0.263
	0.0057	0.0072	0.0077	0.0095
Surface Normal	52.44%	12.47%	0.0721	0.176
	0.0038	0.0026	0.0187	0.0098
Reshading	46.82%	10.42%	0.0845	0.171
	0.0053	0.0023	0.0093	0.0130

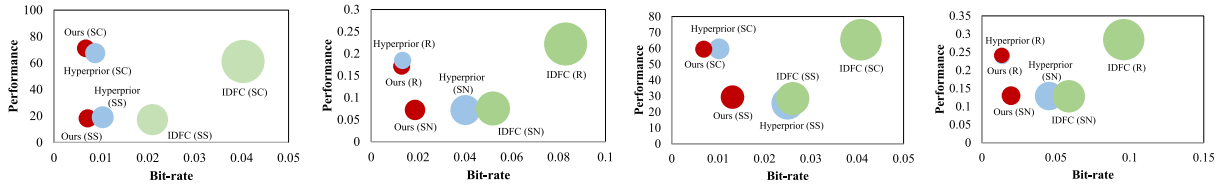
Each table entry include Metric at the top and bpp at the bottom. Adopted metrics include L_1 Distance for Normal and Reshading, Accuracy for Scene class and mIoU for Semantic Segmentation. The best results are denoted in red and the second results are denoted in blue.

R-D trade-offs when coding for different machine tasks, and with the proposed compression scheme, R-D performance of different tasks outperforms *IDFC* with various bit-rates. We also show the bubble chart results comparing *IDFC*, *hyperprior* and *Ours* in R-D performance of various tasks in Fig. 10. It is clearly demonstrated that, *Ours* occupies the smallest areas in bubbles, which shows that *Ours* generates the most compact representations. Besides, it is observed that, *Ours* also achieves comparable or better performance than other methods.

C. Feature Transferability Among Tasks

We also explore the transferability of features among different tasks under the bit-rate constraint. The feature extracted by the encoder of one task is transferred, compressed and then reconstructed to the output prediction by the decoder of another task. The experimental results are shown in Table XII. From the results, we obtain several interesting observations:

- The best performance is achieved when the feature is encoded and decoded by the models of the same task.
- Surface normal is the most generalized feature and its corresponding extracted feature achieves the second best results among different tasks when being applied to handle different tasks.
- The features of scene class and semantic segmentation cannot be generalized to handle surface normal and reshading tasks. The bit-rates of the transferred features are lower



(a) Validation ↑ (b) Validation ↓ (c) Testing ↑ (d) Testing ↓

Fig. 10. Bubble charts of multiple tasks in different evaluation measures. SC, SS, SN, and R short for scene class, semantic segmentation, surface normal, and reshading. ↑ denotes that a larger number signifies a better performance while ↓ denotes that a larger number signifies a worse performance. *The areas in bubbles visualize the bit-rate usage by different methods.*

than the other two tasks, which shows that scene class and semantic segmentation tasks include less information and cannot provide enough information to support the other two tasks.

- The features of surface normal and reshading can be generalized to handle scene class and semantic segmentation tasks, as they include richer information.

D. Task Relationship With Human Vision Task

We also explore feature representative capacities of different tasks when the human vision task, i.e. full pixel reconstruction, is involved in the task aggregation. The compared methods include *Trinity*, *Trinity+* (namely that the tasks in *Trinity* are further grouped with the full pixel reconstruction), and *BPG*, which compresses the images with BPG codec under different QP (43, 47, 51). The experimental results are shown in Table X. It is observed that, *Trinity+* consumes a much larger bit-rate while heavily degrading the analytics performance. BPG can indeed provide better visual reconstruction results. However, the analytics performance is also harmed.

E. Supporting Unseen Tasks

We further explore employing the compressive representation to support external tasks that are not used in R-D training. We conduct experiments with two *Trinity* groups as described in Section V-B, while we train the compression model only for two supervision tasks. The representation is used to train an external decoder for an unseen task.

Following the experimental setting in Section V-B, three training strategies are further adopted in comparisons:

- *Internal:* When evaluating an unseen task, i.e. object classification, the tasks *in an intuitively ideal group*, i.e. scene classification and semantic segmentation, are used for supervision. The same goes for the reshading task, where only the surface normal and curvature tasks are used for supervision.
- *External:* Evaluating in supporting unseen tasks *in intuitively non-optimal groups*, i.e. supporting the reshading task (object classification) via training with the tasks of scene classification and semantic segmentation (the surface normal and curvature).
- *Source+:* In some application scenarios, although the compression component cannot be supervised by an unseen task, *the pre-trained model and extracted feature at the encoder side for that task are available.* Thus, in this setting, the source feature for the unseen task is included in the compression but only the other two tasks are used for supervision.
- *BPG [7]:* The state-of-the-art image compression model that is task-independent.

TABLE XIII
EVALUATION OF COMPRESSION SCHEMES TO SUPPORT UNSEEN TASKS AT THE PLATEAU BIT-RATES

Representation	Bpp ↓	Object Class ↑	Bpp ↓	Reshading ↓
Original	/	60.17%	/	0.221
Internal	**0.0132**	51.06%	0.0229	**0.194**
External	**0.0132**	44.81%	0.0231	0.385
Source+	0.0137	53.50%	**0.0167**	0.205
BPG Image	0.0371	**54.56%**	0.0371	0.222

The best results are denoted in bold.

TABLE XIV
COMPRESSION RESULTS OF TAKING THE COMPRESSION MODELS TRAINED WITH DIFFERENT TASKS

Task	Metric	Customized	*Trinity*	*Trinity*^{s}
Scene Class	Accuracy↑	**71.19%**	71.08%	70.02%
Semantic Seg	mIoU↑	18.19%	18.14%	**23.26%**
Object Class	Accuracy↑	61.55%	**64.19%**	59.32%
Normal	L_1↓	**0.073**	**0.073**	0.074
Reshading	L_1↓	0.173	**0.168**	0.184
Curvature	L_1↓	**0.296**	0.299	0.306
Total Bit-Rate	Bpp Sum↓	0.059	**0.049**	0.063

The Customized, Trinity, Trinity* and Hex settings are as described in the main text. The best results are denoted in bold.

The results on the validation set are shown in Table XIII. As shown, the proposed method can generate compressed visual representations that support external unseen tasks, achieving better performance than utilizing image compression methods. The results (*Source+* versus *Internal* and *External*) also indicate that including the additional feature representation at the encoder side can further bring in further performance gains, e.g., improving classification accuracy for object classification and reducing the bpp for reshading, although R-D optimization is not performed for that task.

F. Unified/Specified Compression/Decompression Models for Multiple Tasks

We have compared our method with another version that adopts a unified compression/decompression model for different tasks as shown in Table XIV. *Trinity*^{s} is a compression setting with a similar group partition to *Trinity*, but only changing the feature extractor/analytics model part and keeping the compression/decompression models unchanged. The results show that, our unified compression/decompression models obtain comparative performance with our proposed method (*Trinity*),

TABLE XV
COMPRESSION RESULTS OF OUR FRAMEWORK WITH THE FIDELITY CONSTRAINT

Test	Evaluation Task	Metric	Train	
			S/S/O	N/R/C
S/S/O	Scene Class	Accuracy↑	71.61%	68.11%
	Semantic Seg	mIoU↑	25.42%	19.82%
	Object Class	Accuracy↑	56.89%	60.59%
	Total Bit-Rate	Bpp Sum↓	0.014	0.036
N/R/C	Normal	L_1 Distance↓	0.079	0.073
	Reshading	L_1 Distance↓	0.225	0.165
	Curvature	L_1 Distance↓	0.302	0.301
	Total Bit-Rate	Bpp Sum↓	0.027	0.031

S/S/O and N/R/C denote that the models are trained with the task bundle respectively (Scene Classification, Semantic Segmentation, and Object Classification) and (Normal, Reshading, and Curvature), whose downstream inference models are known. In training, features of all these features are assumed to be obtained.

which shows our models can be shared among different tasks. It is proved that our method is sufficiently generalized and can actually be adapted to a wide range of visual tasks.

G. Introducing Fidelity Constraints on Features for Generalization

To better support generalization, we make a small improvement to the original framework via enforcing the signal fidelity constraint on the incoming fused feature of the unseen tasks. For the unseen tasks, we assume that, the downstream inference models are unknown, while their features can be obtained, which is also in line with the needs of real application scenarios of feature transmission in smart cities. The loss function in (11) then can be further revised as follows,

$$\mathcal{L} = B\left(R_z\right) + \lambda \sum_{0 \le i \le L} \omega_i l_i + \delta \| F_{Fuse} - \widehat{F}_{Fuse} \|, \quad (12)$$

where λ and δ are the parameters that measure the importance of different terms. F_{Fuse} is the fused feature of F_0 and F_1 (extracted from unseen tasks) where $\widehat{F}_{Fuse}$ is the reconstructed feature that will be split/converted into F_0 and F_1. The constraint equation can be interpreted as the joint optimization of downstream performance on known tasks while imposing a fidelity constraint on the features of unknown tasks, for which the inference models are unseen.

As shown in Table XV, the results clearly demonstrate that, with minor modifications, our method can also show very competitive performance on unseen tasks. For the model trained on S/S/O group (S/S/O's downstream models and N/R/C's features), the model achieves state-of-the-art performance for both S/S/O and N/R/C tasks compared to the models trained on the specified datasets. This is achieved by guidance from downstream models and features associated with the unseen tasks without retraining. These experimental results demonstrate that our approach can effectively handle unseen tasks after training on existing tasks. For training on N/R/C group (N/R/C's downstream models and S/S/O's features), the performance is also competitive for both S/S/O and N/R/C tasks with increased bit-rates. This might be caused by the fact that the information from the input data is redundant, so it becomes more difficult to construct a compact feature representation.

VII. CONCLUSION AND FUTURE DIRECTIONS

This paper formulates and summarizes the problem and solutions of video coding for machines (VCM) in recent years, targeting the collaborative optimization of compressing and transmitting multiple tasks/applications. Several state-of-the-art categories of methods, including feature assisted coding, scalable coding, intermediate feature compression/optimization, and machine vision targeted codec, are reviewed comprehensively. After reviewing existing methods, we raise the new paradigm of compressive analytics taxonomy for VCM, where multi-task performance is revisited under the compression constraint. In particular, we propose a codebook hyperprior to compress the neural network generated features for multi-task applications/tasks. The codebook design helps reduce the dimensionality gap between pixels and features, and an (inverse-)transfer mapping is equipped to generate a unified compact representation. The experiments show the superiority of our codebook-based hyperprior model in handling multi-task applications compared to previous works, which shows a new research/solution direction for VCM.

For the future research, several pending issues need more attention:

- *Joint optimization of video, feature and model streams:* Existing methods mainly focus on video streams and feature streams. As demonstrated in [44], it also has the potential to involve the model in the optimization, as the knowledge might be better reused in the space of the model's parameters, which spans a more representative space and leads to discriminative capability.
- *Theoretic investigation in the relationship of human/machine vision:* As claimed in our work, the reconstruction of full pixels leads to significantly higher bit-rate usage. It is still absent how we make a trade-off between them in different scenarios, and how they correlate and conflict with each other in theory.
- *Consideration in Decoding Complexity:* One of the most important motivations in VCM is "collaborative intelligence" [5] that aims to reduce the burden on the decoder side. However, few works really embody the decoding complexity in the optimization, which shows a critical direction for the future VCM.

In summary, existing VCM efforts bring in abundant practices, including paradigms, solutions, techniques, and systems, and more future endeavours to improve existing methods and explore new directions are expected.

REFERENCES

[1] M. Akbari, J. Liang, and J. Han, "DSSLIC: Deep semantic segmentation-based layered image compression," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2019, pp. 2042–2046.
[2] S.R. Alvar and I. V. Bajić, "Multi-task learning with compressible features for collaborative intelligence," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 1705–1709.
[3] S.R. Alvar and I. V. Bajić, "Bit allocation for multi-task collaborative intelligence," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2020, pp. 4342–4346.
[4] S.R. Alvar and I. V. Bajić, "Pareto-optimal bit allocation for collaborative intelligence," *IEEE Trans. Image Process.*, vol. 30, pp. 3348–3361, 2021.
[5] I. V. Bajic, W. Lin, and Y. Tian, "Collaborative intelligence: Challenges and opportunities," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Process.*, 2021, pp. 8493–8497.

[6] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Representations*, 2018, pp. 1–13.

[7] F. Bellard, "BPG image format," Accessed: May 28, 2021, 2014. [Online]. Available: http://bellard.org/bpg/

[8] L. D. Chamain, F. Racapé, J. Bégaint, A. Pushparaja, and S. Feltman, "End-to-end optimized image compression for machines, a study," in *Proc. IEEE Data Compression Conf.*, 2021, pp. 163–172.

[9] J. Chang et al., "Layered conceptual image compression via deep semantic synthesis," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 694–698.

[10] J. Chang, Z. Zhao, L. Yang, C. Jia, J. Zhang, and S. Ma, "Thousand to one: Semantic prior modeling for conceptual coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6.

[11] D. Chen, Q. Chen, and F. Zhu, "Pixel-level texture segmentation based AV1 video compression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2019, pp. 1622–1626.

[12] Z. Chen, L.-Y. Duan, S. Wang, W. Lin, and A. C. Kot, "Data representation in hybrid coding framework for feature maps compression," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 3094–3098.

[13] Z. Chen, K. Fan, S. Wang, L.-Y. Duan, W. Lin, and A. Kot, "Lossy intermediate deep learning feature compression and evaluation," in *Proc. 27th ACM Int. Conf. ACM Trans. Multimedia*, 2019, pp. 2414–2422.

[14] Z. Chen, K. Fan, S. Wang, L.-Y. Duan, W. Lin, and A. C. Kot, "Toward intelligent sensing: Intermediate deep feature compression," *IEEE Trans. Image Process.*, vol. 29, pp. 2230–2243, 2019.

[15] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 7936–7945.

[16] H.C. Chun, P. Nguyen, and C.-T. Lai, "Learned prior information for image compression," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2019, pp. 7936–7945.

[17] H. Choi and I. V. Bajić, "Scalable image coding for humans and machines," *IEEE Trans. Image Process.*, vol. 31, pp. 2739–2754, 2022.

[18] H. Choi, R. A. Cohen, and I. V. Bajić, "Back-and-forth prediction for deep tensor compression," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 4467–4471.

[19] J. Choi and B. Han, "Task-aware quantization network for JPEG image compression," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2020, pp. 309–324.

[20] T. M. Cover, *Elements of Information Theory*. Hoboken, NJ, USA: Wiley, 1999.

[21] L. Duan, J. Liu, W. Yang, T. Huang, and W. Gao, "Video coding for machines: A paradigm of collaborative compression and intelligent analytics," *IEEE Trans. Image Process.*, vol. 29, pp. 8680–8695, 2020.

[22] W. Gao, S. Liu, X. Xu, M. Rafie, Y. Zhang, and I. Curcio, "Recent Standard Development Activities on Video Coding for Machines. arXiv e-prints," May 2021, *arXiv:2105.12653*.

[23] W. Gao et al., "Digital retina: A way to make the city brain more efficient by visual coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 11, pp. 4147–4161, Nov. 2021.

[24] R. Girshick, "Fast R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2015, pp. 1440–1448.

[25] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[26] WG2 Group, "Draft of white paper on motivation and requirements for video coding for machine," MPEG Technical requirements ISO/IEC JTC 1/SC 29/WG 2:1–13, 2021.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.

[28] T. M. Hoang, J. Zhou, and Y. Fan, "Image compression with encoder-decoder matched semantic segmentation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit. Workshops*, 2020, pp. 619–623.

[29] Y. Hou, L. Zheng, and S. Gould, "Learning to structure an image with few colors," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 10113–10122.

[30] Y. Hu, S. Yang, W. Yang, L.-Y. Duan, and J. Liu, "Towards coding for human and machine vision: A scalable image coding approach," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2020, pp. 1–6.

[31] Y. Hu, S. Xia, W. Yang, and J. Liu, "Sensitivity-aware bit allocation for intermediate deep feature compression," in *Proc. IEEE Vis. Commun. Image Process.*, 2020, pp. 475–478.

[32] Z. Huang, C. Jia, S. Wang, and S. Ma, "Visual analysis motivated rate-distortion model for image coding," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6.

[33] A. Ikusan and R. Dai, "Rate-distortion optimized hierarchical deep feature compression," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6.

[34] T. Karras, S. Laine, M. Aittala, J. Hellsten, J. Lehtinen, and T. Aila, "Analyzing and improving the image quality of StyleGAN," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8107–8116.

[35] S. Kim et al., "Adversarial video compression guided by soft edge detection," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 2193–2197.

[36] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1106–1114.

[37] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, and E. Rahtu, "Image coding for machines: An end-to-end learned approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 1590–1594.

[38] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, H.R. Tavakoli, and E. Rahtu, "Learned image coding for machines: A content-adaptive approach," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6.

[39] H. Li, Y. Guo, Z. Wang, S. Xia, and W. Zhu, "AdaCompress: Adaptive compression for online computer vision services," in *Proc. Int. Conf. ACM Trans. Multimedia*, 2019, pp. 2440–2448.

[40] J. Lin, R. Zhang, F. Ganz, S. Han, and J.-Y. Zhu, "Enhancing unsupervised video representation learning by decoupling the scene and the motion," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 10129–10137.

[41] T.-Y. Lin et al., "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 740–755.

[42] K. Liu, D. Liu, L. Li, N. Yan, and H. Li, "Semantics-to-signal scalable image compression with learned revertible representations," *Int. J. Comput. Vis.*, vol. 129, no. 9, pp. 2605–2621, 2021.

[43] F. Locatello et al., "Challenging common assumptions in the unsupervised learning of disentangled representations," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 4114–4124.

[44] Y. Lou et al., "Towards efficient front-end visual sensing for digital retina: A model-centric paradigm," *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 3002–3013, Nov. 2020.

[45] S. Ma, T. Huang, C. Reader, and W. Gao, "AVS2? Making video coding smarter [standards in a nutshell]," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 172–183, Mar. 2015.

[46] M. Oquab et al., "Low bandwidth video-chat compression using deep generative models," Dec. 2020, *arXiv:2012.00328*.

[47] A. Orzan, A. Bousseau, P. Barla, and J. Thollot, "Structure-preserving manipulation of photographs," in *Proc. Int. Symp. Non-Photorealistic Animation Rendering*, 2007, pp. 103–110.

[48] N. Patwa, N. Ahuja, S. Somayazulu, O. Tickoo, S. Varadarajan, and S. Koolagudi, "Semantic-preserving image compression," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 1281–1285.

[49] R. Prabhakar et al., "Reducing latency and bandwidth for video streaming using keypoint extraction and digital puppetry," in *Proc. Data Compression Conf.*, 2021, pp. 1–11.

[50] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[51] M. A. Shah and B. Raj, "Deriving compact feature representations via annealed contraction," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2020, pp. 2068–2072.

[52] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–14.

[53] S. Singh, S. Abu-El-Haija, N. Johnston, J. Ballé, A. Shrivastava, and G. Toderici, "End-to-end learning of compressible features," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 3349–3353.

[54] S. Singh, S. Abu-El-Haija, N. Johnston, J. Ballé, A. Shrivastava, and G. Toderici, "End-to-end learning of compressible features," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 3349–3353.

[55] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

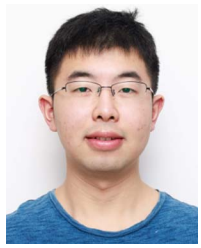
[56] S. Suzuki, M. Takagi, K. Hayase, T. Onishi, and A. Shimizu, "Image pre-transformation for recognition-aware image compression," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 2686–2690.

[57] S. Suzuki, M. Takagi, S. Takeda, R. Tanida, and H. Kimata, "Deep feature compression with spatio-temporal arranging for collaborative intelligence," in *Proc. IEEE Int. Conf. Image Process.*, 2020, pp. 3099–3103.

[58] M. Ulhaq and I. V. Bajić, "Latent space motion analysis for collaborative intelligence," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 8498–8502.
[59] S. Vandenhende, S. Georgoulis, W. Van Gansbeke, M. Proesmans, D. Dai, and L. Van Gool, "Multi-task learning for dense prediction tasks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3614–3633, Jul. 2022.
[60] S. Wang, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Scalable facial image compression with deep feature reconstruction," in *Proc. IEEE Int. Conf. Image Process.*, 2019, pp. 2691–2695.
[61] S. Wang, S. Wang, W. Yang, X. Zhang, S. Wang, and S. Ma, "Teacher-student learning with multi-granularity constraint towards compact facial feature representation," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, 2021, pp. 8503–8507.
[62] S. Wang et al., "Towards analysis-friendly face representation with scalable feature and texture compression," *IEEE Trans. Multimedia*, vol. 24, pp. 3169–3181, 2022.
[63] T.-C. Wang, A. Mallya, and M.-Y. Liu, "One-shot free-view neural talking-head synthesis for video conferencing," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 10039–10049.
[64] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[65] Q. Xia, H. Liu, and Z. Ma, "Object-based image coding: A learning-driven revisit," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2020, pp. 1–6.
[66] S. Xia, K. Liang, W. Yang, L.-Y. Duan, and J. Liu, "An emerging coding paradigm VCM: A scalable coding approach beyond feature and signal," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2020, pp. 1–6.
[67] Z. Xie, Y. Lin, Z. Zhang, Y. Cao, S. Lin, and H. Hu, "Propagate yourself: Exploring pixel-level consistency for unsupervised visual representation learning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 16679–16688.
[68] P. Xing, P. Peng, Y. Liang, T. Huang, and Y. Tian, "Binary representation and high efficient compression of 3D CNN features for action recognition," in *Proc. Data Compression Conf.*, 2020, pp. 400–400.
[69] N. Yan, D. Liu, H. Li, and F. Wu, "Semantically scalable image coding with compression of feature maps," in *Proc. IEEE Int. Conf. Image Process.*2020, pp. 3114–3118.
[70] S. Yang, Y. Hu, W. Yang, L.-Y. Duan, and J. Liu, "Towards coding for human and machine vision: Scalable face image coding," *IEEE Trans. Multimedia*, vol. 23, pp. 2957–2971, 2021.
[71] Z. Yang et al., "Discernible image compression," in *Proc. ACM Trans. Multimedia*, 2020, pp. 1561–1569.
[72] F. Yu et al., "BDD100K: A diverse driving dataset for heterogeneous multitask learning," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 2633–2642.
[73] R. Amir et al., "Taskonomy: Disentangling task transfer learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3712–3722.
[74] J. Zhang and D. Tao, "Empowering things with intelligence: A survey of the progress, challenges, and opportunities in artificial intelligence of things," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7789–7817, May 2021.
[75] Z. Zhang, M. Wang, M. Ma, J. Li, and X. Fan, "MSFC: Deep feature compression in multi-task network," in *Proc. IEEE Int. Conf. Multimedia Expo*, 2021, pp. 1–6.
[76] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 2242–2251.

Wenhan Yang (Member, IEEE) received the BS and PhD degrees (Hons.) in computer science from Peking University, Beijing, China, in 2012 and 2018, respectively. He is currently an associate researcher with Peng Cheng Laboratory, Shenzhen, Guangdong, China. His current research interests include image/video processing/restoration, bad weather restoration, human-machine collaborative coding. He has authored more than 50 technical articles in refereed journals and proceedings, and holds nine granted patents. He received the 2023 IEEE Multimedia Rising Star Runner-Up Award, the IEEE ICME-2020 Best Paper Award, the IFTC 2017 Best Paper Award, the IEEE CVPR-2018 UG2 Challenge First Runner-up Award, and the MSA-TC Best Paper Award of ISCAS 2022. He was the candidate of CSIG Best Doctoral Dissertation Award in 2019. He served as the area chair of IEEE ICME-2021 and 2022, the session chair of IEEE ICME-2021, and the organizer of IEEE CVPR-2019/2020/2021 UG2+ Challenge and Workshop.

Haofeng Huang received the BS degree in computer science from Peking University, Beijing, China in 2021. He is currently working toward the PhD degree with the Wangxuan Institute of Computer Technology. His current research interests include deep-learning based image/video compression, image/video coding for machines, and intelligent visual enhancement.



Yueyu Hu (Student Member, IEEE) received the BS and MS degrees in computer science from Peking University, Beijing, China, in 2018 and 2021, respectively. He is currently working toward the PhD degree with New York University, New York, NY. His current research interests include machine learning inspired 2D and 3D image compression and processing. He received the Best Paper Award at IEEE-ICME 2020.

Ling-Yu Duan (Member, IEEE) received the PhD degree in information technology from the University of Newcastle, Callaghan, NSW, Australia, in 2008. He is currently a full professor with the National Engineering Laboratory of Video Technology, School of Electronics Engineering and Computer Science, Peking University, Beijing, China, and since 2012, he has been the associate director of the Rapid-Rich Object Search Laboratory, a joint lab between Nanyang Technological University, Singapore, and Peking University. Since 2019, he has been with Peng Cheng Laboratory, Shenzhen, China. He has authored or coauthored about 200 research papers. His research interests include multimedia indexing, search, and retrieval, mobile visual search, visual feature coding, and video analytics. He was the co-editor of the MPEG Compact Descriptor for Visual Search Standard (ISO/IEC 15938-13) and MPEG Compact Descriptor for Video Analytics standard (ISO/IEC 15938-15). He is currently an associate editor for *IEEE Transactions on Multimedia*, *ACM Transactions on Intelligent Systems and Technology* and *ACM Transactions on Multimedia Computing, Communications, and Applications*, and the area chair of the ACM MM and IEEE ICME. He is a member of the MSA Technical Committee in IEEE-CAS Society. He was the recipient of the IEEE ICME best paper awards, in 2020 and 2019, the IEEE VCIP best paper award, in 2019, EURASIP Journal on Image and Video Processing Best Paper Award, in 2015, the Ministry of Education Technology Invention Award (First Prize), in 2016, the National Technology Invention Award (Second Prize), in 2017, the China Patent Award for Excellence, in 2017, and the National Information Technology Standardization Technical Committee Standardization Work Outstanding Person Award, in 2015.

Jiaying Liu (Senior Member, IEEE) received the PhD degree (Hons.) in computer science from Peking University, Beijing China, in 2010. She is currently an associate professor, Boya Young fellow with the Wangxuan Institute of Computer Technology, Peking University, China. She has authored more than 100 technical articles in refereed journals and proceedings, and holds 70 granted patents. Her current research interests include multimedia signal processing, compression, and computer vision. She is a senior member of CSIG, and a distinguished member of CCF. She was a visiting scholar with the University of Southern California, Los Angeles, California, from 2007 to 2008. She was a visiting researcher with Microsoft Research Asia, in 2015 supported by the Star Track Young Faculties Award. She has served as a member of Multimedia Systems and Applications Technical Committee (MSA TC), and Visual Signal Processing and Communications Technical Committee (VSPC TC) in IEEE Circuits and Systems Society. She received the IEEE ICME 2020 Best Paper Award and IEEE MMSP 2015 Top10% Paper Award. She has also served as the associate editor of the *IEEE Transaction on Image Processing*, *IEEE Trans. on Circuits Systems for Video Technology* and *Journal of Visual Communication and Image Representation*, the technical program chair of ACM MM Asia-2023/IEEE ICME-2021/ACM ICMR-2021/IEEE VCIP-2019, the area Cchair of CVPR-2021/ECCV-2020/ICCV-2019, ACM ICMR Steering Committee member and the CAS representative with the ICME Steering Committee. She was the APSIPA distinguished lecturer (2016-2017).