Video Coding for Machines: A Paradigm of Collaborative Compression and Intelligent Analytics

Ling-Yu Duan, Jiaying Liu, Wenhan Yang, Tiejun Huang, Wen Gao

Abstract—Video coding, which targets to compress and reconstruct the whole frame, and feature compression, which only preserves and transmits the most critical information, stand at two ends of the scale. That is, one is with compactness and efficiency to serve for machine vision, and the other is with full fidelity, bowing to human perception. The recent endeavors in imminent trends of video compression, *e.g.* deep learning based coding tools and end-to-end image/video coding, and MPEG-7 compact feature descriptor standards, *i.e.* Compact Descriptors for Visual Search and Compact Descriptors for Video Analysis, promote the sustainable and fast development in their own directions, respectively. In this paper, thanks to booming AI technology, *e.g.* prediction and generation models, we carry out exploration in the new area, Video Coding for Machines (VCM), arising from the emerging MPEG standardization efforts[1]. Towards collaborative compression and intelligent analytics, VCM attempts to bridge the gap between feature coding for machine vision and video coding for human vision. Aligning with the rising *Analyze then Compress* instance Digital Retina, the definition, formulation, and paradigm of VCM are given first. Meanwhile, we systematically review state-of-the-art techniques in video compression and feature compression from the unique perspective of MPEG standardization, which provides the academic and industrial evidence to realize the collaborative compression of video and feature streams in a broad range of AI applications. Finally, we come up with potential VCM solutions, and the preliminary results have demonstrated the performance and efficiency gains. Further direction is discussed as well.

Index Terms—Video coding for machine, video compression, feature compression, generative model, prediction model

I. INTRODUCTION

In the big data era, massive videos are fed into machines to realize intelligent analysis in numerous applications of smart cities or Internet of things (IoT). Like the explosion of surveillance systems deployed in urban areas, there arise important concerns on how to efficiently manage massive video data. There is a unique set of challenges (*e.g.* low latency and high accuracy) regarding efficiently analyzing and searching the target within the millions of objects/events captured everyday. In particular, video compression and transmission constitute the basic infrastructure to support these applications from the perspective of *Compress then Analyze*. Over the past decades, a series of standards (*e.g.* MPEG-4 AVC/H.264 [1] and High Efficiency Video Coding (HEVC) [2]), Audio Video coding Standard (AVS) [3] are built to significantly improve the video coding efficiency, by squeezing out the spatial-temporal pixel-level redundancy of video frames based on the visual signal statistics and the priors of human perception. More recently, deep learning based video coding makes great progress. With the hierarchical model architecture and the large-scale data priors, these methods largely outperform the state-of-the-art codecs by utilizing deep-network aided coding tools. Rather than directly targeting machines, these methods focus on efficiently reconstructing the pixels for human vision, in which the spatial-temporal volume of pixel arrays can be fed into machine learning and pattern recognition algorithms to complete high-level analysis and retrieval tasks.

However, when facing big data and video analytics, existing video coding methods (even for the deep learning based) are still questionable, regarding whether such big video data can be efficiently handled by visual signal level compression. Moreover, the full-resolution videos are of low density in practical values. It is prohibitive to compress and store all video data first and then perform analytics over the decompressed video stream. By degrading the quality of compressed videos, it might save more bitrates, but incur the risk of degraded analytics performance due to the poorly extracted features.

To facilitate the high-level machine vision tasks in terms of performance and efficiency, lots of research efforts have been dedicated to extracting those pieces of key information, *i.e.*, visual features, from the pixels, which is usually compressed and represented in a very compact form. This poses an alternative strategy *Analyze then Compress*, which extracts, saves, and transmits compact features to satisfy various intelligent video analytics tasks, by using significantly less data than the compressed video itself. In particular, to meet the demand for large-scale video analysis in smart city applications, the feature stream instead of the video signal stream can be transmitted. In view of the necessity and importance of transmitting feature descriptors, MPEG has finalized the standardization of compact descriptors for visual search (CDVS) (ISO/IEC15938-13) in Sep. 2015 [16] and compact descriptors for video analysis (CDVA) (ISO/IEC15938-15) [17] in July 2019 to enable the interoperability for efficient and effective image/video retrieval and analysis by standardizing the bitstream syntax of compact feature descriptors. In CDVS, hand-crafted local and global descriptors are designed to represent the visual characteristics of images. In CDVA, the deep learning features are adopted to further boost the video analysis performance. Over the course of the standardization process, remarkable improvements are achieved in reducing the size of features while maintaining their discriminative power for machine vision tasks. Such compact features cannot reconstruct the full resolution videos

L.-Y. Duan, J. Liu, W. Yang, T. Huang, and W. Gao are with Peking University, Beijing 100871, China (e-mail: lingyu@pku.edu.cn; liujiaying@pku.edu.cn; yangwenhan@pku.edu.cn; tjhuang@pku.edu.cn; wgao@pku.edu.cn).

[1]https://lists.aau.at/mailman/listinfo/mpeg-vcm

for human observers, thereby incurring two successive stages of analysis and compression for machine and human vision.

For either *Compress then Analyze* or *Analyze then Compress*, the optimization jobs of video coding and feature coding are separate. Due to the very nature of multi-tasks for machine vision and human vision, the intrusive setup of two separate stages is sub-optimal. It is expected to explore more collaborative operations between video and feature streams, which opens up more space for improving the performance of intelligent analytics, optimizing the video coding efficiency, and thereby reducing the total cost. Good opportunities to bridge the cross-domain research on machine vision and human vision have been there, as deep neural network has demonstrated its excellent capability of multi-task end-to-end optimization as well as abstracting the representations of multiple granularities in a hierarchical architecture.

In this paper, we attempt to identify the opportunities and challenges of developing collaborative compression techniques for humans and machines. Through reviewing the advance of two separate tracks of video coding and feature coding, we present the necessity of machine-human collaborative compression, and formulate a new problem of video coding for machines (VCM). Furthermore, to promote the emerging MPEG-VCM standardization efforts and collect for evidences for MPEG Ad hoc Group of VCM, we propose trial exemplar VCM architectures and conduct preliminary experiments, which are also expected to provide insights into bridging the cross-domain research from visual signal processing, computer vision, and machine learning, when AI meets the video big data. The contributions are summarized as follows,

- We present and formulate a new problem of Video Coding for Machines by identifying three elements of *tasks, features*, and *resources*, in which a novel feedback mechanism for *collaborative* and *scalable* modes is introduced to improve the coding efficiency and the analytics performance for both human and machine-oriented tasks.
- We review the state-of-the-art approaches in video compression and feature compression from a unique perspective of standardized technologies, and study the impact of more recent deep image/video prediction and generation models on the potential VCM related techniques.
- We propose exemplar VCM architectures and provide potential solutions. Preliminary experimental results have shown the advantages of VCM collaborative compression in improving video and feature coding efficiency and performance for human and machine vision.

The rest of the article is organized as follows. Section II and III briefly review previous works on video coding and feature compression, respectively. Section IV provides the definition, formulation, and paradigm of VCM. Section V illustrates the emerging AI technique, which provides useful evidence for VCM, After that, in Section VII, we provide potential solutions for VCM problem. In Section VII, the preliminary experimental results are reported. In Section VIII, several issues, and future directions are discussed. In Section IX, the concluding remarks are provided.

II. REVIEW OF VIDEO COMPRESSION: FROM PIXEL FEATURE PERSPECTIVE

Visual information takes up at least 83% of all information [18] that people can feel. It is important for humans to record, store, and view the image/videos efficiently. For past decades, lots of academic and industrial efforts have been devoted to video compression, which is to maximize the compression efficiency from the pixel feature perspective. Below we review the advance of traditional video coding as well as the impact of deep learning based compression on visual data coding for human vision in a general sense.

A. Traditional Hybrid Video Coding

Video coding transforms the input video into a compact binary code for more economic and light-weighted storage and transmission, and targets reconstructing videos visually by the decoding process. In 1975, the *hybrid spatial-temporal coding architecture* [20] is proposed to take the lead and occupy the major proportion during the next few decades. After that, the following video coding standards have evolved through the development of the ITU-T and ISO/IEC standards. The ITU-T produced H.261 [21] and H.263 [22], ISO/IEC produced MPEG-1 [23] and MPEG-4 Visual [24], and the two organizations worked together to produce the H.262/MPEG-2 Video [25], H.264/MPEG-4 Advanced Video Coding (AVC) [26] standards, and H.265/MPEG-H (Part 2) High Efficiency Video Coding (HEVC) [27] standards.

The design and development of all these standards follows the *block-based video coding* approach. The first technical feature is block-wise partition. Each coded picture is partitioned into macroblocks (MB) of luma and chroma samples. MBs will be divided into slices and coded independently. Each slice is further partitioned into coding tree units. After that, the coding unit (CU), prediction unit (PU), and transform unit (TU) are obtained to make the coding, prediction and transform processes more flexible. Based on the block-based design, the intra and inter-predictions are applied based on PU and the corresponding contexts, *i.e.* neighboring blocks and reference frames in the intra and inter modes, respectively. But these kinds of designed patterns just cover parts of the context information, which limits the modeling capacity in prediction. Moreover, the block-wise prediction, along with transform and quantization, leads to the discontinuity at the block boundaries. With the quantization of the residue or original signal in the transform block, the blockness appears. To suppress the artifacts, the loop filter is applied for smoothing.

Another important technical feature is *hybrid video coding*. Intra and inter-prediction are used to remove temporal and spatial statistical redundancies, respectively. For intra-prediction, HEVC utilizes a line of preceding reconstructed pixels above and on the left side of the current PU as the reference for generating predictions. The number of intra modes is 35, including planar mode, DC mode, and 33 angular modes. It is performed in the transform unit. For inter-prediction, HEVC derives a motion-compensated prediction for a block of image samples. The homogeneous motion inside a block is assumed, and the size of a moving object is usually larger than one block. Reference blocks will be searched from previously

coded pictures for inter prediction. For both intra and inter-predictions, the best mode is selected by the Rate-Distortion Optimization (RDO) [28]. However, the multi-line prediction scheme and the block-wise reference block might not provide a desirable prediction reference when the structures and motion are complex. Besides, when the RDO boosts the modeling capacity, the overhead of signaled bits and computation occur. The target of optimizing the rate distortion efficiency is to seek for the trade-off in bitrate and signal distortion. It can be solved via Lagrangian optimization techniques. Coder control finally determines a set of coding parameters that affect the encoded bit-streams.

With tremendous expert efforts, the coding performance is dramatically improved in the past decade that there is the rule of thumb that, one generation of video coding standards almost surpasses the previous one by up to 50% in coding efficiency.

B. Deep Learning Based Video Coding

The deep learning techniques significantly promote the development of video coding. The seminar work [29] in 2015 opened a door to the end-to-end learned video coding. The deep learning based coding tools [30, 31] are developed since 2016. Benefiting from the bonus of big data, powerful architectures, end-to-end optimization, and other advanced techniques, *e.g.* unsupervised learning, the emerging deep learning excel in learning data-driven priors for effective video coding. First, complex nonlinear mappings can be modeled by the hierarchical structures of neural networks, which improves prediction models to make the reconstructed signal more similar to the original one. Second, deep structures, such as PixelCNN and PixelRNN, are capable to model the pixel probability and provide powerful generation functions to model the visual signal in a more compact form.

The deep learning based coding methods do not rely on the partition scheme and support full resolution coding, and thereby removing the blocking artifacts naturally. Since the partition is not required, these modules can access more context in a larger region. As the features are extracted via a hierarchical network and jointly optimized with the reconstruction task, the resulting features tend to comprehensive and powerful for high efficient coding. More efforts are put into increasing the receptive field for better perception of larger regions via recurrent neural network [32, 33] and nonlocal attention [34, 35], leading to improved coding performance. The former infers the latent representations of image/videos progressively. In each iteration, with the previously perceived context, the network removes the unnecessary bits from the latent representations to achieve more compactness. The latter makes efforts to figure out the nonlocal correspondence among endpixels/regions, in order to remove the long-term spatial redundancy.

The efforts in improving the performance of neural network-aided coding tools rely on the excellent prediction ability of deep networks. Many works attempt to effectively learn the end-to-end mapping in a series of video coding tasks, *e.g.*, intra-prediction [7, 8], inter-prediction [4–6, 9, 10], deblocking [11–14], and fast mode decision [15]. With a powerful and unified prediction model, these methods obtain superior R-D performance. For intra-prediction, the diversified modes derived from RDO in the traditional video codec are replaced by a learned general model given a certain kind of context. In [7], based on the multi-line reference context, fully-connected (FC) neural networks are utilized to generate the prediction result. In [36], the strengths of FC and CNN networks are combined. In [8], benefiting from the end-to-end learning and the block-level reference scheme, the proposed predictor employs context information in a large scale to suppress quantization noises. For inter-prediction, deep network-based methods [37, 38] break the limit of block-wise motion compensation and bring in better inter-frame prediction, namely generating better reference frames by using all the reconstructed frames. For loop-filter, the powerful network architectures and the full-resolution input of all reconstructed frames [39, 40] significantly improve the performance of loop filters, say up to 10% BD-rate reduction as reported in many methods.

The end-to-end learning based compression leverages deep networks to model the pixel distribution and generate complex signals from a very compact representation. The pioneering work [33] proposes a parametric nonlinear transformation to well Gaussianize data for reducing mutual information between transformed components and show impressive results in image compression. Meanwhile, in [29], a general framework is built upon convolutional and deconvolutional LSTM recurrent networks, trained once, to support variable-rate image compression via reconstructing the image progressively. Later works [32, 41–43] continue to improve compression efficiency by following the routes. All these methods attempt to reduce the overall R-D cost on a large-scale dataset. Due to the model's flexibility, there are also more practical ways to control the bitrate adaptively, such as applying the attention mask [44] to guide the use of more bits on complex regions. Besides, as the R-D cost is optimized in an end-to-end manner, it is flexible to adapt the rate and distortion to accommodate a variety of end applications, *e.g.* machine vision tasks.

Although video coding performance is improved constantly, some intrinsic problems exist, especially when tremendous volumes of data need to be processed and analyzed. The low-value data volume still constitutes a major part. So these methods of reconstructing whole pictures cannot fulfill the requirement of real-time video content analytics when dealing with large-scale video data. However, the strengths in deep learning based image/video coding, *i.e.* the excellent prediction and generation capacity of deep models and the flexibility of R-D cost optimization, provide opportunities to develop VCM technology to address these challenges.

III. REVIEW OF FEATURE COMPRESSION: FROM SEMANTIC FEATURE PERSPECTIVE

The traditional video coding targets high visual fidelity for humans. With the proliferation of applications that capture video for (remote) consumption by a machine, such as connected vehicles, video surveillance systems, and video capture for smart cities, more recent efforts on feature compression

target low bitrate intelligent analytics (*e.g.*, image/video recognition, classification, retrieval) for machines, as transmission bandwidth for raw visual features is often at a premium, even for the emerging strategy of *Analyze then Compress*.

However, it is not new to explore feature descriptors for MPEG. Back in 1998, MPEG initiated MPEG-7, formally known as Multimedia Content Description Interface, driven by the needs for tools and systems to index, search, filter, and manage audio-visual content. Towards interoperable interface, MPEG-7 [45] defines the syntax and semantics of various tools for describing color, texture, shape, motion, *etc*. Such descriptions of streamed or stored media help human or machine users to identify, retrieve, or filter audio-visual information. Early visual descriptors developed in MPEG-7 have limited usage, as those low-level descriptors are sensitive to scale, rotation, lighting, occlusion, noise, *etc*. More recently, the advance of computer vision and deep learning has significantly pushed forward the standardized visual descriptions in MPEG-7. In particular, CDVS [46] and CDVA [47] have been in the vanguard of the trend of *Analyze then Compress* by extracting and transmitting compact visual descriptors.

A. CDVS

CDVS can trace back to the requirements for early mobile visual search systems by 2010, such as faster search, higher accuracy, and better user experience. Initial research [48–53] demonstrated that one could reduce transmission data by at least an order of magnitude by extracting compact visual features on the mobile device and sending descriptors at low bitrates to a remote machine for search. Moreover, a significant reduction in latency could also be achieved when performing all processing on the device itself. Following initial research, an exploratory activity in the MPEG was initiated at the 91*st* meeting (Kyoto, Jan. 2010). In July 2011, MPEG launched the standardization of CDVS. The CDVS standard (formally known as MPEG-7, Part 13) was published by ISO on Aug. 25, 2015, which specifies a normative bitstream of standardized compact visual descriptors for mobile visual search and augmented reality applications.

Over the course of the standardization process, CDVS has made remarkable improvements over a large-scale benchmark in image matching/retrieval performance with very compact feature descriptors (at six predefined descriptor lengths: 512B, 1KB, 2KB, 4KB, 8KB, and 16KB) [16]. High performance is achieved while stringent memory and computational complexity requirements are satisfied to make the standard ideally suited for both hardware and software implementations [54].

To guarantee the interoperability, CDVS makes a normative feature extraction pipeline including interest point detection, local feature selection, local feature description, local feature descriptor aggregation, local feature descriptor compression, and local feature location compression. Key techniques, *e.g.* low-degree polynomial detector [46, 55, 56], light-weighted interest point selection [57, 58], scalable compressed fisher vector [59, 60], and location histogram coding [61, 62], have been developed by competitive and collaborative experiments within a rigorous evaluation framework [63].

It is worthy to mention that CDVS makes a normative encoder (the feature extraction process is fixed), which is completely distinct from conventional video coding standards in making a normative decoder. The success of CDVS standardization originates from the mature computer vision algorithm (like SIFT [64]) on the reliable image matching between different views of an object or scene. However, when dealing with more complex analysis tasks in the autonomous drive and video surveillances, the normative encoder process incurs more risks of lower performance than those task-specific fine-tuned features directly derived from an end-to-end learning framework. Fortunately, the collaborative compression of video and feature streams is expected to address this issue, as we may leverage the joint advantages of the normative feature encoding process and the normative video decoding process.

B. CDVA

The bold idea of CDVA initiated in the 111*th* MPEG meeting in Feb. 2015 is to have a normative video feature descriptor based on neural networks for machine vision tasks, targeting an exponential increase in the demand for video analysis in autonomous drive, video surveillance systems, entertainment, and smart cities. The standardization at the encoder requires the deterministic deep network model and parameters. However, due to fast-moving deep learning techniques and their end-to-end optimization nature, there is a lack of a generic deep model sufficing for a broad of video analytics tasks. This is the primary challenge for CDVA.

To kick off this standardization, CDVA narrows down to the general task of video matching and retrieval, aims at determining if a pair of videos share the object or scene with similar content, and searching for videos containing similar segments to the one in the query video. Extensive experiments [65] over CDVA benchmark report comparable performances of the deep learning features with different off-the-shelf CNN models (like VGG16 [66], AlexNet [77], ResNet [68]), which provide useful evidence for normative deep learning features for the task of video matching and retrieval. Due to the hardware friendly merits of uniformly sized small filters (3x3 convolution kernel and 2x2 max pooling), and the competitive performance of combining convolution layers of the small filters by replacing a large filter (5×5 or 7×7), VGG16 is adopted by CDVA as the normative backbone network to derive compact deep invariant feature representations.

The CDVA standard (formally known as MPEG-7, Part 15) was published by ISO in July 2019. Based on a previously agreed-upon methodology, key technologies are developed by MPEG experts, including Nested Invariance Pooling (NIP) [65] for deep invariance global descriptor, Adaptive Binary Arithmetic Coding (ABAC) based temporal coding of global and local descriptors [47], the integration of hand-crafted and deep learning features [17]. The NIP method produces compact global descriptors from a CNN model by progressive pooling operations to improve the translation, scale and rotation invariance over the feature maps of intermediate network layers. The extensive experiments have demonstrated that the NIP (derived from the last pooling layer, *i.e.*, pool5, of
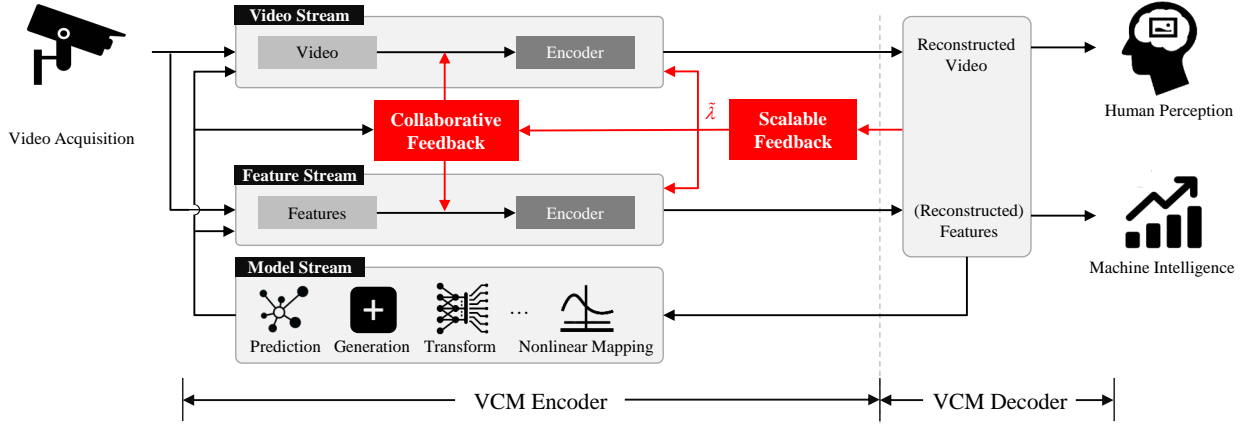
Fig. 1. The proposed video coding for machines (VCM) framework by incorporating the feedback mechanism into the collaborative coding of multiple streams including videos, features, and models, targeting the multi-tasks of human perception and machine intelligence.

VGG16) outperforms the state-of-the-art deep and canonical hand-crafted descriptors with significant gains. In particular, the combination of (deep learning) NIP global descriptors and (hand-crafted) CDVS global descriptors has significantly boosted the performance with a comparable descriptor length.

More recently, great efforts are made to extend the CDVA standard for VCM. Smart tiled CDVA [69] targeting video/image analysis of higher resolution input, can be applied to rich tasks including fine-grained feature extraction network, small object detection, segmentation, and other use cases. Smart sensing [70] is an extension of the CDVA standard to directly process raw Bayer pattern data, without the need of a traditional ISP. This enables all Bayer pattern sensors to be natively CDVA compatible. SuperCDVA [71] is an application of the CDVA standard for video understanding by using the temporal information, in which each CDVA vector is embedded as one block in an image, and the multiple blocks are put in a sequential manner to represent the sequence of video frames for classification by a 2-D CNN model.

In summary, the new MPEG-7 standard CDVA opens the door to exciting new opportunities in supporting machine-only encoding as well as hybrid (machine and human) encoding formats to unburden the network and storage resources for dramatic benefits in various service models and a broad range of applications with low latency and reduced cost.

IV. VIDEO CODING FOR MACHINES: DEFINITION, FORMULATION AND PARADIGM

The emerging requirements of video processing and content analysis in a collaborative manner are expected to support both human vision and machine vision. In practice, two separate streams of compressed video data and compact visual feature descriptors are involved to satisfy the human observers and a variety of low-level and high-level machine vision tasks, respectively. How to further improve the visual signal compression efficiency as well as the visual task performance by leveraging multiple granularities of visual features remains a challenging but promising task, in which the image/video data is considered as a sort of pixel-level feature.

A. Definition and Formulation

Traditional *video coding* targets visual signal fidelity at high bitrates, while *feature coding* targets high performance of vision tasks at very low bitrates. Like CDVS and CDVA, most of the existing feature coding approaches heavily rely on specific tasks, which significantly save bitrate by ignoring the requirement of full-resolution video reconstruction. By contrast, the video codecs like HEVC focus on the reconstruction of full-resolution pictures from the compressed bit stream. Unfortunately, the pixel-level features from data decompression cannot suffice for large-scale video analysis in a huge or a moderate scale camera network efficiently, due to the bottleneck of computation, communication, and storage [72].

We attempt to identify the role of Video Coding for Machines (**VCM**), as shown in Fig. 1, to bridge the gap between coding semantic features for machine vision tasks and coding pixel features for human vision. The scalability is meant to incrementally improve the performance in a variety of vision tasks as well as the coding efficiency by optimizing bit utility between multiple low-level and high-level feature streams. Moreover, VCM is committed to developing key techniques for economizing the use of a bit budget to collaboratively complete multiple tasks targeting humans and/or machines.

VCM aims to *jointly maximize the performance of multiple tasks ranging from low-level processing to high-level semantic analysis, but minimize the use of communication and computational resources*. VCM relates to three key elements:

- *Tasks*. VCM incurs low-level signal fidelity as well as high-level semantics, which may request a complex optimization objective from multiple tasks. It is important to figure out an efficient coding scheme across tasks.
- *Resources*. VCM is committed to tackling the practical performance issue, which cannot be well solved by traditional video coding or feature coding approaches solely, subject to the constraints of resources like bandwidth, computation, storage, and energy.
- *Features*. VCM is to explore a suite of rate distortion functions and optimization solutions from a unified perspective of leveraging the features at different granular-

ities including the pixels for humans to derive efficient compression functions (*e.g.*, transform and prediction).

Here we formulate the VCM problem. The features are denoted by $\mathbf{F} = \{F^0, F^1, ..., F^z\}$ of $(z+1)$ tasks, where $V := F^z$ is the pixel feature, associated with the task of improving visual signal fidelity for human observers. F^i denotes more task (non-)specific semantic or syntactic features if $0 \leq i < z$. A smaller i denotes that the feature is more abstract. The performance of the task i is defined as follows:

$$q^i = \Phi^i(\hat{F}^i), \tag{1}$$

where $\Phi^i(\cdot)$ is the quality metric related to task i and $\hat{F}^i$ is the reconstructed feature undergoing the encoding and decoding processes. Note that, certain compressed domain processing or analysis may not require the complete reconstruction process. We use $C(\cdot|\theta_c), D(\cdot|\theta_d)$, and $\mathcal{G}(\cdot|\theta_g)$ to denote the processes of compression, decompression, feature prediction, where θ_c, θ_d, and θ_g are parameters of the corresponding processes. We define $S(\cdot)$ to measure the resource usage of computing a given feature, namely compressing, transmitting, and storing the feature, as well as further analyzing the feature. By incorporating the elements of tasks, resources, and features, VCM explicitly or non-explicitly models a complex objective function for maximizing the performance and minimizing the resource cost, in which joint optimization applies. Generally, the **VCM optimization function** is described as follows,

$$\operatorname*{argmax}_{\Theta=\{\theta_c, \theta_d, \theta_g\}} \sum_{0 \leq i \leq z} \omega^i q^i,$$
$$\text{subject to: } \sum_{0 \leq i \leq z} \omega^i = 1, \tag{2}$$

$$S\left(R_{F^0}\right) + \sum_{i>0} \min_{0 \leq j < i}\left\{S\left(R_{F_{i \to j}}\right)\right\} + S\left(R_M\right) + S\left(\Theta\right) \leq S_T,$$

where S_T is total resource cost constraint, and ω^i balances the importance of different tasks in the optimization. The first two terms in the resource constraint are the resource cost of compressing and transmitting features of all tasks with feature prediction. Note that, $R_V = \min_j\left\{S\left(R_{F_{z \to j}}\right)\right\}$ is the resource cost to encode videos F^z with feature prediction. The third term in the resource constraint is the resource overhead from leveraging models and updating models accordingly. The last term $S(\Theta)$ calculates the resource cost caused by using a more complex model, such as longer training time, response time delay due to feedbacks, larger-scale training dataset. Principle rules apply to the key elements as below,

$$R_{F^0} = C\left(F^0|\theta_c\right), \tag{3}$$
$$R_{F_{i \to j}} = C\left(F^i - \mathcal{G}\left(F^j, i|\theta_g\right)\right), \text{ for } i \neq 0, \tag{4}$$
$$\widehat{F}^0 = D\left(R_{F^0}|\theta_d\right), \tag{5}$$
$$\widehat{F}^i = D\left(R_{F_{i \to j}}|\theta_d\right) + \mathcal{G}\left(\widehat{F}^j, i|\theta_g\right), \text{ for } i \neq 0, \tag{6}$$

where $\mathcal{G}(\cdot, j|\theta_g)$ projects the input feature to the space of F^j. $\hat{}$ denotes that, the feature decoded from the bit stream would be suffering from compression loss. In practice, the VCM problem can be rephrased in Eq. (2) by designing more task specific terms $C(\cdot|\theta_c)$, $D(\cdot|\theta_d)$ and $\mathcal{G}(\cdot|\theta_g)$.

B. A VCM Paradigm: Digital Retina

As a VCM instance, digital retina [72–74] is to solve the problem of real-time surveillance video analysis collaboratively from massive cameras in smart cities. Three streams are established for human vision and machine vision as follows:

- **Video stream**: Compressed visual data is transmitted to learn data-driven priors to impact the optimization function in Eq. (2). In addition, the fully reconstructed video is optionally for humans as requested on demand.
- **Model stream**: To improve the task-specific performance and hybrid video coding efficiency, hand-crafted or learning based models play a significant role. This stream is expected to guide the model updating and the model based prediction subsequently. The model learning works on the task-specific performance metric in Eq. (1).
- **Feature stream**: This stream consisting of task-specific semantic or syntactic features extracted at the front end devices is transmitted to the server end. As formulated in Eq. (2), a joint optimization is expected to reduce the resource cost of video and feature streams.

Instead of working on video stream alone, the digital retina may leverage multiple streams of features to reduce the bandwidth cost by transmitting task-specific features, and balance the computation load by moving part of feature computing from the back end to the front end. As an emerging VCM approach, the digital retina is revolutionizing the vision system of smart cities. However, for the current digital retina solution, the optimization in Eq. (2) is reduced to minimizing an objective for each single stream separately, without the aid of feature prediction in Eq. (4) and (6), rather than multiple streams jointly. For example, state-of-the-art video coding standards like HEVC are applied to compress video streams. The compact visual descriptor standards CDVS/CDVA are applied to compress visual features. The collaboration between two different types of streams works in a combination mode. There is a lack of joint optimization across streams in terms of task-specific performance and coding efficiency.

Generally speaking, the traditional video coding standardization on HEVC/AVS, together with recent compact feature descriptor standardization on CDVS/CDVA, targets efficient data exchange between human and human, machine and human, and machine and machine. To further boost the functionality of feature streams and open up more room for collaborating between streams, the digital retina introduces a novel model stream, which takes advantage of pixel-level features, semantic features (representation), and even existing models to generate a new model for improving the performance as well as the generality of task-related features. More recent work [73] proposed to reuse multiple models to improve the performance of a target model, and further came up with a collaborative approach [75] to low cost and intelligent visual sensing analysis. However, how to improve the video coding efficiency and/or optimize the performance by collaborating different streams is still an open and challenging issue.

C. Reflection on VCM

Prior to injecting model streams, the digital retina [76] has limitations. First, the video stream and feature stream are handled separately, which limits the utilization of more streams. Second, the processes of video coding and feature coding are uni-directional, thereby limiting the optimization performance due to the lack of feedback mechanism, which is crucial from the perspective of a vision system in smart cities.

Beyond the basic digital retina solution, VCM is supposed to jointly optimize the compression and utilization of feature, video, and model streams in a scalable way by introducing feedback mechanisms as shown in Fig. 1:

1) *Feedback for collaborative mode*: The pixel and/or semantic features, equivalently video and feature streams, can be jointly optimized towards higher coding efficiency for humans and/or machines in a collaborative manner. That is, the features can be fed back to each other between streams for improving the task-specific performance for machines, in addition to optimizing the coding efficiency for humans. Advanced learning-based prediction or generation models may be applied to bridge the gap between streams.

2) *Feedback for scalable mode*: When bit budget cannot suffice for video or feature streams, or namely, the quality of reconstructed feature and video are not desirable, more additional resources are utilized, along with the previously coded streams, to improve the quality of both feature and video streams with a triggered feedback. Therefore, the desired behavior of incrementally improving the performance of those human and machine vision tasks subject to an increasing bit budget can be expected.

V. New Trends and Technologies

The efficient transition between the features of different granularities is essential for VCM. In this section, we review more recent advances in the image/video predictive and generative models.

A. Predictive Models

1) Image Predictive Models: Deep convolutional neural networks [77, 78] have been proven to be effective to predict semantic labels of images/videos. This superior capacity has been witnessed in many visual tasks, *e.g.* image classification [78, 79], object detection [80, 81], semantic segmentation [82], pose estimation [83]. The key to the success of these tasks is to extract discriminative features to effectively model critical factors highly related to the final predictions. Deep networks have hierarchical structures to extract features from low-level to high-level progressively. The features at the deep layer are more compact and contain less redundant information, with very sparse activated regions. As such, the image predictive model can capture the compact and critical feature from a single image, which provides an opportunity to develop efficient compression approaches.

2) Video Predictive Models: The deep networks designed for video analytics pay additional attention to modeling temporal dynamics and utilizing complex joint spatial and temporal

correlations for semantic label prediction of videos. In [84], several approaches extend the temporal connectivity of a CNN to fully make use of local spatio-temporal information for video classification. In [85], a two-stream ConvNet architecture incorporated with spatial appearance and motion information is built for action recognition. Their successive works are based on mixed architectures with both CNN and RNN [86], and 3D convolutional networks [87] for action recognition [88, 89], scene recognition [87], captioning, commenting [90]. These models are capable to extract discriminative and compact joint spatial and temporal features, potential to benefit squeezing out the redundancy in videos.

B. Generative Models

1) Generative Adversarial Networks: The advance of generative adversarial networks (GANs) [91] makes a significant impact in machine vision. Recent years have witnessed the prosperity of image generation and its related field [92–95]. In general, most of the existing methods can be categorized into two classes: supervised and unsupervised methods. In supervised methods [96–98], GANs act as a powerful loss function to capture the visual property that the traditional losses fail to describe. Pix2pix [99] is a milestone work based on conditional GAN to apply the image-to-image translation from the perspective of domain transfer. Later on, more efforts [92, 100] are dedicated to generating high-resolution photo-realistic images with a progressive refinement framework. In unsupervised methods, due to the lack of the paired ground truth, the cycle reconstruction consistency [101–103] is introduced to model the cross-domain mapping.

2) Guided Image Generation: Some works focus on guided image generation, where semantic features, *e.g.* human pose and semantic map, are taken as the guidance input. The early attempt pays attention to pose-guided image generation and a two-stage network PG^2 [104] is built to coarsely generate the output image under the target pose in the first stage, and then refine it in the second stage. In [105], deformable skips are utilized to transform high-level features of each body part to better model shapes and appearances. In [106], the body part segmentation masks are used as guidance for image generation. However, the above-mentioned methods [104–106] rely on paired data. To address the limitation, in [107], a fully unsupervised GAN is designed, inspired by [101, 108]. Furthermore, the works in [109, 110] resort to sampling from the feature space based on the data distribution. These techniques bring in precious opportunities to develop efficient video coding techniques. The semantic feature guidance is much more compact. With the support of the semantic feature, the original video can be well reconstructed economically.

3) Video Prediction and Generation: Another branch of generation models are for video prediction and generation. Video prediction aims to produce future frames based on previous frames of a video sequence in a deterministic manner, in which recurrent neural networks are often used to model the temporal dynamics [111–113]. In [111], an LSTM encoder-decoder network is utilized to learn patch-level video representations. In [112], a convolutional LSTM is built to predict
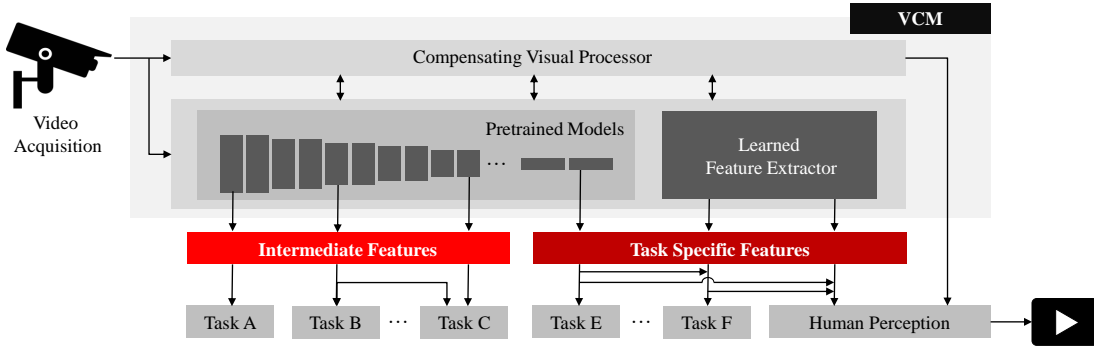
Fig. 2. The key modules of VCM architecture, as well as the relationship between features and human/machine tasks.

video frames. In [113], a multi-layer LSTM is constructed to progressively refine the prediction process. Some methods do not rely on recurrent networks, *e.g.* 3D convolutional neural network [114, 115]. Some other methods [116–118] estimate local and global transforms and then apply these transforms to generate future frames indirectly.

Comparatively, video generation methods aim to produce visually authentic video sequences in a probabilistic manner. In the literature, methods based on GAN [119–122] and Variational AutoEncoder (VAE) [123–127] are built. The above-mentioned methods just predict a few video frames. Later works [128] target long-term video prediction. In [128], up to 100 future frames of an Atari game are generated. The future frame is generated with the guidance of the encoded features by CNN and LSTM from previous frames. In [129], a new model is proposed to generate real video sequences. The high-level structures are estimated from past frames, which are further propagated to those of the future frame via an LSTM. In [130], an unsupervised method is built to extract a high-level feature which is further used to predict the future high-level feature. After that, the model can predict future frames based on the predicted features and the first frame.

Undoubtedly, inter prediction plays a significant role in video coding. The video prediction and generation models are expected to leverage compact features to propagate the context of video for improving coding efficiency.

C. Progressive Generation and Enhancement

Deep learning brings in the wealth of 'deep'. That is, with the hierarchical layer structure, the information is processed and distilled progressively, which benefits both high-level semantic understanding and low-level vision reconstruction. For VCM, an important property is scalability, a capacity of feature prediction ranging from the extreme compactness to the injection of redundant visual data for serving humans and machines, which is closely correlated to the hot topics of deep progressive generation and enhancement.

A series of works have been proposed to generate or enhance images progressively. In [147], a cascade of CNN within a Laplacian pyramid framework is built to generate images in a coarse-to-fine fashion. In [148], a similar framework is applied for single-image super-resolution. Later works refining features progressively at the feature-level, like ResNet [68], or concatenating and fusing features from different levels, like DenseNet [149], lead to better representations of pixels and their contexts for low-level visions. The related beneficial tasks include super-resolution [96, 150–153], rain removal [154, 155], dehazing [151], inpainting [156], compression artifacts removal [157], and deblurring [158]. Zhang *et al.* [151] combined the structure of ResNet and DenseNet. Dense blocks are used to obtain dense local features. All features in each dense block are connected by skip connections, and then fused in the last layer adaptively in a holistic way.

In video coding, there is also a similar tendency to pursue a progressive image and video compression, namely, *scalable image/video coding* [131], affording to form the bit-stream at any bitrate. The bit-stream usually consists of several code layers (one base layer and several enhancement layers). The base layer is responsible for basic but coarse modeling of image/video. The enhancement layers progressively improve the reconstruction quality with additional bit-streams. A typical example is JPEG2000 [132], where an image pyramid from the wavelet transform is build up for scalable image reconstruction based on compact feature representations. Later on, the extension of the scalable video codec is made in the H.264 standard [133]. The base layer bit-stream is formed by compressing the original video frames, and the enhanced layer bit-stream is formed by encoding the residue signal.

VCM is expected to incorporate the scalability into the collaborative coding of multiple task-specific data streams for humans and/or machines. It is worthy to note that, the rapid development of DNNs (deep neural networks) is proliferating scalable coding schemes. In [32], an RNN model is utilized to realize variable-bitrate compression. In [134], bidirectional ConvLSTM is adopted to decompose the bit plane by efficiently memorizing long-term dependency. In [135], inspired by the self-attention, a transformer-based decorrelation unit is designed to reduce the feature redundancy at different levels. More recently, several works [136–138, 140, 141] attempt to jointly compress videos and features in a scalable way, which shows more evidence for the scalability in VCM.

VI. VIDEO CODING FOR MACHINES: POTENTIAL SOLUTIONS

In this section, we present several exemplar VCM solutions: deep intermediate feature compression, predictive coding with

collaborative feedback, and enhancing predictive coding with scalable feedback. Based on the fact that the pretrained deep learning networks, *e.g.* VGG and AlexNet, can support a wide range of tasks, such as classification and object detection, we first investigate the issue of compressing intermediate features (usually before pool5 layer) extracted from off-the-shelf pretrained networks (left part of Fig. 2), less specific to given tasks, via state-of-the-art video coding techniques. In the next step, we explore the solution that learns to extract key points as a highly discriminative image-level feature (right part of Fig. 2) to support the motion-related tasks for both machine and human vision with collaborative feedback in a feature assisted predictive way. Finally, we attempt to pursue a preliminary scheme to offer a general feature scalability to derive both pixel and image-level representations and improve the coding performance incrementally.

A. Deep Intermediate Feature Compression

In the VCM, features are the key bridge for both front and back user-ends as well as high and low-level visions. It naturally raises questions about the optimized feature extraction and model updating in the VCM framework. For deep model-based applications, the feature compression is hindered by that, the models are generally tuned for specific tasks, and that, the top-layer features are very task-specific and hard to be generalized. The work in [139] explores a novel problem: *the intermediate layer feature compression*, reducing the computing burden while being capable to support different kinds of visual analytics applications. In practice, it provides a compromise between the traditional video coding and feature compression and yields a good trade-off among the computational load, communication cost, and the generalization capacity.

As illustrated in Fig. 2, VCM attempts to connect the features of different granularities to the human/machine vision tasks from the perspective of a general deep learning framework. The intermediate layer features are compressed and transmitted instead of the original video or top layer features. Compared with the deep layer features, the intermediate features from shallow layers contain more informative cues in a general sense, as the end-to-end learning usually makes the deep layer features more task-specific with a large receptive field. To accommodate a wide range of machine vision tasks, VCM prefers to optimize the compression of intermediate features for general purposes. An interesting view is that, the shallow layers closer to the input image/video, are supposed to be less important than the deep layers, as the semantics play a significant role in data compressing even for humans. As indicated in Fig. 2, VCM prefers to use the deep layers features in improving coding efficiency for humans.

Beyond that, a further idea is to propose the problem of *feature recomposition*. The features for different tasks are with various granularities. It is worthwhile to explore how to evaluate the feature coding performance of all tasks in a unified perspective, and further decide to recompose the feature of different tasks, sharing common features and organizing the related compression in a more or less scalable way.

B. Predictive Coding with Collaborative Feedback

Fig. 3 (a) shows the overview pipeline of a joint feature and video compression approach [140, 141]. At the encoder side, a set of key frames v_k will be first selected and compressed with traditional video codecs to form the bit-stream B_I. The coded key frames convey the appearance information and are transmitted to the decoder side to synthesize the reconstructed non-key frames. Then, the network learns to represent $V = \{v_1, v_2, ..., v_N\}$ with the learned sparse points $F = \{f_1, f_2, ..., f_N\}$ to describe temporal changes and object motion among frames. We employ prediction model $P(\cdot)$ and generation model $G(\cdot)$ to implement the feature transition operation $\mathcal{G}(\cdot)$ to convert the feature from a redundant form to compact one and vice verse, respectively. More specifically, $P(\cdot)$ and $G(\cdot)$ are the processes to extract key points from videos and generate videos based on the learned key points. To extract the key points, we have:

$$F = P\left(V, \lambda | \theta_{gp}\right), \tag{7}$$

where θ_{gp} is a learnable parameter. F is a compact feature, which only requires very fewer bit streams for transmission and storage. λ is a rate control parameter. The compression model $C_F\left(\cdot | \theta_{cf}\right)$ compresses F into the feature stream B_F:

$$B_F = C_F\left(F | \theta_{cf}\right), \tag{8}$$

where θ_{cf} is a learnable parameter.

Then, a motion guided generation network calculates the motion based on these points and then transfers the appearance from the reconstructed key frames to those remaining non-key frames. Specifically, for the t-th frame to be reconstructed, we denote its previous reconstructed key frame, previous reconstructed key points, and the current reconstructed key points by $\widehat{v}_{\psi(t)}, \widehat{f}_{\psi(t)}$ and $\widehat{f}_t$: where $\psi(t)$ maps the index of the key frame of the t-th frame. The target frame $\widetilde{v}_t \in \widetilde{V}$ is synthesized as follows:

$$\widetilde{v}_t = G\left(\widehat{v}_{\psi(t)}, \widehat{f}_{\psi(t)}, \widehat{f}_t | \theta_{gg}\right), \tag{9}$$

where θ_{gg} is a learnable parameter. After that, the residual video $R = V - \widetilde{V}$ can be calculated, where $\widetilde{V} = \{\widetilde{v}_1, \widetilde{v}_2, ..., \widetilde{v}_N\}$. The sparse point descriptor and residual video will be quantized and compressed to the bit stream B_F and B_V for transmission. That is $B_V = \{B_I, B_R\}$. We can adjust the total bitrate via controlling the bitrates of B_F and B_V.

At the decoder side, the key frames will be first reconstructed from B_I. The sparse point representations are also decompressed as $\widehat{F} = \left\{\widehat{f}_1, \widehat{f}_2, ..., \widehat{f}_N\right\}$ from B_F as follows,

$$\widehat{F} = D_F\left(B_F | \theta_{df}\right), \tag{10}$$

where $D_F\left(\cdot | \theta_{df}\right)$ is a feature decompression model, and θ_{df} is a learnable parameter. The videos are reconstructed via: $\widehat{V} = \widetilde{V} + \widehat{R}$. Finally, $\widehat{F}$ along with $\widehat{V}$ serves machine analysis and human vision, respectively.
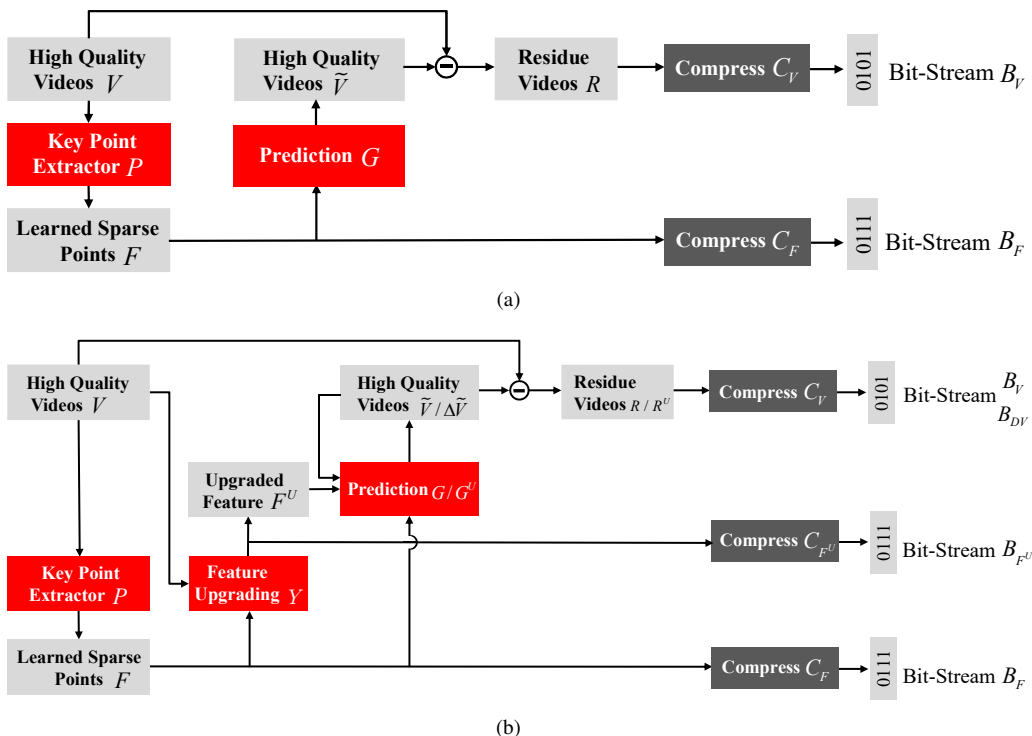
Fig. 3. Two potential VCM solutions: (a) Predictive coding with collaborative feedback, and (b) Enhancing predictive coding with scalable feedback.

C. Enhancing Predictive Coding with Scalable Feedback

Fig. 3 (b) shows an optimized architecture for VCM, by adding scalable feedback. Similarly, the key points of video frames F are extracted. After that, the redundancy of the video and key frames $\widehat{v}_{\psi(t)}$ is removed by applying feature-guided prediction. Then, the residue video R is compressed into the video stream, which is further passed to the decoder.

When the feature and video qualities after decompression do not meet the requirements, a scalable feedback is launched and more guidance features are introduced:

$$\Delta F = Y\left(F, V | \theta_y\right), \tag{11}$$

$$F^U = F + \Delta F, \tag{12}$$

where θ_y is a learnable parameter. With the key points F in the base layer (the result in the scheme in Fig. 3 (a)) and the original video V, we generate the residual feature ΔF to effectively utilize additionally allocated bits to encode more abundant information and form the updated feature F^U.

Then, G^U is used to refine V by taking the reconstructed key points and video as its input:

$$\Delta \widetilde{V} = G^U\left(\widehat{F}, \Delta \widehat{F}, \widehat{V} | \theta_h\right), \tag{13}$$

where θ_h is a learnable parameter. Then, we can infer the incremental residue video: $R^U = V - (\widetilde{V} + \Delta\widetilde{V}) - R$, which is compressed into the bit stream B_{DV}. At the decoder side, $\hat{R}^U$ is decompressed from B_{DV}. Then, the video with a high quality is inferred via: $\widehat{V}^U = \widetilde{V} + \Delta\widetilde{V} + \widehat{R}^U + \widehat{R}$. This allows to introduce more bits via launching scalable feedback.

TABLE I
LOSSY FEATURE COMPRESSION RESULTS FOR DIFFERENT TASKS
(COMP.RATE—FIDELITY[2])

Feature	Classification		Retrieval		Detection	
QP	22	42	22	42	22	42
VGGNet						
conv1	0.080 0.985	0.020 0.839	0.041 0.997	0.006 0.955	0.065 0.954	0.013 0.850
pool1	0.099 0.984	0.023 0.693	0.039 0.996	0.005 0.923	0.085 0.942	0.018 0.820
conv2	0.098 0.972	0.035 0.790	0.069 0.996	0.027 0.955	0.090 0.950	0.030 0.858
pool2	0.138 0.982	0.047 0.745	0.119 0.997	0.037 0.945	0.128 0.953	0.040 0.815
conv3	0.080 0.986	0.034 0.840	0.089 0.998	0.048 0.976	0.033 0.954	0.015 0.845
pool3	0.140 0.981	0.063 0.819	-	-	0.066 0.955	0.032 0.826
conv4	0.053 0.984	0.028 0.865	0.070 0.997	0.043 0.959	0.019 0.960	0.008 0.877
pool4	0.127 0.974	0.065 0.864	-	-	0.041 0.960	0.019 0.847
conv5	0.046 0.989	0.023 0.920	0.041 0.995	0.030 0.952	0.023 0.956	0.005 0.741
pool5	0.129 0.986	0.075 0.908	0.200 0.996	0.146 0.960	-	-
ResNet						
conv1	0.041 0.935	0.005 0.356	0.018 0.964	0.001 0.792	0.029 0.915	0.002 0.713
pool1	0.043 0.937	0.004 0.087	0.025 0.963	0.002 0.720	0.030 0.889	0.002 0.470
conv2	0.095 0.986	0.014 0.765	0.027 0.939	0.001 0.554	0.107 0.949	0.016 0.660
conv3	0.134 0.989	0.028 0.854	0.041 0.986	0.003 0.551	0.118 0.971	0.021 0.684
conv4	0.170 0.992	0.034 0.932	0.035 0.997	0.012 0.799	0.056 0.964	0.008 0.833
conv5	0.131 0.998	0.063 0.961	0.040 0.999	0.014 0.992	-	-

VII. PRELIMINARY EXPERIMENTAL RESULTS

In this section, we provide preliminary experimental results from the perspectives of intermediate deep feature compression and machine-human collaborative compression.

A. Deep Intermediate Feature Compression for Different Tasks

1) Compression Results: We show the compression performance on the intermediate features for different tasks in Table I. The compression rate is calculated by the ratio of original intermediate deep features and the compressed bit-streams. As to the fidelity evaluation[2], the reconstructed

[2]https://github.com/ZoomChen/DeepFeatureCoding/tree/master/Coding_and_Evaluation

TABLE II
LOSSY FEATURE COMPRESSION RESULTS FOR DIFFERENT TASKS.
DD-CHANNEL CONCATENATION DENOTES CHANNEL CONCATENATION BY
DESCENDING DIFFERENCE. (COMP.RATE—FIDELITY[2])

Feature	Channel Concatenation				DD-Channel Concatenation				Channel Tiling			
QP	12		42		12		42		12		42	
conv1	0.100	0.991	0.006	0.666	0.100	0.992	0.006	0.694	0.116	0.998	0.020	0.839
pool1	0.124	0.987	0.006	0.401	0.124	0.986	0.005	0.412	0.145	0.993	0.023	0.686
conv2	0.116	0.989	0.012	0.343	0.116	0.988	0.012	0.455	0.130	0.992	0.035	0.781
pool2	0.166	0.991	0.014	0.260	0.165	0.991	0.014	0.369	0.184	0.996	0.045	0.756
conv3	0.093	0.989	0.013	0.586	0.093	0.990	0.013	0.617	0.100	0.995	0.033	0.835
pool3	0.164	0.990	0.023	0.477	0.164	0.994	0.023	0.541	0.163	0.992	0.053	0.791
conv4	0.059	0.992	0.012	0.700	0.059	0.988	0.012	0.706	0.051	0.993	0.021	0.857
pool4	0.140	0.990	0.030	0.624	0.142	0.991	0.028	0.611	0.100	0.993	0.042	0.855
conv5	0.046	0.995	0.016	0.809	0.046	0.990	0.015	0.812	0.028	0.996	0.010	0.920
pool5	0.127	0.992	0.055	0.776	0.127	0.994	0.053	0.746	0.054	0.996	0.024	0.903

TABLE III
ACTION RECOGNITION ACCURACY
OF DIFFERENT METHODS AND
CORRESPONDING BITRATE COSTS.

Codec	Bitrate (Kbps)	Accu.(%)
HEVC	16.2	65.2
Ours	5.2	74.6

TABLE IV
SSIM COMPARISON BETWEEN
DIFFERENT METHODS AND
CORRESPONDING BITRATE COSTS.

Codec	Bitrate (Kbps)	SSIM
HEVC	33.0	0.9008
Ours	32.1	0.9071

features are passed to their birth-layer of the original neural network to infer the network outputs, which will be compared with pristine outputs to evaluate the information loss of the lossy compression methods. More results and details on the evaluation framework can be found in [142].

From Table I, several interesting observations are reached. First, the potential of lossy compression is inspiring. In the extreme case, for example in image retrieval, ResNet conv2 feature achieves at least $1000\times$ compression ratio at QP 42, while the lossless methods usually provide 2-5×. Second, for each feature type, the fidelity metric decreases with a larger QP value. Third, QP 22 generally provides high fidelity and fair compression ratio. Forth, upper layer features, like conv4 to pool5, tend to be more robust to heavy compression.

2) Channel Packaging: Deep features have multiple channels. It needs to arrange these features into single-channel or three-channel maps and then compress them with the existing video codecs. Three modes are studied: *channel concatenation*, *channel concatenation by descending difference*, and *channel tiling*. For *channel concatenation*, each channel of the feature map corresponds to a frame in the input data of a traditional video encoder. The height and width of the feature map are filled to the height and width that meet the input requirements of the video encoder. The feature map channel order is the original order and remains unchanged. In this mode, inter-coding of HEVC is applied. For *channel concatenation by descending difference*, to obtain higher compression efficiency, the channel of the feature map is reordered before being fed into a traditional video encoder. The first channel is fixed, and the remaining channels are arranged according to the L2 norm of the different between the current channel to the previous one. For *channel tiling*, multiple channels are tiled into a two-dimensional map, serving as an input to a video encoder. The result is presented in Table II. These results are preliminary and more efforts are expected to improve the efficiency of compressing deep feature maps.

B. Joint Compression of Feature and Video

Let us evaluate the effectiveness of the potential solution: feature assisted predictive coding with the collaborative feedback. We show the results of *compression for machine vision*, including *action recognition*, *human detection* and *compression for human vision*, video reconstruction.

1) Experimental Settings: PKU-MMD dataset [143] is used to generate the testing samples. In total, 3317 clips, each sampling 32 frames, are used for training, and 227 clips, each sampling 32 frames, are used for testing. All frames are cropped and resized to 512×512. To verify the coding efficiency, we use HEVC, implemented in FFmpeg version 2.8.15[3], as the anchor for comparison by compressing all frames with the HEVC codec in the constant rate factor mode.

To evaluate the performance of feature assisted predictive coding, the sparse motion pattern [144] is extracted to serve machine vision. For a given input frame, a U-Net followed by softmax activations is used to extract heatmaps for key point prediction. The covariance matrix is additionally generated to capture the correlations between the key points and its neighbor pixels. For each key point, in total 6 float numbers including two numbers indicating the position and 4 numbers in the covariance matrix are used for description. The selected key frames are compressed and transmitted, along with the sparse motion pattern to generate the full picture for human vision. In the testing process, we select the first frame in each clip as the key frames. At the encoder side, the key frame is coded with the HEVC codec in the constant rate factor mode. Besides the key frame, 20 key points of each frame form the sparse motion representation. Each key point contains 6 float numbers. For the two position numbers, a quantization with the step of 2 is performed for compression. For the other 4 float numbers belonging to the covariance matrix, we calculate the inverse of the matrix in advance, and then quantize the 4 values with a step of 64. Then, the quantized key point values are further losslessly compressed by the Lempel Ziv Markov chain algorithm (LZMA) algorithm [145].

2) Action Recognition: We first evaluate the efficiency of the learned key points for action recognition. Although each key point is represented with 6 numbers, we only use two quantized position numbers for action recognition. To align with the bitrate cost of the features, the clips are first downscaled to 256×256 and then compressed with the constant rate factor 51 with HEVC. All 227 clips are used in the testing. Table III has shown the action recognition accuracy and corresponding bitrate costs of different kinds of data. Our method can obtain considerable action recognition accuracy with only 5.2 Kbps bitrate cost, superior to that by HEVC.

3) Human Detection: Apart from action recognition, human detection accuracy is also compared. The original skeleton information in the dataset is used to form the ground truth bounding box and a classical detection algorithm YOLO v3 [159] is adopted for comparison. All 227 clips are used in

---

[3]https://www.ffmpeg.org/

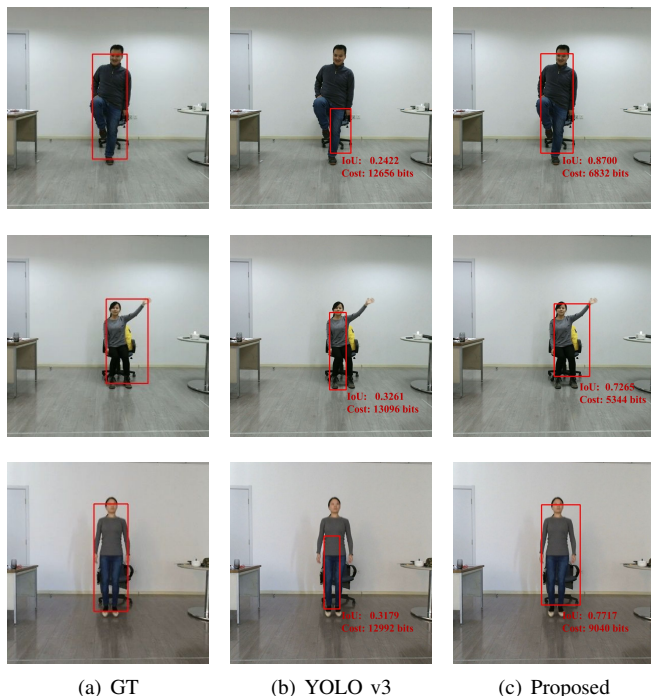(a) GT   (b) YOLO v3   (c) Proposed

Fig. 4. Subjective results of human detection. The coding cost represents the bits required to code the corresponding testing clip.
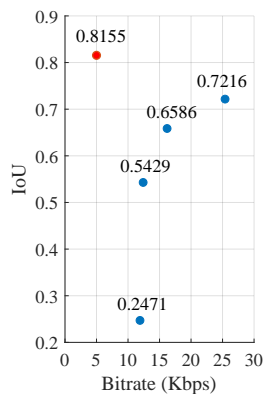

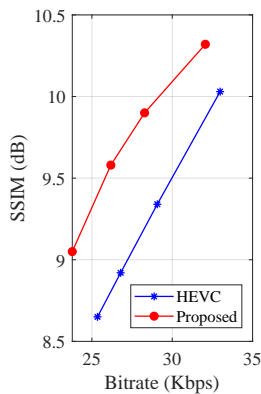Fig. 5. Human detection IoU under different bitrates.

Fig. 6. Rate distortion curves of HEVC and the proposed method.

the testing. The testing clips are down-scaled to different scales of 64×64, 128×128, 256×256 and 512×512 and compressed by HEVC with the constant rate factor 51 to form the input of YOLO v3. Fig. 5 has shown the Intersection over Union (IoU) of different methods and their corresponding bitrates. Our method can achieve much better detection accuracy with lower bitrate costs. Some subjective results of human detection are shown in Fig. 4, we can see that our method can achieve better detection accuracy with fewer bit costs.

4) Video Reconstruction: The video reconstruction quality of the proposed method is compared with that of HEVC. During the testing phase, we compress the key frames with a constant rate factor 32 to maintain a high appearance quality. The bitrate is calculated by considering the compressed key frames and key points. As for HEVC, we compress all frames with a constant rate factor 44 to achieve an approaching bitrate.

Fig. 7. The visual results of the reconstructed videos by HEVC (left panel) and our method (right panel), respectively.

(a) Ground Truth   (b) HEVC   (c) Proposed

Fig. 8. Video reconstruction results of different methods.

Table IV has shown the quantitative reconstruction quality of different methods. SSIM values are adopted for quantitative comparison. It can be observed that, our method can achieve better reconstruction quality than HEVC with lower bitrate cost. The subjective results of different methods are shown in Fig. 8. There are obvious compression artifacts on the reconstruction results of HEVC, which heavily degrade the visual quality. Compared with HEVC, our method can provide far more visually pleasing results.

Moreover, we add a rate distortion curve for comparison. HEVC is used as the anchor undergoing four constant rate factors 44, 47, 49 and 51. For our method, the key frames are compressed respectively under constant rate factors 32, 35, 37 and 40. The rate distortion curve is shown in Fig. 6. Our method yields better reconstruction quality at all bitrates.

VIII. Discussion and Future Directions

A. Entropy Bounds for Tasks

Machine vision tasks rely on features at different levels of granularity. High-level tasks prefer more discriminative and

compact features, while low-level tasks need abundant pixels for fine modeling. As VCM is to explore the collaborative compression and intelligent analytics over multiple tasks, it is valuable to figure out the intrinsic relationship among a variety of tasks in a general sense. There's some preliminary work to reveal the connection between typical vision tasks [146]. However, there is no theoretical evidence to measure the information associated with any given task. We may resort to extensive experiments on the optimal compression ratio vs. the desired performance for each specific task. But the empirical study hinders the collaborative optimization across multiple tasks due to the complex objective and the heavy computational cost. Moreover, the proposed VCM solution benefits from the incorporation of collaborative and scalable feedback over tasks. How to mathematically formulate the connection of different tasks helps to pave the path to the completeness in theory on the feedback mechanism in VCM. In particular, the theoretical study on entropy bounds for tasks is important for VCM to improve the performance and efficiency for machine and human vision in a broad range of AI applications.

B. Bio-Inspired Data Acquisition and Coding

Recently, inspired by the biological mechanism of human vision, researchers invent the bio-inspired spike camera to continuously accumulate luminance intensity and launch spikes when reaching the dispatch threshold. The spike camera brings about a new capacity of capturing the fast-moving scene in a frame-free manner while reconstructing full texture, which provides new insights into the gap between human vision and machine vision, and new opportunities for addressing the fundamental scientific and technical issues in video coding for machines. Valuable works have been done to investigate the spike firing mechanism [160], spatio-temporal distribution of the spikes [161], and lossy spike coding framework for efficient spike data coding [162]. The advance of the spike coding shows other potentials for VCM.

C. Domain Shift in Prediction and Generation

By employing the data-driven methods, the VCM makes more compact and informative features. The risk is that those methods might be trapped in the over-fitting due to the domain shift problem. Targeting more reliable VCM, how to improve the domain generalization of the prediction and generation models, and how to realize the domain adaptation (say, via online learning) are important topics.

IX. CONCLUSION

As a response to the emerging MPEG standardization efforts VCM, this paper formulates a new problem of video coding for machines, targeting the collaborative optimization of video and feature coding for human and/or machine visions. Potential solutions, preliminary results, and future direction of VCM are presented. The state-of-the-art video coding, feature coding, and general learning approaches from the perspective of predictive and generative models, are reviewed comprehensively as well. As an initial attempt in identifying the roles and

principles of VCM, this work is expected to call for more evidence of VCM from both academia and industry, especially when AI meets the big data era.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE TCSVT*, Jul. 2003.

[2] G. J. Sullivan, J. Ohm, W. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE TCSVT*, Dec. 2012.

[3] S. Ma, T. Huang, C. Reader, and W. Gao, "Avs2?making video coding smarter [standards in a nutshell]," *SPM*, 2015.

[4] S. Xia, W. Yang, Y. Hu, S. Ma, and J. Liu, "A group variational transformation neural network for fractional interpolation of video coding," in *IEEE DCC*, Mar. 2018.

[5] J. Liu, S. Xia, W. Yang, M. Li, and D. Liu, "One-for-All: Grouped variation network-based fractional interpolation in video coding," *IEEE TIP*, May 2019.

[6] N. Yan, D. Liu, H. Li, T. Xu, F. Wu, and B. Li, "Convolutional neural network-based invertible half-pixel interpolation filter for video coding," in *IEEE ICIP*, Oct. 2018.

[7] J. Li, B. Li, J. Xu, and R. Xiong, "Efficient multiple-line-based intra prediction for HEVC," *IEEE TCSVT*, Apr. 2018.

[8] Y. Hu, W. Yang, M. Li, and J. Liu, "Progressive spatial recurrent neural network for intra prediction," *IEEE TMM*, Dec. 2019.

[9] J. Liu, S. Xia, and W. Yang, "Deep reference generation with multi-domain hierarchical constraints for inter prediction," *IEEE TMM*, Dec. 2019.

[10] L. Zhao, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Enhanced motion-compensated video coding with deep virtual reference frame generation," *IEEE TIP*, Oct. 2019.

[11] X. Zhang, W. Yang, Y. Hu, and J. Liu, "DMCNN: Dual-domain multi-scale convolutional neural network for compression artifacts removal," in *IEEE ICIP*, Oct. 2018.

[12] C. Jia, S. Wang, X. Zhang, S. Wang, J. Liu, S. Pu, and S. Ma, "Content-aware convolutional neural network for in-loop filtering in high efficiency video coding," *IEEE TIP*, Jul. 2019.

[13] D. Wang, S. Xia, W. Yang, Y. Hu, and J. Liu, "Partition tree guided progressive rethinking network for in-loop filtering of HEVC," in *IEEE ICIP*, Sep. 2019.

[14] Y. Dai, D. Liu, Z. Zha, and F. Wu, "A CNN-based in-loop filter with CU classification for HEVC," in *IEEE VCIP*, Dec. 2018.

[15] Z. Liu, X. Yu, S. Chen, and D. Wang, "CNN oriented fast HEVC intra CU mode decision," in *IEEE ISCAS*, May 2016.

[16] L.-Y. Duan, V. Chandrasekhar, J. Chen, J. Lin, Z. Wang, T. Huang, B. Girod, and W. Gao, "Overview of the MPEG-CDVS standard," *IEEE TIP*, Jan. 2016.

[17] L.-Y. Duan, Y. Lou, Y. Bai, T. Huang, W. Gao, V. Chandrasekhar, J. Lin, S. Wang, and A. C. Kot, "Compact descriptors for video analysis: The emerging MPEG standard," *IEEE TMM*, Apr. 2019.

[18] Y. Xu, C. Lv, S. Li, P. Xin, S. Ma, H. Zou, W. Zhang, "Review of development of visual neural computing," *Computer Engineering and Applications*, 2017.

[19] A. Habibi, "Hybrid coding of pictorial data," *IEEE TOC*, 1974.

[20] R. J. A. and R. G. S., "Combined spatial and temporal coding of digital image sequences," *SPIE*, 1975.

[21] "Video codec for audiovisual services at px64 kbit/s," *ITU-T Rec. H.261*, Nov. 1990.

[22] "Video coding for low bit rate communication," *ITU-T Rec. H.263*, Nov. 1995.

[23] "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 mbit/s–part 2: Video, ISO/IEC 11172-2 (MPEG-1)," *ISO/IEC JTC 1*, 1993.

[24] "Coding of audio-visual objects–part 2: Visual, ISO/IEC 14496-2 (MPEG-4 visual version 1)," *ISO/IEC JTC 1*, Apr. 1999.

[25] "Generic coding of moving pictures and associated audio information–part 2: Video, ITU-T rec. H.262 and ISO/IEC 13818-2 (MPEG 2 video)," *ITU-T and ISO/IEC JTC 1*, Nov. 1994.

[26] "Advanced video coding for generic audio-visual services, ITU-T rec. H.264 and ISO/IEC 14496-10 (AVC)," *ITU-T and ISO/IEC JTC 1*, May 2003.

[27] "H.265: High efficiency video coding," *ITU-T and ISO/IEC JTC 1*, Apr. 2013.

[28] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE SPM*, Nov. 1998.

[29] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density Modeling of Images using a Generalized Normalization Transformation," *arXiv e-prints*, arXiv:1511.06281, Nov. 2015.

[30] W. Park and M. Kim, "CNN-Based in-loop filtering for coding efficiency improvement," in *IEEE IVMSP*, Jul. 2016.

[31] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, and D. Wang, "CU partition mode decision for HEVC hardwired intra encoder using convolution neural network," *IEEE TIP*, Nov. 2016.

[32] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *CVPR*, Jul. 2017.

[33] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable rate image compression with recurrent neural networks," *arXiv e-prints*, arXiv:1511.06085, Nov. 2015.

[34] H. Liu, T. Chen, Q. Shen, and Z. Ma, "Practical stacked non-local attention modules for image compression," in *CVPRW*, Jun. 2019.

[35] H. Liu, T. Chen, P. Guo, Q. Shen, X. Cao, Y. Wang, and Z. Ma, "Non-local attention optimized deep image compression," *arXiv e-prints*, arXiv:1904.09757, Apr. 2019.

[36] T. Dumas, A. Roumy, and C. Guillemot, "Context-adaptive neural network-based prediction for image compression," *IEEE TIP*, 2020.

[37] L. Jiaying, X. Sifeng, , and Y. Wenhan, "Deep reference generation with multi-domain hierarchical constraints for inter prediction," *IEEE TMM*, Dec. 2019.

[38] L. Zhao, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Enhanced CTU-level inter prediction with deep frame rate up-conversion for high efficiency video coding," in *IEEE ICIP*, Oct. 2018.

[39] Y. Dai, D. Liu, Z. Zha, and F. Wu, "A CNN-based in-loop filter with CU classification for HEVC," in *IEEE VCIP*, Dec. 2018.

[40] Y. B. Wang, Z. Han, Y. M. Li, Z. Z. Chen, and S. Liu, "Dense residual convolutional neural network based in-loop filter for HEVC," in *IEEE ICIP*, 2018.

[41] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimization of nonlinear transform codes for perceptual quality," *arXiv e-prints*, arXiv:1607.05006, Jul. 2016.

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

[43] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *arXiv e-prints*, arXiv:1802.01436, Jan. 2018.

[44] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," in *CVPR*, Jun. 2018, pp. 3214–3223.

[45] M. Bober, "MPEG-7 visual shape descriptors," *IEEE TCSVT*, Jun. 2001.

[46] "Information technology on multimedia content description interface part 13: Compact descriptors for visual search," *ISO/IEC 15938-13*, Sep. 2015.

[47] "Information technology on multimedia content description interface part 15: Compact descriptors for video analysis," *ISO/IEC 15938-15*, Jul. 2019.

[48] B. Girod, V. Chandrasekhar, D. M. Chen, N. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. S. Tsai, and R. Vedantham, "Mobile visual search," IEEE SPM, Jul. 2011.

[49] V. Chandrasekhar, G. Takacs, D. M. Chen, S. S. Tsai, Y. Reznik, R. Grzeszczuk, and B. Girod, "Compressed histogram of gradients: A low-bitrate descriptor," *IJCV*, Feb. 2012.

[50] D. M. Chen and B. Girod, "Memory-efficient image databases for mobile visual search," *IEEE MultiMedia*, Jan. 2014.

[51] D. M. Chen, S. S. Tsai, V. Chandrasekhar, G. Takacs, J. Singh, and B. Girod, "Tree histogram coding for mobile image matching," in *IEEE DCC*, Mar. 2009.

[52] R. Ji, L.-Y. Duan, J. Chen, H. Yao, Y. Rui, S.-F. Chang, and W. Gao, "Towards low bit rate mobile visual search with multiple-channel coding," in *ACM MM*, 2011.

[53] R. Ji, L.-Y. Duan, J. Chen, H. Yao, J. Yuan, Y. Rui, and W. Gao, "Location discriminative vocabulary coding for mobile landmark search," *IJCV*, Feb. 2012.

[54] L.-Y. Duan, W. Sun, X. Zhang, S. Wang, J. Chen, J. Yin, S. See, T. Huang, A. C. Kot, and W. Gao, "Fast MPEG-CDVS encoder with GPU-CPU hybrid computing," *IEEE TIP*, May 2018.

[55] "CDVS: Telecom italia's response to CE1 - interest point detection," *ISO/IEC JTC1/SC29/WG11/M31369*, 2013.

[56] C. Loiacono, M. Balestri, G. Cabodi, G. Francini, S. Quer, A. Garbo, and D. Patti, "Accurate and efficient visual search on embedded systems," in *CCIT*, 2015.

[57] F. Gao, X. Zhang, Y. Huang, Y. Luo, X. Li, and L.-Y. Duan, "Data-driven lightweight interest point selection for large-scale visual search," *IEEE TMM*, Oct. 2018.

[58] G. Francini, S. Lepsφy, and M. Balestri, "Selection of local features for visual search," *Signal Processing: Image Communication*, 2013.

[59] Y. Wu, F. Gao, Y. Huang, J. Lin, V. Chandrasekhar, J. Yuan, and L.-Y. Duan, "Codebook-free compact descriptor for scalable visual search," *IEEE TMM*, Feb. 2019.

[60] J. Lin, L.-Y. Duan, Y. Huang, S. Luo, T. Huang, and W. Gao, "Rate-adaptive compact fisher codes for mobile visual search," *IEEE SPL*, Feb. 2014.

[61] S. S. Tsai, D. Chen, G. Takacs, V. Chandrasekhar, J. P. Singh, and B. Girod, "Location coding for mobile image retrieval," in *International ICST Mobile Multimedia Communications Conference*, 2009.

[62] S. S. Tsai, D. Chen, G. Takacs, V. Chandrasekhar, M. Makar, R. Grzeszczuk, and B. Girod, "Improved coding for image feature location information," in *Applications of Digital Image Processing*, 2012.

[63] "Evaluation framework for compact descriptors for visual search," *ISO/IEC JTC1/SC29/WG11/N12202*, 2011.

[64] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, Nov. 2004.

[65] J. Lin, L.-Y. Duan, S. Wang, Y. Bai, Y. Lou, V. Chandrasekhar, T. Huang, A. Kot, and W. Gao, "HNIP: Compact deep invariant representations for video matching, localization, and retrieval," *IEEE TMM*, Sep. 2017.

[66] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015.

[67] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[68] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, Jun. 2016.

[69] "Smart TiledCDVA - MPEG128," *ISO/IEC JTC1/SC29/WG11/M50976*, Oct. 2019.

[70] "Smart sensing - MPEG128," *ISO/IEC JTC1/SC29/WG11/M50966*, Oct. 2019.

[71] "SuperCDVA - MPEG128," *ISO/IEC JTC1/SC29/WG11/M50974*, Oct. 2019.

[72] Y. Lou, L. Duan, S. Wang, Z. Chen, Y. Bai, C. Chen, and W. Gao, "Front-end smart visual sensing and back-end intelligent analysis: A unified infrastructure for economizing the visual system of city brain," *IEEE JCAS*, Jul. 2019.

[73] Y. Lou, L. Duan, Y. Luo, Z. Chen, T. Liu, S. Wang, and W. Gao, "Towards digital retina in smart cities: A model generation, utilization and communication paradigm," in *ICME*, Jul. 2019.

[74] Z. Chen, L. Duan, S. Wang, Y. Lou, T. Huang, D. O. Wu, and W. Gao, "Toward knowledge as a service over networks: A deep learning model communication paradigm," *IEEE JSAC*, June 2019.

[75] Y. Bai, L.-Y. Duan, Y. Luo, S. Wang, Y. Wen, and W. Gao, "Toward intelligent visual sensing and low-cost analysis: A collaborative computing approach," in *IEEE VCIP*, Dec. 2019.

[76] W. Gao, Y. Tian, and J. Wang, "Digital retina: revolutionizing camera systems for the smart city (in chinese)," *Sci Sin Inform*, 2018.

[77] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *NeurIPS*, 2012.

[78] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, Jun. 2016.

[79] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *IJCV*, Dec. 2015.

[80] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *CVPR*, Jun. 2014.

[81] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE TPAMI*, Jun. 2017.

[82] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *CVPR*, Jun. 2015.

[83] S. Kreiss, L. Bertoni, and A. Alahi, "Pifpaf: Composite fields for human pose estimation," in *CVPR*, Jun. 2019.

[84] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[85] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *NeurIPS*, 2014.

[86] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE TPAMI*, Apr. 2017.

[87] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *ICCV*, 2017.

[88] J. Liu, Y. Li, S. Song, J. Xing, C. Lan, and W. Zeng, "Multi-modality multi-task recurrent neural network for online action detection," *IEEE TCSVT*, Sep. 2019.

[89] S. Song, C. Lan, J. Xing, W. Zeng, and J. Liu, "Spatio-temporal attention-based LSTM networks for 3D action recognition and detection," *IEEE TIP*, Jul. 2018.

[90] Y. Li, T. Yao, T. Mei, H. Chao, and Y. Rui, "Share-and-chat: Achieving human-level video commenting by search and multi-view embedding," in *ACM MM*, 2016.

[91] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *NeurIPS*, 2014.

[92] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive growing of GANs for improved quality, stability, and variation," *arXiv e-prints*, arXiv:1710.10196, Oct. 2017.

[93] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional gans," in *CVPR*, Jun. 2018.

[94] Y. Choi, M. Choi, M. Kim, J. Ha, S. Kim, and J. Choo, "StarGAN: Unified generative adversarial networks for multi-domain image-to-image translation," in *CVPR*, Jun. 2018.

[95] W. Xian, P. Sangkloy, V. Agrawal, A. Raj, J. Lu, C. Fang, F. Yu, and J. Hays, "TextureGAN: Controlling deep image synthesis with texture patches," in *CVPR*, Jun. 2018.

[96] C. Ledig, L. Theis, F. Huszzr, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *CVPR*, Jul. 2017.

[97] X. Wang, K. Yu, C. Dong, and C. Change Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *CVPR*, Jun. 2018.

[98] J. Guo and H. Chao, "One-to-many network for visually pleasing compression artifacts reduction," in *CVPR*, Jul. 2017.

[99] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *CVPR*, Jul. 2017.

[100] T. Wang, M. Liu, J. Zhu, A. Tao, J. Kautz, and B. Catanzaro, "High-resolution image synthesis and semantic manipulation with conditional GANs," in *CVPR*, Jun. 2018.

[101] J. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in *ICCV*, Oct. 2017.

[102] Z. Yi, H. Zhang, P. Tan, and M. Gong, "DualGAN: Unsupervised dual learning for image-to-image translation," in *ICCV*, Oct. 2017.

[103] T. Kim, M. Cha, H. Kim, J. K. Lee, and J. Kim, "Learning to discover cross-domain relations with generative adversarial networks," in *ICLR*, 2017.

[104] L. Ma, X. Jia, Q. Sun, B. Schiele, T. Tuytelaars, and L. Van Gool, "Pose guided person image generation," in *NeurIPS*, 2017.

[105] A. Siarohin, E. Sangineto, S. Lathuilire, and N. Sebe, "Deformable GANs for pose-based human image generation," in *CVPR*, Jun. 2018.

[106] G. Balakrishnan, A. Zhao, A. V. Dalca, F. Durand, and J. Guttag, "Synthesizing images of humans in unseen poses," in *CVPR*, Jun. 2018.

[107] A. Pumarola, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "Unsupervised person image synthesis in arbitrary poses," in *CVPR*, Jun. 2018.

[108] S. Reed, Z. Akata, S. Mohan, S. Tenka, B. Schiele, and H. Lee, "Learning what and where to draw," in *NeurIPS*, 2016.

[109] L. Ma, Q. Sun, S. Georgoulis, L. Van Gool, B. Schiele, and M. Fritz, "Disentangled person image generation," in *CVPR*, Jun. 2018.

[110] P. Esser and E. Sutter, "A variational U-Net for conditional appearance and shape generation," in *CVPR*, Jun. 2018.

[111] N. Srivastava, E. Mansimov, and R. Salakhutdinov, "Unsupervised learning of video representations using LSTMs," in *ICML*, 2015.

[112] C. Finn, I. Goodfellow, and S. Levine, "Unsupervised learning for physical interaction through video prediction," in *NeurIPS*, 2016.

[113] W. Lotter, G. Kreiman, and D. Cox, "Deep predictive coding networks for video prediction and unsupervised learning," *arXiv e-prints*, arXiv:1605.08104, May 2016.

[114] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," *arXiv e-prints*, arXiv:1511.05440, Nov. 2015.

[115] S. Aigner and M. Körner, "FutureGAN: Anticipating the future frames of video sequences using spatio-temporal 3D convolutions in progressively growing GANs," *ISPRS*, Sep. 2019.

[116] B. Chen, W. Wang, and J. Wang, "Video imagination from a single image with transformation generation," in *Thematic Workshops of ACM Multimedia*, 2017.

[117] B. De Brabandere, X. Jia, T. Tuytelaars, and L. Van Gool, "Dynamic filter networks," in *NeurIPS*, 2016.

[118] J. van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala, "Transformation-based models of video sequences," *arXiv e-prints*, arXiv:1701.08435, Jan. 2017.

[119] C. Vondrick, H. Pirsiavash, and A. Torralba, "Generating videos with scene dynamics," in *NeurIPS*, 2016.

[120] M. Saito, E. Matsumoto, and S. Saito, "Temporal generative adversarial nets with singular value clipping," in *ICCV*, Oct. 2017.

[121] M. Babaeizadeh, C. Finn, D. Erhan, R. H. Campbell, and S. Levine, "Stochastic variational video prediction," *arXiv e-prints*, Oct. 2017.

[122] E. Denton and R. Fergus, "Stochastic video generation with a learned prior," in *ICML*, Jul. 2018.

[123] T. Xue, J. Wu, K. Bouman, and B. Freeman, "Visual dynamics: Probabilistic future frame synthesis via cross convolutional networks," in *NeurIPS*, 2016.

[124] J. Walker, C. Doersch, A. Gupta, and M. Hebert, "An uncertain future: Forecasting from static images using variational autoencoders," in *ECCV*, 2016.

[125] A. X. Lee, R. Zhang, F. Ebert, P. Abbeel, C. Finn, and S. Levine, "Stochastic Adversarial Video Prediction," *arXiv e-prints*, arXiv:1804.01523, Apr. 2018.

[126] S. Tulyakov, M. Liu, X. Yang, and J. Kautz, "Mocogan: Decomposing motion and content for video generation," in *CVPR*, Jun. 2018.

[127] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing Motion and Content for Natural Video Sequence Prediction," *arXiv e-prints*, arXiv:1706.08033, Jun. 2017.

[128] J. Oh, X. Guo, H. Lee, R. Lewis, and S. Singh, "Action-conditional video prediction using deep networks in atari games," in *NeurIPS*, ser. NIPS, 2015.

[129] R. Villegas, J. Yang, Y. Zou, S. Sohn, X. Lin, and H. Lee, "Learning to generate long-term future via hierarchical prediction," in *ICML*, 2017.

[130] N. Wichers, R. Villegas, D. Erhan, and H. Lee, "Hierarchical long-term video prediction without supervision," *arXiv e-prints*, arXiv:1806.04768, Jun. 2018.

[131] Weiping Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE TCSVT*, Mar. 2001.

[132] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE SPM*, Sep. 2001.

[133] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE TCSVT*, Sep. 2007.

[134] Z. Zhang, Z. Chen, J. Lin, and W. Li, "Learned scalable image compression with bidirectional context disentanglement network," *arXiv e-prints*, arXiv:1812.09443, Dec. 2018.

[135] Z. Guo, Z. Zhang, and Z. Chen, "Deep scalable image compression via hierarchical feature decorrelation," in *IEEE PCS*, 2019.

[136] H. Liu, H. shen, L. Huang, M. Lu, T. Chen, and Z. Ma, "Learned video compression via joint spatial-temporal correlation exploration," *arXiv e-prints*, Dec. 2019.

[137] H. Choi and I. V. Bajic, "Near-lossless deep feature compression for collaborative intelligence," *arXiv e-prints*, arXiv:1804.09963, Apr. 2018.

[138] S. Wang, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Scalable facial image compression with deep feature reconstruction," in *IEEE ICIP*, Sep. 2019.

[139] Z. Chen, K. Fan, S. Wang, L.-Y. Duan, W. Lin, and A. C. Kot, "Intermediate deep feature compression: Toward intelligent sensing," *IEEE TIP*, 2019.

[140] 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," *arXiv e-prints*, arXiv:2001.03004, Jan. 2020.

[141] Y. Hu, S. Yang, W. Yang, L.-Y. Duan, and J. Liu, "Towards coding for human and machine vision: A scalable image coding approach," *arXiv e-prints*, arXiv:2001.02915, Jan. 2020.

[142] Z. Chen, K. Fan, S. Wang, L.-Y. Duan, W. Lin, and A. Kot, "Lossy intermediate deep learning feature compression and evaluation," in *ACM MM*, 2019.

[143] C. Liu, Y. Hu, Y. Li, S. Song, and J. Liu, "PKU-MMD: A large scale benchmark for skeleton-based human action understanding," in *ACM MM*, 2017.

[144] A. Siarohin, S. Lathuilire, S. Tulyakov, E. Ricci, and N. Sebe, "Animating arbitrary objects via deep motion transfer," in *CVPR*, 2019.

[145] I. Pavlov, "Lempel Ziv Markov chain algorithm," in *http://en.wikipedia.org/wiki/LZMA*.

[146] A. R. Zamir, A. Sax, W. Shen, L. J. Guibas, J. Malik, and S. Savarese, "Taskonomy: Disentangling task transfer learning," in *CVPR*, Jun. 2018.

[147] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," *arXiv e-prints*, arXiv:1506.05751, Jun. 2015.

[148] W. Lai, J. Huang, N. Ahuja, and M. Yang, "Fast and accurate image super-resolution with deep laplacian pyramid networks," *IEEE TPAMI*, Nov. 2019.

[149] G. Huang, Z. Liu, L. v. d. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *CVPR*, Jul. 2017.

[150] W. Yang, J. Feng, J. Yang, F. Zhao, J. Liu, Z. Guo, and S. Yan, "Deep edge guided recurrent residual learning for image super-resolution," *IEEE TIP*, Dec. 2017.

[151] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *CVPR*, Jun. 2018.

[152] Y. Tai, J. Yang, X. Liu, and C. Xu, "Memnet: A persistent memory network for image restoration," in *ICCV*, Oct. 2017.

[153] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, "Image super-resolution using very deep residual channel attention networks," in *ICCV*, Jul. 2018.

[154] W. Yang, R. T. Tan, J. Feng, J. Liu, Z. Guo, and S. Yan, "Deep joint rain detection and removal from a single image," in *CVPR*, Jul. 2017.

[155] H. Zhang and V. M. Patel, "Density-aware single image de-raining using a multi-stream dense network," in *CVPR*, Jun. 2018.

[156] W. Dong, M. Yuan, X. Li, and G. Shi, "Joint demosaicing and denoising with perceptual optimization on a generative adversarial network," *arXiv e-prints*, arXiv:1802.04723, 2018.

[157] L. Galteri, L. Seidenari, M. Bertini, and A. Del Bimbo, "Deep generative adversarial compression artifact removal," in *CVPR*, 2017.

[158] O. Kupyn, V. Budzan, M. Mykhailych, D. Mishkin, and J. Matas. "DeblurGAN: Blind motion deblurring using conditional adversarial networks," in *CVPR*, Jun. 2018.

[159] R. Joseph and F. Ali, "YOLOv3: An incremental improvement," *arXiv e-prints*, arXiv:1804.02767, 2018.

[160] S. Dong, Z. Bi, Y. Tian, and T. Huang, "Spike coding for dynamic vision sensor in intelligent driving," *IEEE Internet of Things Journal*, Feb. 2019.

[161] S. Dong, L. Zhu, D. Xu, Y. Tian, and T. Huang, "An efficient coding method for spike camera using inter-spike intervals," in *IEEE DCC*, Mar. 2019.

[162] Y. Fu, J. Li, S. Dong, Y. Tian, and T. Huang, "Spike coding: Towards lossy compression for dynamic vision sensor," in *IEEE DCC*, Mar. 2019.