IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 5 March 2023, accepted 21 March 2023, date of publication 24 March 2023, date of current version 29 March 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3261668

RESEARCH ARTICLE

Slimmable Multi-Task Image Compression for Human and Machine Vision

JIANGZHONG CAO[1], XIMEI YAO[1], HUAN ZHANG[1], JIAN JIN[2], (Member, IEEE), YUN ZHANG[3], (Senior Member, IEEE), AND BINGO WING-KUEN LING[1], (Senior Member, IEEE)

[1]School of Information Engineering, Guangdong University of Technology, Guangzhou 510006, China
[2]Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University, Singapore 639798
[3]School of Electronics and Communication Engineering, Sun Yat-sen University, Shenzhen 518107, China

Corresponding author: Huan Zhang (huanzhang2021@gdut.edu.cn)

This work was supported in part by the Joint Fund of the National Natural Science Foundation of China and Guangdong Province under Grant U1701266, in part by the Guangdong Provincial Key Laboratory of Intellectual Property and Big Data under Grant 2018B030322016, and in part by the Guangdong Basic and Applied Basic Research Foundation under Grant 2021A1515110031.

ABSTRACT In the Internet of Things (IoT) communications, visual data are frequently processed among intelligent devices using artificial intelligence algorithms, replacing humans for analysis and decision-making while only occasionally requiring human scrutiny. However, due to high redundancy of compressive encoders, existing image coding solutions for machine vision are inefficient at runtime. To balance the rate-accuracy performance and efficiency of image compression for machine vision while attaining high-quality reconstructed images for human vision, this paper introduces a novel slimmable multi-task compression framework for human and machine vision in visual IoT applications. Firstly, image compression for human and machine vision under the constraint of bandwidth, latency, and computational resources is modeled as a multi-task optimization problem. Secondly, slimmable encoders are employed for multiple human and machine vision tasks in which the parameters of the sub-encoder for machine vision tasks are shared among all tasks and jointly learned. Thirdly, to solve the feature match between latent representation and intermediate features of deep vision networks, feature transformation networks are introduced as decoders of machine vision feature compression. Finally, the proposed framework is successfully applied to human and machine vision tasks' scenarios, e.g., object detection and image reconstruction. Experimental results show that the proposed method outperforms baselines and other image compression approaches on machine vision tasks with higher efficiency (shorter latency) in two vision tasks' scenarios while retaining comparable quality on image reconstruction.

INDEX TERMS Image compression, feature compression, collaborative compression, intelligent analytics, machine vision.

I. INTRODUCTION

Tn recent years, Internet of Things (IoT) devices have been deployed with deep learning-based models and are getting smarter, which may make decisions and analyses independently or collaboratively even without human intervention. However, most intelligent devices suffer from insufficient storage capacity and computation power; thus cloud servers equipped with deep neural networks are introduced into the IoT environment to share the storage and computation burdens and help with analyzing the data that is sent from intelligent devices. The communication between IoT devices and servers, termed ''machine-machine communication'', thus becomes increasingly more frequent and dominant than conventional human/machine-human communication. Under normal circumstances, few scenarios require human intervention, and only when an exceptional case emerges, for example, manual authentication is needed if there is a verification error in facial recognition using machines, as shown in Fig. 1. Therefore, effective and efficient image

The associate editor coordinating the review of this manuscript and approving it for publication was Mahdi Zareei.

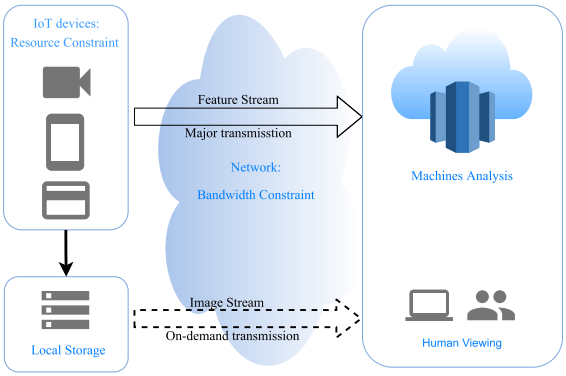

FIGURE 1. Visual Internet of Things communication system, which is dominant with feature stream transmission for machine vision tasks, supplemented by the transmission of image streams for human perception.

compression that satisfies both machine vision tasks and human perception [1], [2] is desired.

Traditional image codecs, such as JPEG [3], HEVC [4], and DNN-based image codecs [5], [6], [7] could compress images to a needed quality for the human perception at a certain bit rate. The compressed images can be further used for machine vision tasks. This type of approach is called "Compress then Analyze (CTA)". However, the whole explosive growth of data is required to be compressed and delivered, which increases the computation burden on the encoder side and causes network jams during transmission. It is not inefficient as most multimedia data are not for humans. Key features may be enough for machine vision.

By contrast, feature coding aims to achieve great performance for machine vision tasks at a specific bit rate. This type of approach is called "Analyze then Compress (ATC)", which extracts compact and analytic-friendly visual features first, and then compresses them for transmission using existing image compression techniques. Examples of such an approach include compact descriptors for visual search (CVDS) [8] and compact descriptors for video analysis (CDVA) [9], which extracted and compressed hand-crafted and deep-learning compact feature descriptors, respectively. However, the abstraction of intermediate features makes it difficult to reconstruct the original image for human perception.

Researchers have recently looked into novel collaborative compression techniques that attempt to integrate feature and image coding. The concept of video coding for machines (VCM) has been described in [1], which aims to satisfy both machine vision and human vision while using the possible minimal amount of computing and communication resources. In certain studies [10], [11], [12], [13], [14], intelligent analysis was performed directly in the compressed domain using multi-task learning techniques. A collaborative image compression and classification framework was proposed in [10]. The method combines image compression with semantic inference using multi-task learning and introduces an adversarial loss for optimization. However, the latent representation is shared by several tasks, yet each task has distinct needs for the information contained in the latent representation, which may cause conflicts.

Scalable methods have become more popular in VCM recently. In [15], the face's edge features served as the base layer, while the color information served as the enhancement layer. And a Generative Adversarial Net (GAN) was used to reconstruct face images suitable for face recognition and human perception from corresponding layers. In order to better serve human and machine vision and to find a better trade-off between computational load and generalization capabilities, some scholars consider a scheme in which image signals and features of different layers are simultaneously compressed and transmitted [16]. However, most methods need to be supplemented with auxiliary modules to produce scalable bit streams, and each feature needs to be encoded by an independent encoder. This makes the whole architecture obese, which requires high computational storage capacity for edge devices and ultimately limits their applications in IoT.

Therefore, IoT applications require image compression algorithms that can be used for both machine vision and human vision, as well as low latency during compression and inference for machine vision tasks. Inspired by [17] and [18], we propose a slimmable multi-task image compression framework by controlling the encoder network's width to adjust latent representation for human and machine vision tasks. For human vision tasks, we use larger encoder widths to ensure the visual quality of the images. For machine vision tasks, we use smaller encoder widths to reduce the bit rate and transmission latency of the latent codes. And the low latency of corresponding latent codes compression could be achieved due to the reduced width of the sub-encoders. Compared with existing methods, our framework can execute encoders at different widths, enabling smooth switching of latent code for different vision tasks. We also explore the effect of the width of the sub-encoder on the performance of machine vision tasks. Our main contributions are summarized as follows

(1) A multi-task image compression framework with slimmable encoders for human and machine vision is proposed, in which slimmable encoders are served for various vision tasks. The proposed method can achieve better rate-accuracy performance on machine vision tasks in two vision task scenarios while maintaining comparable reconstructed image quality than other benchmarks.

(2) A slimmable network that can produce variable-size latent representation for several vision tasks is proposed, and the smallest size of sub-encoder is assigned to the frequently used machine vision application, which could reduce latency on machine vision inference tasks and save bandwidth during IoT communication. And the rate accuracy performance for machine vision could be boosted by learning jointly with compression for human vision.

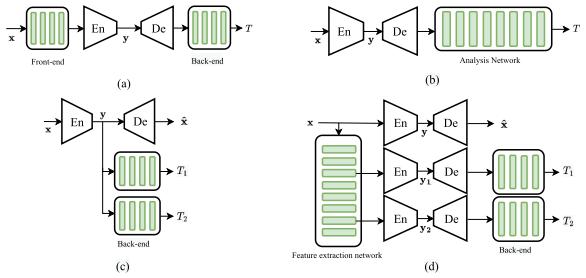

FIGURE 2. Various frameworks of DNN-based compression system. x represents the input image, while $\hat{x}$ represents the reconstructed image. y indicates latent representation. And T_i stand for the specific machine vision inference task i. (a) Analyze then Compress. (b) Compress then Analyze. (c) Multi-task with jointly optimized scalable bit stream with latent-space scalability. (d) Multi-task with separately optimized scalable bit stream.

(3) In addition to the full utilization of communication and computing resources in the IoT communication scenario, as shown in Fig. 1, the proposed framework could somehow protect the users' privacy since the feature stream and image stream are separate.

II. RELATED WORK

Currently, image compression methods for human vision and machine vision can be broadly classified into four categories. Fig. 2 shows the general framework of these four approaches.

A. FEATURE CODING

One is called "Analyze then Compress (ATC)", as shown in Fig. 2(a). Earlier works compress and transmit features that can be used directly for machine vision tasks. MPEG has completed the standardization of compact descriptors for visual search (CVDS) [8] and compact descriptors for video analysis (CDVA) [9], which standardize hand-crafted and deep-learning compact feature descriptors, respectively. This greatly advances feature compression. However, this kind of highly concentrated feature coding may be limited to specific tasks and scenarios [1]. Meanwhile, the development of deep learning has prompted the emergence of new feature compression schemes, which mainly transmit the intermediate features of deep models for machine vision analysis tasks and which layers to transmit depending on the subsequent tasks [19]. In this scenario, intelligent analysis networks are divided into two parts, and one is deployed at the edge as a feature extraction network, called the front-end network. The other is deployed in the cloud, termed back-end network. There is a lot of research devoted to improving the compression efficiency of intermediate features [20], [21], [22], [23], [24]. For example, Singh et al. [23] proposed an end-to-end learning approach that jointly optimized the bit rate and task objective. Choi et al. [24] proposed a compression method that selectively compressed a subset of deep feature tensors and restored the original deep feature tensors with the proposed back-and-forth (BaF) predictor to complete the analysis task in the cloud.

B. IMAGE CODING

Another is "Compress then Analyze (CTA)", and its structure is shown in Fig. 2(b), which combines compression and machine vision analysis network structure and devises joint optimization strategies. Some methods [25], [26], [27], [28], [29], [30], which are based on existing learned image compression frameworks, obtain the reconstructed image more appropriate for analysis through joint learning. However, in most cases, the quality of the image suffers. For instance, in [27] and [29], the compression module and the analysis network are successively coupled and trained together, and accuracy (e.g., mAP) of machine vision task increases with optimization, but the quality (e.g., PSNR) of reconstructed image declines. In [28] and [30], the network structure and machine vision task-friendly optimization method result in a severely compressed background in the image, which negatively affects the perceptual quality of the background. In addition, since these methods still reconstruct images after decoding, a complete cloud analysis network is still required to perform machine vision tasks. It greatly increases the resource consumption of the cloud. The aforementioned two strategies in feature coding are helpful for machine vision tasks while ineffective for human perception.

The image coding and feature coding both treat the compression processes and vision analysis as two individual tasks though some work [27], [29] jointly train the two tasks, and they either favor human perception or machine vision. Therefore, both image coding and feature coding for human and machine vision require the total computing consumption of compression processes and vision analysis and may not be effective for both two tasks. How to combine image coding and feature coding to improve resource utilization and compression efficiency becomes an important issue.

C. SCALABLE CODING

There is a popular approach to jointly train multiple vision compression tasks and perform machine vision tasks directly on the compressed domain [10], [11], [12], [13], [14], as shown in Fig. 2(c). For instance, in [12], the training was carried out using a multi-task loss consisting of classification loss, reconstruction loss and rate loss, and the classification is performed on the quantized representation. Another important approach is to make latent codes scalable. Some work [15], [31], [32], [33] separates the compressed bitstream into two layers, i.e., the base layer and the enhancement layer. The base layer is used for intelligent analysis, and the enhancement layer is utilized to fuse with the base layer to reconstruct the input image for human perception. In [31] and [32], the Facenet network's deep features were utilized as the base layer, and the coarse input image was first reconstructed from the base layer. The fine input image was obtained by adding the coarse input image with the enhancement layer, which was the decompressed residual of the original input image and the coarse input image.

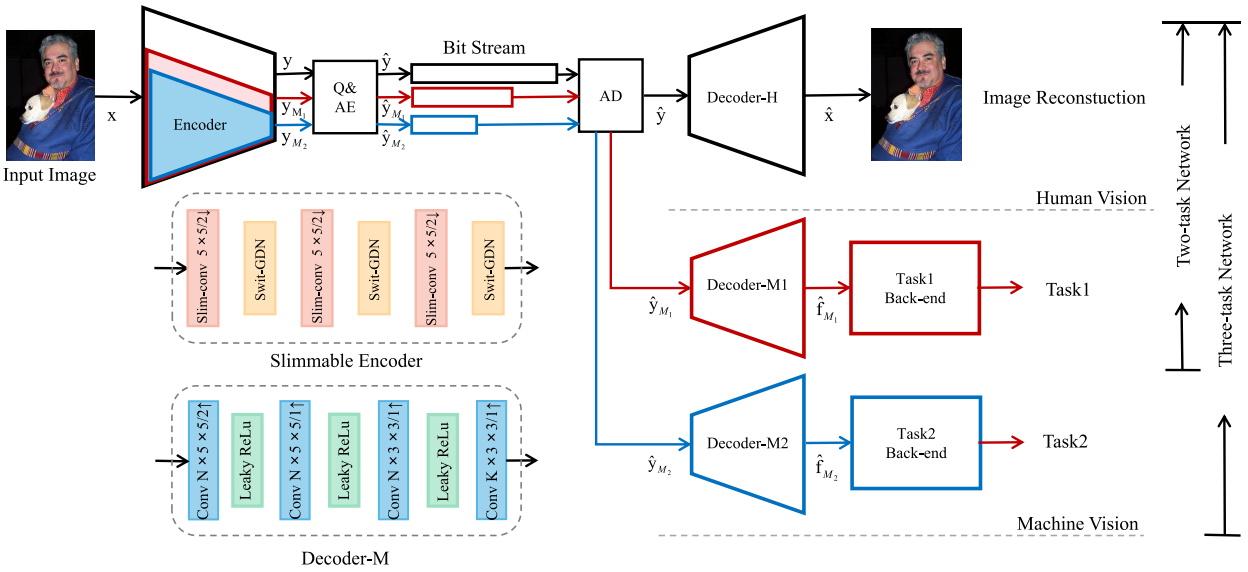

FIGURE 3. An example implementation of the proposed slimmable multi-task image compression framework. "Q" represents quantization. "AE" and "AD" stand for arithmetic encoder and arithmetic decoder, respectively. The network architectures of the slimmable encoder and decoder-M are illustrated at the bottom of the picture. *N* and *K* are the number of channels of convolution, which are chosen according to the task. And the filters' kernel size and stride are followed by the number of channels. ↑ and ↓ indicate upsampling and downsampling in each convolutional layer, respectively. Slim-conv represents the slimmable convolutional layer and swit-GDN represents the switchable GDN layer.

Another popular approach is to compress and transfer the image signal and features simultaneously [16], [34]. There have been some studies extending to more visual analysis tasks, the structure is shown in Fig. 2(d). For instance, in [34], the authors proposed a method for compressing multiple deep feature maps, which are intermediate representations of deep networks. The deep-to-shallow feature maps will be used for the coarse-to-fine analysis task.

III. PROBLEM FORMULATION AND MOTIVATION

In IoT applications, the goal of image compression for human and machine vision to minimize the accuracy loss for machine analysis and the reconstruction loss for human eyes within the resource constraint of intelligent devices, e.g., memory, storage, and computational cost limits, while demanding for low latency for visual analysis.

$$\begin{aligned} &arg \min_{\{\phi,\varphi\}} D^M + D^H, \\ &\text{s.t.}\, f(R_M, R_H) \le R_T, \\ &\quad g(\xi_M, \xi_H) \le \xi_T, \\ &\quad h(\gamma_M, \gamma_H) \le \gamma_T, \\ &\quad l(C_M, C_H) \le C_T, \end{aligned} \tag{1}$$

where D^M denotes the accuracy performance for machine vision, and D^H denotes the signal reconstruction performance degradation for human vision. 'M' and 'H' represent machine vision and human vision, respectively. The bit rates, memory, storage, and computational costs of image compression are under the constraints of R_T, ξ_T, γ_T, C_T, respectively. ϕ, φ are the parameters of encoders and decoders to be optimized. $f(.,.)$ calculates the bit rates of encoded latent codes, $g(.,.)$, $h(.,.)$ compute the operation memory, storage the encoders need, respectively, $l(.,.)$ counts the computational costs of encoders consume. The specific way of $f(.,.)$, $g(.,.)$, $h(.,.)$, $l(.,.)$ integrating machine vision and human vision terms depend on the compression network structures.

In IoT application, a captured image $\mathbf{x}$ could be compressed as latent representation y_M or the feature maps of backbones f are transmitted for visual analysis tasks T in most cases. In addition, the image $\mathbf{x}$ will be compressed as latent representation y for reconstructing $\hat{\mathbf{x}}$ occasionally required by human observers. To make the most of the resources and achieve effective and efficient image coding for visual analysis while guaranteeing the reconstructed image quality, different solutions can be attempted to address this issue. Scalable image coding is a possible solution. However, the whole encoders are used for image coding for both machine and human vision in existing scalable image coding solutions, thus the efficiency of the encoder for machine vision is confined by that of the encoder for human vision. How to improve the efficiency of image coding for machine vision with negligible rate accuracy performance degradation needs to be explored.

Suppose the reconstructed image $\hat{\mathbf{x}}$ is approximately equal to the input image $\mathbf{x}$, the inference tasks T can be conducted on reconstructed image $\hat{\mathbf{x}}$. From [33] it can be inferred that the processing chain can be described in Markov chain as $\mathbf{x} \rightarrow \mathbf{y} \rightarrow \hat{\mathbf{x}} \rightarrow \mathbf{f} \rightarrow T$. And their mutual information relationship can be written as

$$I(\mathbf{x}; \mathbf{y}) \ge I(\mathbf{x}; \mathbf{f}), \tag{2}$$

where $I(.;.)$ calculates the mutual information between two vectors. It indicates the mutual information between $\mathbf{x}$ and

f is smaller than that between **x** and **y**. Thus, we assume that the necessary information need to extracted from **x** to generate **f** for visual analysis is less than that for **y** for human vision. For example, edges information are more necessary for image detection than color information for image quality. Therefore, we conjecture that compressive encoders with appropriate fewer channels are enough to extract necessary key features for generating **f** for visual analysis. Motivated by the above analysis and inspired by [17] and [18], we propose a slimmable multi-task image compression method for human and machine vision, in which reduced-size encoders are splitted as sub-encoders for machine vision and the original size encoder is for human vision. The machine and human vision tasks are jointly learned to optimize Eq (1).

IV. PROPOSED SLIMMABLE MULTI-TASK IMAGE COMPRESSION FOR HUMAN AND MACHINE VISION

A. THE PROPOSED LEARNED IMAGE COMPRESSION FRAMEWORK

Basically, a learned image compression framework includes four modules, i.e., an encoder, a quantizer, an entropy model, and a decoder. The overall framework of our proposed learned image compression for human and machine vision is shown in Fig. 3. The major difference is that slimmable encoders with multiple sub-encoders are proposed for several vision tasks, and the corresponding number of entropy models, and decoders for vision tasks are configured. The basic structure of the proposed method is based on two learned image compression models, i.e., the factorized model called bmshj2018-factorized and the hyperprior model called bmshj2018-hyperprior from [5]. For simplicity, we denote bmshj2018-factorized and bmshj2018-hyperprior models as BFM and BHM. For BFM and BHM, the difference is that the entropy model of BFM is a factorized-prior model, while that of BHM is a hyperprior model. The two proposed networks based on the two compression models are called S-BFM and S-BHM.

B. THE PIPELINE OF SLIMMABLE IMAGE COMPRESSION

As shown in Fig. 3, the encoder g_a transforms the input image **x** to the latent representation y, which is quantized by the quantizer Q and then transmitted after entropy coding. y is then decoded and transformed by the decoder g_s to obtain the reconstructed input $\hat{x}$. In our method, the input reconstruction operation, the same as that in [5], can be described as

$$\mathbf{y} = g_a(\mathbf{x}; \phi), \tag{3}$$
$$\hat{\mathbf{y}} = Q(\mathbf{y}), \tag{4}$$
$$\hat{\mathbf{x}} = g_s(\hat{\mathbf{y}}; \varphi), \tag{5}$$

where ϕ and φ are the parameters of the encoder and decoder, respectively.

For image compression for machine vision, a certain layer in a backbone analysis network is chosen and taken as the separatrix to divide the analysis network into two parts, i.e., the front-end network and the back-end network. The intermediate features $\mathbf{f}_{M_i}$ generated by the intermediate layer of the front-end network are compressed and transmitted to serve as the input to the back-end networks in [24]. Different from this strategy, in our work, the front-end network and feature compression processes are replaced by a process in which the intermediate features $\mathbf{f}_{M_i}$ are directly generated from images. For the analysis task i, the sub-encoder g_{a_i}, which is a reduced-width sub-encoder embedded in the native encoder, produces the latent variable $\mathbf{y}_{M_i}$. To perform the analysis task i, a decoder module decoder-M$_i$ g_{s_i} is builted to reconstruct the intermediate features $\hat{\mathbf{f}}_{M_i}$, which can be described as

$$\mathbf{y}_{M_i} = g_{a_i}(\mathbf{x}; \phi_i), \tag{6}$$
$$\hat{\mathbf{y}}_{M_i} = Q(\mathbf{y}_{M_i}), \tag{7}$$
$$\hat{\mathbf{f}}_{M_i} = g_{s_i}(\hat{\mathbf{y}}_{M_i}; \theta), \tag{8}$$

where the sub-encoder g_{a_i} shares parameters with the original encoder. The individual modules of the proposed slimmable compression framework will be discussed in the following subsections.

C. SLIMMABLE COMPRESSIVE ENCODERS

As has been analyzed in Section III, the larger width of compressive encoders entails higher dimensional feature embedding with more details, and there exists an enough (reduced-size) width of encoder to compress images into latent codes for visual analysis. In addition, it can be inferred from [33] that more information is required for image reconstruction than for visual analysis tasks. Consequently, slimmable compressive encoders for human and machine vision are proposed, in which the original-size encoder is for image reconstruction, while sub-encoders with reduced width are assigned for low-latency visual analysis tasks. Assume the encoder has S sub-encoders, which enable it to perform S intelligent analysis tasks. An encoder with a smaller width will share its parameters with an encoder with a larger width. And the parameters of the sub-encoder with the smallest width are shared among all tasks. For instance, in the network with a machine vision task, the width of the sub-encoder g_{a_1} for machine vision task is smaller than the native encoder g_a for image reconstruction task, then ϕ_1 is part of ϕ. That is, the two encoders share the parameters ϕ_1. The specific structure of slimmable encoders is shown in Fig 3.

To realize the slimmablity of compressive encoders, similar to [17], convolutional layers are set to be slimmable convolutional layers which could discard a few channels of layers during operation, thus the slimmable encoders can be mapped to serve multiple vision tasks. GDN layers are set to be switchable GDN layers in our slimmable compressive encoders so that independent normalization computations of feature map distributions are switched among different subnetworks.

D. DECODER MODULES FOR IMAGE AND FEATURE RECONSTRUCTION

To make a distinction between decoders for human and machine vision, the decoder for image reconstruction is denoted as decoder-H, and the decoders for visual analysis are denoted as decoder-M$_i$. The structure of decoder-H in the proposed compression framework is the same as that in [5], which consists of deconvolution layers and IGDN layers. For visual analysis, a transformation network, also decoder-M$_i$, is constructed to transform the latent variables $\hat{\mathbf{y}}_{M_i}$ into the intermediate features $\hat{\mathbf{f}}_{M_i}$ of the analysis network. The structure of decoder-M$_i$ is demonstrated in Fig. 3. The difference between decoder-M$_i$ from decoder-H is that the configuration in each layer of decoder-M$_i$, i.e., the channel number, kernel size, and stride, will change depending on the analysis task to ensure that its output matches the input of the back-end network. For example, the selected intermediate features in visual analysis network for instance segmentation may vary from that for object detection. Therefore, to match the dimensions of the corresponding intermediate features, output dimensions of the convolutional layers of the decoder-M$_i$ modules for different tasks should be custom-set. Additionally, LeakyReLU layers are used rather than IGDN layers so that the output features of the decoder-M$_i$ can match the dynamic range of the features in visual analysis network.

E. LOSS FUNCTION

Our goal of the proposed slimmable compression method is to optimize the averaged rate-distortion (or accuracy) performance over image reconstruction and vision analysis tasks, especially improving rate-accuracy performance of image vision analysis while retaining rate-distortion performance of image reconstruction. Thus, the total loss function can be written as

$$L_{total} = \alpha_H L_H + \sum_{i=1}^{S} \alpha_{M_i} L_{M_i}, \tag{9}$$

where L_H represents the rate-distortion loss of image reconstruction task and L_{M_i} represents rate-feature-distortion loss of machine vision task i. And the parameters of α_H and α_{M_i} control the direction of optimization. Combining Eq. (1), the corresponding Lagrangian form loss function can be constructed as

$$L = R_\Phi + \lambda_\Phi D^\Phi, \tag{10}$$

where $\Phi \in \{H, M\}$, R_Φ denotes the bit rate, D^Φ represents the distortion or accuracy of vision task Φ, and λ_Φ is the Lagrangian multiplier. Specifically, for image reconstruction, the loss function can be formulated as

$$L_H = E_{x\sim p_x}[-log_2(Q(g_a(\mathbf{x};\phi)))] + \lambda_H MSE(\mathbf{x}, \hat{\mathbf{x}}), \tag{11}$$

where x is the input image and $\hat{x}$ is the reconstructed image. The first term estimates the rate of the quantized latent code of $\mathbf{x}$ for human vision. $MSE(.)$ is used to measure how well the predicted value $\hat{x}$ matches the original x by calculating Mean Square Error (MSE) between $\hat{x}$ and x. For the machine vision task, the loss function can be written as

$$L_{M_i} = E_{f\sim p_f}[-log_2(Q(g_{a_i}^{(c_i)}(\mathbf{x};\phi_i)))] + \lambda_{M_i} MSE(\mathbf{f}_{M_i}, \hat{\mathbf{f}}_{M_i}), \tag{12}$$

where the first term estimates the rate of the quantized latent code $\hat{\mathbf{y}}_{M_i}$ of $\mathbf{x}$ for machine vision. The second term computes the MSE between output features $\hat{\mathbf{f}}_{M_i}$ of the $g_{s_i}^{(c_i)}$ and intermediate feature $\mathbf{f}_{M_i}$ of the pre-trained analysis network for machine vision task i. c_i represents the width of the sub-encoder $g_{a_i}^{(c_i)}$ of machine vision task i. The determination of c_i affects the latency and performance of machine vision tasks, and also the performance of image reconstruction. How to select a suitable c_i is of importance, and we will discuss it in the experimental section.

V. EXPERIMENTS

In this section, the configuration of experiments is first introduced. The selection of the width of the sub-encoder for machine vision tasks is discussed afterward, then the performance of our proposed slimmable encoders for two vision tasks is evaluated and compared. Finally, the encoder's efficiency for machine vision tasks is compared.

A. EXPERIMENTAL SETTING

1) TRAINING SETTING

Inspired by [18], we mainly adopted the training strategy in [18], which optimizes its loss averaged from all switches. During training, the width of our proposed slimmable encoders is switched once for each batch of data. We compute the respective rate-distortion loss for encoders with different widths and perform a parameter update. In every batch of training, the encoder and decoder-H for human vision are first trained with the parameter α_H set to 1 and α_{M_i} set to 0. Next, the sub-encoder with reduced width and the decoder-M is trained for the object detection or instance segmentation tasks with the parameter α_H set to 0 and α_{M_i} set to 1. Each batch of training process is alternated between the respective task losses. To speed up training, a high-rate model is trained first, and then the low-rate models are fine-tuned based on the pre-trained high-rate model's weight. COCO train2014 data set [36] are used to train the network with two vision tasks. The training image is initially resized to 256 $\times$ 256 before being fed into the model for training, and the batch size is set to 64. The models are optimized using Adam optimizer with learning rate of 1e-4 on encoders and decoders, and learning rate of 1e-3 on entropy model. And the ReduceLROnPlateau learning strategy is chosen, which reduces the learning rate when the loss is no longer decreasing. The values of λ_ϕ, shown in Table 1, are used in the loss function to produce six versions of trained models. The hyper-parameters λ_H and λ_M for rate-distortion model were selected based on alternate optimization by fixing one parameter and changing the other. Each pair of λ_H and λ_M with the smallest loss were chosen.

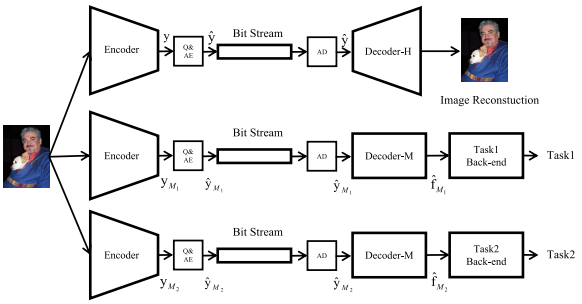

FIGURE 4. Individual compression. Images are compressed separately for human and machine vision tasks.

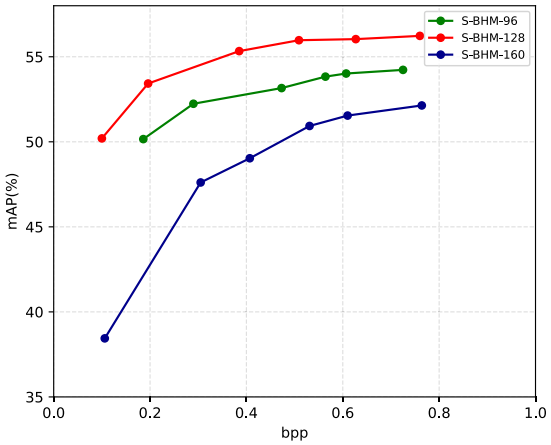

FIGURE 5. Comparison of object detection performance among the proposed S-BHM networks with various widths of sub-encoder. Note that only the feature bitrate is calculated in this figure.

The code of the proposed network is developed on PyTorch. The model was trained on a workstation with a 2.3GHz dual-core processor (Intel Xeon E5-2686 v4) and two TITAN RTX GPUs.

In the proposed compressive encoders, the original encoder generates a 192-dimensional latent representation for image reconstruction, and a sub-encoder with a width of 128 dimensions is set up for the object detection task. The selection of the width for sub-encoder is discussed in Section V-B. And the output of the decoder-M is used as the 14th layer's input of YOLOv3 network [35] at the decoding end, and the object detection result is acquired after the back-end network calculation. COCO2014 dataset's validation set [36], which includes 80 different object categories, was used as the test dataset for object detection and image reconstruction tasks.

2) EVALUATION METRICS

Bit per pixel (bpp) is the coding length required for per-pixel coding. The mean average precision (mAP) metrics, which are the average of the Average precision (AP) over IoU thresholds from 50% to 95%, are used to represent the object detection performances. For image reconstruction, Peak Signal to Noise Ratio (PSNR) and Mean Structural Similarity (MS-SSIM) are the evaluation metrics used to assess image quality.

3) COMPARISON METHODS

To demonstrate the effectiveness of our proposed slimmable multi-task compression framework, a baseline model is compared, as shown in Fig. 4. Specifically, in the baseline method, two independent encoders are trained separately for human and machine vision tasks. The structures of these two independent encoders are the same as those of the encoders used for human and machine vision tasks in the slimmable encoder. Corresponding to the proposed S-BFM and S-BHM, the baseline models can be termed as Ba-BFM or Ba-BHM, respectively. For machine vision, Equation (12) is used as the loss function for baseline models. λ_M values used in Equation (12) are shown in Table 3. The pretained models from [37] are used as the baseline model for human vision.

In addition to the baseline Ba-BFM and Ba-BHM models, a comparison method proposed is called latent space scalability, dubbed as LSS, is introduced to test whether the proposed slimmable encoders, i.e., reusable encoder, are more effective than the LSS, i.e., reusable latent codes. The LSS method uses an encoder to compress images into latent codes, of which 128 dimensions of which are extracted for intermediate feature representation for object detection, and all the latent codes are used for image reconstruction. Corresponding to the proposed S-BFM, for a fair comparison to the proposed S-BFM, the LSS method is adapted by employing the BFM [5] as the compression model instead of the compression model in [14]. Besides, the input image format of the adapted LSS is changed from YUV to RGB. We denote the adapted LSS as LSS-BFM.

Conventional codecs like JPEG and HEVC are also included in the comparison methods. The quantization parameters of HEVC are set to 22, 25, 28, 31, 34, and 37 in the experiments. And the JPEG quality level ranges from 10 to 60 in steps of 10.

B. SELECTION OF THE WIDTH OF SUB-ENCODERS FOR MACHINE VISION TASKS

It is important to choose a suitable c_i of sub-encoder $g_{a_i}^{(c_i)}$ for machine vision task i. When c_i is small, fewer channels are assigned for joint machine vision and human vision feature compression, and more channels are left exclusively for detail construction for human vision. We mainly explore the effects of c_i on human and machine vision performance in two tasks, i.e., object detection and image reconstruction. The proposed slimmable compression models S-BHM with 96, 128, 160 of c_1 are compared. In addition, the original compression model BHM is counted as slimmable encoders S-BHM with 0 channel for machine vision. Thus, four models S-BHM-0, S-BHM-96, S-BHM-128, and S-BHM-160 were compared. Since 0 channel is used for the machine vision task, there are no results of S-BHM-0 for machine vision task. All tested compression models were tested on a series of six

TABLE 1. λ_H and λ_M values settings in the loss function for the proposed slimmable compression models.

Model Index	1		2		3		4		5		6	
	λ_H	λ_M	λ_H	λ_M	λ_H	λ_M	λ_H	λ_M	λ_H	λ_M	λ_H	λ_M
S-BFM	0.003	0.00007	0.005	0.0001	0.008	0.0002	0.01	0.0003	0.02	0.0004	0.03	0.0008
S-BHM	0.001	0.00005	0.002	0.00007	0.005	0.0001	0.01	0.0003	0.02	0.0005	0.03	0.0008

TABLE 2. λ_H and λ_M values settings in the loss function for the proposed slimmable compression models with various widths of sub-encoder for machine vision.

Model Index	1		2		3		4		5		6	
	λ_H	λ_M	λ_H	λ_M	λ_H	λ_M	λ_H	λ_M	λ_H	λ_M	λ_H	λ_M
S-BHM-96	0.005	0.0001	0.01	0.0002	0.02	0.0005	0.03	0.0008	0.04	0.001	0.06	0.0015
S-BHM-128	0.001	0.00005	0.002	0.00007	0.005	0.0001	0.01	0.0003	0.02	0.0005	0.03	0.0008
S-BHM-160	0.005	0.0001	0.01	0.0002	0.02	0.0005	0.03	0.0008	0.04	0.001	0.06	0.0015

TABLE 3. λ_M values for training baseline models for object detection.

Model Index	1	2	3	4	5	6
Ba-BFM	0.0001	0.0002	0.0004	0.0008	0.001	0.002
Ba-BHM	0.00005	0.0001	0.0003	0.0005	0.001	0.002

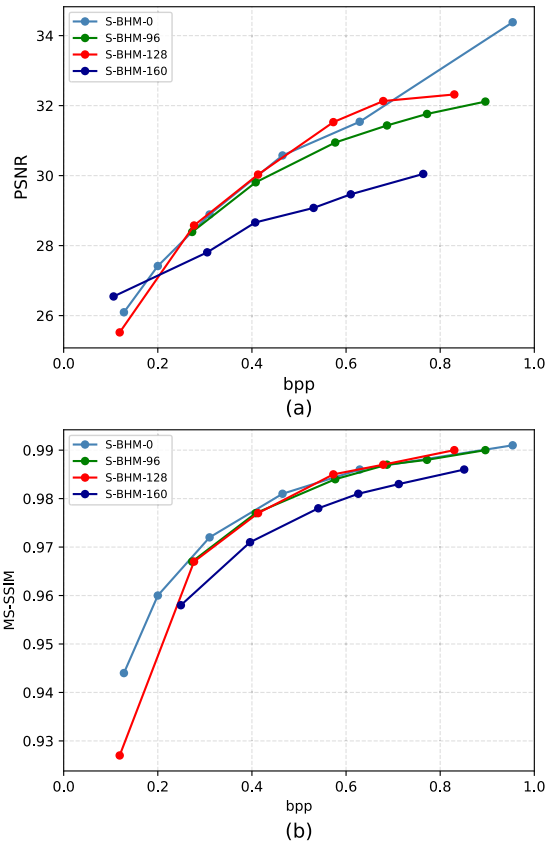

FIGURE 6. Comparison of image reconstruction performance among the proposed S-BHM networks with various width of sub-encoder. Note that only the image bitrate is calculated in this figure.

bitrates by setting λ_H to six different values, which are shown in Table 2.

Fig. 5 demonstrates the bpp-mAP results of S-BHM-96, S-BHM-128, and S-BHM-160 models, and Fig. 6 demonstrates the bpp-PSNR and bpp-MS-SSIM

TABLE 4. Performance of the proposed slimmable compression models S-BFM/S-BHM for object detection against baselines with BD metrics.

	S-BFM		S-BHM	
Benchmarks	BD-Bitrate	BD-mAP	BD-Bitrate	BD-mAP
Ba-BFM	-24.220	1.140	-	-
Ba-BHM	-	-	-36.232	0.393
LSS-BFM	9.983	-0.317	-	-
JPEG	-75.759	10.998	-96.594	11.686
HEVC	-54.492	4.479	-44.960	4.752

results of S-BHM-0, S-BHM-96, S-BHM-128, and S-BHM-160 models. It can be observed that the bpp-mAP curves obtained by S-BHM-96 and S-BHM-160 are below the bpp-mAP curve obtained by S-BHM-128, and obviously the S-BHM-128 model performs the best, and the S-BHM-160 model does the worst among the three models. It may be inferred that as c_1 increases, the rate-accuracy performance of object detection first increases and then decrease. For image reconstruction, the bpp-PSNR and bpp-MS-SSIM curves obtained by the four models S-BHM-0, S-BHM-96, S-BHM-128 are close while S-BHM-160 model is relatively inferior. It may be inferred that sparing some channels for joint human and machine vision feature compression may not influence the image reconstruction performance greatly, and when the number of channels for sharing is too large, the image reconstruction performance becomes worse. The reason could be that when c_1 is too large, fewer channels are assigned for individual image reconstruction, thus affecting the image reconstruction performance directly and vision analysis performance (object detection) indirectly through joint learning. Therefore, considering both the visual analysis and image reconstruction performance, and also the latency for vision analysis, c_1 is simply set as 128 for object detection in our proposed slimmable encoders.

C. PERFORMANCE ON HUMAN-VISION PERCEPTION AND MACHINE-VISION OF TWO TASKS

1) OBJECT DETECTION

Fig. 7 shows the mAP results in the range of [0-1] bpp. In this case, the black dashed line is the result of the YOLOv3 network loaded with pre-trained parameters for an input image size of 512 × 512, which has a value of 57.68%. The

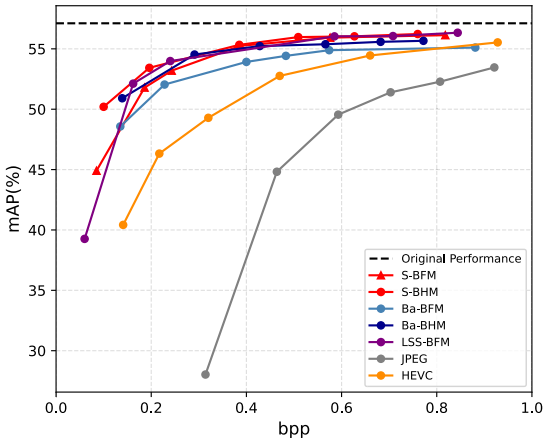

FIGURE 7. Comparison of object detection performance among proposed network and benchmarks. Note that only the feature bitrate is calculated in this figure except that the performance of JPEG and HEVC are listed as anchor.

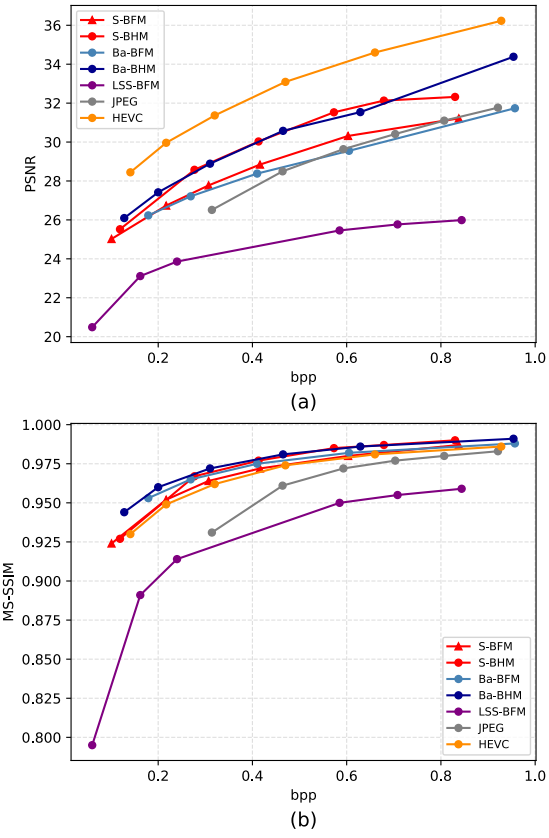

FIGURE 8. Comparison of image reconstruction performance among proposed network and benchmarks. Note that only the image bitrate is calculated in this figure except that the performance of JPEG and HEVC are listed as anchor.

red lines are the results of the proposed networks S-BFM and S-BHM. Table 4 details the performance of BD-mAP, which is the extended BD metrics. The average increase or decrease in mAP at a given bit rate is represented by BD-mAP. The BD-bitrate is expressed as an average percentage of bit savings at the same precision, and negative numbers represent savings. It can be observed that the bpp-mAP

TABLE 5. Performance of the proposed slimmable compression models S-BFM/S-BHM for image reconstruction against baselines with BD metrics.

Benchmarks	S-BFM		S-BHM	
	BD-Bitrate (PSNR)	BD-Bitrate (MS-SSIM)	BD-Bitrate (PSNR)	BD-Bitrate (MS-SSIM)
Ba-BFM	-9.631	17.984	-	-
Ba-BHM	-	-	-0.308	7.114
LSS-BFM	-78.696	-67.473	-81.974	-67.294
JPEG	-18.038	-36.025	-37.044	-53.099
HEVC	156.312	-7.072	82.149	-21.607

curves of S-BFM and S-BHM are above the curves of their respective baselines, JPEG and HEVC. Specifically, the S-BFM performs better than JPEG and HEVC in mAP by 10.998% and 4.479% improvement, respectively. And S-BHM also improves 11.686% and 4.752% mAP over JPEG and HEVC. Compared to baseline Ba-BFM, our S-BFM achieves 1.140% mAP improvement and -24.220% bit savings. And the S-BHM network improves 0.393% mAP and saves -36.332% bits over Ba-BHM. The results show that object detection task performed well with jointly learning on slimmable models than independent training. The improved performance may be attributed to the knowledge sharing introduced by parameter sharing. In addition, the proposed S-BFM obtained slightly worse performance on object detection than the LSS-BFM method with -0.317% BD-mAP and 9.983% BD-bitrate. This is because the parameter setting of the loss function of LSS-BFM makes the optimization more inclined in the direction of machine vision.

Fig. 9 shows the reconstructed images and object detection results at around two levels of bitrates. The corresponding bitrates, PSNR, and MS-SSIM values are given below the reconstructed images, and the bounding box results of the object detection task are also visualized and displayed. For JPEG and HEVC methods, object detection is conducted after the input image has been compressed and reconstructed, so the bitrate of the object detection task is the same as that of image reconstruction. For other comparisons and our proposed methods, the object detection task separates from the image reconstruction, and the bounding box results are displayed on a black background. The results of original images are listed as anchor. For the first rate example, it can be observed that S-BFM detects more true objects (bounding boxes) than Ba-BFM and JPEG at similar or lower bitrates. Compared with HEVC and LSS-BFM methods, the bit rate of S-BFM could achieve similar object detection performance at lower bit rate. While S-BHM and BHM have the same number of detected objects, the object confidence scores of S-BHM are higher. A similar performance can also be observed in the second example.

2) IMAGE RECONSTRUCTION

Fig. 8 shows the corresponding bpp-PSNR and bpp-MS-SSIM curves of the proposed network, baseline approaches, and LSS-BFM method. Table 5 shows the BD-bitrate of our two implementations, compared to their respective baseline

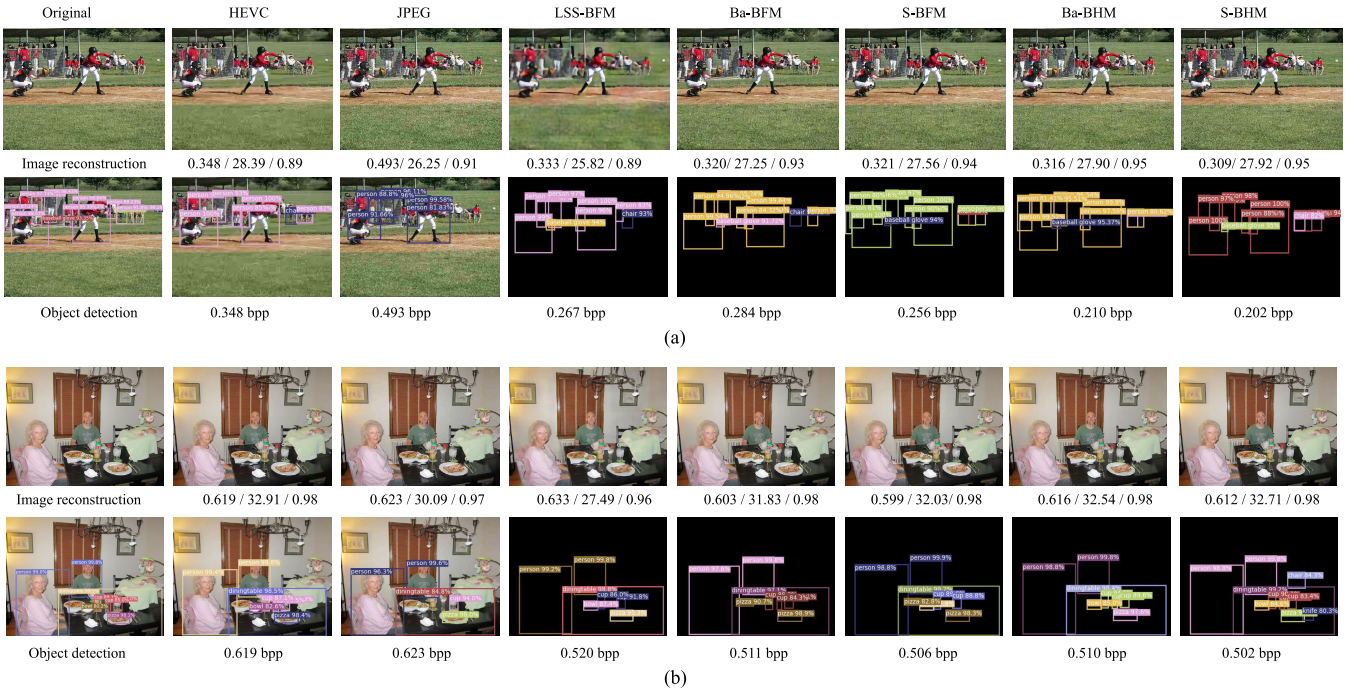

FIGURE 9. Visual examples of our models' outputs compared with several benchmarks. For visualization purposes, we only highlighted objects with a confidence score of over 80%. The bit rate (bpp), PSNR (dB), and MS-SSIM values are displayed sequentially below the reconstructed image.

models and LSS-BFM. As can be shown, the bpp-PSNR curves of the S-BFM and S-BHM are close to the bpp-PSNR curves of Ba-BFM and Ba-BHM. For PSNR, the bitrate saving is −9.631% and −0.308% for S-BFM and S-BHM compared to their respective baseline models. On MS-SSIM, our method is slightly lower than its baseline, with 17.984% and 7.114% increased for S-BFM and S-BHM, respectively. The results demonstrate that the proposed method can maintain the reconstructed image quality of the baseline model to a certain extent while improving the performance of machine vision tasks. In comparison to method LSS-BFM, S-BFM achieves −78.696% and −67.473% bitrate savings in PSNR and MS-SSIM compared to the method of LSS-BFM with a factorized entropy model. The reconstructed image quality of S-BFM is superior to that of LSS-BFM, while the machine vision performance of S-BFM is comparable to LSS-BFM, demonstrating the effectiveness of the proposed slimmable encoder. Compared with HEVC, the proposed method outperforms HEVC in MS-SSIM but is inferior to HEVC in PSNR. It is validated that DNN-based codecs perform very well on MS-SSIM in [33]. Our proposed method can be applied to other DNN-based compression models and is expected to achieve similar PSNR and MS-SSIM performance as the model it is based on.

As can be seen from Fig. 9, at a similar bit rate, the PSNR and MS-SSIM of the reconstructed image by S-BFM are comparable to that of BFM. It is a similar case of S-BHM against BHM. Compared with JPEG and LSS-BFM, the proposed S-BFM and S-BHM achieve better PSNR and MS-SSIM values at lower bit rates.

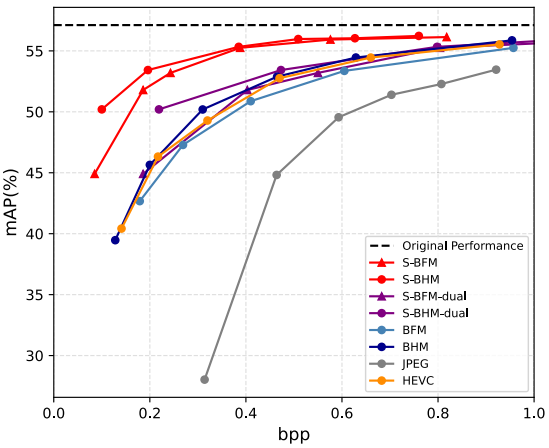

FIGURE 10. Performance of object detection among proposed two-task network and DNN-based image compression methods. Note that the bitrates for S-BFM-dual and S-BHM-dual are calculated by adding the feature stream and the image stream.

D. EVALUATION FOR BOTH IMAGE SIGNAL AND FEATURE COMPRESSION

In the above experimental comparison, we compare the image bitstream and feature bitstream separately, which is due to the characteristics of a large number of edge-cloud communications and a small amount of edge-human communications in the visual IoT. However, there are still rare cases where the feature stream and the image stream need to be transmitted at the same time. In this section, we will discuss the comparison with two DNN-based image

TABLE 6. Params (millions of params), computational cost (billions of FLOPs) and encoding latency (ms) of comparsions among the proposed slimmable compression models and other compared methods. The values are calculated for a 512 × 512 input image on a TITAN RTX GPU.

	Parameters			FLOPs			Latency		
	Image reconstruction	Object detection	Image reconstruction and object detection	Image reconstruction	Object detection	Image reconstruction and object detection	Image reconstruction	Object detection	Image reconstruction and object detection
Ba-BFM	2.781M	1.239M	4.020M	6.794G	3.073G	9.867G	2.50	2.29	4.79
LSS-BFM	2.781M	2.781M	2.781M	6.794G	6.794G	6.794G	2.50	2.50	5.00
S-BFM	2.781M	1.239M	2.781M	6.794G	3.073G	9.867G	2.37	2.01	4.38

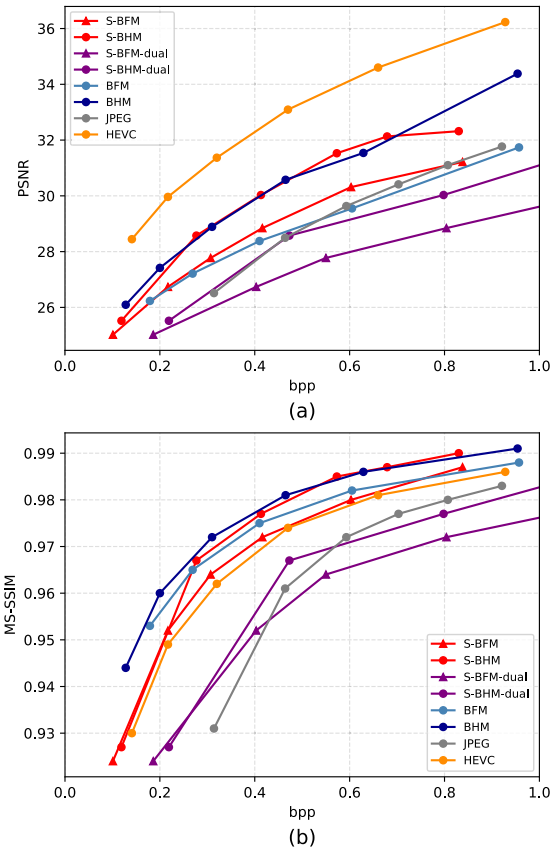

FIGURE 11. Comparison of image reconstruction performance among proposed network and DNN-based image compression methods.

compression methods, i.e., BFM [5], BHM [5], in the case of transmitting the feature stream and image stream simultaneously. We take the network for two vision tasks as an example. For DNN-based image compression methods, the test images were first resized to 512 × 512, after which the rebuilt images were fed into the pre-trained YOLOv3 network.

When the image bitstream and feature bitstream are transmitted concurrently, the rate-distortion curves of the suggested two-task network and the learning-based image method are shown in Figs. 10 and 11. The purple lines are the rate-performance curve after adding the code rates of the characteristic bit stream and the image bit stream of the proposed scheme. For object detection, the S-BFM network outperforms the BFM network marginally. The same goes for the S-BHM network, which is higher than BHM. But for image reconstruction, the image quality is much worse than the other learning-based image compression methods at the same bit rate. Although the proposed scheme does not perform well when the image stream and feature stream are transmitted together, the proposed method only needs to deploy half of the inference network at the decoding end, which reduces the computational burden. On the other hand, the image reconstruction branch of the proposed scheme can do the same machine vision task analysis as the neural network-based image algorithm, that is, connect the whole inference network after reconstructing the image. Therefore, we provide an alternative solution for machine vision analysis.

E. COMPUTATIONAL COMPLEXITY

We assessed the efficiency of the proposed slimmable encoder in terms of parameters (in MB), computational cost in floating point operations (FLOPs) and encoding latency (in ms). These values are calculated for 512 × 512 input images, where the latency is the average of the average latency of the six different bitrate models on the COCOval2014 dataset. And encoding latency is calculated on a Titan RTX GPU, excluding data loading, writing and arithmetic coding. We compare the Ba-BFM, Ba-BHM, and LSS-BFM methods, whose compression processes differ only in the encoding process. Since there is no benefit in latency, CTA schemes like BFM are not used as comparison methods because they require a full inference network, which increases the inference time for machine vision applications beyond our proposed solution.

The resource-saving of the proposed network is displayed in Table 6. Because S-BFM's and S-BHM's encoder structures are identical, their theoretically corresponding computational complexity is also the same. Here, we use S-BFM as an illustration. When only feature streams are transmitted, the encoder width of the proposed S-BFM is reduced. Compared to the LSS-BFM model with the original size encoder, the FLOPs reduction on the object detection task is calculated to be around 55%. The reduction also results in lower latency during encoding by around 0.5 ms. This greatly reduces the computational burden on the encoding side and maximizes resource utilization in the IoT environment where machine-to-machine communication is common. When the image stream and feature stream are transmitted at the same time, the proposed S-BFM saves about 30.8% of the parameters compared to Ba-BFM. Compared with LSS-BFM, the proposed network increases the parameters and computational cost of one sub-encoder but with lower latency

and better image reconstruction quality with comparable machine vision analysis performance.

VI. CONCLUSION

The collaborative compression of human and machine vision is one solution to meet the human and machine vision needs of IoT visual communication. Existing VCM solutions for the IoT, though, are constrained in their practical implementation due to their complexity or inefficiency. In this paper, we introduce a multi-task compression framework for human and machine vision based on a slimmable encoder. The slimmable encoder can be reduced to a smaller sub-encoder by adjusting its width to serve various compression tasks. Moreover, feature transformation networks were introduced as decoders, which map the latent representation to the intermediate features of machine vision inference networks. The proposed framework shows better performance and is more friendly to edge devices due to the lightweight encoder. At the same time, the privacy of users is somewhat safeguarded because only characteristic information is transmitted.

Despite the promising results of our framework, there are still some limitations and challenges that need to be addressed in future work. For example, our framework does not explicitly formulate a mathematical optimization problem. It would be interesting to explore optimization strategies for the model. In our future work, the proposed method can be generalized to other types of image and machine vision tasks, such as segmentation, human pose detection, etc. In addition, the impact of different inference networks on the performance of machine vision tasks can be explored in the future.

REFERENCES

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

[2] W. Yang, H. Huang, Y. Hu, L.-Y. Duan, and J. Liu, "Video coding for machine: Compact visual representation compression for intelligent collaborative analytics," 2021, *arXiv:2110.09241*.

[3] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30–44, 1991.

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

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

[6] D. Minnen, J. Ballé, and G. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–10.

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

[8] L.-Y. Duan, V. Chandrasekhar, J. Chen, J. Lin, Z. Wang, T. Huang, B. Girod, and W. Gao, "Overview of the MPEG-CVDS standard," *IEEE Trans. Image Process.*, vol. 25, no. 1, pp. 179–194, Jan. 2016.

[9] 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 MultiMedia*, vol. 26, no. 2, pp. 44–54, Apr./Jun. 2019.

[10] B. Du, Y. Duan, H. Zhang, X. Tao, Y. Wu, and C. Ru, "Collaborative image compression and classification with multi-task learning for visual Internet of Things," *Chin. J. Aeronaut.*, vol. 35, no. 5, pp. 390–399, May 2022.

[11] J. Liu, H. Sun, and J. Katto, "Learning in compressed domain for faster machine vision tasks," in *Proc. Int. Conf. Vis. Commun. Image Process. (VCIP)*, Dec. 2021, pp. 1–5.

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

[13] R. Torfason, F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Towards image understanding from deep compression without decoding," 2018, *arXiv:1803.06131*.

[14] H. Choi and I. V. Bajic, "Latent-space scalability for multi-task collaborative intelligence," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2021, pp. 3562–3566.

[15] S. Yang, Y. Hu, W. Yang, L.-Y. Duan, and J. Liu, "Towards coding for human and machine vision: Scalable face image coding," *IEEE Trans. Multimedia*, vol. 23, pp. 2957–2971, 2021.

[16] H. Tu, L. Li, W. Zhou, and H. Li, "Semantic scalable image compression with cross-layer priors," in *Proc. 29th ACM Int. Conf. Multimedia*, Oct. 2021, pp. 4044–4052.

[17] F. Yang, L. Herranz, Y. Cheng, and M. G. Mozerov, "Slimmable compressive autoencoders for practical neural image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 4998–5007.

[18] J. Yu, L. Yang, N. Xu, J. Yang, and T. S. Huang, "Slimmable neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–12.

[19] I. V. Bajic and L. W. T. Yonghong, "Collaborative intelligence: Challenges and opportunities," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Jun. 2021, pp. 8493–8497.

[20] Z. Chen, K. Fan, S. Wang, L. Duan, W. Lin, and A. C. Kot, "Toward intelligent sensing: Intermediate deep feature compression," *IEEE Trans. Image Process.*, vol. 29, pp. 2230–2243, 2020.

[21] Z. Zhang, M. Wang, M. Ma, J. Li, and X. Fan, "MSFC: Deep feature compression in multi-task network," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2021, pp. 1–6.

[22] S. R. Alvar and I. V. Bajic, "Multi-task learning with compressible features for collaborative intelligence," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 1705–1709.

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

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

[25] C. Gao, D. Liu, L. Li, and F. Wu, "Towards task-generic image compression: A study of semantics-oriented metrics," *IEEE Trans. Multimedia*, vol. 25, pp. 721–735, 2023.

[26] S. Wang, Z. Wang, S. Wang, and Y. Ye, "End-to-end compression towards machine vision: Network architecture design and optimization," *IEEE Open J. Circuits Syst.*, vol. 2, pp. 675–685, 2021.

[27] L. D. Chamain, F. Racape, J. Begaint, A. Pushparaja, and S. Feltman, "End-to-end optimized image compression for machines, a study," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2021, pp. 163–172.

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

[29] L. D. Chamain, F. Racapé, J. Bégaint, A. Pushparaja, and S. Feltman, "End-to-end optimized image compression for multiple machine tasks," 2021, *arXiv:2103.04178*.

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

[31] S. Wang, S. Wang, W. Yang, and X. Zhang, "Towards analysis-friendly face representation with scalable feature and texture compression," *IEEE Trans. Multimedia*, vol. 24, pp. 3169–3181, 2021.

[32] S. Wang, S. Wang, X. Zhang, S. Wang, S. Ma, and W. Gao, "Scalable facial image compression with deep feature reconstruction," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2691–2695.

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

[34] N. Yan, D. Liu, H. Li, and F. Wu, "Semantically scalable image coding with compression of feature maps," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3114–3118.

[35] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*.
[36] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2014.
[37] J. Bégaint, F. Racapé, S. Feltman, and A. Pushparaja, "CompressAI: A PyTorch library and evaluation platform for end-to-end compression research," 2020, *arXiv:2011.03029*.

JIANGZHONG CAO received the Ph.D. degree in communication and information system from the School of Information Science and Technology, Sun Yat-sen University, China, in 2013. He is currently an Associate Professor with the School of Information Engineering, Guangdong University of Technology, Guangzhou, China. His research interests include computer vision, pattern recognition, and deep learning.

XIMEI YAO received the B.S. degree in communication engineering from Guangzhou Maritime University, Guangzhou, China, in 2020. She is currently pursuing the M.S. degree with the School of Information Engineering, Guangdong University of Technology, Guangzhou. Her research interests include image coding and machine learning.

HUAN ZHANG received the B.S. degree from the Civil Aviation University of China, Tianjin, China, in 2010, the M.S. degree from Tsinghua University, Beijing, China, in 2013, and the Ph.D. degree from the University of Chinese Academy of Sciences, in 2021. She is currently with the School of Information Engineering, Guangdong University of Technology, Guangzhou, China. Her research interests include image restoration, 3-D image/video quality assessment, and learned image compression.

JIAN JIN (Member, IEEE) received the Ph.D. degree in signal and information processing from the Institute of Information Science, Beijing Jiaotong University, China, in 2019. He was a joint Ph.D. Student with Simon Fraser University, Canada, from 2016 to 2018. He is currently a Research Fellow with the Alibaba-NTU Singapore Joint Research Institute, Nanyang Technological University, Singapore. His research interests include visual perceptual modeling, image/video/feature compression, and visual quality assessment. He received the Excellent Ph.D. Thesis Award from the Chinese Institute of Electronics (CIE), in 2019.

YUN ZHANG (Senior Member, IEEE) received the B.S. and M.S. degrees in electrical engineering from Ningbo University, Ningbo, China, in 2004 and 2007, respectively, and the Ph.D. degree in computer science from the Institute of Computing Technology (ICT), Chinese Academy of Sciences (CAS), Beijing, China, in 2010. From 2009 to 2014, he was a Postdoctoral Researcher with the Department of Computer Science, City University of Hong Kong, Kowloon, Hong Kong. Since 2010, he has been an Assistant Professor, an Associate Professor, and a Full Professor with the Shenzhen Institutes of Advanced Technology (SIAT), CAS. In 2022, he joined Sun Yat-sen University as a Professor. His research interests include video compression, 3-D video processing, and visual perception.

BINGO WING-KUEN LING (Senior Member, IEEE) received the B.Eng. degree in electronic and computer engineering and the M.Phil. degree in electronic engineering from The Hong Kong University of Science and Technology, in 1997 and 2000, respectively, and the Ph.D. degree in electronic and information engineering from Hong Kong Polytechnic University, in 2003. His research interests include time frequency analysis, optimization theory, artificial intelligence, human signal processing, and multimedia signal processing. He is very active in various professional societies, such as serving in various technical committees in the IEEE Circuits and Systems Society and the IEEE Industrial Electronics Society. He has also organized many IEEE international conferences, such as the ICASSP, in 2019, the ICCE, USA, in 2016, the ICCE, China, in 2016, the ICCE, China, in 2015, and the ICCE, China, in 2014. He also served as an Associate Editor for various international journals, such as the *Journal of Franklin Institute*, the *International Journal of Bifurcation and Chaos*, the *IET Signal Processing*, the *Circuits, Systems and Signal Processing*, the *Frontiers in Signal Processing*, the *Measurement*, the *Measurement: Sensors*, and the *Journal of Industrial and Management Optimization*.