International Journal of Computer Vision (2021) 129:2605–2621
https://doi.org/10.1007/s11263-021-01491-7

Semantics-to-Signal Scalable Image Compression with Learned Revertible Representations

Kang Liu[1] · Dong Liu[1] · Li Li[1] · Ning Yan[1] · Houqiang Li[1]

Received: 20 December 2020 / Accepted: 9 June 2021 / Published online: 22 June 2021

Abstract
Image/video compression and communication need to serve both human vision and machine vision. To address this need, we propose a scalable image compression solution. We assume that machine vision needs less information that is related to semantics, whereas human vision needs more information that is to reconstruct signal. We then propose semantics-to-signal scalable compression, where partial bitstream is decodeable for machine vision and the entire bitstream is decodeable for human vision. Our method is inspired by the scalable image coding standard, JPEG2000, and similarly adopts subband-wise representations. We first design a trainable and revertible transform based on the lifting structure, which converts an image into a pyramid of multiple subbands; the transform is trained to make the partial representations useful for multiple machine vision tasks. We then design an end-to-end optimized encoding/decoding network for compressing the multiple subbands, to jointly optimize compression ratio, semantic analysis accuracy, and signal reconstruction quality. We experiment with two datasets: CUB200-2011 and FGVC-Aircraft, taking coarse-to-fine image classification tasks as an example. Experimental results demonstrate that our proposed method achieves semantics-to-signal scalable compression, and outperforms JPEG2000 in compression efficiency. The proposed method sheds light on a generic approach for image/video coding for human and machines.

Keywords Deep learning · Image compression · Lifting structure · Machine vision · Scalable coding

1 Introduction

Image/video contains rich semantic information, being one of the main information sources for human. With the explosive growth of image/video data, it has become impossible to fully rely on limited manpower to understand massive images. Instead, the development of machine vision algorithms, especially with the help of deep learning technologies, has greatly improved the efficiency of machine understanding of images. Nonetheless, machine vision could not completely replace human observation, so human–machine collaborative judgment will last for a while. Note that human can directly view images, but machine vision generally needs to convert pixels to compact semantic representations, namely features. In particular, various machine vision tasks generally require different features.

In order to reduce the cost of storage and transmission, images in typical scenarios are compressed with information loss. Traditional image compression algorithms do not consider feature fidelity, so the compression artifacts under low bit rates seriously affect the semantic analysis accuracy (Poyser et al. 2020; Dejean-Servières et al. 2017; Dodge and Karam 2016). A feasible solution is to obtain and compress compact feature representations extracted from the original image. Note that it is difficult to reconstruct the image only based on the features. For human–machine collabora-

Communicated by Dong Xu.

✉ Dong Liu
dongeliu@ustc.edu.cn

✉ Houqiang Li
lihq@ustc.edu.cn

Kang Liu
kangliu@mail.ustc.edu.cn

Li Li
lil1@ustc.edu.cn

Ning Yan
nyan@mail.ustc.edu.cn

[1] CAS Key Laboratory of Technology in Geo-Spatial Information Processing and Application System, University of Science and Technology of China, Hefei 230027, China

tive judgment, it appears necessary to compress and transmit both the features and the image itself simultaneously.

Scalable image compression is an promising approach for jointly compressing the image and features. Scalable compression means that the compressed bitstream can be partially decoded to obtain meaningful output. Note that the features are compact representations of the image and the information contained in the features is a subset of the image information. Accordingly, the base layer can be the feature bitstream, which contains less information of the image about its semantics. The enhancement layer may contain richer semantics-related information, and/or the information that is used to reconstruct the signal. In addition, there is redundancy between the image and its features, so the enhancement layer may refer to the base layer to improve the compression efficiency.

Some studies (Wang et al. 2019; Hu et al. 2020) propose scalable image compression methods based on feature extraction and image generation to fulfill the need of human–machine collaborative judgment. Feature extraction is a process of information contraction (Simonyan and Zisserman 2014; Zhao et al. 2019; Latif et al. 2019), during which a large amount of task-independent information related to the original image is lost, resulting in the lack of clear correspondence between high-level features and low-level pixels, namely *semantic gap* (Kwaśnicka and Jain 2018). The semantic gap makes it difficult to predict an image based on its sparse and compact features, and further affects the flexibility of increasing semantics-related information through partial decoding of the scalable bitstream. In addition, the extracted features may not be compact enough (Zhao et al. 2020; Ruder 2017; Baxter 1997), which incurs unnecessary bitstream cost.

In this paper, we target at the image compression for human–machine collaborative judgment from the perspective of jointly considering feature extraction, feature compression, and image compression. Firstly, to bridge the semantic gap between image and features, inspired by the scalable image coding standard, JPEG2000 (Christopoulos et al. 2000), we design a learned revertible transform, namely a hierarchical signal representation method, as illustrated in Fig. 1. The transform is trainable so that semantic tasks can be well performed by intercepting partial features. The information contained in the partial features is a subset of image information, which enhances the interpretability of the scalable structure. Secondly, we design an end-to-end encoding/decoding network to achieve layered compression of the features. Our specific contributions are summarized as follows.

– First, we propose a task-driven learned revertible transform that converts an image into compact features and achieves hierarchical representations of image information. With increasing features, semantics-related information contained in the features is gradually enriched.
– Second, we design a layered compression network to compresses the multiple features into a scalable bitstream. We use the end-to-end optimization strategy to achieve the joint optimization of semantic accuracy, signal fidelity, and compression ratio.
– Third, we use coarse-to-fine image classification and image reconstruction as a motivating example to conduct experiments, and verify the effectiveness of the proposed semantics-to-signal scalable image compression.

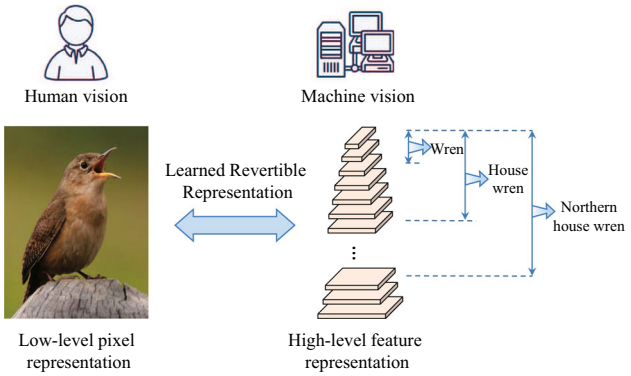


Fig. 1 Conceptual illustration of the proposed semantics-to-signal scalability with learned revertible representations. Image signal, i.e. pixels, is transformed into a set of features. The transform is revertible in the sense that the image can be perfectly reconstructed using all the features. Using partial features we can perform semantic analysis tasks; using more features, we have more semantic information. In this figure, we show a coarse-to-fine image classification task: with more features we perform finer-grained classification, from "Wren", "House wren", to "Northern house wren"

The remainder of this paper is organized as follows. Related work is presented in Sect. 2. We elaborate on the hierarchical representation of the proposed semantics-to-signal scalable bitstream in Sect. 3. In Sect. 4, we introduce the proposed feature representation network. Section 5 demonstrates the proposed layered compression network for the learned representations. In Sect. 6, we show the experimental results about semantic analysis accuracy and compression efficiency. Finally, Sect. 7 concludes the paper.

2 Related Work

Image and features need to be compressed simultaneously for human–machine collaborative judgment. In this section, we introduce in detail the related work from two perspectives: compression and image representation.

2.1 Image and Feature Compression

Studies in Johnston et al. (2018), Dejean-Servières et al. (2017), Dodge and Karam (2016) have shown that compression artifacts have an impact on machine vision tasks. Taking classification as an example, compression has little impact on the classification accuracy at high bit rates, but at low bit rates, compression will seriously lower the classification accuracy.

To make images serve both human vision and machine vision, signal distortion and semantic analysis accuracy need to be considered in the compression. Ma et al. conduct a systematic review, analyze the joint compression of image and features (Ma et al. 2018), and explain the advantages of the joint image-feature compression for reconstruction quality and analysis accuracy. Wang et al. propose a scalable image coding framework for face recognition task (Wang et al. 2019), where base layer and enhancement layer are used to represent face features and signal residuals, respectively. The framework uses traditional coding technologies, such as quantization and entropy coding, on the face features, and uses a network-based compression scheme for the signal residuals. Its compression performance surpasses JPEG and JPEG2000 while maintaining semantic analysis accuracy. Hu et al. (2020) extend the strategy to face keypoint detection task, and further unify the entire compression scheme with a deep neural network. Yan et al. propose an image compression method based on semantic scalability in Yan et al. (2020). The multi-layer features of the deep network are compressed into a scalable bitstream, which can serve multi-grained classification tasks, verifying the advantage of joint compression for significant bits saving. Besides, some researchers (Zhang et al. 2016; Duan et al. 2020; Xia et al. 2020) extend the concept of scalability to videos and verify the effectiveness of joint video-feature compression.

In the recent years, end-to-end optimized image compression based on deep neural networks has demonstrated more flexible and efficient image compression capabilities. Toderici et al. propose the first end-to-end image coding method based on recurrent neural network (RNN) (Toderici et al. 2015), and achieve scalable coding by iteratively invoking the RNN-based encoder to compress the image or residual. Ballé et al. propose the first end-to-end image coding method based on convolutional neural network (CNN) (Ballé et al. 2016). After that, hyper-prior model (Ballé et al. 2018) and autoregressive model (Minnen et al. 2018; Lee et al. 2018) are introduced for more efficient entropy coding. By using the non-local attention module (Li et al. 2020; Zhou et al. 2019; Chen et al. 2019), the compression efficiency is further improved. Nowadays, CNN-based end-to-end coding methods outperform the state-of-the-art non-deep image coding scheme, Better Portable Graphics (BPG),[1] significantly. Our encoding/decoding network is also based on CNN and adopts the hyper-prior model as well as the non-local attention module. The existing end-to-end compression methods, similar to non-deep methods, focus on improving the objective or subjective quality of reconstructed images, but ignore the fidelity of semantic information. Torfason et al. propose to use the same bitstream for machine vision and image reconstruction (Torfason et al. 2018). Intuitively, machine vision tasks (e.g. classification, detection, recognition) usually require less information quantity than image reconstruction does. Using the scheme of Torfason et al. (2018), the number of bits suitable for machine vision may be too few to reconstruct visually pleasing image, and the number of bits suitable for image reconstruction may be too redundant for machine vision. Different from their work, our designed bitstream is scalable; in our bitstream, the bits used for machine vision occupy only a small fraction (e.g. less than 10%, see Sect. 6.3.1) of the entire bitstream.

[1] https://bellard.org/bpg/.

2.2 Image Representation via Transforms

Signal representation is an important approach for image analysis and processing. In image compression, discrete cosine transform (DCT) (Akansu and Liu 1991) and discrete wavelet transform (DWT) (Mallat 1989) are the most commonly used signal representation methods. DCT linearly maps the signal from the spatial domain into the frequency domain while keeping the resolution unchanged. DWT uses orthogonal basis functions to decompose the original signal into multi-resolution coefficients with a pyramid structure. Further, nonlinear wavelets (Goutsias and Heijmans 2000; Heijmans and Goutsias 2000) are proposed based on the lifting structure (Sweldens 1998), which brings the advantages of perfect reconstruction and non-redundant representation. However, DCT and DWT directly decompose low-frequency and high-frequency components in the signal from the perspective of energy distribution (Akansu et al. 2001). These relatively low-level coefficients are difficult to directly serve image understanding.

How to obtain high-level semantics-oriented features from raw image pixels remains an open problem. Feature representation based on deep learning is the current main research trend and has achieved outstanding performance in various machine vision tasks. Several popular neural networks such as VGGNet (Simonyan and Zisserman 2014), ResNet (He et al. 2016), DenseNet (Huang et al. 2017) are believed to have outstanding feature extraction capabilities. Based on the concept of information bottleneck (Tishby et al. 2000), Tishby et al. interpret the learning process of the deep neural network (DNN) as constantly forgetting information about the input while obtaining an efficient expression of the label (Tishby and Zaslavsky 2015; Shwartz-Ziv and Tishby 2017). The process of forgetting is irrevertible, which can easily lead

to a semantic gap (Kwaśnicka and Jain 2018), i.e. the lack of explicit connection between features and image. Semantic gap brings more difficulties to the joint compression of image and features.

Convolutions and nonlinear activations in DNN are the main reasons for irreversibility. It is a possible direction to combine DNN with traditional transforms. In Lo et al. (2003), Ma et al. (2019, 2020), linear and nonlinear neural networks are introduced into the lifting structure, where the studies are focused on signal fidelity instead of preserving semantic information. He et al. (2019) replace partial modules in the ResNet with the lifting structure, which surpasses ResNet in remote sensing classification task, and show the effectiveness of the lifting structure for feature representation. Another wavelet-based network is designed in Rodriguez et al. (2020), where wavelet coefficients obtained by transform are directly used for classification, but the network does not have the revertible characteristic. i-RevNet is a completely revertible nonlinear convolutional neural network proposed in Jacobsen et al. (2018) on the basis of Gomez et al. (2017). In i-RevNet, the input information is always fully retained, and the features based on multi-level mapping are used for object classification and signal reconstruction. However, i-RevNet has the drawback that the feature extraction process and the classifier always use entire image information, which ignores the compactness of features.

3 Hierarchical Representation for Semantics-to-Signal Scalability

For human–machine collaborative judgment, an image needs to be provided to both human and machines. We now introduce an application scenario that motivates the following discussions and experiments. In the considered scenario, there are multiple semantic analysis tasks, for example coarse-grained classification and fine-grained classification. Here, coarse-grained classification has less optional classes, and fine-grained classification has more and finer classes. For example, we may want to classify each image as "dog" or "cat," and if there is a cat, we ask which species the cat belongs to. The dog-cat problem and the dog/cat species problem are coarse-grained and fine-grained, respectively. In addition, image shall be reconstructed for human viewing.

It is intuitive that machine vision needs less, semantics-related information, whereas human vision needs more, signal-reconstructive information. For classification tasks with different granularities, the information required for coarse-grained classification is intuitively a subset of the information required for fine-grained classification. Meanwhile, features required by machine vision tasks are generally compact representations of an image, that is to say, the information required by machine vision tasks is a subset of image

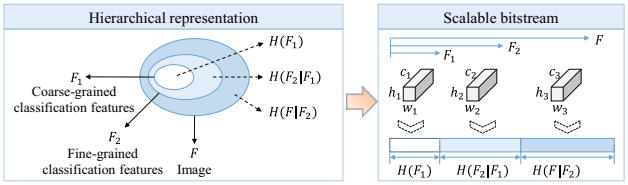


Fig. 2 Theoretical interpretation of the semantics-to-signal scalability. Left: the entropy of image may be represented in a hierarchy. Right: the hierarchical representations can be naturally compressed into a scalable bitstream

information. We now use symbols F_1, F_2, and F to denote features for coarse-grained classification, features for fine-grained classification, and image, respectively, as shown in Fig. 2. In the language of information theory, we know that

$$H\left(F_1|F_2\right) = 0 \tag{1}$$
$$H\left(F_2|F\right) = 0 \tag{2}$$

then we have

$$I\left(F_2; F\right) = H\left(F_2\right) - H\left(F_2|F\right) = H\left(F_2\right) \tag{3}$$
$$I\left(F_1; F_2\right) = H\left(F_1\right) - H\left(F_1|F_2\right) = H\left(F_1\right) \tag{4}$$

and thus

$$\begin{aligned} H(F) &= H\left(F|F_2\right) + I\left(F_2; F\right) \\ &= H\left(F|F_2\right) + H\left(F_2\right) \\ &= H\left(F|F_2\right) + H\left(F_2|F_1\right) + I\left(F_1; F_2\right) \\ &= H\left(F|F_2\right) + H\left(F_2|F_1\right) + H\left(F_1\right) \end{aligned} \tag{5}$$

which is depicted in Fig. 2. In other words, the image information can be decomposed into a series of entropies or conditional entropies of the features or the image. The hierarchical structure of entropy naturally leads to a scalable bitstream, where the base layer is corresponding to $H(F_1)$, and the enhancement layers correspond to $H(F_2|F_1)$ and $H(F|F_2)$, respectively. It also implies that the enhancement layers may be compressed by using prediction from the previous layer(s).

The hierarchical structure of entropy inspires us that features may also have a hierarchical characteristic. Motivated by JPEG2000, we design a revertible feature representation method, which distributes the image information into the feature space without any information loss and achieves the compact representations for machine vision tasks by constraining the amount of information contained in the features. By gradually adding features, the semantic information can be continuously augmented, then the hierarchical representations have the characteristic of semantics-to-signal scalability. Note that the existing feature extraction methods like Simonyan and Zisserman (2014), He et al. (2016) only

consider whether the feature is useful to express semantic information, but do not consider whether the feature is able to reconstruct image; the existing neural image compression methods like Ballé et al. (2018), Minnen et al. (2018), Chen et al. (2019) only consider whether the coded representation (feature) is useful to reconstruct image, but do not consider whether the feature carries important semantic information. Different from them, we consider signal reconstruction and semantic analysis simultaneously to obtain compact features.

4 Proposed Learned Revertible Representation

In this section, we propose a Lifting-based Feature Representation Network (LFRNet) to convert an image into multiple features. The conversion is revertible. Using LFRNet, we achieve semantics-to-signal scalability through the hierarchical representation of the image information.

4.1 Lifting-Based Feature Representation Network

4.1.1 Overview

The proposed LFRNet is a *fully convolutional network* based on the lifting structure (Sweldens 1998). The network structure is shown in Fig. 3a. Specifically, input to the network is the image I, or rigorously speaking, the pixels. LFRNet converts I into feature representations with a hierarchical structure. The conversion can be formulated as

$$\left\{F_K^m, F_K^d, F_{K-1}^d, \ldots, F_2^d, F_1^d\right\} = \overrightarrow{\mathbb{F}}(I|\Theta) \tag{6}$$

where $\overrightarrow{\mathbb{F}}$ stands for the forward-transform of LFRNet, Θ is the set of trainable parameters of LFRNet, and K is the order of transform. $K = 5$ in the experiments by default. The features, $\{F_K^m, F_K^d, F_{K-1}^d, \ldots, F_2^d, F_1^d\}$, are derived from multiple Revertible Feature Representation Units (RFRUs).

RFRU is the basic unit of LFRNet as shown in Fig. 3b. It is designed based on the lifting structure and CNN. RFRU includes three basic operations: *Split*, *Predict*, and *Update*. The input of $RFRU_F^k$ ($k \in [1, K]$), the kth forward-transform unit of LFRNet, is F_{k-1}^m. The Split operation decomposes F_{k-1}^m into the main branch subband $\tilde{F}_k^m$ and the dual branch subband $\tilde{F}_k^d$. In particular, the splitting process is revertible:

$$\left(\tilde{F}_k^m, \tilde{F}_k^d\right) = \text{Split}\left(F_{k-1}^m\right) \tag{7}$$

Then, we use $\tilde{F}_k^m$ to predict $\tilde{F}_k^d$, and use the prediction residual F_k^d to update F_k^m:

$$\begin{cases} F_k^d = \tilde{F}_k^d - \text{Predict}\left(\tilde{F}_k^m\right) \\ F_k^m = \tilde{F}_k^m + \text{Update}\left(F_k^d\right) \end{cases} \tag{8}$$

Equation (8) is revertible. Its inverse process is:

$$\begin{cases} \tilde{F}_k^m = F_k^m - \text{Update}\left(F_k^d\right) \\ \tilde{F}_k^d = F_k^d + \text{Predict}\left(\tilde{F}_k^m\right) \end{cases} \tag{9}$$

In other words, F_{k-1}^m can be perfectly reconstructed when F_k^m and F_k^d are known.

The revertibility of RFRU implies that LFRNet is revertible, provided that the parameters used for the forward-transform are also used for the inverse-transform. Here, we do not consider the information loss due to numeric computations. When all the features are known, the input image I can be reconstructed by the inverse operation $\overleftarrow{\mathbb{F}}$:

$$I = \overleftarrow{\mathbb{F}}\left(F_K^m, F_K^d, F_{K-1}^d, \ldots, F_2^d, F_1^d|\Theta\right). \tag{10}$$

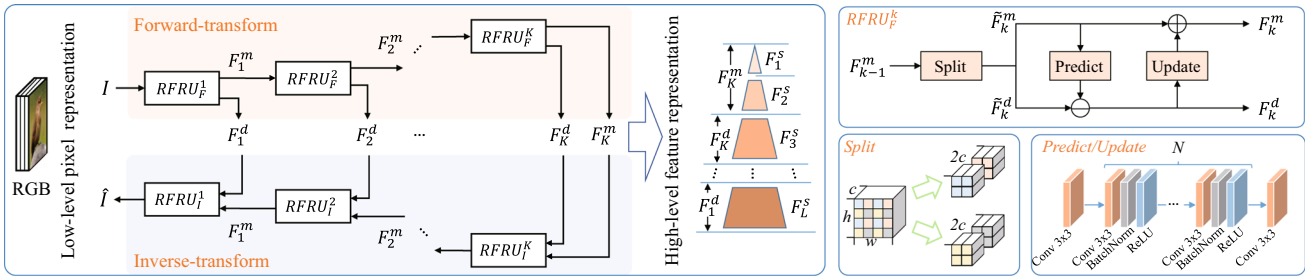
4.1.2 Structure of RFRU

Note that RFRU follows the general lifting structure, which was proposed initially for efficient implementation of the wavelet transform (Sweldens 1998). While the lifting structure remains the same, our RFRU is different from the traditional wavelets, because in RFRU the Predict and Update operations are implemented by trained networks.

Specifically, *Predict* and *Update* in (8) use the same network structure, but have different parameters. Besides, different RFRUs in LFRNet do not share parameters. Each Predict/Update network has three parts: redundant representation, feature extraction, and feature shrinkage. The redundant representation is using a convolutional layer with kernel size equal to 3×3 to expand in the channel dimension to 8 times. The feature extraction is using N repetitive units, each of which has a convolution, a batch normalization, and a Rectified Linear Unit (ReLU). N is the order of nonlinearity, and is set to $\{2, 2, 3, 3, 3\}$ for the $K = 5$ RFRUs. The feature shrinkage is using a convolutional layer with kernel size equal to 3×3 to shrink in the channel dimension back to that of the input.

4.1.3 Structure of Hierarchical Representations

LFRNet achieves the mapping from an image I to a set of features $\{F_K^m, F_K^d, F_{K-1}^d, \ldots, F_2^d, F_1^d\}$ through the cascade of RFRUs. Next, these features are bound with machine



(a) LFRNet **(b)** RFRU

Fig. 3 The proposed lifting-based feature representation network (LFRNet) and revertible feature representation unit (RFRU)

vision tasks. Usually machine vision tasks need a compact set of representations for the sake of computational simplicity. Thus, we may further decompose the features into subsets and choose for example a subset of features for a task. For example, we may decompose at the subband level, or may decompose at the channel level.

In LFRNet, subband decomposition is implanted into the cascade of RFRUs. Specifically, while one RFRU outputs two subbands: main branch and dual branch, the following RFRU deals with the main branch only. Thus, the deeper RFRUs process the less data, with the hope of extracting compact features. It is also worth noting that the degree of nonlinearity also increases with deeper RFRUs.

In addition, we may also select partial channels from a subband as a subset. Because the proposed RFRU reduces spatial resolution but increases number of channels, the main-branch and dual-branch features have more and more channels. Possibly, for a machine vision task, we may need only a subset of a subband. Channel decomposition is a convenient way to achieve this.

In this paper we consider image classification task, which benefits from relatively deeper CNN that has more nonlinearity. Thus, we use the last main-branch feature F_K^m to perform classification. For coarse-grained classification, we perform channel decomposition on F_K^m to obtain a subset. This is illustrated in Fig. 3a, where we decompose F_K^m into F_1^s and F_2^s. Here we use a notation slightly different from that in Sect. 3: F_1^s is equivalent to F_1 in Sect. 3, and refers to the features for coarse-grained classification. $\{F_1^s, F_2^s\}$ is equivalent to F_2 in Sect. 3, and refer to the features for fine-grained classification.

We would like to remark that the proposed LFRNet has the following advantages. First, the image information is redistributed in the feature space without any information loss. Second, by using a part/all of the features, compact/complete representations are obtained, respectively. Third, compared with i-RevNet that always performs feature extraction upon entire image information (Jacobsen et al. 2018), the main branches of LFRNet discard information gradually, which makes the feature extraction more interpretable and reduces number of network parameters and computational cost. Fourth, the inverse-transform directly uses the parameters of the forward-transform, avoiding additional modeling and training.

4.2 Task-Oriented Optimization

The proposed LFRNet can be optimized for specific machine vision tasks. Note that LFRNet produces a series of features, which can be input to different networks to fulfill various machine vision tasks. LFRNet and the task-specific networks may be jointly optimized to ensure the usability of the features. Here, we first give a general formulation, and then present the specific formulation for the scenario of coarse-to-fine image classification.

In general, we may have T tasks, for each task Task_t ($t = 1, \ldots, T$), we assign a subset of features $F_t \subseteq \{F_K^m, F_K^d, F_{K-1}^d, \ldots, F_2^d, F_1^d\}$ to perform the task. Let Θ be the set of trainable parameters of LFRNet, and Θ_t be the set of trainable parameters of the task-specific network, the optimization problem can be defined as

$$\min_{\Theta, \Theta_1, \ldots, \Theta_T} \sum_{t=1}^{T} \lambda_t \mathcal{L}_t \left(\Theta, \Theta_t\right) \tag{11}$$

where λ_t is the weight of the tth task and $\mathcal{L}_t$ measures the task-specific loss.

Specifically, for the considered coarse-to-fine image classification, there are two tasks: coarse-grained classification and fine-grained classification. There are respectively two task-specific networks dedicated to classification. For example, we use three fully-connected layers to build a classification network. The two classification networks are denoted by $\mathbb{N}_c$ and $\mathbb{N}_f$, and their parameters are Θ_c and Θ_f, respectively. In addition, we have mentioned that we use F_1^s for coarse-grained classification and we use $\{F_1^s, F_2^s\}$ for fine-grained classification. Therefore, the specific optimization problem

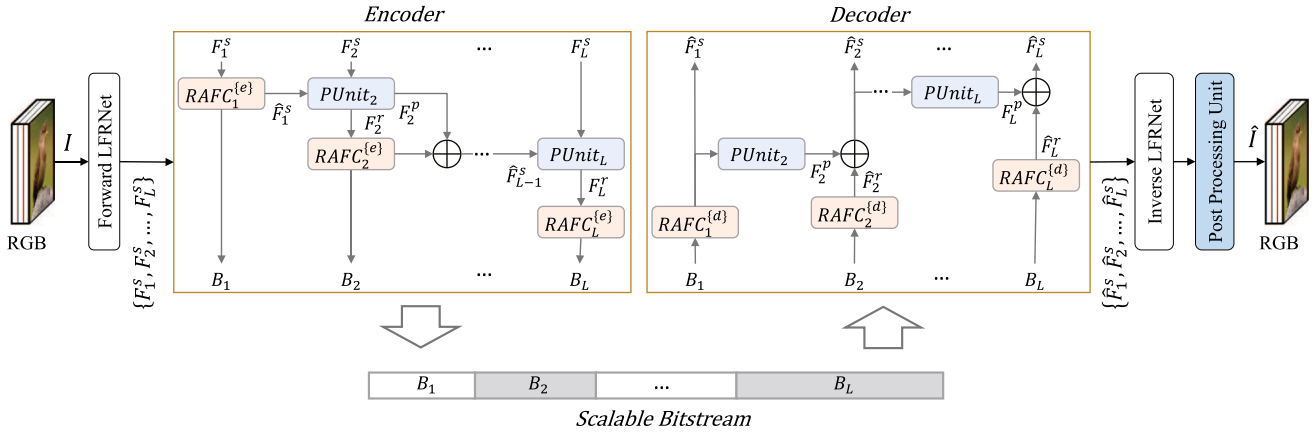


Fig. 4 The proposed layered compression network (LCNet). $RAFC_l$ stands for the resolution adaptive feature compression (RAFC) unit for layer l. The superscripts $\{e\}$ and $\{d\}$ stand for encoding and decoding, respectively. "PUnit" is the inter-layer prediction unit. $\{F_1^s, F_2^s, \ldots, F_L^s\}$ are the features to be compressed. $\{\hat{F}_1^s, \hat{F}_2^s, \ldots, \hat{F}_L^s\}$ are the reconstructed features after compression. $\{B_1, B_2, \ldots, B_L\}$ form a scalable bitstream

becomes:

$$\begin{aligned}&\min_{\Theta,\Theta_c,\Theta_f} \lambda_c H\left(\mathbb{N}_c\left(F_1^s|\Theta_c\right), L_c\right)\\&+\lambda_f H\left(\mathbb{N}_f\left(\left\{F_1^s, F_2^s\right\}|\Theta_f\right), L_f\right)\end{aligned} \tag{12}$$

where $H(\cdot,\cdot)$ is the cross-entropy loss function, L_c and L_f are the ground-truth labels for coarse-grained and fine-grained classification, respectively. $\lambda_c = \lambda_f = 1$ in the experiments because we assign equal importance to the two tasks.

5 Proposed Layered Compression Network

In this section, we propose a layered compression network to compress the multi-layer features into a scalable bitstream. The compression network is optimized end-to-end to achieve the joint optimization of compression ratio, signal distortion, and semantic analysis accuracy.

5.1 Problem Formulation

Based on the hierarchical representations generated by LFR-Net, we propose a Layered Compression Network (LCNet) to compress all the features. To use a unified notation, hereafter we use F_l^s $(l = 1, \ldots, L)$ to replace the previous symbols $\{F_K^m, F_K^d, F_{K-1}^d, \ldots, F_2^d, F_1^d\}$, as shown in Fig. 3. Note that we have split F_K^m into F_1^s and F_2^s, so $L = K + 2 = 7$ in the experiments. When optimizing LCNet, we assume the parameters of LFRNet had been trained and keep unchanged. So the optimization problem can be defined as

$$\begin{aligned}&\min_{\Omega} \text{Distortion} + \lambda \times \text{Rate}\\&= \lambda_c H\left(\mathbb{N}_c\left(\hat{F}_1^s|\Omega\right), L_c\right) + \lambda_f H\left(\mathbb{N}_f\left(\left\{\hat{F}_1^s|\Omega, \hat{F}_2^s|\Omega\right\}\right), L_f\right)\\&+ \lambda_D \mathcal{L}\left(\overleftarrow{\mathbb{F}}\left(\hat{F}_1^s, \ldots, \hat{F}_L^s|\Omega\right), I\right) + \lambda \sum_{l=1}^{L} R_l(\Omega)\end{aligned} \tag{13}$$

where λ is the Lagrangian multiplier for rate-distortion trade-off, $\hat{F}_l^s$ is the lossily compressed and reconstructed version of F_l^s and is dependent on the parameters Ω, $\mathcal{L}(\cdot,\cdot)$ measures the signal distortion and λ_D is its weight, R_l is the rate of F_l^s and is dependent on the parameters Ω. Note that Θ, Θ_c, Θ_f are omitted in the optimization problem because they all keep unchanged.

The joint optimization of bitrate, signal distortion, and semantic analysis accuracy has twofold benefits. First, it effectively retains the semantic information required by machine vision tasks, thereby ensuring the accuracy of semantics. Second, it can tradeoff between bitrate and signal distortion by more or less compressing features that are irrelevant to machine vision tasks.

5.2 Layered Compression Network

Under the guidance of the optimization problem defined in (13), we construct the layered compression network as shown in Fig. 4. The entire LCNet consists of three parts: Encoder, Decoder, and Post-Processing Unit. During the end-to-end optimization, LFRNet is involved but the parameters of LFR-Net are fixed.

The Encoder in LCNet has multiple resolution-adaptive feature compression (RAFC) units as well as multiple inter-feature prediction units (PUnits). Each RAFC (encoding) unit deals with one feature F_l^s, either compressing the feature (when $l = 1$) or compressing the residual of the feature (when $l \geq 2$) into a part of bitstream B_l. Each PUnit predicts the feature F_l^s (when $l \geq 2$) from the previously reconstructed features $\hat{F}_1^s, \ldots, \hat{F}_{l-1}^s$; the prediction is denoted by F_l^p and the corresponding residual is denoted by $F_l^r = F_l^s - F_l^p$. The compressed and reconstructed residual is denoted by $\hat{F}_l^r$. Then, the reconstructed feature is $\hat{F}_l^s = F_l^p + \hat{F}_l^r$. PUnit effectively reduces the inter-feature redundancy.

The Decoder in LCNet also has multiple RAFC units and multiple PUnits. Each RAFC (decoding) unit decodes the partial bitstream B_l, reconstructs the feature (when $l = 1$) or the residual (when $l \geq 2$). The residual is added with the prediction to obtain the reconstructed feature when $l \geq 2$. Note that the Decoder can decode partial bitstream, which is a nature of scalable coding. For example, if one wants to perform coarse-grained classification, it is sufficient to decode B_1 into $\hat{F}_1^s$, and then use the task-specific network. The entire bitstream is decoded only when we want to reconstruct the image.

Since the features are lossily compressed, the compression artifacts may deteriorate the quality of the reconstructed image, especially at low bit rates. Thus, we use a post-processing unit to repair the reconstructed image for human vision.

5.3 Basic Modules

5.3.1 Resolution-Adaptive Feature Compression Unit

Our resolution-adaptive feature compression (RAFC) unit is a simplified and adapted version of the network of Non-Local Attention optimization and Improved Context modeling-based image compression (NLAIC) (Chen et al. 2019). NLAIC is based on Ballé's pioneering work about hyper-prior model (Ballé et al. 2018). Compared to Ballé's work, NLAIC introduces the non-local attention optimization and the improved context model: the attention mechanism together with the nonlocal operations are used to process multi-layer features adaptively; both hyper-prior model and neighboring reconstructed features are used to improve the efficiency of context modeling. Compared to NLAIC, our RAFC has simplified the network structure and adapted some hyper-parameters for different features. Figure 5 shows the core modules of RAFC.

The compression unit for the lth feature is denoted by $RAFC_l$. $RAFC_l^{\{e\}}$ and $RAFC_l^{\{d\}}$ represent its encoding part and decoding part, respectively. $RAFC_l^{\{e\}}$ is a combination of $\{\mathbb{E}_m, \mathbb{E}_h, \mathbb{D}_h, Q, AE, CM\}$. $RAFC_l^{\{d\}}$ is a combination of

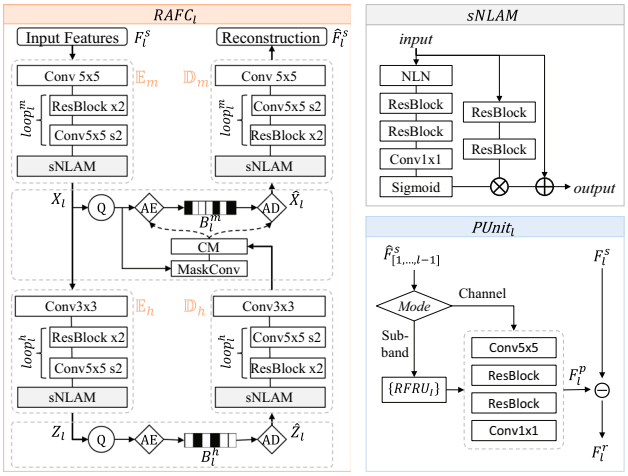


Fig. 5 Left: the proposed resolution adaptive feature compression (RAFC) unit. "Conv5x5 s2" indicates a convolutional layer using a kernel of size 5×5 and a stride of 2. "$loop_l^m$" and "$loop_l^h$" respectively represent the number of stacking the specified module. Right: the proposed prediction unit (PUnit). Each ResBlock contains two convolutional layers and a rectified linear unit (ReLU) inside the two, as well as a residual connection, i.e. $f(x) = \text{Conv}_2(\text{ReLU}(\text{Conv}_1(x))) + x$

$\{\mathbb{D}_m, \mathbb{D}_h, AD, CM\}$. In the encoder, we first use the main transformation $\mathbb{E}_m$ to obtain representations X_l, then we use the secondary transformation $\mathbb{E}_h$ to obtain representations Z_l. Z_l passes quantization (Q) and arithmetic-encoder (AE) to become a bitstream B_l^h. B_l^h is decoded by arithmetic-decoder (AD) to obtain $\hat{Z}_l$, which is then sent to the secondary inverse-transformation $\mathbb{D}_h$. The output of $\mathbb{D}_h$, together with the neighboring reconstructed features that pass a masked convolution (MaskConv), is used by the context modeling (CM), which provides probability models for the AE to encode the quantized version of X_l into another bitstream B_l^m. So far, we obtain the bitstream B_l composed by B_l^m and B_l^h. Note that the rate of B_l, in other words, R_l, can be estimated according to these probability models as calculated in Chen et al. (2019). Finally, we can decode B_l^m to get $\hat{X}_l$, and use the main inverse-transformation $\mathbb{D}_m$ to obtain the reconstruction.

Compared to Chen et al. (2019), our RAFC uses a simplified non-local attention module (sNLAM). Specifically, we remove one residual-connection block (ResBlock) and reduce the number of channels from 192 to 128. This is observed efficient for computation and do not incur much compression performance loss.

In addition, all the features use RAFC units but the features have different resolutions. Accordingly, the RAFC unit for each feature slightly differs from one another, notably in the down-sampling scales of $\mathbb{E}_m$ and $\mathbb{E}_h$. The scales are controlled by $loop_l^m$ and $loop_l^h$. Denote the resolution of F_l^s by (C, S, S) , where C is number of channels and $S = 2^j$ is width/height. We set $loop_l^m$ to $floor(\frac{\log_2(S)-1}{2})$, and set

$loop_l^h$ to 2. In addition, the width/height and number of channels of X_l and Z_l are set to

$$\begin{aligned} S_m &= S \cdot 2^{-floor\left(\frac{\log_2(S)-1}{2}\right)} \\ C_m &= \frac{C \cdot S^2}{4 \cdot (S_m)^2} \\ S_h &= \frac{S_m}{4} \\ C_h &= \frac{C_m}{1.5} \end{aligned} \tag{14}$$

S_m and C_m are for X_l, and S_h and C_h are for Z_l, respectively. Therefore, the size of X_l is 1/4 of the size of the input feature, and the size of Z_l is 1/24 of the size of X_l.

5.3.2 Inter-Feature Prediction Unit

The inter-feature prediction has two modes depending on how the features are decomposed. F_1^s and F_2^s are different channels of the same subband. The following features belong to different subbands. As shown in Fig. 5, PUnit first distinguishes the two cases. If it is channel decomposition, the feature to be predicted and the feature used to predict have the same spatial resolution, so a fully convolutional network is directly used. If it is subband decomposition, the feature used to predict shall be processed by an RFRU so that the spatial resolution is aligned, and then passes a fully convolutional network.

Each PUnit has a lightweight six-layer CNN. Its first convolutional layer uses kernel size 5×5 and 128 channels for redundant representation. The following four layers are organized into two ResBlocks. Each ResBlock has two convolutional layers with kernel size equal to 3×3 and a ReLU between the two, and a residual connection. The last convolutional layer uses kernel size 1×1 and 1 channel to output.

5.3.3 Post-Processing Unit

A post-processing unit is added to filter the entire reconstructed image, so as to reduce compression artifacts and enhance signal reconstruction quality. In particular, the post-processing unit uses the same network structure as the PUnit.

6 Experiments

6.1 Tasks and Datasets

In this paper, we consider coarse-to-fine image classification as the targeted machine vision tasks, where one image can be classified into a coarse category (the number of optional categories is small), or a fine category (the number of optional categories is large). Note that our scheme is to provide a scalable bitstream, which can be partially decoded to obtain features that then serve the classification. For the classification tasks, one may imagine a compression scheme that performs classification at the *encoder* side and transmits only the classification results to the *decoder* side. This scheme is feasible, but may have severe limitations: the transmitted data may be useless if the decoder side wants to perform another classification task with a different category set; the encoder side may not have sufficient computational resource to perform the classification. In view of these limitations, we do not experimentally compare with the imaginary scheme of "transmitting classification results."

There are several datasets that support our designed study. For example, CUB200-2011 (Wah et al. 2011) is a bird image dataset consisting of 11,788 images, where 5994 images are for training and 5794 are for test. All the images are divided into 200 categories. According to ornithology systematics, the 200 categories can be merged into 122 coarse categories or 37 coarser categories. For CUB200-2011, we use coarse-grained to refer to 37-category classification, fine-grained to refer to 200-category classification, and intermediate-grained to refer to 122-category classification. For another example, FGVC-Aircraft (Maji et al. 2013) is an aircraft image dataset consisting of 10,000 images, where 6667 images are for training and 3333 are for test. The images can be divided by manufacturer, family, variant, into 30, 70, 100 categories, respectively. For FGVC-Aircraft, we use coarse-grained, intermediate-grained, and fine-grained to refer to the 30-category, 70-category, and 100-category classification, respectively. All these category labels are available in the dataset, so the classification accuracy can be directly calculated. Note that our proposed LFRNet and LCNet are optimized only for coarse-grained and fine-grained classification, but we will test their generalization ability for intermediate-grained classification.

Since the content of the above two datasets is relatively homogeneous, we use the ILSVRC dataset (Russakovsky et al. 2015) (often known as ImageNet) for pre-training of LFRNet. ILSVRC contains 1000 categories, and each category has about 1000 images.

All the images in the used datasets have the resolution of 256×256, or have been resized to this resolution, to ensure a fair comparison with other methods. Fig. 6 presents some images of the used datasets.

We evaluate the proposed method and compare with the others in both semantic analysis accuracy and compression efficiency. Top-1 accuracy and top-5 accuracy are used to evaluate the classification results. Compression efficiency is evaluated from three indicators: compression ratio (or bitrate in bit-per-pixel, bpp), peak signal-to-noise ratio (PSNR), and multi-scale structural similarity (MS-SSIM).

(a) ILSVRC **(b)** CUB200-2011 **(c)** FGVC-Aircraft

Fig. 6 Exemplar images of the used datasets. ILSVRC (often known as ImageNet) is used for network pre-training. Both CUB200-2011 and FGVC-Aircraft are used for coarse-to-fine image classification

Table 1 Training hyper-parameters

Network	LFRNet	LCNet
Epoch number	128	128
Batch size	256	24
Initial lr	1e−2	1e−5
Step lr	0.1×/32 epochs	0.75×/32 epochs
Weight decay	0.0005	0.0005
Momentum	0.9	0.9

"lr" is short for learning rate

6.2 Experimental Settings

Our implementation is based on PyTorch (Paszke et al. 2017). All the training and test are conducted on a cluster of GTX1080Ti graphics processing units (GPUs). Four GPUs are used for fast training.

In the training stage, we use the stochastic gradient descent algorithm for gradient back-propagation and parameter optimization. The weights in (13) are set to $\lambda_c = \lambda_f = 1$, $\lambda_D = 0.5$, and $\lambda \in \{0.0067, 0.04, 0.2, 2\}$, respectively. Four values are used for λ to achieve different bit rates. Compared with several neural image compression networks such as Ballé et al. (2018), Chen et al. (2019) that use patches for training, our networks are trained with complete images due to the considered machine vision tasks–image classification. Note that our LFRNet and LCNet are separately trained: we train LFRNet first, then fix the parameters of LFRNet and train LCNet. Our LFRNet is pre-trained on ILSVRC and then further trained on either CUB200-2011 or FGVC-Aircraft. LCNet is not pre-trained. Table 1 summarizes the hyper-parameters for network training.

6.3 Performance of Semantics-to-Signal Scalable Compression

Four compression models are trained by setting different values for λ. These models are referring to the same LFRNet model. Table 2 shows the average bit rate of compressed images with different models. On CUB200-2011, the maximal average compression ratio is 471 (0.051 bpp) and the minimal one is 34 (0.711 bpp). On FGVC-Aircraft, the maximal one is 889 (0.027 bpp) and the minimal one is 51 (0.474 bpp).

6.3.1 Semantics-to-Signal Scalability

First, we examine the scalable coding functionality of the proposed method. Figure 7 shows the partial decoding results on CUB200-2011 using the model with $\lambda = 0.0067$. It also displays some reconstructed images of partial decoding, taking one picture in the CUB200-2011 test set as an example. Clearly, with more bits decoded, the classification accuracy and PSNR both increase, and the visual quality of the reconstructed images becomes better. Also, after a certain rate (e.g. 0.011 bpp for coarse-grained), the classification accuracy becomes stable. This rate is called "critical rate" for the machine vision task. Obviously, the critical rate for fine-grained classification ($\sim$ 0.02 bpp) is higher than that for coarse-grained, but it is still far less than the rate needed for image reconstruction. As shown in Fig. 7, to reconstruct visually pleasing image, the bitrate shall be higher than 0.2 bpp. In this example, the bits required for the classification tasks occupy less than 10% of the entire bitstream that provides visually acceptable image reconstruction.

In Fig. 8, we randomly select two images from the CUB200-2011 test set and the FGVC-Aircraft test set, respectively, and display some reconstructed images of partial decoding. When the decoded bit rate is very low, the reconstructed images are hard to recognize, but human may

Table 2 Classification accuracy results

Dataset	Task	Bitrate	Top-1 Accuracy					Top-5 Accuracy				
			JPEG 2000	Ours (Image)	BPG	NLAIC	Ours (Feature)	JPEG 2000	Ours (Image)	BPG	NLAIC	Ours (Feature)
CUB200-2011	Coarse-grained	Orig.	88.9	88.9	88.9	88.9	89.3	98.7	98.7	98.7	98.7	98.4
		0.711	83.3	84.7	85.0	87.0	**87.9**	97.1	97.4	97.8	**98.2**	98.1
		0.343	74.2	77.0	76.0	81.1	**87.9**	93.8	94.7	95.1	94.5	**98.2**
		0.153	57.8	58.5	62.5	63.1	**87.4**	85.8	85.3	88.9	88.2	**98.1**
		0.051	26.6	30.7	40.6	–	**87.6**	62.5	59.6	73.0	–	**98.0**
	Intermediate-grained	Orig.	81.9	81.9	81.9	81.9	80.7	96.4	96.4	96.4	96.4	96.0
		0.711	73.0	76.2	76.5	**79.2**	77.2	92.3	93.9	94.1	**95.1**	94.9
		0.343	61.3	67.2	65.4	71.0	**76.8**	85.6	89.2	88.9	92.1	**94.5**
		0.153	40.9	46.7	48.7	50.2	**76.4**	69.2	74.4	78.2	77.5	**94.7**
		0.051	12.2	20.8	23.9	–	**76.8**	33.5	43.4	50.9	–	**94.5**
	Fine-grained	Orig.	75.8	75.8	75.8	75.8	75.5	94.0	94.0	94.0	94.0	93.3
		0.711	68.6	69.9	70.2	72.4	**72.6**	90.4	91.7	91.4	**92.9**	92.3
		0.343	57.0	61.5	60.2	65.5	**72.4**	83.4	87.1	86.4	89.3	**92.3**
		0.153	36.1	41.8	46.3	44.6	**72.2**	66.5	70.4	75.7	73.5	**92.3**
		0.051	9.9	16.0	22.5	–	**72.1**	29.6	36.4	48.3	–	**92.4**
FGVC-Aircraft	Coarse-grained	Orig.	93.3	93.3	93.3	93.3	92.6	98.5	98.5	98.5	98.5	98.6
		0.474	82.3	83.4	85.5	86.4	**90.9**	95.7	96.1	96.5	96.7	**98.1**
		0.206	58.9	70.2	75.0	79.6	**90.4**	85.2	88.8	92.4	94.0	**98.1**
		0.093	22.4	41.0	49.4	57.2	**90.9**	49.4	68.0	77.4	79.9	**98.0**
		0.027	6.7	10.2	16.1	–	**89.6**	21.2	28.3	38.7	–	**97.5**
	Intermediate-grained	Orig.	89.7	89.7	89.7	89.7	88.4	96.8	96.8	96.8	96.8	96.3
		0.474	75.6	79.6	80.1	81.3	**85.2**	92.6	94.6	94.6	95.0	**95.1**
		0.206	50.2	69.7	70.8	75.9	**85.1**	78.3	89.1	91.4	93.2	**95.0**
		0.093	18.5	46.2	49.7	57.4	**84.3**	43.4	73.8	77.9	81.3	**94.6**
		0.027	5.3	12.0	14.7	–	**81.3**	18.0	32.9	36.9	–	**94.2**
	Fine-grained	Orig.	89.6	89.6	89.6	89.6	89.0	97.0	97.0	97.0	97.0	97.0
		0.474	76.7	80.5	80.7	82.0	**87.1**	92.7	94.1	94.5	94.9	**96.3**
		0.206	55.0	71.2	74.1	78.7	**86.6**	80.5	90.3	92.4	93.7	**95.9**
		0.093	20.4	47.7	56.0	61.5	**86.4**	43.8	73.0	80.8	83.5	**96.0**
		0.027	4.3	11.7	18.3	–	**85.1**	14.8	34.4	41.0	–	**95.5**

"Ours (Feature)" and "Ours (Image)" respectively indicate decoded feature-based and reconstructed image-based classification results. "Orig." indicates using original uncompressed images, whose results are underlined to distinguish

Bold indicates the best accuracy for each bitrate

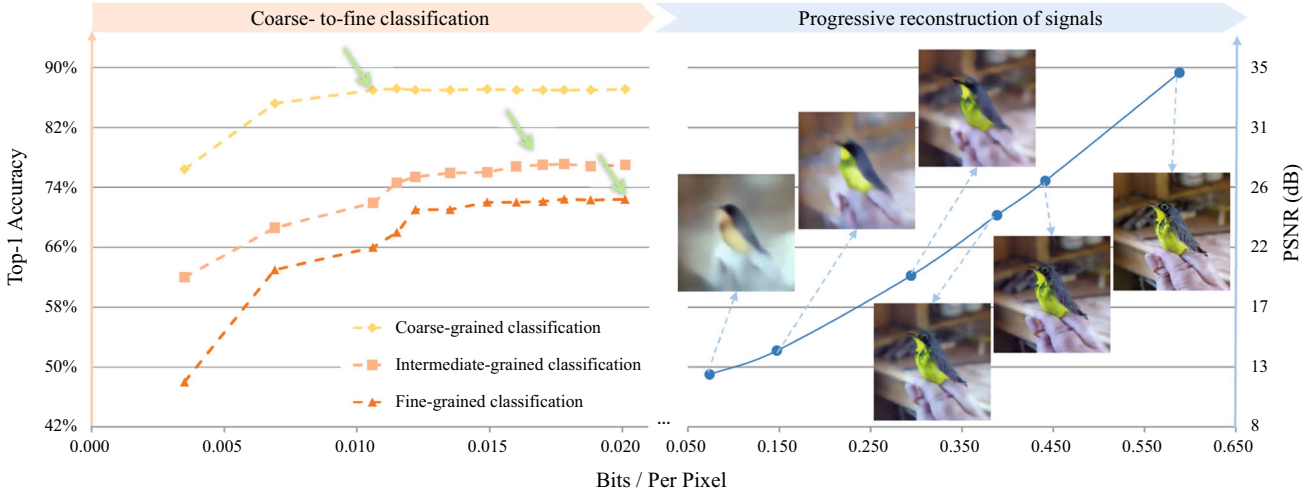


Fig. 7 Quantitative evaluation of the semantics-to-signal scalability ($\lambda = 0.0067$). The horizontal axis represents bitrate (bit-per-pixel, bpp), and the left and right vertical axes represent top-1 classification accuracy and reconstruction quality in PSNR, respectively. The classification accuracy is not displayed in the high bitrate range because it becomes stable after the indicated (by the green arrow) point. The reconstruction quality is not displayed in the (extremely) low bitrate range because it is too low to be useful (Color figure online)

still be able to identify the shape of the object. With more bits decoded, colors, edges, and textures become more and more clear. When all the bits are decoded, reconstructed images appear similar to the original images.

6.3.2 Compression Performance

We choose three image compression methods for comparison. The first is JPEG2000, which is the widely used standard for scalable image compression. The second is BPG, which represents state-of-the-art of non-learned compression. We use the default configuration of BPG, that is to compress with YUV420 format. The third is NLAIC, a CNN-based end-to-end learned image compression method proposed in Chen et al. (2019). We choose NLAIC because our compression network also borrows some ideas from it. For JPEG2000 and BPG, we have adjusted the quantization parameter to achieve similar compression ratios as our method. For NLAIC, we use pre-trained models, so the bit rates are not well aligned to those of the other methods.

First, we show the classification results. Here for CUB200-2011 and FGVC-Aircraft, we respectively train a VGG16 model with uncompressed images, and use the same model on reconstructed images with different methods (JPEG2000, ours, BPG, NLAIC) and different bit rates. These results are summarized in Table 2. In addition, we also report the classification results of our method not using reconstructed images but using *features that are decoded from partial bitstream*. It can be observed that with the decrease of bit rate, the classification accuracy of reconstructed images usually drops significantly. However, our results using features are quite stable across different bit rates. For example on CUB200-2011, for coarse-grained classification, when bitrate is around 0.051 bpp, the top-1 accuracy of JPEG2000 reconstructed images is 26.6%, but our result using features is 87.6%. Note that the classifier for features is a three-layer fully-connected network and it is not stronger than VGG16. Therefore, the results show that compressing and transmitting features for machine vision tasks is a good choice at low bit rates.

Second, we compare the results of different methods on PSNR, MS-SSIM, and bitrate. Figure 9 shows the rate-PSNR/MS-SSIM curves of our method and JPEG2000, BPG, and NLAIC. Our proposed method significantly surpasses JPEG2000 in both PSNR and MS-SSIM. In addition, our method achieves comparable MS-SSIM than BPG and NLAIC at high bit rates, but does not catch up with BPG and NLAIC in PSNR. We believe the result is mainly due to the less compression efficiency of our entropy coding method. JPEG2000 compresses all the subbands simultaneously, and it has dedicated, highly efficient coding tools like zero-tree (Taubman 2000). BPG has an advanced context-adaptive binary arithmetic coding (CABAC) engine for entropy coding (Marpe et al. 2003). NLAIC compresses all the features and optimizes the uniform entropy coder in an end-to-end fashion. In our scheme, the features are compressed one by one using prediction from each to the next. The correlation among non-adjacent features is not fully exploited. As a result, our method performs less well especially in PSNR at high bit rates where entropy coding has big impact. Nonetheless, our method performs well in MS-SSIM at high bit rates;

Fig. 8 Reconstructed images of our method. Numbers shown below each image indicate bitrate and PSNR. The top two rows correspond to $\lambda = 0.04$ and the bottom two rows correspond to $\lambda = 0.2$, respectively

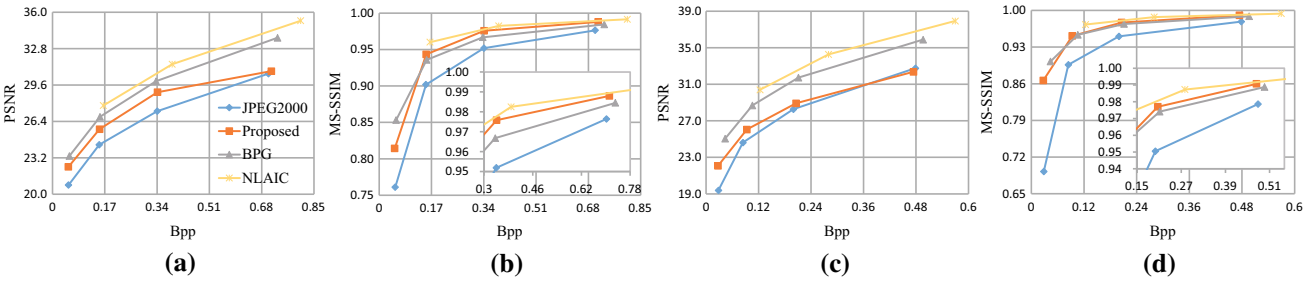


Fig. 9 Rate-distortion curves of the proposed method compared to JPEG2000, BPG, and NLAIC (Chen et al. 2019). **a**, **b** Correspond to the CUB200-2011 dataset. **c**, **d** Correspond to the FGVC-Aircraft dataset

this is probably due to the effectiveness of multi-scale decomposition of our method.

6.3.3 Complexity Analysis

We report the average running time of each module in Table 3. The time was measured on a GTX1080Ti GPU. Note that our scheme enables partial decoding, so we report the time needed for coarse-grained classification, fine-grained classification, and image reconstruction, respectively. It can be observed that the slowest module in our scheme is the decoding module, which is attributed to the autoregressive context model (Chen et al. 2019). In the future, we may simplify the context model to accelerate the decoding process. It is also noticeable that, excluding the decoding, the analysis modules at the decoder side are computationally efficient, reducing more than 90% computational time than the image-based classification.

6.4 Performance of LFRNet

LFRNet converts an image into hierarchical feature representations in a revertible manner. Partial features may serve as a compact representation of the information needed for

Table 3 Average running time of each module for one image

Module	Running time (s)
Encoder	
LFRNet	0.0099
Encoding of F_1^s	0.0264
Encoding of F_2^s	0.0241
Encoding of the other features	0.1931
Total time	0.2535
Decoder	
Decoding of $\hat{F}_1^s$	$t_1 = 0.8244$
Decoding of $\hat{F}_2^s$	$t_2 = 2.3098$
Decoding of the other features	$t_3 = 47.3683$
Analysis of $\hat{F}_1^s$	$t_4 = 0.0002$
Analysis of $\{\hat{F}_1^s, \hat{F}_2^s\}$	$t_5 = 0.0003$
Inverse LFRNet	$t_6 = 0.0099$
Total time for coarse-grained	$t_1 + t_4 = 0.8246$
Total time for fine-grained	$t_1 + t_2 + t_5 = 3.1345$
Total time for image reconstruction	$t_1 + t_2 + t_3 + t_6 = 50.5124$
Image-based classification	
Coarse-grained	0.0036
Fine-grained	0.0043

Table 4 Classification accuracy on ILSVRC dataset

Network	Top-1 Acc.	Top-5 Acc.	# Parameters
VGG16	73.36	91.51	1.38e8
i-RevNet	74.02 (↑0.66)	91.59 (↑0.08)	1.25e8 (↓ 9%)
Ours	72.84 (↓0.52)	90.32 (↓1.19)	0.54e8 (↓61%)

a machine vision task. The image can be perfectly reconstructed based on all features. In this section, we ask: how many features are sufficient for a given task? We provide empirical results to address this question. In addition, we compare with other networks, notably VGG16 (Simonyan and Zisserman 2014) and i-RevNet (Jacobsen et al. 2018).

6.4.1 Performance of Pre-trained Models

We have pre-trained LFRNet on the ILSVRC training set. For fair comparison, we similarly pre-train VGG16 and i-RevNet on the same training set. Then we test the three models on the ILSVRC validation set. The tested classification accuracy, as well as number of parameters, is shown in Table 4. The three models achieve comparable top-1 and top-5 accuracy, but our LFRNet has greatly reduced the number of parameters. Note that our LFRNet gradually discards information in the network, which is quite different from VGG16 and i-RevNet. It also confirms that the information required for classification may be compactly represented by a small set of features.

6.4.2 Performance in Coarse-to-Fine Classification

Based on the pre-trained LFRNet model and a given dataset/task (e.g. CUB200-2011, coarse-grained classification), we want to identify how many features are sufficient for the task. Here we perform a grid search using k channels of F_K^m, where $k \in \{8, 16, 24, \ldots, 96\}$. For each k we fine-tune LFRNet with the given dataset/task, and draw the curves of top-1 and top-5 accuracy with respect to k in Fig. 10. It can be observed that the accuracy becomes stable when k is larger than a "critical number." According to these results, we use 24 channels for coarse-grained classification and 96 channels for fine-grained classification, respectively. In other words, F_1^s has 24 channels, and F_2^s has 72 channels.

Based on the above settings, we now fine-tune LFRNet for two classification tasks simultaneously to optimize (12). For comparison, we also fine-tune VGG16 and i-RevNet, but for coarse-grained and fine-grained classification individually. Table 5 presents the classification results of different fine-tuned models on the corresponding test set. It is observed that our method achieves comparable classification accuracy, but our method uses much less features, again demonstrating the advantage of compact representation.

6.4.3 Performance in Intermediate-Grained Classification

We go one step further to ask: may LFRNet perform well on non-trained machine vision tasks? To answer this

Table 5 Classification accuracy results of uncompressed features

Dataset	Method	Coarse-grained classification			Fine-grained classification		
		Feature	Top-1 Acc.	Top-5 Acc.	Feature	Top-1 Acc.	Top-5 Acc.
CUB200-2011	VGG16	(512, 8, 8)	88.9	98.7	(512, 8, 8)	75.8	94.0
	i-RevNet	(1024, 8, 8)	89.7(↑0.8)	98.5(↓0.2)	(1024, 8, 8)	76.7(↑0.9)	93.3(↓0.7)
	Ours	(24, 8, 8)	89.3(↑0.4)	98.4(↓0.3)	(96, 8, 8)	75.5(↓0.3)	93.3(↓0.7)
FGVC-Aircraft	VGG16	(512, 8, 8)	93.3	98.5	(512, 8, 8)	89.6	97.0
	i-RevNet	(1024, 8, 8)	92.1(↓1.2)	98.5(↑0.0)	(1024, 8, 8)	90.0(↑0.4)	97.2(↑0.2)
	Ours	(24, 8, 8)	92.6(↓0.7)	98.6(↑0.1)	(96, 8, 8)	89.0(↓0.6)	97.0(↑0.0)

The dimensions of used features are shown in numbers like (512,8,8), which is short for $512 \times 8 \times 8$, i.e. 512 feature maps with spatial resolution 8×8

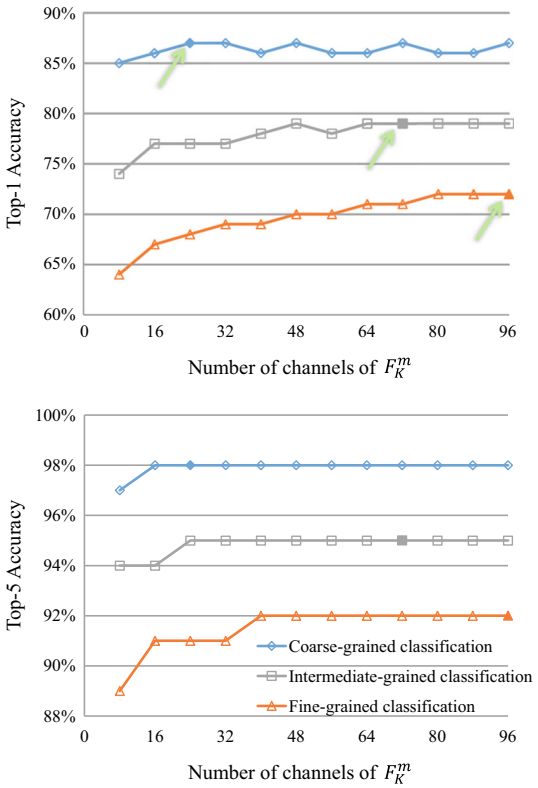


Fig. 10 Relation between classification accuracy and number of channels of F_K^m. In each curve, there is a solid marker showing that the top-1 accuracy becomes stable after that point (also indicated by a green arrow) (Color figure online)

question, we introduce a third classification task, namely the intermediate-grained as mentioned before. Note that intermediate-grained classification is not considered during the LFRNet training.

Again we need to identify how many features are sufficient for this task. We now fix all the parameters of LFRNet, set different values of k, and train the classifier (also a three-layer fully-connected network) for the intermediate-grained classification task. By grid search, we found $k = 72$ is appropriate, as shown in Fig. 10.

Table 6 Classification accuracy results of uncompressed features for the intermediate-grained classification task

Dataset	Method	Top-1 Acc.	Top-5 Acc.
CUB200-2011	VGG16	81.9	96.4
	i-RevNet	84.2(↑2.3)	96.4(↑0.0)
	Ours	80.7(↓1.2)	96.0(↓0.4)
FGVC-Aircraft	VGG16	89.7	96.8
	i-RevNet	89.2(↓0.5)	97.1(↑0.3)
	Ours	88.4(↓1.3)	96.3(↓0.5)

Note that in our method, the features ($72 \times 8 \times 8$) are a subset of the features ($96 \times 8 \times 8$) prepared for the fine-grained classification. Nonetheless, the intermediate-grained classification was not considered during the training

For comparison, we also fine-tune VGG16 and i-RevNet for the intermediate-grained classification. The results are given in Table 6. It is observed that LFRNet still achieves very competitive results, especially taken into account that VGG16 and i-RevNet are specifically trained for the task. These results demonstrate that LFRNet has extracted compact features that work well for trained tasks and meanwhile are generalizable to similar tasks.

7 Conclusion

In this paper, we have presented a semantics-to-signal scalable image compression framework with learned revertible representations. We have proposed LFRNet to learn effective and efficient features oriented to machine vision tasks. We have also proposed LCNet to compress the features into a scalable bitstream, so as to achieve a joint optimization of compression ratio, signal reconstruction quality, and semantic analysis accuracy. As a concrete example of human–machine collaborative judgment, we study coarse-

to-fine image classification and image reconstruction as the targets for image compression. Our experimental results have verified the effectiveness of the proposed method, which outperforms JPEG2000 significantly.

In the future, our work can be extended in several directions. First, we may further investigate revertible networks to enhance the feature learning capabilities. Second, we may design advanced methods for more efficient compression of the features. Third, we may consider video coding in a similar way but need to address motion carefully.

Acknowledgements This work was supported by the National Key Research and Development Program of China under Grant 2018YFA07 01603, by the Natural Science Foundation of China under Grant 61772483, and by the Fundamental Research Funds for the Central Universities under Contract WK3490000005. We acknowledge the support of the GPU cluster built by MCC Lab of the School of Information Science and Technology of USTC.

References

Akansu, A. N., Haddad, P. A., Haddad, R. A., & Haddad, P. R. (2001). *Multiresolution signal decomposition: Transforms, subbands, and wavelets*. New York: Academic Press.

Akansu, A. N., & Liu, Y. (1991). On-signal decomposition techniques. *Optical Engineering*, *30*(7), 912–921.

Ballé, J., Laparra, V., & Simoncelli, E. P. (2016). *End-to-end optimized image compression*. Technical Report. arXiv preprint arXiv:1611.01704

Ballé, J., Minnen, D., Singh, S., Hwang, S. J., & Johnston, N. (2018). *Variational image compression with a scale hyperprior*. Technical Report. arXiv preprint arXiv:1802.01436

Baxter, J. (1997). A bayesian/information theoretic model of learning to learn via multiple task sampling. *Machine Learning*, *28*(1), 7–39.

Chen, T., Liu, H., Ma, Z., Shen, Q., Cao, X., & Wang, Y. (2019). *Neural image compression via non-local attention optimization and improved context modeling*. Technical Report. arXiv preprint arXiv:1910.06244

Christopoulos, C., Skodras, A., & Ebrahimi, T. (2000). The JPEG2000 still image coding system: An overview. *IEEE Transactions on Consumer Electronics*, *46*(4), 1103–1127.

Dejean-Servières, M., Desnos, K., Abdelouahab, K., Hamidouche, W., Morin, L., & Pelcat, M. (2017). *Study of the impact of standard image compression techniques on performance of image classification with a convolutional neural network*. Technical Report. hal-01725126. https://hal.archives-ouvertes.fr/hal-01725126

Dodge, S., & Karam, L. (2016). Understanding how image quality affects deep neural networks. In *International conference on quality of multimedia experience* (pp. 1–6). IEEE.

Duan, L., Liu, J., Yang, W., Huang, T., & Gao, W. (2020). Video coding for machines: A paradigm of collaborative compression and intelligent analytics. *IEEE Transactions on Image Processing*, *29*, 8680–8695.

Gomez, A. N., Ren, M., Urtasun, R., & Grosse, R. B. (2017). The reversible residual network: Backpropagation without storing activations. In: *Advances in neural information processing systems* (pp. 2214–2224).

Goutsias, J., & Heijmans, H. J. (2000). Nonlinear multiresolution signal decomposition schemes (I) Morphological pyramids. *IEEE Transactions on Image Processing*, *9*(11), 1862–1876.

He, C., Shi, Z., Qu, T., Wang, D., & Liao, M. (2019). Lifting scheme-based deep neural network for remote sensing scene classification. *Remote Sensing*, *11*(22), 2648.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *CVPR* (pp. 770–778).

Heijmans, H. J., & Goutsias, J. (2000). Nonlinear multiresolution signal decomposition schemes (II) Morphological wavelets. *IEEE Transactions on Image Processing*, *9*(11), 1897–1913.

Hu, Y., Yang, S., Yang, W., Duan, L. Y., & Liu, J. (2020). Towards coding for human and machine vision: A scalable image coding approach. In *ICME* (pp. 1–6). IEEE.

Huang, G., Liu, Z., Van Der Maaten L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *CVPR* (pp. 4700–4708).

Jacobsen, J. H., Smeulders, A., & Oyallon, E. (2018). *i-Revnet: Deep invertible networks*. Technical Report. arXiv preprint arXiv:1802.07088

Johnston, P., Elyan, E., & Jayne, C. (2018). Spatial effects of video compression on classification in convolutional neural networks. In *IJCNN* (pp. 1–8).

Kwaśnicka, H., & Jain, L. C. (2018). *Bridging the semantic gap in image and video analysis*. Berlin: Springer.

Latif, A., Rasheed, A., Sajid, U., Ahmed, J., Ali, N., Ratyal, N. I., et al. (2019). Content-based image retrieval and feature extraction: A comprehensive review. *Mathematical Problems in Engineering*, *2019*, 1–21.

Lee, J., Cho, S., & Beack, S. K. (2018). *Context-adaptive entropy model for end-to-end optimized image compression*. Technical Report. arXiv preprint arXiv:1809.10452

Li, M., Zhang, K., Zuo, W., Timofte, R., & Zhang, D. (2020). *Learning context-based non-local entropy modeling for image compression*. Technical Report. arXiv preprint arXiv:2005.04661

Lo, S. C., Li, H., & Freedman, M. T. (2003). Optimization of wavelet decomposition for image compression and feature preservation. *IEEE Transactions on Medical Imaging*, *22*(9), 1141–1151.

Ma, H., Liu, D., Xiong, R., & Wu, F. (2019). iWave: CNN-based wavelet-like transform for image compression. *IEEE Transactions on Multimedia*, *22*, 1667–1679.

Ma, H., Liu, D., Yan, N., Li, H., & Wu, F. (2020). End-to-end optimized versatile image compression with wavelet-like transform. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,. https://doi.org/10.1109/TPAMI.2020.3026003.

Ma, S., Zhang, X., Wang, S., Zhang, X., Jia, C., & Wang, S. (2018). Joint feature and texture coding: Toward smart video representation via front-end intelligence. *IEEE Transactions on Circuits and Systems for Video Technology*, *29*(10), 3095–3105.

Maji, S., Rahtu, E., Kannala, J., Blaschko, M., & Vedaldi, A. (2013). *Fine-grained visual classification of aircraft*. Technical Report. arXiv preprint arXiv:1306.5151.

Mallat, S. G. (1989). A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *11*(7), 674–693.

Marpe, D., Schwarz, H., & Wiegand, T. (2003). Context-based adaptive binary arithmetic coding in the h.264/AVC video compression standard. *IEEE Transactions on Circuits and Systems for Video Technology*, *13*(7), 620–636.

Minnen, D., Ballé, J., & Toderici, G. D. (2018). Joint autoregressive and hierarchical priors for learned image compression. In *Advances in neural information processing systems* (pp. 10771–10780).

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., & Lerer A. (2017). *Automatic differentiation in PyTorch*. Technical report. OpenReview.net, https://openreview.net/forum?id=BJJsrmfCZ.

Poyser, M., Atapour-Abarghouei, A., & Breckon, T. P. (2020). *On the impact of lossy image and video compression on the performance*

of deep convolutional neural network architectures. Technical Report. arXiv preprint arXiv:2007.14314.

Rodriguez, M. X. B., Gruson, A., Polania, L., Fujieda, S., Prieto, F., Takayama, K., & Hachisuka, T. (2020). Deep adaptive wavelet network. In *IEEE Winter conference on applications of computer vision* (pp. 3111–3119).

Ruder, S. (2017). *An overview of multi-task learning in deep neural networks*. Technical Report. arXiv preprint arXiv:1706.05098

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., et al. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*(3), 211–252.

Shwartz-Ziv, R., & Tishby, N. (2017). *Opening the black box of deep neural networks via information*. Technical Report. arXiv preprint arXiv:1703.00810

Simonyan, K., & Zisserman, A. (2014). *Very deep convolutional networks for large-scale image recognition*. Technical Report. arXiv preprint arXiv:1409.1556.

Sweldens, W. (1998). The lifting scheme: A construction of second generation wavelets. *SIAM Journal on Mathematical Analysis*, *29*(2), 511–546.

Taubman, D. (2000). High performance scalable image compression with EBCOT. *IEEE Transactions on Image Processing*, *9*(7), 1158–1170.

Tishby, N., Pereira, F. C., & Bialek, W. (2000). *The information bottleneck method*. Technical Report. arXiv preprint arXiv:physics/0004057.

Tishby, N., & Zaslavsky, N. (2015). Deep learning and the information bottleneck principle. In *IEEE information theory workshop* (pp. 1–5).

Toderici, G., O'Malley, S. M., Hwang, S. J., Vincent, D., Minnen, D., Baluja, S., Covell, M., & Sukthankar, R. (2015). *Variable rate image compression with recurrent neural networks*. Technical Report. arXiv preprint arXiv:1511.06085

Torfason, R., Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., & Van Gool, L. (2018). *Towards image understanding from deep compression without decoding*. Technical Report. arXiv preprint arXiv:1803.06131

Wah, C., Branson, S., Welinder, P., Perona, P., & Belongie, S. (2011). *The Caltech-UCSD Birds-200-2011 Dataset*. Technical Report. CNS-TR-2011-001, California Institute of Technology.

Wang, S., Wang, S., Zhang, X., Wang, S., Ma, S., & Gao, W. (2019). Scalable facial image compression with deep feature reconstruction. In *ICIP* (pp. 2691–2695). IEEE.

Xia, S., Liang, K., Yang, W., Duan, L. Y., & Liu, J. (2020). An emerging coding paradigm VCM: A scalable coding approach beyond feature and signal. In *ICME* (pp. 1–6). IEEE

Yan, N., Liu, D., Li, H., & Wu, F. (2020). Semantically scalable image coding with compression of feature maps. In *ICIP*, IEEE (pp. 3114–3118).

Zhang, X., Ma, S., Wang, S., Zhang, X., Sun, H., & Gao, W. (2016). A joint compression scheme of video feature descriptors and visual content. *IEEE Transactions on Image Processing*, *26*(2), 633–647.

Zhao, J., Peng, Y., & He, X. (2020). Attribute hierarchy based multi-task learning for fine-grained image classification. *Neurocomputing*, *395*, 150–159.

Zhao, Z. Q., Zheng, P., & Xu St, Wu X. (2019). Object detection with deep learning: A review. *IEEE Transactions on Neural Networks and Learning Systems*, *30*(11), 3212–3232.

Zhou, L., Sun, Z., Wu, X., & Wu, J. (2019). End-to-end optimized image compression with attention mechanism. In *CVPR workshops* (pp. 1–4).