End-to-End Image Classification and Compression with variational autoencoders

Lahiru D. Chamain, *Student Member, IEEE,* Siyu Qi, and Zhi Ding, *Fellow, IEEE*

***Abstract*—The past decade has witnessed the rising dominance of deep learning and artificial intelligence in a wide range of applications. In particular, the ocean of wireless smartphones and IoT devices continue to fuel the tremendous growth of edge/cloud-based machine learning (ML) systems including image/speech recognition and classification. To overcome the infrastructural barrier of limited network bandwidth in cloud ML, existing solutions have mainly relied on traditional compression codecs such as JPEG that were historically engineered for human-end users instead of ML algorithms. Traditional codecs do not necessarily preserve features important to ML algorithms under limited bandwidth, leading to potentially inferior performance. This work investigates application-driven optimization of programmable commercial codec settings for networked learning tasks such as image classification. Based on the foundation of variational autoencoders (VAEs), we develop an end-to-end networked learning framework by jointly optimizing the codec and classifier without reconstructing images for given data rate (bandwidth). Compared with standard JPEG codec, the proposed VAE joint compression and classification framework achieves classification accuracy improvement by over 10% and 4%, respectively, for CIFAR-10 and ImageNet-1k data sets at data rate of 0.8 bpp. Our proposed VAE-based models show 65%−99% reductions in encoder size, $\times$1.5−$\times$13.1 improvements in inference speed and 25%−99% savings in power compared to baseline models. We further show that a simple decoder can reconstruct images with sufficient quality without compromising classification accuracy.**

***Index Terms*—Variational autoencoders, End-to-end, Classification, Compression, Reconstruction.**

I. Introduction

THE concept "Internet of Everything (IoE)" generalizes the idea of internet of things (IoT) to a broader paradigm. IoE nodes are connected by networks capable of communicating data generated from sensing and processing [1]. The power of IoE has already bolstered learning-intensive technologies such as self-driving cars, surveillance cameras, smart cities and smart transportation where deep learning (DL) applications are increasingly integrated with network services. As part of networked learning paradigm, these DL applications together with network services facilitate machine-to-machine and machine-to-cloud server communications [2]–[6].

One such example of networked learning is illustrated in Fig. 1. In automated inference systems such as pedestrian detection or object tracking for applications like self-driving, media data (image or video) collected by a source node is transported to remote network cloud nodes to carry out more complex inference processes. The data transmission can be for multiple inference tasks (such as vehicle detection, traffic monitoring, or risk warning) at the cloud servers or visualization purposes at cloud or end nodes. For instance, video/image frames captured by a street camera are sent to a server for vehicular detection, crowd activity monitoring, etc., and used by smart vehicles with an obstructed view for assisted/self-driving-related visualizations/decision-making.

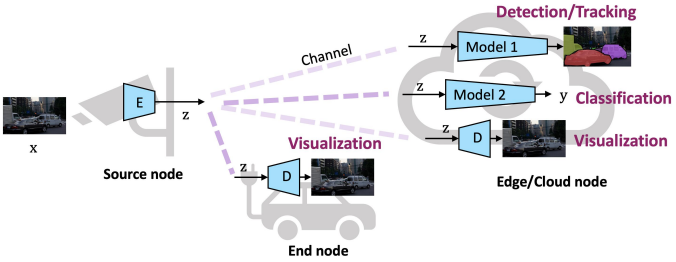


Fig. 1. A networked ML instance: video/image frames from a camera are sent to a server for vehicle detection, traffic monitoring, and used by vehicles with an obstructed view for assisted/self-driving-related visualizations/decision-making. E: Encoder, D: Decoder.

Given the explosive growth and deployment of IoT devices [7] and massive data collected by such devices [8], traditional cloud computing struggles to keep up with the demands of large IoT networks. Typical IoT configuration features source devices only as data collectors while cloud nodes are responsible for processing and analysis, hence, is limited by network link capacity, delay and losses, leading to long latency, unreliable inference and scalability issues [9]–[11]. Edge computing addresses these issues by deploying computing services in proximity to IoT devices [12]. Many image and video sensing devices such as smart phones, vehicular sensors and home/street cameras have sufficient resources to collect data and perform preliminary pre-processing and feature extraction, but not enough to implement full machine learning algorithms on-device [10]. With this new configuration of networked AI coupled with edge computing, such source devices can share a part of the computational burden by performing pre-processing and feature extraction, and deliver the extracted features to more powerful and resource rich edge/fog nodes for complex machine learning tasks.

When designing communication networks specifically targeting networked AI over edge/cloud for above such time-sensitive tasks, source data transmitted to the edge/cloud may fulfill two main requirements. On one hand, the data transport must achieve high coding efficiency ensuring low power usage, low latency and high bandwidth efficiency. On the other hand, source data compression for transport must

Our code is publicly available at https://github.com/chamain/AE-classifier.

The authors are with the Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616 e-mail: hdchamain@ucdavis.edu, syqi@ucdavis.edu, zding@ucdavis.edu.

Manuscript received April 25, 2021.

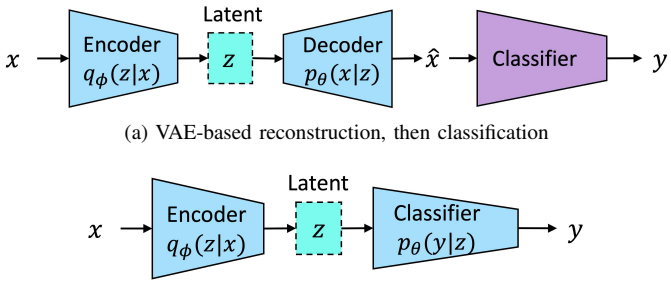
maintain high learning (task) accuracy. Therefore, there is a clear design trade-off between source coding rate and learning accuracy. This work designs a variational autoencoder (VAE)-based compression and learning framework to achieve a better coding rate and learning accuracy trade-off ensuring low power usage, low latency and low memory usage.

Our focused investigation in this work is on an efficient image codec for classification applications. Conventional classification applications begin by reconstructing images of interest at the decoder before inference [13]–[15]. See Fig 2(a). We note the following two observations where the reconstruction can introduce inefficiency in cloud-based AI applications. First, when performing classification inference on the images compressed with image codecs that are originally optimized for rate-distortion performance targeting visualization applications, the classification accuracy suffers at high compression ratios under limited data rate [16]–[20]. Second, inference on images compressed with standard compression codecs such as JPEG2000 demonstrates non-negligible inference speed and accuracy degradation compared to with end-to-end-optimized joint classification and compression frameworks [17], [18] that bypass image reconstruction at decoders. Motivated by these observations, the proposed VAE-based classifier bypasses the reconstruction and is designed to naturally support end-to-end optimization for rate-accuracy performance. See Fig. 2(b).

In terms of device complexity, conventional video/image compression codecs such as VVC [21], HEVC [22] and JPEG [23] featuring a more complex encoder and a simpler decoder could be less desirable on low cost source devices. Instead, a more desirable data compression and learning model for massive deployment in a network-AI edge/cloud should feature a simple encoder with low power and memory usage and a relatively more complex decoding and learning edge/cloud node. The trained encoder can reside in source devices with low computational power and memory to encode and compress data collections. Further, they can selectively compress and transmit task-important underlying features (e.g., latent) of the raw source data to other edge/cloud servers dedicated for decoding and learning ensuring rate-accuracy performance.

The approaches in [16]–[18] are likely sub-optimal in terms of coding efficiency by optimizing only a part of pre-engineered JPEG2000 encoding-decoding pipeline. Addressing this, as part of end-to-end optimization of the codec, some recent DL-based image/video compression codecs showcase a learning-based encoder (autoencoder) that can be optimized to minimize a given loss function [20], [24]–[28]. They demonstrated the importance of learning code word distribution for efficient compression instead of using fixed code word tables as in JPEG. However, the image encoders used in above works are still high in complexity, demanding high memory and power and resulting in lower inference speeds. Hence, they are not suitable for broad IoT deployment. Thus, we propose to use light-weight encoders and to model code word distributions for VAE-based classifier achieving high coding efficiency.

To achieve the aforementioned objectives in a networked AI environment, we propose a VAE-based joint compression and classification model (shown in Fig. 2(b)) that enables learning on the latent feature space to efficiently encode/compress and effectively classify images through end-to-end training. We aim to achieve high accuracy, fast inference, low bandwidth usage, and low latency over network links. Furthermore, our proposed framework features simple encoders while ensuring re-usability of the transmitted features.



(a) VAE-based reconstruction, then classification

(b) Proposed: VAE-based classification

Fig. 2. (a). VAE-based reconstruction before classification: Encoder transforms source image $\boldsymbol{x}$ to low dimensional $\boldsymbol{z}$. Decoder reconstructs image $\hat{\boldsymbol{x}}$ based on $\boldsymbol{z}$. (b). Proposed joint compression and classification: Without image reconstruction, classifier directly generates class label y based on $\boldsymbol{z}$.

Our two main contributions of this work are:

- A new information theoretical formulation of a VAE-based end-to-end, joint compression and classification framework.
- Development of power saving, low-complexity-encoder and faster-inference-classifier for edge/cloud-based networked image classification applications.

Furthermore, because encoded images may serve the dual-use of human visual perception and machine learning in certain practical applications, we investigate the feasibility of adapting the VAE-based classifier as a dual-use codec for efficiently encoding image features delivered to the cloud for both accurate image classification and image reconstruction of sufficiently high quality (PSNR).

We organize this manuscript as follows. Sec. II presents the basic problem formulation and review the concept of variational autoencoders. Sec. III introduces our new proposal of a VAE-based end-to-end solution for classification by describing the basic learning model and the rate-accuracy function for joint classification-compression. In Sec. IV we describe our experiment setup and present our test results. Additionally, Sec. V presents benchmark comparison of model complexity, inference speed, and power consumption for our proposed models. In Sec. VI, we discuss how to adapt the proposed models for joint classification and reconstruction. Sec. VII further provides theoretical basis that connects the proposed VAE-based approaches to frameworks such as information bottleneck [29], [30].

II. Related Works

A. End-to-End Learning Framework

Originally designed to target remote human vision applications over resource constrained channels, popular image compression codecs such as JPEG [31] and JPEG2000 [32] feature an engineered data processing workflow. This workflow mainly consists of a level offset, a color transformation

TABLE I
SUMMARY OF NOTATIONS

Notation	Meaning
$\mathcal{X}$	set of input images
$\mathbf{x}$	input image
$\hat{\mathbf{x}}$	reconstructed image
y_{gt}	ground truth label of image $\mathbf{x}$
y	estimated label of image $\mathbf{x}$
$\mathbf{z}$	latent vector representation of $\mathbf{x}$
$\hat{\mathbf{z}}$	quantized $\mathbf{z}$
d_ℓ	quantization level
$\mathcal{D}$	set of quantization levels d_ℓ
$p_{\mathbf{x},\boldsymbol{\theta}}$	probability distribution of $\mathbf{x}$ parameterized by $\boldsymbol{\theta}$
$p_{\mathbf{z}\|\mathbf{x},\boldsymbol{\theta}}$	conditional distribution of $\mathbf{z}$ parameterized by $\boldsymbol{\theta}$
$q_{\mathbf{z}\|\mathbf{x},\boldsymbol{\phi}}$	approximated distribution of $p_{\mathbf{z}\|\mathbf{x},\boldsymbol{\theta}}$ parameterized by $\boldsymbol{\phi}$
$\hat{\rho}_{\boldsymbol{\theta}}(y\|\boldsymbol{x})$	estimated class label distribution with a model parameterized by $\boldsymbol{\theta}$ given the latent $\mathbf{x}$
$\eta_{\boldsymbol{\theta}}(y\|\boldsymbol{z})$	estimated class label distribution with a model parameterized by $\boldsymbol{\theta}$ given the latent $\mathbf{z}$
$P_{i,\ell_i}(\hat{\mathbf{z}}\|\boldsymbol{x})$	conditional probability of the latent value $\hat{z}_i$ that takes the value of the quantization level d_{ℓ_i}
$Q_{i,\ell_i}(y_{\text{gt}}\|\hat{\mathbf{z}})$	class label probability for each $\hat{z}_i$ that takes the value of d_{ℓ_i}
$\text{KL}(p,q)$	KL divergence between distributions p and q
$\text{CE}(y_{\text{gt}},y)$	cross entropy between distribution of y_{gt} and y
$H(p_{\mathbf{x},\boldsymbol{\theta}})$	entropy of variable $\mathbf{x}$ with distribution $p_{\mathbf{x},\boldsymbol{\theta}}$
β,γ	trade-off coefficients for rate, distortion.

(e.g., RGB to YCbCr), a domain transformation such as Discrete Cosine Transformation (DCT) or Discrete Wavelet Transformation (DWT), a quantizer and a block encoder (e.g., Huffman or Arithmetic) to convert the quantized transform domain coefficients to a bit-stream for transmission.

These engineered blocks of conventional image codecs can often achieve good reconstruction results for a given data rate. Similarly, for a given image quality/distortion, such codecs can generate low rate bit-streams for transmission over low bandwidth network channels. However, these popular image/video codecs such as JPEG and JPEG2000 are not optimized for remote learning tasks. Thus, a naïve and direct adoption of these codecs for cloud-based learning applications do not guarantee good performance or high spectrum efficiency. Conceptually, source codecs should be customized and optimized for specific cloud-based learning tasks [18], [20]. Therefore, our objective in this work aims to optimize rate-accuracy performance in terms of encoding/compression rate and classification accuracy.

B. Source Data Reconstruction with VAEs

When designing codecs for source compression to achieve optimized rate-distortion performance, accurate modeling of the data distribution is of vital importance. Depending on the underlying input distribution, the process of "generative modeling" can be challenging. Often, the input data ($\mathcal{X}$) manifold is high-dimensional and is complex to characterize.

Several likelihood-based methods have been proposed in the literature for generative modeling, such as auto-regressive models [33], flow-based methods [34], [35] and VAEs [36]. In this work, we leverage the concept of VAE for joint classification and compression. Without relying on strong assumptions, VAE exhibits fast training with back-propagation [37]. Such property is advantageous in comparison with model based approaches relying on strong assumptions or requiring high computation complexity such as the Markov Chain Monte Carlo (MCMC) [36]. Further, VAE-based models are amenable to naturally interpretable loss terms that are directly related to rate-distortion trade-off in lossy compression [38], as shown later.

We can capture the general concept of VAE via Fig. 2(a). A VAE consists of an encoder for mapping the high dimensional input $\mathbf{x}$ into a latent representation $\mathbf{z}$, followed by a decoder in charge of reconstructing the input that is denoted as input estimate $\hat{\mathbf{x}}$. The encoder's output $\mathbf{z}$ is a low-dimensional latent vector representing distinct features from the input data $\mathbf{x}$. The encoder functionality is to compress and to extract critical features by mapping input $\mathbf{x}$ into $\mathbf{z}$. The decoder can rely on $\mathbf{z}$ to reconstruct $\mathbf{x}$ for various remote applications. Thus, the VAE encoder makes it possible to store and transmit $\mathbf{z}$ instead of $\mathbf{x}$ to preserve bandwidth and storage.

In the VAE formulation, the observable input data set $\mathcal{X}$ is assumed to consist of i.i.d. samples of $\mathbf{x}$ which are generated by some random processes that involves an unobserved random variable $\mathbf{z}$ and generative model parameters $\boldsymbol{\theta}$ via

$$p_{\mathbf{x},\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z}) p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z}. \tag{1}$$

Neither the true parameters $\bar{\boldsymbol{\theta}}$ nor $\mathbf{z}$ are known.

Within this framework [36], the encoder can be viewed as to provide an efficient posterior estimate of the latent vector $\mathbf{z}$ from an observed input $\mathbf{x}$ for a given parameter setting $\boldsymbol{\theta}$. The decoder provides an approximate marginal inference of $\mathbf{x}$ upon reception of the latent vector $\mathbf{z}$ from the encoding transmitter.

C. Variational Bound for Reconstruction

VAE considers the general case when the posterior $p_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x})$ is intractable for which an approximation $q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ parameterized by $\boldsymbol{\phi}$ is introduced to act as the encoder. For an arbitrary distribution $q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$, we can derive the "variational bound" that follows the outlines of [36], [37] with detailed proof below for both reconstruction and classification. Throughout the presentation, we denote an instance of the random variables $\mathbf{x}$ as $\boldsymbol{x}$. Further in Table. I, we summarize the notations used in the current work. Assume $\boldsymbol{x} \in \mathcal{X}$ is a random sample of the random variable $\mathbf{x}$ which follows a generative model parameterized by $\boldsymbol{\theta}$ from an unobserved random variable $\mathbf{z}$. We can write:

$$p_{\mathbf{x},\boldsymbol{\theta}}(\boldsymbol{x}) = \int p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z}) p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z}. \tag{2}$$

For an arbitrary distribution $q_{\mathbf{z},\boldsymbol{\phi}}(\boldsymbol{z})$, we can write,

$$\begin{aligned} p_{\mathbf{x},\boldsymbol{\theta}}(\boldsymbol{x}) &= \int p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z}) p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) d\boldsymbol{z} \\ &= \int p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z}) \frac{q_{\mathbf{z},\boldsymbol{\phi}}(\boldsymbol{z})}{q_{\mathbf{z},\boldsymbol{\phi}}(\boldsymbol{z})} d\boldsymbol{z} \\ &= E_{q_{\mathbf{z},\boldsymbol{\phi}}}\left[p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \frac{p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z})}{q_{\mathbf{z},\boldsymbol{\phi}}(\boldsymbol{z})} \right]. \end{aligned} \tag{3}$$

Taking $-\log$ of both sides leads to

$$-\log p_{\mathbf{x},\boldsymbol{\theta}}(\boldsymbol{x}) = -\log E_{q_{\mathbf{z},\boldsymbol{\phi}}}\left[p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \frac{p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z})}{q_{\mathbf{z},\boldsymbol{\phi}}(\boldsymbol{z})} \right]. \tag{4}$$

Applying Jensen's inequality to (4), we write,

$$\begin{aligned} -\log p_{\mathbf{x},\boldsymbol{\theta}}(\boldsymbol{x}) &\leq E_{q_{\mathbf{z},\boldsymbol{\phi}}} - \log \left[p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \frac{p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z})}{q_{\mathbf{z},\boldsymbol{\phi}}(\boldsymbol{z})} \right] \\ &\leq -E_{q_{\mathbf{z},\boldsymbol{\phi}}} \log p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) \\ &\quad - E_{q_{\mathbf{z},\boldsymbol{\phi}}} \log \left[\frac{p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z})}{q_{\mathbf{z},\boldsymbol{\phi}}(\boldsymbol{z})} \right]. \end{aligned} \tag{5}$$

Following the definition of KL divergence, we write,

$$-\log p_{\mathbf{x},\boldsymbol{\theta}}(\boldsymbol{x}) \leq -E_{q_{\mathbf{z},\boldsymbol{\phi}}} \log p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) + \mathrm{KL}(q_{\mathbf{z},\boldsymbol{\phi}}|p_{\mathbf{z},\boldsymbol{\theta}}). \tag{6}$$

Since $q_{\mathbf{z},\boldsymbol{\phi}}(\boldsymbol{z})$ is arbitrary, we can replace $q_{\mathbf{z},\boldsymbol{\phi}}(\boldsymbol{z})$ with conditional density $q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})$ and write [36], [37],

$$\begin{aligned} -\log p_{\mathbf{x},\boldsymbol{\theta}}(\boldsymbol{x}) \leq -E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\left[\log p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\right] \\ + \mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z},\boldsymbol{\theta}}) \end{aligned} \tag{7}$$

where KL denotes the Kullback Leibler divergence.

This variational bound can serve as an optimization surrogate when direct minimization of $-\log p_{\mathbf{x},\boldsymbol{\theta}}(\boldsymbol{x})$ is intractable. Functionally, $q_{\mathbf{z}|\boldsymbol{x},\boldsymbol{\phi}}$ models the probabilistic encoder and $p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}$ models the reconstruction decoder as in Fig. 2(a). The first RHS term in Eq. (7) is the conditional entropy of $\mathbf{x}$ given $\mathbf{z}$ which quantifies the reconstruction loss. The second RHS term $\mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z},\boldsymbol{\theta}})$ is related to the coding cost of the latents as shown later. Hence, the variational bound can be utilized to form the classical variational loss for reconstruction [36].

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) = E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\big[-\log p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\big] + \mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z},\boldsymbol{\theta}}) \tag{8}$$

When minimizing the variational bound as the loss function given in Eq. (8) that consists of two parts, a hard constraint can be imposed on the coding cost $\mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z},\boldsymbol{\theta}})$ while minimizing the reconstruction loss. To this extent, the authors of [39] further suggested adding a trade-off parameter $\beta(\geq 1)$ to Eq. (8) to reformulate a β-VAE loss function also adopted in [14], [27], [38] as

$$\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) = E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\big[-\log p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z})\big] + \beta \mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z},\boldsymbol{\theta}}). \tag{9}$$

Note that we can leverage the definition of cross entropy (CE) to rewrite

$$\mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z},\boldsymbol{\theta}}) = \mathrm{CE}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}, p_{\mathbf{z},\boldsymbol{\theta}}) - H(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}).$$

The term $\mathrm{CE}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}, p_{\mathbf{z},\boldsymbol{\theta}})$ averages the entropy $-\log(p_{\mathbf{z},\boldsymbol{\theta}})$ over the encoder distribution $q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}$ and captures the average encoding "cost" of latent representation $\mathbf{z}$. Following the approach of [38], if we consider only the deterministic encoder $\mathbf{z} = \boldsymbol{\phi}(\mathbf{x})$ for which $q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) = \delta(\boldsymbol{z} - \boldsymbol{\phi}(\boldsymbol{x}))$ and $H(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}) = 0$, we have

$$\begin{aligned} \mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z},\boldsymbol{\theta}}) &= \mathrm{CE}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}, p_{\mathbf{z},\boldsymbol{\theta}}) \\ &= \mathrm{CE}(\delta(\boldsymbol{z} - \boldsymbol{\phi}(\boldsymbol{x})), p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z})) \\ &= -\log p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{\phi}(\boldsymbol{x})). \end{aligned} \tag{10}$$

From Eq. (10), the β-VAE loss function for a deterministic encoder can be expressed as [38],

$$\begin{aligned} \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}) &= E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\left[-\log p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{z}) - \beta \log p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{z})\right] \\ &= -\log p_{\mathbf{x}|\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{x}|\boldsymbol{\phi}(\boldsymbol{x})) - \beta \log p_{\mathbf{z},\boldsymbol{\theta}}(\boldsymbol{\phi}(\boldsymbol{x})). \end{aligned} \tag{11}$$

When $\beta = 1$, Eq. (11) corresponds to the VAE loss function in Eq (9). Setting $\beta \geq 1$ can impose a hard constraint on the latent representation [39] to limit the coding cost. In short, the approximated β-VAE loss is the sum of the reconstruction loss and the cost of encoding $\mathbf{z}$ weighted by β. The weight β facilitates a rate-distortion trade-off parameter. One extreme case would be to encode nothing, i.e., $\mathbf{z} = 0$, in which case the encoding cost is zero whereas the reconstruction loss from $\mathbf{z} = 0$ would be gigantic. The other extreme case would be to encode $\mathbf{x}$ directly, i.e., $\mathbf{z} = \mathbf{x}$, in which case the reconstruction loss is 0 whereas coding cost would be high.

D. VAE-Based Classifier

When reviewing the VAE framework, data reconstruction from $\mathbf{z}$ is typically the learning objective. We note, however, that for automated learning tasks such as image classification, reconstruction of $\mathbf{x}$ may not always be the end goal [20], [40] and may be unnecessary in many cases. For example, a practical security-related automated object detection system would rely on AI or DL algorithm trained on the latent vector $\mathbf{z}$ for object detection and/or recognition. Hence, the reconstruction of image frame $\hat{\mathbf{x}}$ is unnecessary for inference. Only at rare occasions such as evidence collection or inspection, the system may have to recover RGB frames for human visualization. Thus, the image reconstruction step can be performed only when required from the stored latent vectors.

Considering these practical AI/learning applications, we propose a VAE-based classifier by removing the VAE decoder in Fig. 2(a) to directly connect the latent code $\mathbf{z}$ with a data classifier to perform the learning task, as shown in Fig. 2(b). In this framework, variable y denotes the label of the output class based on $\mathbf{z}$. Our framework aims to jointly optimize the encoder and the classifier with end-to-end training.

Before presenting details of our proposed VAE framework, we first examine some related recent works on the joint learning of compression and classification based on autoencoders. Among others, one approach constructs a loss function by combining the latent entropy $H(\mathbf{z})$ with the classification cross entropy loss $\mathcal{L}_{\mathrm{CE}}$ resulting from classification $\nu_0(\hat{\boldsymbol{x}})$ according to the reconstructed $\hat{\boldsymbol{x}}(\mathbf{z})$ from $\mathbf{z}$ (as shown in Fig. 2(a)).

$$\mathcal{L}(\boldsymbol{x}, y_{\mathrm{gt}}) = \mathcal{L}_{\mathrm{CE}}(y_{\mathrm{gt}}, \nu_0(\hat{\boldsymbol{x}}(\mathbf{z})) + \beta H(\mathbf{z}|\boldsymbol{x}) \tag{12}$$

Here y_{gt} denoted the ground truth class of image sample $\boldsymbol{x}$. Clearly, this approach requires a reconstruction step prior to classification [41], [42].

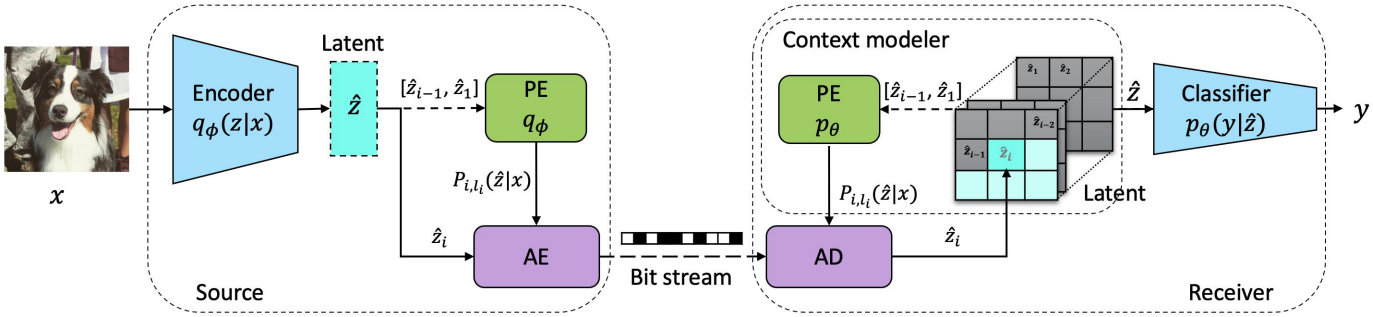


Fig. 3. Overview of the proposed VAE classifier during inference: Quantized latent vector $\hat{\mathbf{z}}$ is encoded into bit stream by a context-adaptive arithmetic encoder (AE) assisted by probability estimator (PE). At receiver, probability of each symbol $\hat{z}_i$ (shown in cyan) is estimated by using a learned PE based on previously decoded latents $\hat{z}_{i-1}, \cdots, \hat{z}_1$ (shown in gray). Without groundtruth distribution of the latent elements $q_{\boldsymbol{\phi}}(\hat{\mathbf{z}}|\boldsymbol{x})$ at the receiver, PE learns to approximate $p_{\boldsymbol{\theta}} \approx q_{\boldsymbol{\phi}}$ during training.

Another approach is to construct the loss function by combining the latent entropy $H(\mathbf{z})$ with the classification CE loss based directly on the latent representation $\mathbf{z}$. Hence, the loss considering the classification $\nu_1(\mathbf{z})$ directly based on encoded $\mathbf{z}(\boldsymbol{x})$ is,

$$\mathcal{L}(\boldsymbol{x}, y_{\text{gt}}) = \mathcal{L}_{\text{CE}}(y_{\text{gt}}, \nu_1(\mathbf{z}(\boldsymbol{x}))) + \beta H(\mathbf{z}|\boldsymbol{x}). \tag{13}$$

Compared with the first approach, this formulation can potentially generate faster and more efficient training/inference models by skipping the reconstruction step [43]. For example, the authors of [16], [18] introduced an image classification model by jointly training to optimize both the JPEG2000 encoder and the CNN classifier without image reconstruction. While achieving faster inference and higher accuracy, there is limited improvement since the proposed end-to-end framework only optimizes the quantization block of the encoder. Further, the majority of existing VAE-based image classification frameworks either still perform this unnecessary reconstruction step [13]–[15] or neglect the bandwidth aspect of the latent representation [44] making them impractical for wide IoT deployment in cloud-based image classification.

III. A New VAE Classification Framework

In this work, we focus on the second approach described in Eq. (13) and develop a novel VAE model for joint compression and classification encoding. Specifically, we design a VAE-based classifier for direct image classification based on the latent vectors without image reconstruction, as seen from Fig. 2. Targeting cloud-based classification over bandwidth-limited network data links, we implement adaptive entropy coders with context modeling aimed at improving overall rate-accuracy performance trade-off. Further, we demonstrate that spatial domain RGB images can also be reconstructed for practical applications by fine-tuning a separate decoder under the proposed classification setting.

A. Variational Bound and Loss for Classification

The performance of a classifier can be measured by cross entropy loss from labeling. Consider an i.i.d. data sample $\boldsymbol{x}$ and corresponding label y that belongs to a set $\mathcal{Y}$ of unique labels, with $\{(\boldsymbol{x}, y)|\boldsymbol{x} \in \mathcal{D}, y \in \mathcal{Y}\}$. We can denote $\eta_{\boldsymbol{\theta}}(y|\mathbf{z})$ as the estimated label distribution with a model parameterized by $\boldsymbol{\theta}$ given the latent $\mathbf{z}$. For a latent $\mathbf{z}$ mapped from given $\boldsymbol{x}$, the estimated label distribution is simply

$$\hat{\rho}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = \int \eta_{\boldsymbol{\theta}}(y|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}. \tag{14}$$

Similar to (3), we introduce an arbitrary function $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ parameterized by $\boldsymbol{\phi}$.

$$\begin{aligned}\hat{\rho}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) &= \int \eta_{\boldsymbol{\theta}}(y|\boldsymbol{z}) p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} \\ &= E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\left[\eta_{\boldsymbol{\theta}}(y|\mathbf{z}) \frac{p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right].\end{aligned}$$

Once again, taking $-\log$ on both sides leads to

$$-\log \hat{\rho}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) = -\log E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\left[\eta_{\boldsymbol{\theta}}(y|\mathbf{z}) \frac{p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right]. \tag{15}$$

Applying Jensen's inequality, we write,

$$\begin{aligned}-\log \hat{\rho}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) &\leq -E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}} \log \left[\eta_{\boldsymbol{\theta}}(y|\mathbf{z}) \frac{p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right] \\ &\leq -E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}} \log \eta_{\boldsymbol{\theta}}(y|\mathbf{z}) \\ &\quad - E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}} \log \left[\frac{p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x})}{q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x})}\right].\end{aligned} \tag{16}$$

Following the definition of KL divergence, we write,

$$-\log \hat{\rho}_{\boldsymbol{\theta}}(y|\boldsymbol{x}) \leq -E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}} \log \eta_{\boldsymbol{\theta}}(y|\mathbf{z}) + \text{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}). \tag{17}$$

We denote the ground truth class of sample $\boldsymbol{x}$ as y_{gt}. The true distribution ρ of y given $\boldsymbol{x}$ can be written as following by using the Kronecker delta function $\delta[\cdot]$

$$\rho(y|\boldsymbol{x}) = \delta[y - y_{\text{gt}}] = \begin{cases} 1, & y = y_{\text{gt}} \\ 0, & y \neq y_{\text{gt}} \end{cases} \tag{18}$$

With the above notion, we write the cross entropy between the true and the estimated label distributions as:

$$\text{CE}_{y|\boldsymbol{x}}(\rho, \hat{\rho}_{\boldsymbol{\theta}}) = -\sum_{c \in \mathcal{Y}} \rho(y = c|\boldsymbol{x}) \log \hat{\rho}_{\boldsymbol{\theta}}(y = c|\boldsymbol{x}). \tag{19}$$

From (17) and (19), we can derive the variational bound for classification.

$$\begin{aligned}\mathrm{CE}_{y|\boldsymbol{x}}(\rho,\hat{\rho}_{\boldsymbol{\theta}}) &= -\sum_{c\in\mathcal{Y}}\rho(y=c|\boldsymbol{x})\log\hat{\rho}_{\boldsymbol{\theta}}(y=c|\boldsymbol{x})\\ &\le -\sum_{c\in\mathcal{Y}}\rho(y=c|\boldsymbol{x})\{E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\log\eta_{\boldsymbol{\theta}}(y=c|\mathbf{z})\}\\ &\quad+\sum_{c\in\mathcal{Y}}\rho(y=c|\boldsymbol{x})\{\mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}})\} \quad (20)\end{aligned}$$

The bound can be further simplified into

$$\begin{aligned}\mathrm{CE}_{y|\mathbf{x}}(\rho,\hat{\rho}_{\boldsymbol{\theta}}) \le &-\sum_{c\in\mathcal{Y}}\rho(y=c|\boldsymbol{x})\{E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\log\eta_{\boldsymbol{\theta}}(y=c|\mathbf{z})\}\\ &+\mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}). \quad (21)\end{aligned}$$

Using (18), we obtain the variational bound for classification

$$\begin{aligned}\mathrm{CE}_{y|\boldsymbol{x}}(\rho,\hat{\rho}_{\boldsymbol{\theta}}) \le &-E_{q_{\mathbf{z}|\boldsymbol{x},\boldsymbol{\phi}}}\log\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z})\\ &+\mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}|p_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}) \quad (22)\end{aligned}$$

Following the approach [38] discussed in Sec. II-C, we can define the β-VAE loss function for a single sample $\boldsymbol{x}$ as

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x},y_{\mathrm{gt}}) = &E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\left[-\log\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z})\right]\\ &+\beta E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\left[-\log p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x})\right], \quad (23)\end{aligned}$$

with a control parameter β. We can interpret the first term as the conditional classification loss and the second term as the coding cost or rate.

B. Learning Model

Our end-to-end compression-classification learning model consists of an encoder (**E**), a probability estimator (**PE**) and a classifier (**CL**) as shown in Fig. 3. Following the approach in [27], we use a context model to estimate the conditional probabilities in **PE**. Consider the following setting.

$$\mathbf{x}\xrightarrow[\boldsymbol{\phi}]{\mathrm{E}}\mathbf{z}\xrightarrow{\mathrm{Q}}\hat{\mathbf{z}}\xrightarrow[\boldsymbol{\theta}]{\mathrm{CL}}y \quad (24)$$

$\mathbf{x}$ is the input image to the model with dimensions $(w\times h\times 3)$. The encoder, parameterized by $\boldsymbol{\phi}$, maps $\mathbf{x}$ to the latent representation $\mathbf{z}$ of dimension $(\frac{w}{s}\times\frac{w}{s}\times K=m)$. A quantizer (Q) further maps $\mathbf{z}$ to $\hat{\mathbf{z}}$ by assigning each element z_i $(i=1,\cdots,m)$ to the closest quantization levels $\mathcal{D}=\{d_\ell,\ \ell=1,\cdots,L\}$. For instance, multiple z_i values can be mapped to the same d_ℓ such that,

$$\hat{z}_i = Q(z_i) = \arg\min_{d\in\mathcal{D}}\|z_i-d\|_2. \quad (25)$$

The quantization centers $\{d_\ell\}$ are learned through training. The **CL**, parameterized by $\boldsymbol{\theta}$, takes the $\hat{\mathbf{z}}$ as the input and predicts the class label y for the given image sample $\boldsymbol{x}$. Taking quantization into account for practical applications, Eq. (23) can be rewritten as follows.

$$\begin{aligned}\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x},y_{\mathrm{gt}}) = &E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}}\left[-\log\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\hat{\mathbf{z}})\right]\\ &+\beta E_{q_{\hat{\mathbf{z}}|\mathbf{x},\boldsymbol{\phi}}}\left[-\log p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}|\boldsymbol{x})\right] \quad (26)\end{aligned}$$

We now examine how to obtain each term for the loss function in Eq. (26).

C. Rate Loss

The second RHS term of Eq. (26) represents the cross entropy between two conditional distributions of $\mathbf{z}$: $p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}|\boldsymbol{x})$ and $q_{\boldsymbol{\phi}}(\hat{\mathbf{z}}|\boldsymbol{x})$. The distribution $q_{\boldsymbol{\phi}}(\hat{\mathbf{z}}|\boldsymbol{x})$ is deterministic and readily available after the deterministic encoding step (**E**). During training, we estimate the distribution $p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}|\mathbf{x})$ with a **PE** using a conditional context model following [27], [38]. The latent representation $\hat{\mathbf{z}}$ as discussed in Sec. III-B is a 3D tensor containing m number of latent elements. We can index this 3D tensor $\hat{\mathbf{z}}$ given $\boldsymbol{x}$ as a vector in raster-scan order and write $p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}|\boldsymbol{x})$ as

$$p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}|\boldsymbol{x}) = \prod_{i=1}^{m} p_{\boldsymbol{\theta}}(\hat{z}_i|\hat{z}_{i-1},\cdots,\hat{z}_1,\boldsymbol{x}). \quad (27)$$

With **PE**, we can efficiently estimate the conditional probability of the latent value $\hat{z}_i$ that takes the value of the quantization center d_{ℓ_i} which can be defined as $P_{i,\ell_i}(\hat{\mathbf{z}}|\boldsymbol{x}) = p_{\boldsymbol{\theta}}(\hat{z}_i = d_{\ell_i}|\hat{z}_{i-1},\cdots,\hat{z}_1,\boldsymbol{x})$ for notional simplicity. With this notation, can write the approximate coding rate ($\mathcal{L}_{\mathrm{R}}$) as follows.

$$\begin{aligned}\mathcal{L}_{\mathrm{R}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x}) &= \mathrm{CE}_{\hat{\mathbf{z}}|\boldsymbol{x}}(q_{\boldsymbol{\phi}},p_{\theta})\\ &= E_{q_{\hat{\mathbf{z}}|\mathbf{x},\boldsymbol{\phi}}}\left[-\log p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}|\boldsymbol{x})\right]\\ &= -\sum_{i=1}^{m}\sum_{\ell_i=1}^{L} q_{\boldsymbol{\phi}}(\hat{z}_{i,\ell_i}|\boldsymbol{x})\log P_{i,\ell_i}(\hat{\mathbf{z}}|\boldsymbol{x}) \quad (28)\end{aligned}$$

Here $q_{\boldsymbol{\phi}}(\hat{z}_{i,\ell_i}|\boldsymbol{x})$ is the probability of $\Pr(\hat{z}_i = d_{\ell_i}|\boldsymbol{x})$. Any $\hat{z}_i$ can take only one particular quantization value d_{ℓ_i}.

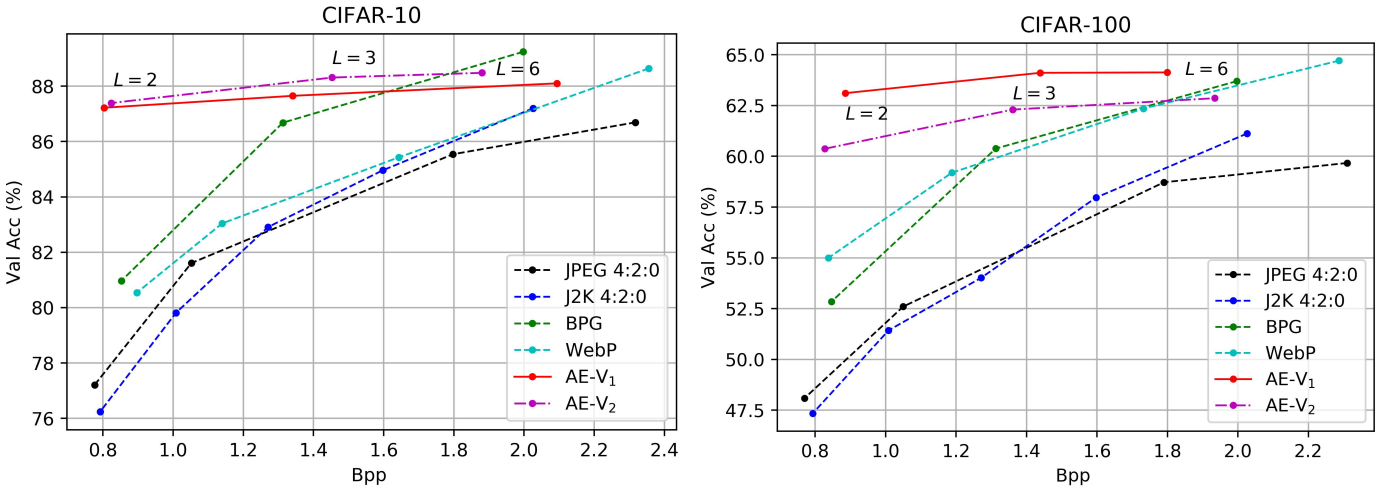
Similar to [27], through end-to-end training, we learn the distribution $p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}|\boldsymbol{x})$ in order to optimally approximate $p_{\boldsymbol{\theta}}\approx q_{\boldsymbol{\phi}}$. Through context-based adaptive arithmetic encoding process (CABAC) [45], $q_{\boldsymbol{\phi}}(\hat{\mathbf{z}}|\boldsymbol{x})$ is available for the probability estimation, only at the encoder. Once the encoded bitstream is received at the context-based adaptive arithmetic decoder, the learned conditional probability $p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}|\boldsymbol{x})$ is used for probability estimation to recover $\hat{\mathbf{z}}$ whereas $q_{\boldsymbol{\phi}}(\hat{\mathbf{z}}|\boldsymbol{x})$ is not available [46]. For this reason, we learn the distribution $p_{\boldsymbol{\theta}}(\hat{\mathbf{z}}|\boldsymbol{x})$ to approximate $p_{\boldsymbol{\theta}}\approx q_{\boldsymbol{\phi}}$ which makes $p_{\boldsymbol{\theta}}$ the "context modeler" as commonly referred in the literature [45], [46]. See Fig. 3. When the learning succeeds by training, with Eq. (28), we observe that the coding rate $\mathcal{L}_{\mathrm{R}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x})$ reduces to the entropy $H(\hat{\mathbf{z}}|\boldsymbol{x})$.

D. Classification Loss

We interpret the first RHS term of Eq. (26) as the classification loss $\mathcal{L}_{\mathrm{CL}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x},y_{\mathrm{gt}})$. Following the rate approach as given in Sec. III-C, we can rewrite the classification loss term as follows.

$$\begin{aligned}\mathcal{L}_{\mathrm{CL}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x},y_{\mathrm{gt}}) &= E_{q_{\hat{\mathbf{z}}|\mathbf{x},\boldsymbol{\phi}}}\left[-\log\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\hat{\mathbf{z}})\right]\\ &= -\sum_{i=1}^{m}\sum_{\ell_i=1}^{L} q_{\boldsymbol{\phi}}(\hat{z}_{i,\ell_i}|\boldsymbol{x})\log Q_{i,\ell_i}(y_{\mathrm{gt}}|\hat{\mathbf{z}})\end{aligned}$$

Here, we have to estimate the class label probability for each $\hat{z}_{i,\ell_i}$ which is defined as $Q_{i,\ell_i}(y_{\mathrm{gt}}|\hat{\mathbf{z}}) = \eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\hat{z}_i = d_{\ell_i})$. We note that estimating the conditional label probability given each individual $\hat{z}_i$ can be challenging.



(a) CIFAR-10 (b) CIFAR-100

Fig. 4. Classification accuracy vs rate results for end-to-end compression and classification on (a) CIFAR-10 and (b) CIFAR-100 data sets. The proposed VAE based compression and classification framework outperforms popular commercial image compression codecs in terms of rate-accuracy, at lower bandwidths.

On the other hand, given $\hat{\mathbf{z}}$, we can efficiently predict the class label using a trained CNN classifier such as ResNet [47]. Hence, we can easily estimate $-\log \eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\hat{\mathbf{z}})$ given the entire sequence of $\hat{\mathbf{z}}$ rather than for one individual element $\hat{z}_i$ therein. In standard stochastic gradient optimizers, we have only one sequence sample $\hat{\mathbf{z}}$ at a time. Hence, the stochastic expectation of $E_{q_{\hat{\mathbf{z}}|\boldsymbol{x},\boldsymbol{\phi}}}[-\log \eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\hat{\mathbf{z}})]$ can be approximated simply by $-\log \boldsymbol{\eta}_{\boldsymbol{\theta}}(y_{\text{gt}}|\hat{\mathbf{z}})$ [37]. We can reduce this approximation error by iterating sufficiently over same $\boldsymbol{x}$. This simple approximation leads to general classification loss,

$$\begin{aligned}\mathcal{L}_{\text{CL}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x}, y_{\text{gt}}) &= E_{q_{\hat{\mathbf{z}}|\boldsymbol{x},\boldsymbol{\phi}}}[-\log \eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\hat{\mathbf{z}})] \\ &\approx -\log \eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\hat{\mathbf{z}}). \end{aligned} \tag{29}$$

With this formulation, we express the total loss $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}, y_{\text{gt}})$ of the joint classification and compression model for image sample $\boldsymbol{x}$ as,

$$\begin{aligned}\mathcal{L}_{\theta,\boldsymbol{\phi}}(\boldsymbol{x}, y_{\text{gt}}) &= \mathcal{L}_{\text{CL}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x}, y_{\text{gt}}) + \beta \mathcal{L}_{\text{R}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x}) \\ &\approx -\log \eta_{\boldsymbol{\theta}}(y_{\text{gt}}|\hat{\mathbf{z}}) \\ &\quad - \beta \sum_{i=1}^{m} \sum_{\ell_i=1}^{L} q_{\boldsymbol{\phi}}(\hat{z}_{i,\ell_i}|\boldsymbol{x}) \log P_{i,\ell_i}(\hat{\mathbf{z}}|\boldsymbol{x}). \end{aligned} \tag{30}$$

Similar to the rate-distortion loss in reconstruction [27], [38], [48], β controls the rate versus mis-classification trade-off.

IV. Experiments and Results

This section presents several experiment results of classification using the proposed VAE approach of Sec. III. We tested the accuracy of classification on three well-known data sets: CIFAR-10 [49], CIFAR-100 [50] and ImageNet-1k [51]. CIFAR-10 consists of 10 classes with 5000 images for training and 1000 images for testing per class. CIFAR-10 images are in RGB format of size 32×32. Similar to CIFAR-10, CIFAR-100 data set contains 50k training and 10k test RGB images of size 32×32 in 100 classes. ImageNet-1k consists of 1000 classes, each of which consists of up to 1300 training images and 50 validation images of mostly 256×256 RGB.

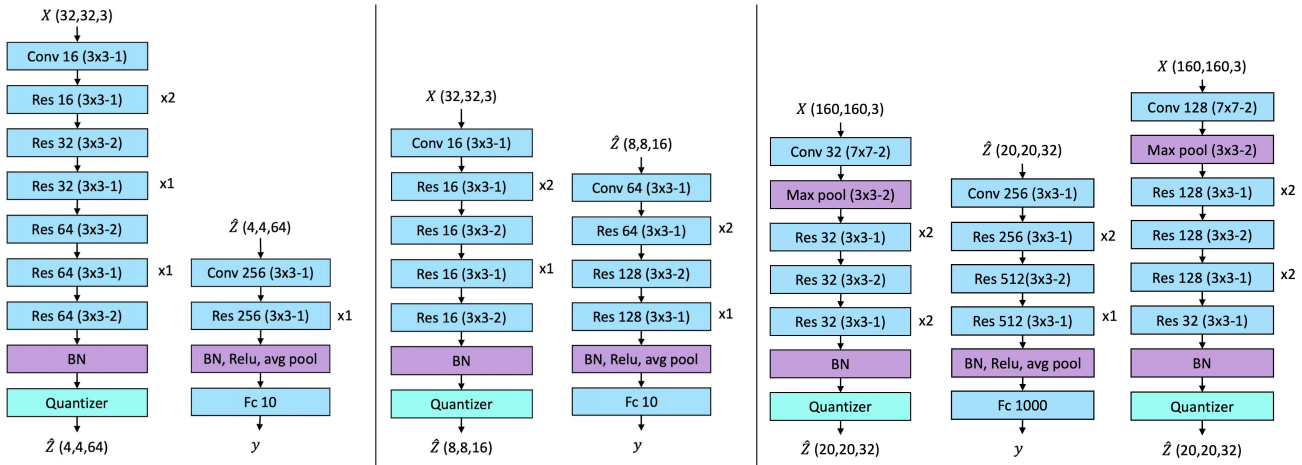
Our experiments used the same quantizer and the context model as [27] to estimate probabilities. Instead of separately training the encoder-decoder and the context model (**PE**) as in [27], the training of **PE** takes place simultaneously with encoder and classifier training.

A. VAE-based Joint Compression and Classification models

When designing the joint model, we began with a ResNet [47] and split into 2 parts as Encoder (**E**) and classifier (**CL**). Next, we added a batch normalization layer and a quantizer block to the encoder and further modified the number of filters and strides. For each CIFAR-10, CIFAR-100 and ImageNet-1k data sets, we considered 2 different encoders-classifier combinations. Figs. 5(a)-(g) depict the structural details of 4 proposed VAE models labeled as AE-V_1, AE-V_2, AE-V_3 and AE-V_4. We use typical notations in CNNs to denote their components. For example, "Conv 16 (3×3-2)" represents a 2D convolution block with 32 filters of size 3 × 3 and a stride of 2. "Res 16 (3×3-2)" denotes a basic ResNet block with a down-sampling factor of 2.

To compare the performance gains of the proposed approach, we mainly used JPEG (4:2:0)[1] as the benchmark technique to compress images within the data sets at different quality (Q) values. One can vary the quality value Q ∈ [1, 100] during compression to set image quality. We then trained a ResNet classifier based on these compressed images for each Q value. To calculate the required channel bandwidth (i.e., image size), we averaged the bits-per-pixel (bpp) for each coded image over the data set. For fair comparison, we only consider the data bits of the images according to bpp without counting packet headers. Following similar steps, we generated rate-accuracy performance for popular standard codecs: JPEG2000[2], WebP[3] and BPG[4], to be used as baselines.

[1] http://www.openjpeg.org/
[2] https://kakadusoftware.com/
[3] https://developers.google.com/speed/webp/download
[4] https://bellard.org/bpg/



(a) AE-V$_1$ Encoder (b) AE-V$_1$ Classifier (c) AE-V$_2$ Encoder (d) AE-V$_2$ Classifier (e) AE-V$_3$ Encoder (f) AE-V$_3$/V$_4$ Classifier (g) AE-V$_4$ Encoder

Fig. 5. Encoders and classifiers for the proposed end-to-end compression and classification models AE-V$_1$, AE-V$_2$, AE-V$_3$ and AE-V$_4$. "Conv 16 (3x3-2)" represents a 2D convolution block with 32 filters of size 3×3 and stride of 2. "Res 16 (3x3-2)" represents a basic ResNet block [47] with down-sampling factor 2.

In the joint compression and classification model, we vary the number of quantization centers (L) to compress images at different BPP values without changing the **E** and **CL** model architectures. As proposed in [27], one can alter the encoder-decoder/classifier architecture to generate latent of different dimensions to obtain the requisite bpp as well.

We selected $\beta = 2$ for all experiments. During model training, we used a 'Momentum' Optimizer for the **E-CL** part of the network and an 'Adam' optimizer for the **PE** with a learning rate of 0.0005 and reduced the learning rate by $\times 0.1$ at 0.25 and 0.75 of total epochs over 90 epochs. In contrast to [27], we concurrently train both **E-CL** and **PE** to also enable back-propagation from **PE** to **E**. This configuration facilitates model convergence at lower bpp.

B. CIFAR-10 Experiments

For CIFAR-10, we used the models AE-V$_1$ and AE-V$_2$ of different complexities. Figs. 5(a)-(d) provide the structural details of both VAE models. As the baseline classification model, we used a ResNet-18. Fig. 4(a) shows the result for CIFAR-10 data set with different combinations of **E-CL** in comparison with popular standard codecs: JPEG, JPEG2000, WebP and BPG. Both AE-V$_1$ and AE-V$_2$ achieve substantial performance improvement over the traditional JPEG (4:2:0) in terms of rate (bpp) and accuracy trade-off. At 0.8 bit/pixel, our joint compression-classification accuracy improves from 77% to 87%. More importantly, this performance improvement is achieved with significantly lighter-weight models, resulting impressive inference speed improvements and power savings compared to the baseline classifiers as will be discussed in Sec. V. Further, compared to JPEG2000, WebP and BPG, the proposed VAE-based codecs maintain the rate-accuracy performance specially at lower bandwidth end. In particular, both VAE-based models demonstrate over 5% accuracy improvement at 0.9 bpp compared to BPG.

C. CIFAR-100 Experiments

As the baseline classification model, we used a ResNet-18 model with a fully connected layer of 100 neurons instead of 10 in the original model designed for CIFAR-10. For the VAE models for classification, we used AE-V$_1$ and AE-V$_2$, each with same encoder (shown in Figs. 5(a), (c)) and the Classifier (Figs. 5(b), (d)) with a fully connected layer of 100 neurons (Fc-100) instead of Fc-10. For training, we follow the same settings as in CIFAR-10. Fig. 4(b) compared the rate-accuracy performance of AE-V$_1$ and AE-V$_2$ models for CIFAR-100 validation set. The proposed method achieves similar improvement as observed in CIFAR-10.

D. ImageNet Experiments

For ImageNet experiments, we followed the same process as used for CIFAR-10 to generate the JPEG (4:2:0) baseline. As a part of data augmentation, we resized the images to 256×256

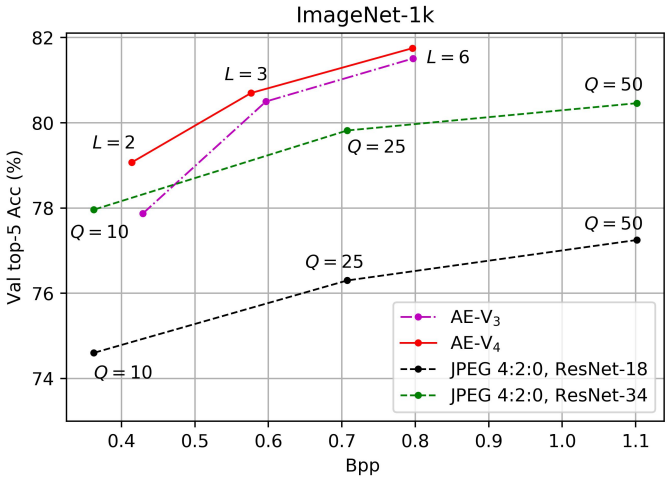


Fig. 6. Classification accuracy vs rate on ImageNet-1k for end-to-end compression and classification. The proposed VAE based compression and classification framework (AE-V$_4$) significantly outperforms JPEG commercial image compression codecs in terms of rate-accuracy.

and cropped them down to 160×160 in training and testing to obtain the desired dimensions of the latent $\hat{\mathbf{z}}$. We propose modified VAE models AE-V$_3$ and AE-V$_4$ for ImageNet. See Figs. 5(e)-(g) for model architectures.

For a fair performance comparison, we fine-tuned the baseline ResNet models with the JPEG encoded images at each quality level (Q) to record the best accuracy achievable at given Q as benchmark results. For the same reason, we do not include packet header in bpp calculation.

Testing the **E-CL** combinations AE-V$_3$ and AE-V$_4$ on ImageNet generates the results of Fig. 6. The performance of the two shallower VAEs is compared against the benchmark ResNet-18 and ResNet-34 operated on JPEG encoding. Our results demonstrate clear performance improvement in terms of rate-accuracy trade-off, without increasing complexity as discussed in Sec. V. In fact, the proposed VAE delivers comparable performance as conventional ResNet-34 that requires 21.8M parameters which is twice as many compared to AE-V$_3$ and AE-V$_4$ models.

V. Complexity Comparison

Simpler encoder models are practically more favorable for wide deployment on low cost devices, which are severely constrained in memory, power, and computation capacity. In this section, we compare the complexity of the proposed models in terms of model size (number of parameters), inference speed and power consumption to the baseline models.

A. Model size comparison

Considering practical constraints in low cost sensors, our encoder design has significantly fewer number of parameters in comparison with the typical cloud-based classifier. See Table II for a complexity comparison in terms of number of parameters for the VAE models used in CIFAR-10 and ImageNet-1k experiments. Percentages provided in the table indicates the reduction of the number of parameters compared to the baseline classifier.

TABLE II
Complexity comparison of the models for CIFAR-10 and ImageNet. All parameters are in millions (M). The percentage reduction is calculated compared to the baseline.

Data set	Model	Encoder	Classifier	Baseline
CIFAR-10	AE-V_1	0.245 (**65%**↓)	0.372 (**47%**↓)	0.704 ResNet-18
	AE-V_2	0.025 (**96%**↓)	0.687 (**2%**↓)	
ImageNet-1k	AE-V_3	0.100 (**99%**↓)	11.35 (**3%**↓)	11.7 ResNet-18
	AE-V_4	1.550 (**87%**↓)	11.35 (**3%**↓)	
ImageNet-1k	AE-V_3	0.100 (**99%**↓)	11.35 (**48%**↓)	21.8 ResNet-34
	AE-V_4	1.550 (**99%**↓)	11.35 (**48%**↓)	

All the proposed encoder and classifier models show parameters savings compared to their corresponding baseline models. The achieved parameter savings range from 65%−99% for encoders and 2%−48% for classifiers. Such balance of complexity reductions are highly practical for wide IoT deployment since classifiers at the server side are equipped with more resources compared to the source encoders at the embedded devices.

TABLE III
Speed comparison in terms of "ips". The inference results are based on a NVIDIA Titan-V GPU. The proposed VAE-based classifier is ×1.5 and ×2.25 faster compared to ResNet-18 and ResNet-34 baselines.

Model	Encode (**E**)	Classifier (**CL**)	**E-CL**	Baseline **CL**
AE-V_1	1702	20283 (×**13.1**↑)	1570(↑)	1547 ResNet-18
AE-V_2	2114	6524 (×**4.2**↑)	1596(↑)	
AE-V_3	4691	5122 (×**1.5**↑)	2448(↓)	3330 ResNet-18
AE-V_4	2417		1642(↓)	
AE-V_3	4691	5122 (×**2.25**↑)	2448(↑)	2268 ResNet-34
AE-V_4	2417		1642(↓)	

1) Task vs Model size: When comparing the model complexity versus task difficulty, more challenging classification tasks often require more complex encoders. Note that both models AE-V$_3$ and AE-V$_4$ use the same network architecture as the classifier for ImageNet data set. Furthermore, from Table II, it is clear that the proposed VAE classifiers of AE-V$_3$ and AE-V$_4$ have the same complexity in terms of the number of parameters while AE-V$_4$ model uses a relatively more complex encoder than AE-V$_3$. Since complex encoders with more parameters are capable of extracting more discriminate features, Fig. 6 shows AE-V$_4$ with higher rate-classification performance than AE-V$_3$, especially when handling more challenging tasks such as ImageNet classification with 1000 classes.

We observe similar trend in rate-classification in Fig. 4(b) for CIFAR-100 data set with 100 classes: A relatively increase of encoder complexity may achieve better performance than a simple encoder through feature extraction for classification. Note in Table II that the model complexity in number of parameters is ×10 smaller for AE-V$_2$ when compared to AE-V$_1$. However, for relatively simple classification task such as CIFAR-10 with only 10 classes, a simpler encoder used in AE-V$_2$ is stronger to preserve enough features to generate the latent vector. In fact, even in such case, having a relatively complex classifier with more parameters to process the extracted features has some benefit, although the resulting rate-classification performance improvement is rather modest.

B. Inference speed comparison

We also observe considerable reduction in inference time by directly classifying on the latent/feature maps without reconstruction. In Table III we list the inference speed in terms of average images per second (ips) for CIFAR-10 test set and ImageNet-1k validation set.

The proposed end-to-end-trained AE-V_1 and AE-V_2 classifiers record ×13.1 and ×4.2 speed gains on CIFAR-10 data set, respectively, compared to the ResNet-18 classifier. Image patches used at inference have the size of 32×32. Similarly, our proposed classifier is ×1.5 faster in comparison with the ResNet-18 baseline and ×2.25 faster in comparison with ResNet-34 baseline, on ImageNet-1k data set where image

TABLE IV
NUMBER OF FLOPS ($\times 10^9$) COMPARISON FOR THE PROPOSED MODELS.

Model	Encoder	Classifier	Baseline
Input: 32×32×3			
AE-V_1	0.037 (**93%**↓)	0.005 (**99%**↓)	0.557
AE-V_2	0.013 (**97%**↓)	0.013 (**97%**↓)	ResNet-18
Input: 160×160×3			
AE-V_3	0.107 (**88%**↓)	1.229 (**32%**↑)	0.928
AE-V_4	1.399 (**50%**↑)	1.229 (**32%**↑)	ResNet-18
Input: 160×160×3			
AE-V_3	0.107 (**94%**↓)	1.229 (**34%**↓)	1.873
AE-V_4	1.399 (**25%**↓)	1.229 (**34%**↓)	ResNet-34

patches have the size of 160×160. Since the inference speed of the encoder for the baseline can vary based on the codec (JPEG, BPG etc.), we only highlight the inference speed gains of the classifiers (**CL**) for fairness. Moreover, due to the simplicity of the proposed AE-V_1 and AE-V_2 encoders, the inference speed including the encoding time (**E-CL** column) surpasses the baseline CL speed calculated even without the encoding time.

C. Power savings comparison

When measured on the same device (GPU), the number of floating point operations per second (FLOPS) is directly proportional to the power consumption. The proposed encoders are not only less complex in terms of number of parameters, but also demand significantly lower number of arithmetic operations which is an essential feature for IoT, mobile and wireless applications [52]. In Table IV we list the required computational operations in floating point operations per second (FLOPs) for the proposed encoders. We indicate the power savings as a percentage compared to the baseline.

The proposed encoders and classifiers of AE-V_1 and AE-V_2 models achieve power savings in the range of 93%−97% compared to ResNet-18 classifier. AE-V_3 and AE-V_4 models with larger input image sizes show less power savings in the classifier compared to the encoder due to the high number of parameters. However, this low power savings are compensated by the resource rich servers at the cloud. More importantly, the proposed encoders display significant power savings that enable wide deployment of power constrained low-end source embedded devices. Modern smart phones are capable of providing over 10×10^9 FLOPs [52], [53].

VI. DISCUSSION

A. Joint Compression and Classification with Simultaneous Reconstruction

Thus far, our proposed joint compression and classification model allows the classifier to operate directly in latent space. We bypass the image reconstruction step unlike the image/video systems [13]–[15], [20], [41] that perform classification on reconstructed data. In practical systems, however, there also exist several application scenarios which may require both autonomous image classification and image reconstruction for users. For example, RGB images may need to be stored on the cloud to be retrieved later after successful classification. These reconstructed and stored RGB images may be required for other subsequent learning tasks such as object detection and segmentation trained on RGB images. To accommodate such potential dual compression objectives of simultaneous classification and reconstruction, we further modify the current VAE to incorporate a parallel decoder at the remote node (or cloud).

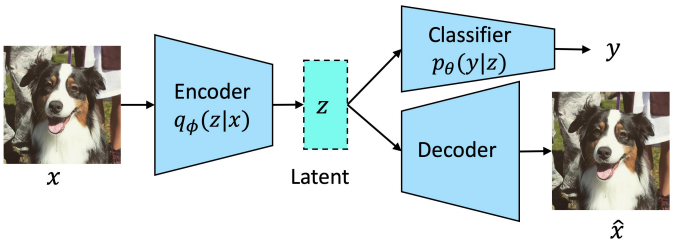


Fig. 7. Proposed joint classification and compression model with reconstruction. Encoder transforms input $\boldsymbol{x}$ to a latent vector $\boldsymbol{z}$ that is optimized for rate-classification-distortion performance.

As shown in Fig. 7, we include a basic DL-based image decoder (**D**) to the proposed joint compression and classification model. This framework of joint compression and classification model with reconstruction (**E-CL-D**) can be trained similarly to the proposed VAE. In consideration of the reconstruction accuracy, we construct a modified loss function of

$$\mathcal{L} = \mathcal{L}_{\mathrm{C}} + \beta\mathcal{L}_{\mathrm{R}} + \gamma\mathcal{L}_{\mathrm{D}}. \tag{31}$$

We use the mean square error (MSE) between $\mathbf{x}$ and $\hat{\mathbf{x}}$ as the distortion loss $\mathcal{L}_{\mathrm{D}}$ and empirically choose a control parameter γ smaller than β to favor high classification accuracy over rate-distortion.

In our experiment, we applied a deep learning decoder similar to the architecture of [27] with modified numbers of convolution kernels to reduce the number of parameters. The DNN decoder has 0.308 M parameters which is only 44% of the ResNet-18 baseline model. We fine-tuned the training of the proposed joint architecture with the modified loss. Fig. 8 shows the classification accuracy for **E-CL-D** model for CIFAR-10. Note that the classification accuracy loss due to reconstruction is smaller compared to the rate-accuracy gain from standard JPEG, JPEG2000, WebP and BPG codecs. For joint compression classification and reconstruction experiments on ImageNet-1k, we would freeze the Encoder-Classifier (**E-CL**) and **PE** parameters optimized for joint compression and classification as described in Eq. (30) for AE-V_3 model and re-trained only the decoder to minimize

TABLE V
RECONSTRUCTION QUALITY FOR CIFAR-10 AND IMAGENET-1K. L IS THE NUMBER OF QUANTIZATION CENTERS.

Data set	L=2	L=3	L=6
		PSNR (dB)	
CIFAR-10	19.24	21.42	22.84
ImageNet-1k	20.18	21.68	22.22
		MS-SSIM	
ImageNet-1k	81.04	86.73	88.35

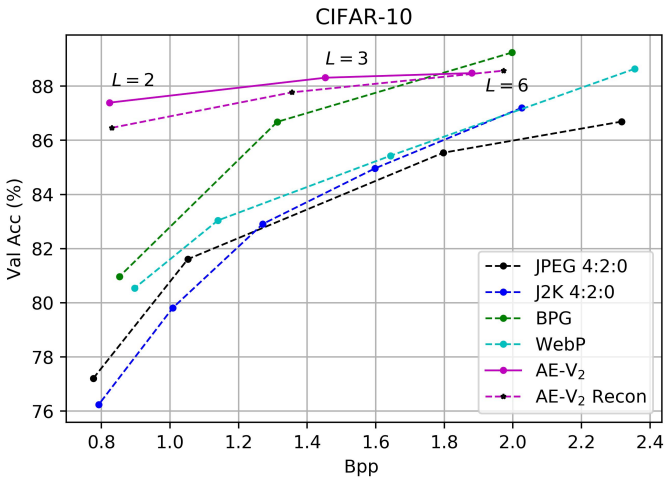
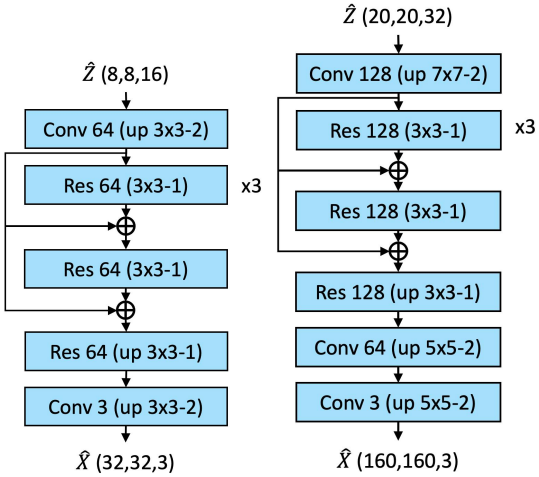


Fig. 8. Classification accuracy vs rate on CIFAR-10 for joint compression-classification with reconstruction: AE-V_2-Recon. Note that rate-accuracy performance of AE-V_2-Recon is still sufficiently good compared to popular commercial codecs, at lower rates.

the MSE between $\mathbf{x}$ and $\hat{\mathbf{x}}$. A frozen **E-CL** and **PE** guarantees the same rate-classification performance as in Fig. 6 with reconstruct-ability. Table V lists reconstruction PSNR and MS-SSIM values for the two test data sets at three bpp settings. The observed lower image quality, observed for the images reconstructed from the rate-accuracy optimized latents, is consistent with the reconstruction quality recorded in [20] for COCO-2017 [54] data set when the codec is optimized for joint compression and object detection.

B. Visualization of reconstructed images

As the decoder models, we used modified architectures of the decoders proposed in [27]. See Fig. 9. 'up' indicated the up-sampling with transposed convolution. Fig. 10(a) and Fig. 10(b) show some of the reconstructed images from 'airplane' class of CIFAR-10 [49] and ImageNet-1k [51]. The reconstructed results look blurry compared to the original images since the MSE loss is minimized by the average of the image. We observe significantly higher distortion when only 2



(a) AE-V_2 Decoder (b) AE-V_3 Decoder

Fig. 9. Decoder design for AE-V_2 and AE-V_3. 'up' indicates up-sampling with transposed convolutions.

quantization levels are used during compression even though the classification accuracy is less compromised (in Fig. 8) as required.

C. Robustness to visual corruptions

In this section we demonstrate the robustness of the proposed model against common visual corruptions of images based on CIFAR-10-C [55] data set. CIFAR-10-C data set provides corrupted versions of CIFAR-10 [49] test data set for visual corruptions such as noise, blur, pixelation etc. at 5 different severity levels of each corruption.

We selected the proposed AE-V_2 VAE model with $L = 6$ trained on CIFAR-10 data as discussed in Sec. IV-A during inference on CIFAR-10-C test data set. For 9 types of corruptions namely **Fog, Motion Blur, Defocus Blur, Frost, Pixelation, Elastic transformation, Impulse noise, Gaussian noise and Shot noise**, we repeated the above inference for classification at 5 severity levels. Figures 11(a)-(c) show the results. 'RGB' baseline corresponds to the inference accuracy with ResNet-20 classifier trained on CIFAR-10 training set.

We note that AE-V_2 VAE classifier outperforms ResNet-20 baseline classifier at inference on images corrupted with fog, motion blur and defocus blur at 3-5 severity levels by a significant margin. Further the proposed model performs slightly worse than the baseline classifier on images corrupted with impulse, Gaussian and shot noise.

D. Visualization of latent maps

Figs. 12(a), (b) and (c) show the latent maps $\hat{\mathbf{z}}$ of a sample image $\mathbf{x}$ for $L = 6, 3, 2$ related to the model AE-V_3. For each sub figure, top-left color image is the original and bottom left color image is the reconstructed from the latent maps visualized next to them. We visualize 32 latent maps (each of size 20×20) before ($\mathbf{z}$) and after quantization maps ($\hat{\mathbf{z}}$). Starting from the first row, the $\mathbf{z}$ maps are given in every other row. The corresponding quantized $\hat{\mathbf{z}}$ is given right below each $\mathbf{z}$ map. Each map is normalized before visualization.

Observe that for each number of quantization centers L, the quantized maps $\hat{\mathbf{z}}$ has only L different colors. Some of the $\hat{\mathbf{z}}$ maps for $L = 6$ show clearly identifiable snow leopard figures implying that high level information of $\mathbf{x}$ is preserved during training and helpful for classification.

E. Implementation details: Effect of β adjustment

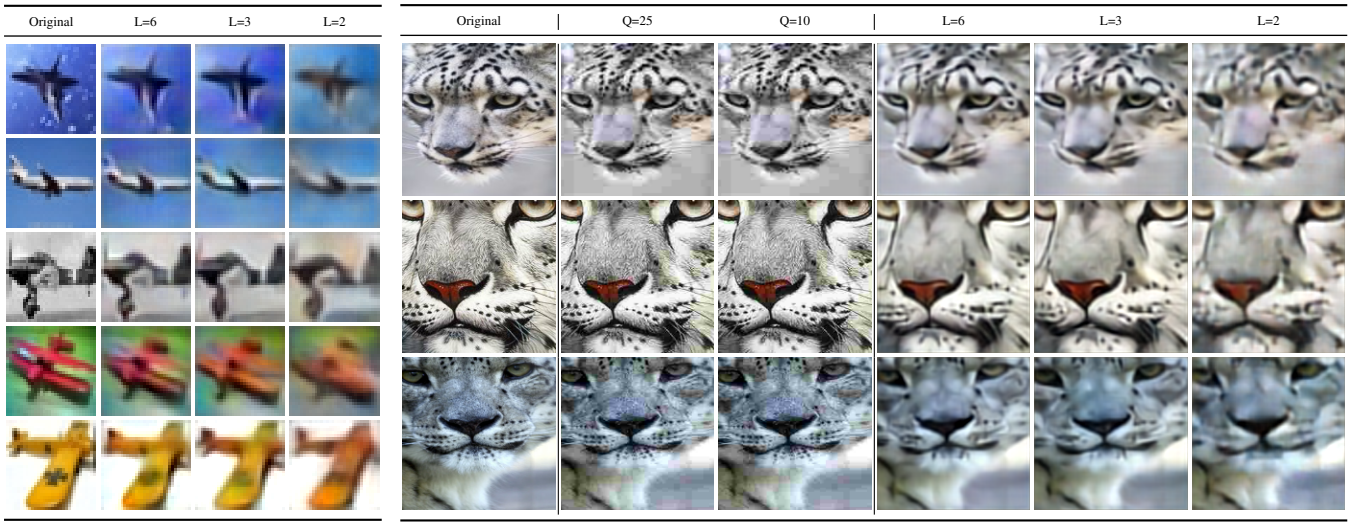
Recall that the VAE loss of the proposed method for a sample $\boldsymbol{x}$ is given by,

$$\mathcal{L}_{\theta,\phi}(\boldsymbol{x}, y_{\text{gt}}) = \mathcal{L}_{\text{C}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x}, y_{\text{gt}}) + \beta \mathcal{L}_{\text{R}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x}) \quad (32)$$

where $\mathcal{L}_{\text{R}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x})$ is the bandwidth of $\mathbf{z}$ transmitted over a network link. We measure the bandwidth (rate) in BPP in our experiments. The effect of the trade-off parameters β for CIFAR-10 data set is shown in Fig. 13(a).

In order to achieve optimal rate-classification accuracy performance at a given rate (r_t), we minimize the following loss instead of the loss given in Eq. (32).

$$\mathcal{L}_{\theta,\phi}(\boldsymbol{x}, y_{\text{gt}}) = \mathcal{L}_{\text{C}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x}, y_{\text{gt}}) + \beta \max\left(\mathcal{L}_{\text{R}(\boldsymbol{\theta},\boldsymbol{\phi})}(\hat{\mathbf{z}}|\boldsymbol{x}) - r_t, 0\right) \quad (33)$$



(a) CIFAR-10 (b) ImageNet-1k

Fig. 10. Examples of (a) CIFAR-10 and (b) ImageNet-1k reconstructed images from the latent space at L =6, 3 and 2. In (b), Q=25 and Q=10 show JPEG compressed images. We observe significantly higher distortion when only 2 quantization levels are used during compression.

We observed better classification accuracy by this approach at a rate slightly above the requirement r_t. In Fig. 13(b) we illustrate the loss curves for a selected β and an r_t value.

VII. Further Theoretical Explanations

A. Relationship to the Information Bottleneck

In this section, we theoretically derive the relationship between the proposed VAE framework for classification and the well-known Information Bottleneck principle [29], [30]. Consider the random variables $\mathbf{Y}$, $\mathbf{x}$ and $\mathbf{z}$ related according to a Markov chain as follows.

$$\mathbf{Y} \rightarrow \mathbf{x} \xrightarrow[\boldsymbol{\phi}]{\mathrm{E}} \mathbf{z} \tag{34}$$

In the context of image classification with auto-encoders $\mathbf{Y}$, $\mathbf{x}$ and $\mathbf{z}$ can be viewed as variables corresponding to image label, image and latent encoding of the image respectively. $\mathbf{z}$ is mapped from $\mathbf{x}$ with an Encoder ($\mathbf{E}$) parameterized by $\boldsymbol{\phi}$. Hence, we can denote $\mathbf{z}$ as $\mathbf{z}(\boldsymbol{\phi})$, which is often written as $\mathbf{z}$ for simplicity.

We can write the objective function of the Information Bottleneck (IB) principle [29], [30] as the following.

$$\max_{\boldsymbol{\phi}} \mathrm{IB}_{\mathbf{x},\mathbf{z},\mathbf{Y}}(\boldsymbol{\phi}) = I(\mathbf{z}(\boldsymbol{\phi}); \mathbf{Y}) - \beta I(\mathbf{x}; \mathbf{z}(\boldsymbol{\phi})), \quad \beta \geq 0 \tag{35}$$

$I(\mathbf{x}; \mathbf{z})$ is the Mutual Information (MI) between the random variables $\mathbf{x}$ and $\mathbf{z}$. We rewrite this as a minimization objective to match the above VAE narrative.

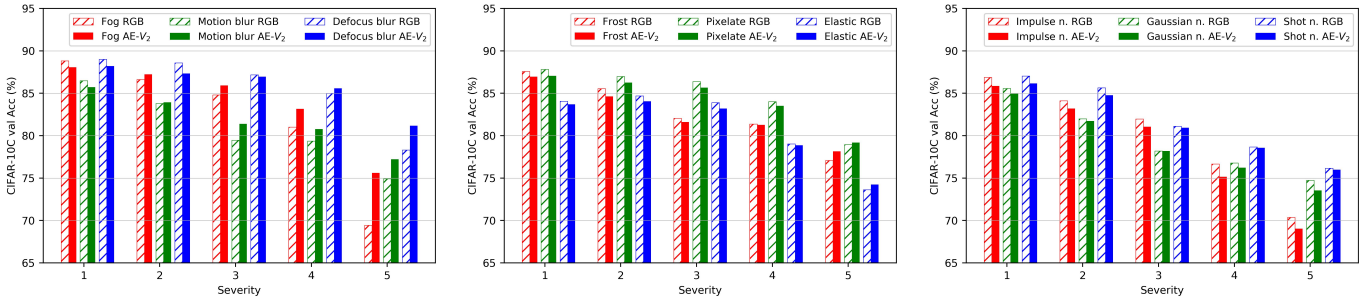
$$\min_{\boldsymbol{\phi}} -I(\mathbf{z}(\boldsymbol{\phi}); \mathbf{Y}) + \beta I(\mathbf{x}; \mathbf{z}(\boldsymbol{\phi})), \quad \beta \geq 0 \tag{36}$$

Following the definition of MI we write,

$$I(\mathbf{z}; \mathbf{Y}) = \int p(y, \boldsymbol{z}) \log \frac{p(y|\boldsymbol{z})}{p(y)} dy \; d\boldsymbol{z}. \tag{37}$$

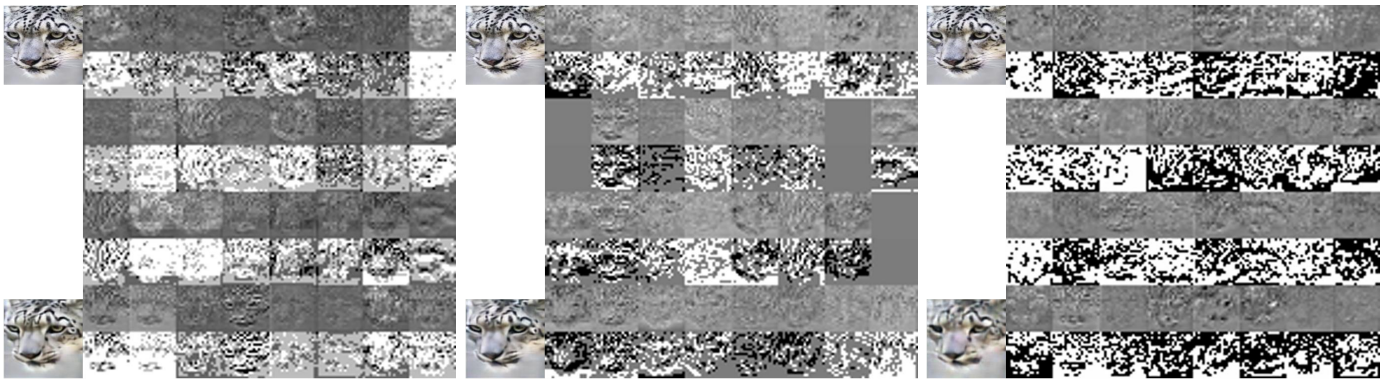
In order to track the distribution $p(y|\mathbf{z})$ we employ a classifier ($\mathbf{CL}$) parameterized by $\boldsymbol{\theta}$ and estimate the conditional label distribution $\eta_{\boldsymbol{\theta}}(y|\mathbf{z})$ as in Eq. (14). Since $\mathrm{KL}(p(\mathbf{Y}|\mathbf{z})|\eta_{\boldsymbol{\theta}}(\mathbf{Y}|\mathbf{z})) \geq 0$ we can write the inequality,

$$\int p(y|\boldsymbol{z}) \log p(y|\boldsymbol{z}) dy \geq \int p(y|\boldsymbol{z}) \log \eta_{\boldsymbol{\theta}}(y|\boldsymbol{z}) dy. \tag{38}$$



(a) fog, motion blur and defocus blur (b) frost, pixelation and elastic transformation (c) impulse noise, Gaussian noise and shot noise

Fig. 11. Robustness comparison for visual corruptions: fog, motion blur, defocus blur, frost, pixelation, elastic transformation, impulse noise, Gaussian noise, shot noise on CIFAR-10-C data set.



(a) $L = 6$ (b) $L = 3$ (c) $L = 2$

Fig. 12. Latent maps $\hat{\mathbf{z}}$ of a sample image $\mathbf{x}$ for $L = 6$, 3 and 2 related to the model AE-V$_3$. For each sub figure, top-left color image is the original and bottom left color image is the reconstructed from the latent maps visualized next to them. Some of the $\hat{\mathbf{z}}$ maps for $L = 6$ show clearly identifiable snow leopard figures implying that high level information of $\mathbf{x}$ is preserved during training and helpful for classification.

We utilize this inequality to bound for $I(\mathbf{z};\mathbf{Y})$ in Eq. (37).

$$\begin{aligned} I(\mathbf{z};\mathbf{Y}) &= \int p(y,\boldsymbol{z}) \log p(y|\boldsymbol{z}) dy \; d\boldsymbol{z} \; - \int p(y) \log p(y) dy \\ &\geq \int p(y,\boldsymbol{z}) \log \eta_{\boldsymbol{\theta}}(y|\boldsymbol{z}) dy \; d\boldsymbol{z} \; + \mathrm{H}(\mathbf{Y}) \end{aligned} \tag{39}$$

Leveraging the Markov assumption in Eq. (34) , $p(y,\boldsymbol{z},\mathbf{x}) = p(\mathbf{x},y)q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ with the encoder distribution $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, we can express the bound in Eq. (39) for $I(\mathbf{z};\mathbf{Y})$ as the following.

$$\begin{aligned} I(\mathbf{z};\mathbf{Y}) &\geq \int p(y,\boldsymbol{z},\boldsymbol{x}) \log \eta_{\boldsymbol{\theta}}(y|\boldsymbol{z}) d\boldsymbol{x} \; dy \; d\boldsymbol{z} + \mathrm{H}(\mathbf{Y}) \\ &\geq \int p(\boldsymbol{x},y) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \eta_{\boldsymbol{\theta}}(y|\boldsymbol{z}) d\boldsymbol{x} \; dy \; d\boldsymbol{z} \\ &\quad + \mathrm{H}(\mathbf{Y}) \end{aligned} \tag{40}$$

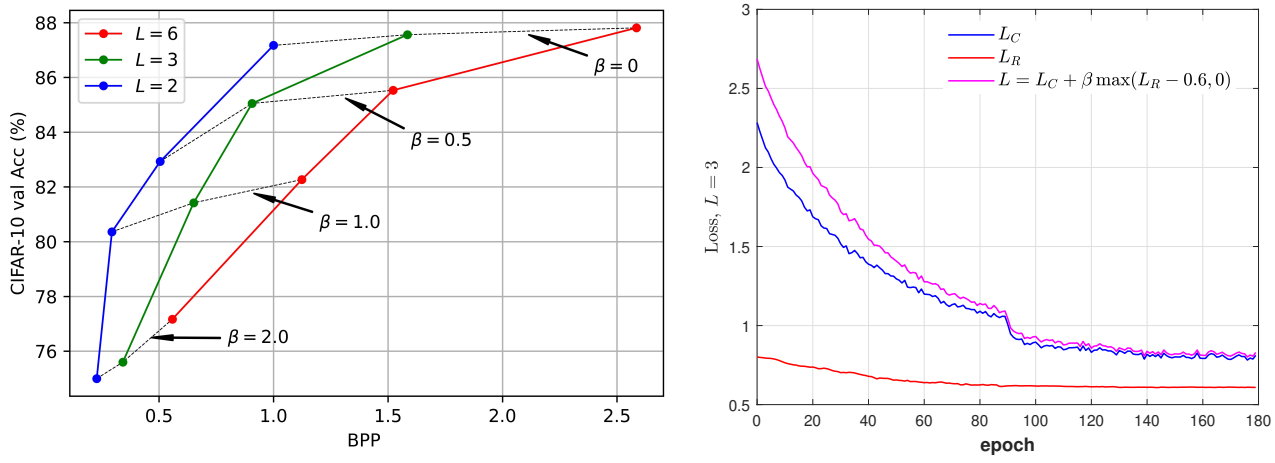
For the second term of the IB objective given in Eq. (36), following the definition of MI, we write,

$$\begin{aligned} I(\mathbf{x};\mathbf{z}) &= \int q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z}) \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z} \; d\boldsymbol{x} \\ &\quad - \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) \log q_{\boldsymbol{\phi}}(\boldsymbol{z}) d\boldsymbol{z}. \end{aligned} \tag{41}$$

With the definition of KL divergence, for an arbitrary $r(\mathbf{z})$, we write the following inequality similar to Eq. (38).

$$\int q_{\boldsymbol{\phi}}(\boldsymbol{z}) \log q_{\boldsymbol{\phi}}(\boldsymbol{z}) d\boldsymbol{z} \geq \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) \log r(\boldsymbol{z}) d\boldsymbol{z}. \tag{42}$$

We combine Eq. (41) and (38) to obtain an upper bound for



(a) Effect of β (b) Convergence of losses

Fig. 13. (a). CIFAR-10 results for end-to-end compression and classification for different β values. Black dash lines connect the points for AE models with the same β value. (b). CIFAR-10 average validation losses obtained with Eq. (33) at $\beta = 2$ and $r_t = 0.6$ for AE-V$_1$, $L = 3$ model. Note the smooth convergence of the rate L_{CL} around 0.6 at higher number of epochs.

$I(\mathbf{x};\mathbf{z})$.

$$\begin{aligned} I(\mathbf{x};\mathbf{z}) &\leq \int q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z}) \log q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) d\boldsymbol{z}\ d\boldsymbol{x} \\ &\quad - \int q_{\boldsymbol{\phi}}(\boldsymbol{z}) \log r(\boldsymbol{z}) d\boldsymbol{z} \\ &= \int q_{\boldsymbol{\phi}}(\boldsymbol{x},\boldsymbol{z}) \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{r(\boldsymbol{z})} d\boldsymbol{z}\ d\boldsymbol{x} \\ &= \int p(\boldsymbol{x}) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{r(\boldsymbol{z})} d\boldsymbol{z}\ d\boldsymbol{x} \quad (43) \\ &= \int p(\boldsymbol{x},y) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{r(\boldsymbol{z})} d\boldsymbol{x}\ dy\ d\boldsymbol{z} \quad (44) \end{aligned}$$

Eq. (44) follows from Eq. (43) since $p(\boldsymbol{x}) = \int p(\boldsymbol{x},y)dy$. Similar to [56], we approximate $p(\boldsymbol{x},y)$ with an empirical distribution based on data samples $(\boldsymbol{x}_n, y_n) \in \mathcal{S}$ s.t.,

$$p(\boldsymbol{x},y) = \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \delta_{\boldsymbol{x}_n}(\boldsymbol{x}) \delta_{y_n}(y). \quad (45)$$

Hence the approximated IB loss for minimization can be written as the following.

$$\begin{aligned} &\mathrm{IB}_{\mathbf{x},\mathbf{z},\mathbf{Y}}(\boldsymbol{\phi},\boldsymbol{\theta}) \\ &\leq -\int p(\boldsymbol{x},y) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \eta_{\boldsymbol{\theta}}(y|\boldsymbol{z}) d\boldsymbol{x}\ dy\ d\boldsymbol{z} \\ &\quad - \mathrm{H}(\mathbf{Y}) \\ &\quad + \beta \int p(\boldsymbol{x},y) q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}) \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x})}{r(\boldsymbol{z})} d\boldsymbol{x}\ dy\ d\boldsymbol{z} \quad (46) \\ &\approx \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \{ -\int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n) \log \eta_{\boldsymbol{\theta}}(y_n|\boldsymbol{z}) d\boldsymbol{z} \\ &\quad + \beta \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n) \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n)}{r(\boldsymbol{z})} d\boldsymbol{z} \} - \mathrm{H}(\mathbf{Y}) \quad (47) \\ &= \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \{ -\int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n) \log \eta_{\boldsymbol{\theta}}(y_n|\boldsymbol{z}) d\boldsymbol{z} \\ &\quad + \beta \int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n) \log \frac{q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n)}{p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_n)} d\boldsymbol{z} \} - \mathrm{H}(\mathbf{Y}) \quad (48) \end{aligned}$$

From Eq. (47) to Eq. (48), we replace the arbitrary $r(\boldsymbol{z})$ with $p_{\boldsymbol{\theta}}(\boldsymbol{z}|\boldsymbol{x}_n)$.

Using the definition of KL divergence, we re-write Eq. (48) as the following which is upper bounded by the β-VAE loss $\mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}$ for classification.

$$\begin{aligned} &\mathrm{IB}_{\mathbf{x},\mathbf{z},\mathbf{Y}}(\boldsymbol{\phi},\boldsymbol{\theta}) \\ &\leq \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \{ -\int q_{\boldsymbol{\phi}}(\boldsymbol{z}|\boldsymbol{x}_n) \log \eta_{\boldsymbol{\theta}}(y_n|\boldsymbol{z}) d\boldsymbol{z} \\ &\quad + \beta\ \mathrm{KL}[q_{\boldsymbol{\phi}}(\mathbf{z}|\boldsymbol{x}_n), p_{\boldsymbol{\theta}}(\mathbf{z}|\boldsymbol{x}_n)] \} - \mathrm{H}(\mathbf{Y}) \\ &\leq \frac{1}{|\mathcal{S}|} \sum_{n=1}^{|\mathcal{S}|} \{ \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x_n}, y_n) \} - \mathrm{H}(\mathbf{Y}) \quad (49) \end{aligned}$$

A Similar result has been shown in [56] for image reconstruction.

B. An alternative problem formulation

Assuming the same setting as we discussed above, we provide an alternative derivation the loss function for joint image classification and compression as follows.

We start with the classification cross entropy loss between the true and estimated label distributions. From Eqs. (18)(19), we can write the cross entropy loss as:

$$\begin{aligned} \mathrm{CE}_{y|\boldsymbol{x}}(\rho, \hat{\rho}_{\boldsymbol{\theta}}) &= -\sum_{c \in \mathcal{Y}} \rho(y=c|\boldsymbol{x}) \log \hat{\rho}_{\boldsymbol{\theta}}(y=c|\boldsymbol{x}) \\ &= -\log \hat{\rho}_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\boldsymbol{x}). \quad (50) \end{aligned}$$

For the setting with an Encoder (**E**) with the density $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$, we have

$$\mathbf{x} \xrightarrow[\boldsymbol{\phi}]{\mathrm{E}} \mathbf{z} \xrightarrow[\boldsymbol{\theta}]{\mathrm{CL}} y. \quad (51)$$

Considering different instances of $\mathbf{z}$ for a given sample $\boldsymbol{x}$, Eq. (14) can be re-written to formulate the following problem to maximize $\hat{\rho}_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\boldsymbol{x})$ in order to minimize classification cross entropy.

$$\max_{\boldsymbol{\theta},\boldsymbol{\phi}} E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}} [\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z})] \quad (52)$$

Since the density $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$ is not available at the Decoder (**D**), we employ a Probability Estimator $p_{\boldsymbol{\theta}}(\mathbf{z}|\mathbf{x})$ to closely estimate $q_{\boldsymbol{\phi}}(\mathbf{z}|\mathbf{x})$. This introduces a constraint to the problem in Eq. (52) with $\epsilon \geq 0$ [26], [39].

$$\begin{aligned} &\max_{\boldsymbol{\theta},\boldsymbol{\phi}} E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}} [\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z})] \\ &\text{s.t.} \quad 0 \leq \mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}} | p_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}) \leq \epsilon \quad (53) \end{aligned}$$

This can be re-written as a minimization problem by introducing a monotonic $-\log()$ function.

$$\begin{aligned} &\min_{\boldsymbol{\theta},\boldsymbol{\phi}} -\log E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}} [\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z})] \\ &\text{s.t.} \quad \mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}} | p_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}) \leq \epsilon \quad (54) \end{aligned}$$

According to Jensen's inequality, we can write,

$$-\log E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}} [\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z})] \leq E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}} [-\log(\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z}))]. \quad (55)$$

Instead of directly minimizing the optimization problem in Eq. (54), we can minimize its upper bound as follows.

$$\begin{aligned} &\min_{\boldsymbol{\theta},\boldsymbol{\phi}} E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}} [-\log(\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z}))] \\ &\text{s.t.} \quad \mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}} | p_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}) \leq \epsilon \quad (56) \end{aligned}$$

By re-writing the problem in Eq (56) as a Lagrangian under KKT [57] conditions for the Lagrangian multiplier $\beta \geq 0$, we write,

$$\begin{aligned} \mathcal{F}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}, y_{\mathrm{gt}}) &= E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}} [-\log(\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z}))] \\ &\quad + \beta \left[\mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}} | p_{\mathbf{z}|\mathbf{X},\boldsymbol{\theta}}) - \epsilon \right]. \quad (57) \end{aligned}$$

With this we can arrive at the β-VAE loss function for classification.

$$\begin{aligned} \mathcal{F}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}, y_{\mathrm{gt}}) \leq \mathcal{L}_{\boldsymbol{\theta},\boldsymbol{\phi}}(\boldsymbol{x}, y_{\mathrm{gt}}) &= E_{q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}}} [-\log(\eta_{\boldsymbol{\theta}}(y_{\mathrm{gt}}|\mathbf{z}))] \\ &\quad + \beta \mathrm{KL}(q_{\mathbf{z}|\mathbf{x},\boldsymbol{\phi}} | p_{\mathbf{z}|\mathbf{x},\boldsymbol{\theta}}). \quad (58) \end{aligned}$$

VIII. Conclusions

In this work, we propose a VAE-based end-to-end framework for joint compression and classification that autonomously learns on the latent features to efficiently compress and classify images in a networked AI edge/cloud environment. Starting from the classification cross entropy loss, we present the theoretical foundation of the solution from information theory perspective to define a rate-classification loss similar to rate-distortion in image reconstruction. Test results of our VAE learning networks on CIFAR-10, CIFAR-100, and ImageNet-1k data sets demonstrate significant improvement of image classification accuracy at the same bit rate. In particular, the proposed VAE-based joint compression and classification framework demonstrates over 10% and 4% classification accuracy improvement for CIFAR-10 and ImageNet-1k data sets respectively, at data rate of 0.8 bpp in comparison with the popular JPEG standard codec. We also achieve over $\times 3$ bandwidth reduction for a given classification accuracy on the two data sets over JPEG. Extending the optimization of rate-classification performance of the proposed framework to address computation complexity is an interesting future direction to explore.

Acknowledgment

This material is based on works supported by the National Science Foundation under Grant No. 2002927 and 2002937.

References

[1] M. H. Miraz, M. Ali, P. S. Excell, and R. Picking, "A review on internet of things (IoT), internet of everything (IoE) and internet of nano things (IoNT)," in *2015 Internet Technologies and Applications (ITA)*. IEEE, 2015, pp. 219–224.

[2] Q. Rao and J. Frtunikj, "Deep learning for self-driving cars: chances and challenges," in *Proceedings of the 1st International Workshop on Software Engineering for AI in Autonomous Systems*, 2018, pp. 35–38.

[3] G. Sreenu and M. S. Durai, "Intelligent video surveillance: a review through deep learning techniques for crowd analysis," *Journal of Big Data*, vol. 6, no. 1, p. 48, 2019.

[4] Z. Yang and L. S. Pun-Cheng, "Vehicle detection in intelligent transportation systems and its applications under varying environments: A review," *Image and Vision Computing*, vol. 69, pp. 143–154, 2018.

[5] J. Kaur, M. A. Khan, M. Iftikhar, M. Imran, and Q. E. U. Haq, "Machine learning techniques for 5g and beyond," *IEEE Access*, vol. 9, pp. 23 472–23 488, 2021.

[6] B. Ji, Y. Wang, K. Song, C. Li, H. Wen, V. G. Menon, and S. Mumtaz, "A survey of computational intelligence for 6G: Key technologies, applications and trends," *IEEE Transactions on Industrial Informatics*, 2021.

[7] C. De Alwis, A. Kalla, Q.-V. Pham, P. Kumar, K. Dev, W.-J. Hwang, and M. Liyanage, "Survey on 6G frontiers: Trends, applications, requirements, technologies and future research," *IEEE Open Journal of the Communications Society*, vol. 2, pp. 836–886, 2021.

[8] V. Cisco, "Cisco visual networking index: Forecast and trends, 2017–2022," *White Paper*, vol. 1, 2018.

[9] W. Yu, F. Liang, X. He, W. G. Hatcher, C. Lu, J. Lin, and X. Yang, "A survey on the edge computing for the internet of things," *IEEE access*, vol. 6, pp. 6900–6919, 2017.

[10] F. Samie, L. Bauer, and J. Henkel, "From cloud down to things: An overview of machine learning in internet of things," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4921–4934, 2019.

[11] A. A. Diro, N. Chilamkurti, and Y. Nam, "Analysis of lightweight encryption scheme for fog-to-things communication," *IEEE Access*, vol. 6, pp. 26 820–26 830, 2018.

[12] C. Gong, F. Lin, X. Gong, and Y. Lu, "Intelligent cooperative edge computing in internet of things," *IEEE Internet of Things Journal*, vol. 7, no. 10, pp. 9372–9382, 2020.

[13] H. Wang, C. Tao, J. Qi, H. Li, and Y. Tang, "Semi-supervised variational generative adversarial networks for hyperspectral image classification," in *IGARSS 2019-2019 IEEE International Geoscience and Remote Sensing Symposium*. IEEE, 2019, pp. 9792–9794.

[14] Y. Luo and H. Pfister, "Adversarial defense of image classification using a variational auto-encoder," *arXiv preprint arXiv:1812.02891*, 2018.

[15] X. Chen, Y. Sun, M. Zhang, and D. Peng, "Evolving deep convolutional variational autoencoders for image classification," *IEEE Transactions on Evolutionary Computation*, vol. 25, no. 5, pp. 815–829, 2020.

[16] L. D. Chamain, S. S. Cheung, and Z. Ding, "Quannet: Joint image compression and classification over channels with limited bandwidth," in *2019 IEEE International Conference on Multimedia and Expo (ICME)*, 2019, pp. 338–343.

[17] L. D. Chamain and Z. Ding, "Faster and accurate classification for JPEG2000 compressed images in networked applications," *arXiv preprint arXiv:1909.05638*, 2019.

[18] ——, "Improving deep learning classification of jpeg2000 images over bandlimited networks," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 4062–4066.

[19] S. Dodge and L. Karam, "Understanding how image quality affects deep neural networks," in *2016 eighth international conference on quality of multimedia experience (QoMEX)*. IEEE, 2016, pp. 1–6.

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

[21] J. Chen, Y. Ye, and S. H. Kim, "JVET-Q2002 Algorithm description for Versatile Video Coding and Test Model 8 (VTM 8)," Jan. 2020.

[22] G. J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012. [Online]. Available: http://ieeexplore.ieee.org/abstract/document/6316136/

[23] G. K. Wallace, "The JPEG still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[24] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. V. Gool, "Soft-to-hard vector quantization for end-to-end learning compressible representations," *Advances in neural information processing systems*, vol. 30, 2017.

[25] M. Li, W. Zuo, S. Gu, D. Zhao, and D. Zhang, "Learning convolutional networks for content-weighted image compression," *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun 2018. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2018.00339

[26] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," *International Conference on Learning Representations (ICLR)*, 2018.

[27] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, "Conditional probability models for deep image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4394–4402.

[28] G. Toderici, D. Vincent, N. Johnston, S. Jin Hwang, D. Minnen, J. Shor, and M. Covell, "Full resolution image compression with recurrent neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5306–5314.

[29] N. Tishby, F. C. Pereira, and W. Bialek, "The information bottleneck method," *arXiv preprint physics/0004057*, 2000.

[30] N. Tishby and N. Zaslavsky, "Deep learning and the information bottleneck principle," in *2015 IEEE Information Theory Workshop (ITW)*. IEEE, 2015, pp. 1–5.

[31] C. R. T.81(1992), "Information technology - Digital compression and coding of continuous-tone still images - Requirements and guidelines." CCITT, Standard, 1992.

[32] D. S. Taubman and M. W. Marcellin, *JPEG2000 Image Compression Fundamentals, Standards and Practice*. Springer Science & Business Media, 2012, vol. 642.

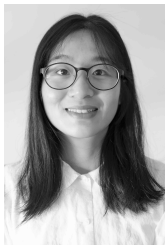
[33] A. Van Den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. W. Senior, and K. Kavukcuoglu, "Wavenet: A generative model for raw audio." *SSW*, vol. 125, p. 2, 2016.

[34] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

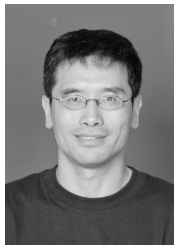
[35] D. P. Kingma and P. Dhariwal, "Glow: Generative flow with invertible 1x1 convolutions," in *Advances in Neural Information Processing Systems*, 2018, pp. 10 215–10 224.

[36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," *arXiv preprint arXiv:1312.6114*, 2013.
[37] C. Doersch, "Tutorial on variational autoencoders," *arXiv preprint arXiv:1606.05908*, 2016.
[38] A. Habibian, T. v. Rozendaal, J. M. Tomczak, and T. S. Cohen, "Video compression with rate-distortion autoencoders," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7033–7042.
[39] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," *International Conference on Learning Representations (ICLR)*, 2017.
[40] S. Singh, S. Abu-El-Haija, N. Johnston, J. Ballé, A. Shrivastava, and G. Toderici, "End-to-end learning of compressible features," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3349–3353.
[41] X. Luo, H. Talebi, F. Yang, M. Elad, and P. Milanfar, "The rate-distortion-accuracy tradeoff: JPEG case study," *arXiv preprint arXiv:2008.00605*, 2020.
[42] M. Weber, C. Renggli, H. Grabner, and C. Zhang, "Lossy image compression with recurrent neural networks: from human perceived visual quality to classification accuracy," *arXiv preprint arXiv:1910.03472*, 2019.
[43] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, "Faster neural networks straight from JPEG," in *Advances in Neural Information Processing Systems*, 2018, pp. 3937–3948.
[44] E. Schonfeld, S. Ebrahimi, S. Sinha, T. Darrell, and Z. Akata, "Generalized zero-and few-shot learning via aligned variational autoencoders," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 8247–8255.
[45] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 620–636, 2003.
[46] V. Sze and D. Marpe, "Entropy coding in HEVC," in *High Efficiency Video Coding (HEVC)*. Springer, 2014, pp. 209–274.
[47] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
[48] F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. V. Gool, "Practical full resolution learned lossless image compression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 10 629–10 638.
[49] A. Krizhevsky, V. Nair, and G. Hinton, "The cifar-10 dataset," *online: http://www. cs. toronto. edu/kriz/cifar. html*, vol. 55, 2014.
[50] A. Krizhevsky, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
[51] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.
[52] Y. Deng, "Deep learning on mobile devices: a review," in *Mobile Multimedia/Image Processing, Security, and Applications 2019*, vol. 10993. International Society for Optics and Photonics, 2019, p. 109930A.
[53] N. D. Lane, S. Bhattacharya, A. Mathur, P. Georgiev, C. Forlivesi, and F. Kawsar, "Squeezing deep learning into mobile and embedded devices," *IEEE Pervasive Computing*, vol. 16, no. 3, pp. 82–88, 2017.
[54] 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 *European conference on computer vision*. Springer, 2014, pp. 740–755.
[55] D. Hendrycks and T. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," *International Conference on Learning Representations (ICLR)*, 2019.
[56] A. A. Alemi, I. Fischer, J. V. Dillon, and K. Murphy, "Deep variational information bottleneck," *Proceedings of the International Conference on Learning Representations (ICLR)*, 2016.
[57] H. W. Kuhn, "Nonlinear programming: a historical view," in *Traces and Emergence of Nonlinear Programming*. Springer, 2014, pp. 393–414.

Lahiru D. Chamain is currently a PhD candidate in Electrical & Computer Engineering (ECE) at University of California Davis, USA. He received his M.Sc. degree in ECE from UC Davis and in 2020 and B.Sc. degree from University of Moratuwa, Sri Lanka in 2016 majoring Electronics and Telecommunication Engineering. His research interests align with image/video compression for deep learning, internet of things, detection and estimation theory.



Siyu Qi is a PhD candidate in the Department of Electrical and Computer Engineering at University of California Davis, USA. She received her B.Sc in Instrument Science and Engineering from Southeast University, China, in 2016. Her research interests lie in the fields of image compression, deep learning, time series forecasting.



Zhi Ding (S'88-M'90-SM'95-F'03) is with the Department of Electrical and Computer Engineering at the University of California, Davis, where he currently holds the position of distinguished professor. He received his Ph.D. degree in Electrical Engineering from Cornell University in 1990. From 1990 to 2000, he was a faculty member of Auburn University and later, University of Iowa. Prof. Ding has held visiting positions in Australian National University, Hong Kong University of Science and Technology, NASA Lewis Research Center and USAF Wright Laboratory. Prof. Ding has active collaboration with researchers from various universities in Australia, Canada, China, Finland, Hong Kong, Japan, Korea, Singapore, Taiwan, and USA.

Dr. Ding is a Fellow of IEEE and has been an active member of IEEE, serving on technical programs of several workshops and conferences. He was associate editor for IEEE Transactions on Signal Processing from 1994-1997, 2001-2004, and associate editor of IEEE Signal Processing Letters 2002-2005. He was a member of technical committee on Statistical Signal and Array Processing and member of technical committee on Signal Processing for Communications (1994-2003). Dr. Ding was the General Chair of the 2016 IEEE International Conference on Acoustics, Speech, and Signal Processing and the Technical Program Chair of the 2006 IEEE Globecom. He was also an IEEE Distinguished Lecturer (Circuits and Systems Society, 2004-06, Communications Society, 2008-09). He served on as IEEE Transactions on Wireless Communications Steering Committee Member (2007-2009) and its Chair (2009-2010). Dr. Ding is a coauthor of the text: Modern Digital and Analog Communication Systems, 5th edition, Oxford University Press, 2019. Prof. Ding received the IEEE Communication Society's WTC Award in 2012 and the IEEE Communication Society's Education Award in 2020.