Towards End-to-End Image Compression and Analysis with Transformers

Yuanchao Bai\textsuperscript{1,2\dagger}, Xu Yang\textsuperscript{1\dagger}, Xianming Liu\textsuperscript{1,2\dagger}, Junjun Jiang\textsuperscript{1,2\dagger}, Yaowei Wang\textsuperscript{2}, Xiangyang Ji\textsuperscript{3}, Wen Gao\textsuperscript{2,4}\dagger

\textsuperscript{1}Harbin Institute of Technology, \textsuperscript{2}Peng Cheng Laboratory, \textsuperscript{3}Tsinghua University, \textsuperscript{4}Peking University

yuanchao.bai@gmail.com, 20s103274@stu.hit.edu.cn, \{csxm,jiangjunjun\}@hit.edu.cn

wangyw@pcl.ac.cn, xyi@tsinghua.edu.cn, wgao@pku.edu.cn

Abstract

We propose an end-to-end image compression and analysis model with Transformers, targeting to the cloud-based image classification application. Instead of placing an existing Transformer-based image classification model directly after an image codec, we aim to redesign the Vision Transformer (ViT) model to perform image classification from the compressed features and facilitate image compression with the long-term information from the Transformer. Specifically, we first replace the patchify stem (i.e., image splitting and embedding) of the ViT model with a lightweight image encoder modelled by a convolutional neural network. The compressed features generated by the image encoder are injected convolutional inductive bias and are fed to the Transformer for image classification bypassing image reconstruction. Meanwhile, we propose a feature aggregation module to fuse the compressed features with the selected intermediate features of the Transformer, and feed the aggregated features to a convolutional neural network for image reconstruction. The aggregated features can obtain the long-term information from the self-attention mechanism of the Transformer and improve the compression performance. The rate-distortion-accuracy optimization problem is finally solved by a two-step training strategy. Experimental results demonstrate the effectiveness of the proposed model in both the image compression and the classification tasks.

Introduction

Vision Transformer (ViT) (Dosovitskiy et al. 2021) and its variations (Touvron et al. 2021; Wu et al. 2021; Yuan et al. 2021; Chen et al. 2021b; Liu et al. 2021), inherited from Transformer architecture ( Vaswani et al. 2017) in natural language processing (NLP), have recently demonstrated outstanding performance on a board range of image analysis tasks, such as image classification (Dosovitskiy et al. 2021), segmentation (Zheng et al. 2021) and object detection (Fang et al. 2021). With the self-attention mechanism, these models are capable of capturing long-range dependencies in the image data, but inevitably result in high computational cost. In practice, Transformer-based models are usually deployed in the cloud-based paradigm and executed remotely. For example, massive image data is acquired by the frontend devices, such as mobile phones or surveillance cameras, and transmitted to the cloud (i.e., data center) for further analysis, sharing and storage. Image compression serves as a fundamental infrastructure for data communication between the frontend and the cloud.

In the traditional paradigm of cloud-based applications, image compression is considered independent of image analysis, and adopts lossy image compression standards designed for human vision, such as JPEG (Wallace 1992). In particular, the raw images are first transformed to the frequency domain with Discrete Cosine Transform (DCT). The frequency coefficients are then quantized to discard high frequencies that are less sensitive to human eyes. The quantized coefficients are encoded to bitstreams with entropy encoding and are transmitted to the cloud. On the cloud side, the quantized coefficients are recovered from the received bitstreams, which are then inversely transformed to reconstruct images. The reconstruction distortions are minimized with respect to Peak Signal-to-Noise Ratio (PSNR). However, if the reconstructed images optimized by PSNR are fed into the downstream image analysis tasks, which are tailored to machine vision instead, the corresponding results may be inaccurate, because the principle of machine vision is different from human vision (Wang et al. 2020). Besides, the traditional image codecs are comprised of hand-crafted modules with complex dependencies. It is difficult to optimize the sophisticated compression frameworks together with subsequent machine analysis tasks.

Recently, learning-based image compression emerges as an active research area in computer vision community. A number of learning-based image codecs, such as (Toderici et al. 2016; Theis et al. 2017; Li et al. 2018; Ballé et al. 2018; Minnen, Ballé, and Toderici 2018; Cheng et al. 2020; Ma et al. 2020; Hu et al. 2021), have achieved comparable or even better perceptual performance than traditional image codecs for human vision. Besides, by replacing the hand-crafted modules with deep neural networks (DNNs), learning-based image compression can be integrated with high-level tasks and end-to-end optimized for machine vision (Torfason et al. 2018; Chamain et al. 2021; Le et al. 2021). However, compared with image compression for human vision, image compression for machine vision is still in its infancy, because it is challenging to achieve the best of both worlds for low-level and high-level tasks.
In this paper, we propose a novel paradigm that is friendly for both human vision and machine vision, which integrates learning-based image compression with Transformer-based image analysis. The derived end-to-end image compression and analysis model leads to the synergy effect of these two tasks. Instead of placing an existing Transformer-based image classification model directly after an image codec, we redesign the ViT model to perform image classification from the compressed features (Torfason et al. 2018) and facilitate image compression with the long-term information from the Transformer. Specifically, we replace the patchify stem (i.e., image splitting and embedding) of the ViT model with a lightweight image encoder modelled by a convolutional neural network (CNN). The compressed features generated by the image encoder are injected convolutional inductive bias and are more expressive than the features extracted by the patchify stem from the decoded images. When transmitted to the cloud, the compressed features are fed to the Transformer for image classification bypassing image reconstruction. We further propose a feature aggregation module to fuse the compressed features with the selected intermediate features of the Transformer, and feed the aggregated features to a deconvolutional neural network for image reconstruction. The aggregated features obtain the long-term information from the Transformer and effectively improve the compression performance. We interpret the corresponding rate-distortion-accuracy optimization problem based on variational auto-encoder (VAE) (Theis et al. 2017; Ballé et al. 2018) and information bottleneck (IB) (Tishby, Pereira, and Bialek 2000; Alemi et al. 2017), and finally solve it with a two-step training strategy.

The main contributions are summarized as follows:

• We propose an end-to-end image compression and analysis model, which performs image classification from the compressed features. We interpret the rate-distortion-accuracy optimization problem based on VAE and IB.
• We design the network by integrating learning-based image compression with ViT-based image analysis, which leads to the synergy between the two tasks.
• In terms of rate-distortion, the proposed model achieves PSNR performance close to BPG (Bellard 2014). In terms of rate-accuracy, the proposed model outperforms ResNet50 (He et al. 2016), DeiT-S (Touvron et al. 2021) and Swin-T (Liu et al. 2021) classification from the decoded images, while significantly reduces the computational cost under equivalent number of parameters.

Proposed Method

Problem Formulation

We aim to perform image analysis from the compressed features. Given a raw image \( x \) and its label \( y \), our goal is to learn a compressed representation \( z \) that facilitates both image decoding (reconstruction) and analysis, as sketched in Fig. 1. Since the compressed representation \( z \) is extracted from the image \( x \) while not accessing the label \( y \), we assume that \( x, y, z \) form a Markov chain \( x \leftrightarrow y \leftrightarrow z \), leading to \( p(z|x, y) = p(z|x) \).

Image Compression. We first formulate the lossy image compression model without taking image analysis into consideration. Following the standard framework of variational auto-encoder based image compression (Theis et al. 2017; Ballé et al. 2018), the latent representation \( z \) is transformed from the raw image \( x \) by an encoder and is quantized to the discrete-valued \( \hat{z} \). Then, \( \hat{z} \) is losslessly compressed with entropy encoding techniques (Witten, Neal, and Cleary 1987;
Because the entropy $H$ is relaxed by adding noise from uniform distribution $\mathcal{U}(-\frac{1}{2^N}, \frac{1}{2^N})$, we have $q(z|x) = \prod_i \mathcal{U}(z_i - \frac{1}{2}, z_i + \frac{1}{2})$ and thus the first term $\log q(z|x) = 0$. The second term of (1) is interpreted as the expected distortion between $x$ and $\hat{x}$, and the third term is interpreted as the cost of encoding $z$, leading to the rate-distortion trade-off (Shannon 1948).

**Image Analysis.** We then turn to consider image analysis. We propose to maximize the mutual information $I(\hat{z}, y)$ between the compressed representation $\hat{z}$ and the label $y$, inspired by information bottleneck (Tishby, Pereira, and Bialek 2000; Alemi et al. 2017). The mutual information $I(\hat{z}, y)$ is the reduction in the uncertainty of $y$ due to the knowledge of $\hat{z}$:

$$I(\hat{z}, y) = H(y) - H(y|\hat{z}) = H(y) + \sum_{y, \hat{z}} p(y, \hat{z}) \log p(y|\hat{z})$$

where $H(\cdot)$ denotes the entropy. Because the true posterior $p(y|\hat{z})$ is also intractable, we propose a variational approximation $q(y|\hat{z})$, which is the decoder for image analysis apart from the decoder for image reconstruction. Since $D_{KL}[p(y|\hat{z})||q(y|\hat{z})] \geq 0$, we have $\sum_{y, \hat{z}} p(y|\hat{z}) \log p(y|\hat{z}) \geq \sum_{\hat{z}} p(\hat{z}) \log q(y|\hat{z})$ and thus

$$I(\hat{z}, y) \geq H(y) + \sum_{y, \hat{z}} p(y, \hat{z}) \log q(y|\hat{z})$$

Because the entropy $H(y)$ is independent of $\hat{z}$, we can maximize $\sum_{y, \hat{z}} p(y, \hat{z}) \log q(y|\hat{z})$ as a proxy for $I(\hat{z}, y)$. Based on the Markov chain assumption, we replace $p(y, \hat{z})$ with $\sum_{\hat{y}} p(x, y, \hat{z}) = \sum_{\hat{y}} p(x, y)p(\hat{z}|x)$, and can rewrite $\sum_{\hat{y}, \hat{z}} p(\hat{y}, \hat{z}) \log q(y|\hat{z})$ as

$$E_{p(x, y)} E_{p(\hat{z}|x)} \log q(y|\hat{z})$$

With $q(y|\hat{z})$, we can generate the estimated label $\hat{y}$ from $\hat{z}$.

**Joint Optimization.** With (1) and (4), we further formulate the joint optimization of both image compression and analysis since $p(\hat{z}|x)$ in (4) is intractable, we share the inference model $q(\hat{z}|x)$ in (1) as the approximation, and minimize the approximated negative (4) together with (1):

$$\mathbb{E}_{p(x,y)} \left\{ \alpha \mathbb{E}_{p(\hat{z}|x)} \log q(y|\hat{z}) + D_{KL}[q(\hat{z}|x)||p(\hat{z}|x)] \right\}$$

where $\alpha$ is a trade-off parameter. Suppose that $p(x|\hat{z})$ is given by $\mathcal{N}(x|\hat{x}, (2/\beta)^{-1} I)$, we can finally rewrite (5) to the objective function:

$$\mathbb{E}_{p(x,y)} \mathbb{E}_{q(\hat{z}|x)} \left[ -\alpha \log q(y|\hat{z}) + \beta \|x - \hat{x}\|^2 - \log p(\hat{z}) \right]$$

The first term of (6) weighted by $\alpha$ can be interpreted as the cross-entropy loss for image analysis, such as image classification or segmentation, based on the types of label $y$. In this work, we choose image classification as the target task. The second term weighted by $\beta$ is the mean square error (MSE) distortion loss. The third term is the rate loss.

In contrast to the image compression models (Theis et al. 2017; Ballé et al. 2018), the compressed representation $\hat{z}$ in (6) is also optimized for image analysis tasks. The complexity of $\hat{z}$ in (6) is controlled by minimizing the cost of encoding $\hat{z}$, rather than controlled by minimizing $I(\hat{z}, x)$ in the information bottleneck models (Tishby, Pereira, and Bialek 2000; Alemi et al. 2017).

**Transformer-based Network Architecture**

We realize the theoretical framework in Fig. 1 by proposing an end-to-end image compression and analysis model with Transformers. The proposed model can promote the synergy between the two tasks.

**Encoder.** The network architecture of the proposed encoder is illustrated in Fig. 2. Similar to the setting of (Ballé et al. 2018), we employ four stride-2 $5 \times 5$ convolutional layers to extract features with gradually reduced spatial resolution from the input image $x \in \mathbb{R}^{H \times W \times 3}$. We use LeakyReLU as the activation function instead of using the Generalized Divisive Normalization (GDN) (Ballé, Laparra, and Simoncelli 2016), because GDN results in convergence

$$\mathbb{E}_{p(x)} D_{KL}[q(\hat{z}|x)||p(\hat{z}|x)] = \mathbb{E}_{p(x,y)} D_{KL}[q(\hat{z}|x)||p(\hat{z}|x)]$$
problem when training with the Transformer blocks in the proposed model.

The resulting feature \( z \in \mathbb{R}^{H \times \frac{W}{4} \times M} \) is then quantized to the discrete-valued \( \tilde{z} \). We employ the hyper-prior module (Ballé et al. 2018; Minnen, Ballé, and Toderici 2018) to estimate \( p(\tilde{z}|h_2) \) for the entropy encoding of \( \tilde{z} \), where \( h_2 \) denotes the hyper-prior of \( \tilde{z} \). We do not use the serial autoregressive module (Minnen, Ballé, and Toderici 2018), because its corresponding decoding time is too long on large-scale image classification datasets.

From the perspective of Transformer architecture design, the proposed encoder can be considered as the replacement of the patchify stem, \( i.e., \) a stride-16 16 \( \times \) 16 convolutional layer, applied to the input image in the ViT model (Dosovitskiy et al. 2021). Several concurrent works (Wu et al. 2021; Yuan et al. 2021; Chen et al. 2021b; Xiao et al. 2021) also replace the patchify stem with a stack of convolutional layers, in order to improve the performance of image classification. In the experiments, we observe that the proposed encoder is capable of extracting compressed features suitable for both image decoding reconstruction and classification through joint optimization. Note that the proposed encoder is relatively lightweight, thus can be deployed at the frontend, such as mobile phones or surveillance cameras.

**Decoder-Classifier.** The decoder receives the bitstream from the encoder and adopts the shared hyper-prior module \( p(\tilde{z}|h_2) \) to recover \( \tilde{z} \). For image classification, we directly feed \( \tilde{z} \) to an inference network, instead of the common subsequent approach—first reconstruct the decoded image \( \hat{x} \) and then conduct inference on \( \hat{x} \).

Specifically, we adopt the standard Transformer blocks in the ViT model (Dosovitskiy et al. 2021) with the number of parameters equivalent to ResNet50 (He et al. 2016), as shown in Fig. 3. We expand the channel dimension of \( \tilde{z} \) to \( C \) with a 1 \( \times \) 1 convolutional layer, and reshape the resulting feature \( \tilde{z}_0 \in \mathbb{R}^{\frac{H}{16} \times \frac{W}{16} \times C} \) to a sequence \( z_0 \in \mathbb{R}^{\frac{H \cdot W}{16^2} \times C} \). To maintain the spatial information of the feature \( \tilde{z}_0 \), we add learnable position embeddings \( p \) to \( z_0 \) leading to \( \tilde{z}_0 = z_0 + p \). Following (Dosovitskiy et al. 2021), we prepend a learnable class embedding \( \bar{c}_0 \), and feed the sequence \( [\bar{c}_0; \tilde{z}_0] \in \mathbb{R}^{(\frac{H \cdot W}{16^2} + 1) \times C} \) to the Transformer consisting of \( L \) Transformer blocks. The architecture of each Transformer block is illustrated in Fig. 3. The computation process can be formulated as

\[
\begin{align}
[c_i; z_i] &= \text{MSA}(\text{LN}([c_{i-1}; z_{i-1}])) + [c_{i-1}; z_{i-1}] \\
[c_i; z_i] &= \text{FFN}(\text{LN}([c_i; z_i])) + [c_i; z_i] \quad i = 1, \ldots, L
\end{align}
\]

where MSA(\cdot) denotes the multi-head self-attention module, FFN(\cdot) denotes the feed forward network and LN(\cdot) denotes the layer normalization (Ba, Kiros, and Hinton 2016), respectively.

With the self-attention mechanism, the class embedding \( \bar{c}_i \) interacts with the image feature \( z_i \), and the final output \( c_L \) is used to compute \( q(y|\tilde{z}) \) for image classification:

\[
q(y|\tilde{z}) = \text{Softmax}(\text{FFN}(\text{LN}(c_L)))
\]

where Softmax(\cdot) denotes softmax operation. The FFN(\cdot) is the classifier head mapping the embedding dimension from \( C \) to the number of classes.

**Decoder-Reconstructor.** Reconstructing image \( \hat{x} \) directly from \( \tilde{z} \) (or \( z_0 \)) ignores the global spatial correlations among the latent features. Recent image compression works (Qian et al. 2021; Guo et al. 2021) demonstrate that leveraging global context information during entropy coding can improve the compression performance. Transformers naturally capture the global spatial information among the latent features, which can also benefits low-level tasks, such as image processing (Chen et al. 2021a) and image generation (Jiang, Chang, and Wang 2021). Motivated by these works, we aim to extract the intermediate features \( \tilde{z}_i \)’s of the Transformer and incorporate them into image reconstruction.

Specifically, we select \( z_0 \) and \( [\tilde{z}_1, \tilde{z}_2, \tilde{z}_3] \), and propose a feature aggregation module to fuse these features, similar to (Zheng et al. 2021). Selecting \( [\tilde{z}_1, \tilde{z}_2, \tilde{z}_3] \) means that the first three Transformer blocks are also involved in the image reconstruction process. Since image reconstruction may work independently of image classification, we avoid using \( \{\tilde{z}_i(i > 3)\} \) that involve too many Transformer blocks in the image reconstruction, in order to reduce the computational complexity. The feature aggregation module is illustrated in Fig. 3. The computational process can be formulated as

\[
\begin{align}
\tilde{z}_0'' &= \text{Conv}^1_0(z_0), \quad \tilde{z}_1'' &= \text{Conv}^1_1(z_1) \\
\tilde{z}_2'' &= \text{Conv}^1_2(z_2), \quad \tilde{z}_3'' &= \text{Conv}^1_3(z_3) \\
\tilde{z}_f &= \text{Conv}_2([\tilde{z}_0''; \tilde{z}_1''; \tilde{z}_2''; \tilde{z}_3''])
\end{align}
\]

where \( \text{Conv}^1_i(\cdot) \) denotes a \( 1 \times 1 \) convolutional layer reducing the channel dimension of the input to \( \frac{C}{4} \). \( \text{Conv}_2(\cdot) \) is another \( 1 \times 1 \) convolutional layer with \( C \) channels fusing the four concatenated input features. \( \tilde{z}_f \) is the fused feature.

Finally, we input the fused feature \( \tilde{z}_f \) to four stride-2 \( 5 \times 5 \) deconvolutional layers gradually increasing the spatial resolution, leading to the reconstructed RGB image \( \hat{x} \).

**Training Strategy**

We observe that the one-step training strategy, \( i.e., \) minimizing (6) to train the encoder and decoder from scratch, leads to convergence problem in the experiments. Instead, we employ a two-step training strategy:

1. We pretrain the proposed model without considering the quantization of \( z \) and the hyper-prior module of \( \tilde{z} \). We remove the rate loss in (6) temporarily, and minimize the cross-entropy loss together with the MSE loss. Because the value of the cross-entropy loss is much smaller than that of the MSE loss, we set \( \alpha = 1 \) and \( \beta = 0.001 \) in (6) to balance the contributions of the two losses.

2. We load the pretrained parameters and minimize (6) to train the entire network including the quantization of \( z \) and the hyper-prior module of \( \tilde{z} \). The \( \alpha \) and \( \beta \) in (6) are tuned with fixed \( \beta \) to achieve different bit rates.

**Experiments**

**Experimental Settings**

**Datasets.** We perform extensive experiments on the ImageNet dataset (Deng et al. 2009) and iNaturalist19 (iNat19)
Figure 3: The network architecture of the decoder. We set $C = 384$, $L = 12$ and $N = 128$. The number of parameters is equivalent to ResNet50 (He et al. 2016). The decoder is deployed on the cloud to classify or reconstruct images from the received bitstreams. AD: Arithmetic Decoder.

dataset (Horn and Aodha 2019). ImageNet is a well-known image classification dataset containing 1000 object classes with 1,281,167 training images and 50,000 validation images. INat19 is a fine-grained classification dataset containing 1010 species of plants and animals with 265,213 training images and 3030 validation images.

Pretraining w/o Compression. As aforementioned, we pretrain the proposed model without the quantization of $z$ and the hyper-prior module of $\hat{z}$. We remove the rate loss, and minimize the cross entropy loss together with the MSE loss. We set $\alpha = 1$ and $\beta = 0.001$, respectively. We set the input size to 224×224 and adopt the same data augmentation as DeiT (Touvron et al. 2021), except for the Exponential Moving Average (EMA) (Polyak and Juditsky 1992), which do not enhance the performance of the proposed model. The input images are normalized with ImageNet default mean and standard deviation, and are denormalized during image reconstruction. We observe that random erasing (Zhong et al. 2020), mixup (Zhang et al. 2017) and cutmix (Yun et al. 2019) designed for the training of image classification are also compatible with the training of image reconstruction in our experiments.

On the ImageNet dataset, we train the proposed network from scratch. We use AdamW optimizer (Loshchilov and Hutter 2019) for 300 epochs with minibatches of size 1024. We set the initial learning rate to 0.0001 and use a cosine decay learning rate scheduler with 5 epochs warm-up.

On the INat19 dataset, we initialize the network with the pretrained parameters on the ImageNet dataset. The classifier head is adjusted to the class number of INat19. We use AdamW optimizer for 100 epochs with minibatches of size 512. We set the initial learning rate to 0.0005 and use a cosine decay learning rate scheduler with 2 epochs warm-up.

Experimental Results

Pretrained Model. Table 1(a) reports the experimental results of our pretrained model without compression on ImageNet. We compare with the existing image classification models including CNN-based models, such as ResNet50 (He et al. 2016) and RegNetY-4G (Radosavovic et al. 2020), and Transformer-based models, such as ViT-B (Dosovitskiy et al. 2021), DeiT-S (Touvron et al. 2021), CvT-13 (Wu et al. 2021), CeiT-S (Yuan et al. 2021), Visformer-S (Chen et al. 2021b), Swin-T (Liu et al. 2021) and ViT$_C$-4GF (Xiao et al. 2021). We select the specific settings of the models with the number of parameters closest to ResNet50.

Table 1(b) reports the experimental results of our pretrained model on INat19 dataset, compared with ResNet50,
DeiT-S and Swin-T. ResNet50 is finetuned based on (He et al. 2016). DeiT-S and Swin-T are finetuned using the same setting as our pretrained model.

From Table 1, our pretrained model can achieve comparable or better Top-1 accuracies than the existing image classification models, which demonstrates the efficacy of the image encoder for the Transformer-based image classification. In terms of image reconstruction evaluated by PSNR, our pretrained model achieves 31.7 dB and 31.5 dB, respectively. All these results demonstrate that the pretrained model can provide a satisfactory initialization for the following training with compression.

Full Model. In Fig. 4, we report the rate-distortion and rate-accuracy results of our full model on ImageNet and INat19. We compare with the existing image codecs and the image classification methods applied to the reconstructed RGB images. JFT means joint finetune.

We select the traditional image codecs including JPEG (Wallace 1992), JPEG2000 (Skodras, Christopoulos, and Ebrahimi 2001), BPG (Bellard 2014), and the learning-based image codecs including bmshj (Ballé et al. 2018) and mbt-m (Minnen, Ballé, and Toderici 2018). The mbt-m removes the serial autoregressive module, avoiding long decoding time on the large-scale datasets. The sophisticated learning-based image codecs with complex entropy models and network architectures, such as (Hu, Yang, and Liu 2020; Cheng et al. 2020; Qian et al. 2021; Guo et al. 2021), are too time-consuming to be evaluated on the large-scale datasets, despite their good compression performance.

We select the decoded images of the best performed traditional and learning-based image codecs in our experiments, i.e., BPG and mbt-m, and adopt the image classification models ResNet50, DeiT-S and Swin-T to compute the Top-1 accuracies in comparison with the proposed model. Moreover, we jointly finetune mbt-m together with ResNet50, DeiT-S and Swin-T by minimizing (6) with $\alpha = 100$, same as the proposed model.

In terms of the rate-distortion performance, the proposed model significantly outperforms the traditional image codecs JPEG and JPEG2000. It is comparable to BPG at relatively low bit-rates but is surpassed by BPG at high bit-rates. The proposed model outperforms the learning-based image codecs bmshj and mbt-m. The rate-distortion performance of the jointly finetuned mbt-m is similar to the original mbt-m. The proposed model also outperforms the jointly finetuned mbt-m.

In terms of the rate-accuracy performance, the proposed model outperforms ResNet50, DeiT-S and Swin-T applied to the decoded images of BPG and mbt-m, because the image classification models trained on the original datasets are not robust to the decoded images at low bit-rates. Although Swin-T outperforms DeiT-S on the original datasets (Table. 1), DeiT-S outperforms Swin-T on the decoded images at low bit-rates. After jointly finetuned, Swin-T surpasses DeiT-S on the decoded images at the corresponding bit-rates. The proposed model also outperforms the jointly
finetuned ResNet50, Swin-T and DeiT-S, because the image reconstruction constrained by MSE loss damages the high-frequency information effective for the image classification. The proposed model bypasses the image reconstruction process and directly do inference from the compressed features, which can utilize these high-frequency information.

In Fig. 5, we show an illustrative example of the experimental results. Under similar PSNRs, our model achieves comparable or less bit-per-pixel (bpp) and the correct classification result, compared with the other methods.

**Ablation Studies**

**Rate-Distortion-Accuracy Trade-off.** When the bit-rates of \( \hat{z} \) is constrained, minimizing (6) with different \( \frac{\alpha}{\beta} \) leads to bit allocation between image classification and reconstruction. We empirically set \( \frac{\alpha}{\beta} \in \{50, 100, 200\} \) and test the rate-distortion-accuracy trade-off as shown in Fig. 6. We can observe that larger \( \frac{\alpha}{\beta} \) leads to better Top-1 accuracy but sacrifices PSNR. In contrast, smaller \( \frac{\alpha}{\beta} \) leads to better PSNR but lower Top-1 accuracy.

**Feature Aggregation.** In Fig. 6, we compare with the proposed model without the feature aggregation module, i.e., directly feeding \( z_0 \) to the deconvolutional neural network for image reconstruction. Although the feature aggregation module is only applied to the image reconstruction process, it can benefit both the image classification and reconstruction through rate-distortion-accuracy optimization. The reduced bit-rates from the image reconstruction are potentially allocated to the image classification, leading to the improvement of both tasks. Although the improvement of Top-1 accuracy is more obvious than PSNR in Fig. 6, the decrease of the cross entropy loss \( -\alpha \log q(y | \hat{z}) \) is actually similar to that of the MSE loss \( \beta \| x - \hat{x} \|_2^2 \) in (6) in the experiments.

**Computational Cost.** In Fig. 7, we compare the computational cost of the proposed model with the concatenation of the learning-based image codec mbt-m and the image classification methods including ResNet50, DeiT-S and Swin-T. The architecture of the proposed encoder is similar to the low-rate setting of mbt-m, thus their computational costs of image encoding are similar. In terms of image reconstruction on the decoder side, our image reconstructor needs more FLOPs than mbt-m due to the feature aggregation module. In terms of image classification on the decoder side, our image classifier directly performs inference from the compressed features without the image reconstruction process, and thus needs far less computational cost compared with the inference from reconstructed RGB images.

**Conclusion**

In this paper, we learn an end-to-end image compression and analysis model with Transformers, targeting to the cloud-based image classification application. At the frontend, a CNN-based image encoder extracts compressed features from raw images and transmits them to the cloud. On the cloud, the compressed features injected convolutional inductive bias are directly fed to the Transformer for image classification bypassing image reconstruction. Meanwhile, the intermediate features of the Transformer capturing global information are aggregated with the compressed features for image reconstruction. Experimental results demonstrate the effectiveness of the proposed model in both rate-distortion and rate-accuracy performance.
Acknowledgements

This work was supported by National Key Research and Development Project under Grant 2019YFE0109600, National Natural Science Foundation of China under Grants 61922027, 61827804, 61971165 and U20B2052, PCNL Key Project under Grant PCL2021A07, China Postdoctoral Science Foundation under Grant 2020M682826.

References


