HNR-ISC: Hybrid Neural Representation for Image Set Compression

Pingping Zhang, Shiqi Wang, *Senior Member, IEEE,*, Meng Wang, Peilin Chen, Wenhui Wu, Xu Wang, *Member, IEEE,*, Sam Kwong, *Fellow, IEEE*

***Abstract*—A common approach for image set compression (ISC) is to remove the redundancy among images at either signal or frequency domain. A predominant problem of this approach is the inefficiency in handling complex geometric deformations across different images. While many methods have been proposed to compress the images/videos with end-to-end deep neural networks, little work has been dedicated to the high efficiency compression of image sets by mining the cross-image redundancies with neural networks. Here, we propose a new Hybrid Neural Representation for ISC (HNR-ISC), including an implicit neural representation for Semantically Common content Compression (SCC) and an explicit neural representation for Semantically Unique content Compression (SUC). For SCC, the underlying principle is converting the semantically common contents into a small-and-sweet neural representation plus embeddings that can be conveyed as the bitstream. For SUC, an invertible module is designed for removing intra-image redundancies. The feature level combination between SCC and SUC naturally forms the final image set. Experimental results demonstrate the robustness and generalization capability of HNR-ISC in terms of perceptual quality and accuracy for the downstream analysis task.**

***Index Terms*—Image set compression, implicit neural representation, image redundancy.**

I. Introduction

RECENT years have witnessed the exponentially growing services of digital images and videos which have increasingly increased the demand for image compression techniques. In principle, these techniques aim to achieve highly efficient representations of images and videos by exploiting various forms of redundancies (e.g., spatial, perceptual, and statistical redundancies) [1], [2], [3], [4]. The image sets, which are formed in an automatic or handcrafted way, could serve as the fundamental management structure in various products (e.g., Google Photos and iCloud Photo Library). In addition to intra-image redundancy, image set compression (ISC) typically leverages the inter-image redundancies [5] across different images. This is typically beyond the commonly-known intra-redundancy, as different images in the set hold similarities in multiple granularities.

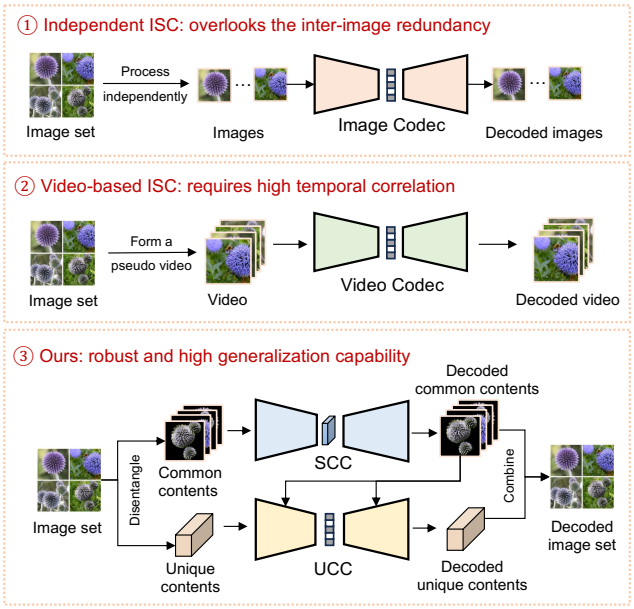

Fig. 1. The comparisons of three ISC methods: 1) Independent ISC methods compress each image individually via the image codec, overlooking inter-image redundancy; 2) Video-based ISC methods form a pseudo video from the image set and then compresses it using a video codec, requiring high temporal correlation; 3) Our proposed scheme demonstrates robust and high generalization capability with the hybrid neural representation.

The inter-image redundancies in image sets are typically removed via signal or frequency domain predictions [5], [6], [7], [8], [9]. Based on their prediction structures, these methods can be categorized into central prediction [6], [7], [8] and sequential prediction methods [9], [10]. Central prediction methods initially select or construct one or more representative images from the set. These representative images are then independently compressed using image compression methods. Subsequently, the remaining images can be compressed by referring to the decoded representative images, with only the prediction residuals being coded. The methods based on sequential prediction structures leverage a video coding framework by reorganizing similar images into a sequence according to prediction costs. As such, each image can be decoded sequentially. These methodologies are effective when dealing with an image set acquired in a single scene, where the background remains consistent. Unfortunately, images within a set are often loosely correlated, especially when they are

This work was supported by the Hong Kong Innovation and Technology Commission (InnoHK Project CIMDA), in part by the Hong Kong GRF-RGC General Research Fund under Grant 11209819 (CityU 9042816) and Grant 11203820 (CityU 9042598).

Pingping Zhang, Shiqi Wang, Meng Wang, and Peilin Chen are with the Department of Computer Science, City University of Hong Kong, Hong Kong, China (email: ppingyes@gmail.com; shiqwang@cityu.edu.hk; mwang98-c@my.cityu.edu.hk; plchen3@cityu.edu.hk).

Wenhui Wu is with the College of Electronics and Information Engineering, Shenzhen University, Shenzhen, China (email: wuwenhui@szu.edu.cn).

Xu Wang is with the College of Computer Science and Software Engineering, Shenzhen University, China, and also with Guangdong Laboratory of Artificial Intelligence and Digital Economy (SZ), Shenzhen University, Shenzhen, China, (email: wangxu@szu.edu.cn).

Sam Kwong is with the Department of Computing and Decision Sciences, Lingnan University, Hong Kong, China (email: samkwong@ln.edu.hk).

grouped automatically based on the foreground objects in a photo album.

The fundamental challenge in ISC is identifying common and unique contents, and compressing the respective content in an effective way. The common contents which share similar semantic information pertain to consistent objects or akin attributes that exist across multiple images. Instead of selecting representative images as the common information, we emphasize on recognizing and comprehending semantic information and identifying the common contents by detecting semantic objects, enabling more accurate extraction of common contents. Regarding common content compression, the relationships among images with low explicit correlations are characterized by an implicit neural representation. Compared to signal-level prediction [9], [10], [11] which may fail in complex geometric deformations, our approach models the inter-image relationship and facilitates the compact representation from a new perspective based on network learning. With respect to the semantically unique content, the considerable degree of intra-image redundancy is removed via an explicit image compression scheme by an invertible neural network.

More specifically, we propose a hybrid neural representation for ISC (HNR-ISC), and the advantages of the proposed scheme are shown in Fig. 1. The proposed scheme includes an implicit neural representation for Semantically Common content Compression (SCC) and an explicit neural representation for Semantically Unique content Compression (SUC), as illustrated in Fig. 2. To this end, we decompose the images into two distinct components: the common content and the unique content. In SCC, common contents are compactly represented through content-adaptive embeddings and a lightweight network shared by all images. In SUC, the unique content is obtained by removing the common content at the feature level, and an invertible neural network is learned to compress the unique content. The contributions of this work are summarized as follows,

- We propose a new image set compression framework HNR-ISC, including an SCC model to remove inter-image redundancies, and an SUC model for intra-image redundancy removal. The proposed framework achieves superior performance in terms of signal quality, perceptual quality, and accuracy on the downstream task.
- We develop an SCC model that efficiently compresses the semantically common content in an image set. SCC leverages a lightweight yet efficient neural representation for modeling common objects. As such, it is highly adaptive to different scenes, leading to promising efficiency in common content compression.
- We develop an SUC model that aims to compactly represent the unique content with the latent code from the invertible network. This enables the removal of the intra-image redundancy and ensures the robustness of the proposed compression framework.

II. Related Works

A. Image Set Compression

ISC aims to compress a collection of similar images. Unlike traditional image compression methods that focus on compressing individual images, ISC considers removing both inter-image and intra-image redundancies. Many existing approaches organize the images into a pseudo video, which is then compressed through existing codecs [9], [10], [11]. For example, Shi *et al.* [9] proposed a methodology that establishes a minimal-cost prediction structure through feature-based k-means clustering, followed by SIFT-based minimum spanning tree searching to generate a video. The video is then compressed via an existing video codec. Although this approach has proven effective for highly correlated images, its performance might be suboptimal for images with lower correlation. Additionally, Shi *et al.* introduced the pioneering multi-model prediction (MoP) method for ISC, which significantly reduces inter-image redundancy [10]. Zhang *et al.* [12] proposed a rate-distortion optimized sparse coding scheme, employing a reordered dictionary specifically designed for ISC. In contrast to the aforementioned algorithms, Wang *et al.* [13] presented a novel deep correlated ISC scheme based on distributed source coding and multi-scale image fusion.

B. Image Compression

Image compression aims to represent image signals compactly for efficient transmission and storage. Over the past decades, numerous image compression standards have been developed, such as JPEG [14], JPEG2000 [15], HEVC (Intra)[16], [17], and VVC (Intra) [18]. These standards typically rely on prediction, transform coding, and entropy coding techniques to reduce redundancies in the images.

Learning-based image compression has demonstrated remarkable advancements in compression performance, showcasing the capacity of neural networks to nonlinearly model visual signals, thereby enhancing compression efficiency [1], [19]. Researchers have been exploring various possibilities for the transform module in image compression [1], [2], [20], [3], [21], [22]. VAE-based models have gained prominence due to their exceptional performance and architectural stability [1], [2], [23]. However, these models do not directly tackle the issue of information loss during the encoding process. Invertible neural networks are generative models that simplify complex distributions, enabling precise and efficient estimation of probability density. Incorporating the invertible concept into the encoding-decoding process presents a promising solution to address the issue of information loss. By employing a bijective input-to-output mapping and strictly invertible characteristics, invertible neural networks offer a suitable framework for image compression [20], [3]. Xie *et al.* proposed a highly invertible architecture that significantly mitigates information loss during image compression [20]. Inspired by this work, Cai *et al.* implemented an invertible activation transformation module in a mathematically invertible manner, demonstrating its ability to achieve fine variable-rate control while better preserving image fidelity [3].

C. Implicit Neural Representation for Compression

Implicit neural representation (INR) has garnered significant attention for its ability to model various types of signals. It is achieved by parameterizing a signal with a function that

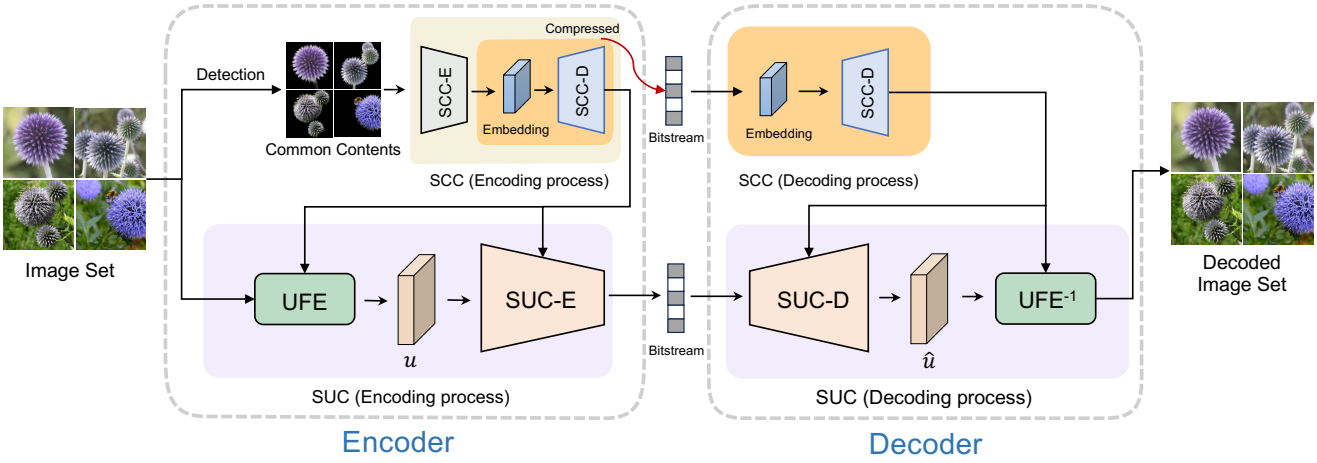

Fig. 2. Illustration of the framework of HNR-ISC, including SCC and SUC. The encoder includes semantically common content detection and the encoding process of SCC and SUC. On the decoder side, both the embedding and the parameters of SCC are extracted from the bitstream in the decoding process of SCC. These components work together to reconstruct common contents, enabling the decoding of unique content and resulting in the final decoded image set. Here, SCC-E and SCC-D represent the encoder and decoder of SCC, respectively. Analogously, SUC-E and SUC-D represent the encoder and decoder of SUC, respectively. u and $\hat{u}$ denote the unique feature and decoded unique feature, respectively.

generates desired properties based on the input. Consequently, the signal is implicitly encoded within the parameters of the network.

In image compression, INR has emerged as a promising approach that harnesses the power of neural networks to compress and decompress images without explicitly encoding pixel values [24], [25]. Strumpler *et al.* [24] introduced meta-learned initializations for INR-based compression, which can improve rate-distortion performance. Subsequently, they proposed a simple yet highly effective modification to the network architecture compared to previous works. The COIN model proposed by Dupont *et al.* [26] stores the weights of an overfitted neural network instead of storing RGB values for each pixel in an image. Furthermore, they developed COIN++, an advanced neural compression framework that seamlessly handles a wide range of data modalities [27]. In the field of video compression, INR-based video compression schemes have made significant advancements. Chen *et al.* [25] proposed a novel neural representation for videos (NeRV), which encodes videos in neural networks. Then, they proposed a hybrid neural representation to store videos. This approach offers decoding advantages in terms of speed and flexibility when compared to traditional codecs.

After obtaining INR, another key issue is model compression as models govern the bitstream. Model compression aims to reduce the size and complexity of neural networks [28], [29]. Model quantization is an essential part of model compression. In particular, it typically reduces the precision of weights and activations in the model. By representing numerical values with fewer bits with fixed-point quantization and dynamic range quantization, the size of the model can be significantly reduced [25], [30].

III. The Proposed Method

A. Overview

We propose a hybrid neural representation approach for compressing image sets. This approach leverages both implicit and explicit neural representations to effectively reduce inter- and intra- image redundancy. By disentangling images within a set into common and unique contents, we achieve efficient compression using two distinct models: the SCC model for common content compression and the SUC model for unique content compression. Our proposed scheme exhibits high adaptability to various types of image sets and application scenarios.

At the encoder side, we begin by extracting the semantically common contents using a saliency detection approach, followed by the extraction of semantically unique contents with the assistance of the common content through the UFE module. To achieve this, we utilize U^2-Net [31], in combination with the alpha matting technique [32] for extracting the common contents. Then, the semantically common contents are represented through the SCC model, which consists of an SCC encoder (SCC-E) for generating the content-adaptive embedding, and an SCC decoder (SCC-D). Only the embedding and SCC-D are compressed and transmitted as the final bitstream. Following this vein, the semantically unique contents are compressed with guidance from the decoded common contents through the SUC encoder (SUC-E), as illustrated in Fig. 3.

For the decoder, the received SCC decoder (SCC-D) reconstructs the common contents from the compact embedding. Subsequently, the SUC decoder (SUC-D) decompresses the semantically unique contents with the help of the common contents. Finally, the images are reconstructed by merging the decoded common contents and unique contents with the UFE^{-1} module.

B. Semantically Common Content Compression

Generally speaking, the different deformations of common contents make it challenging to model the explicit inter-image correlation. Thus, we develop the INR model for SCC. In contrast to explicit compression methods [1], [3], INR based compression methods store all information implicitly in the

network weights θ. Typically, they take the coordinate $(\mathbf{x}, \mathbf{y})$ as the input. The encoding process is akin to training the INR, which can be represented as:

$$\theta^\star = \arg\min_{\theta} \mathcal{L}\left(\mathbf{c}, f_\theta(\mathbf{x}, \mathbf{y})\right), \tag{1}$$

where $f_\theta(*)$ represents the INR network parameterized by θ, and $\theta^\star$ denotes the optimized weight. Then, the model weights $\theta^\star$ undergo compression and decompression procedures. The decoding process involves restoring decoding weights into the network to generate the RGB color from the coordinate (x, y). This can be expressed as,

$$\widehat{\mathbf{c}} = f_{\theta^\star}(x, y), \tag{2}$$

where $\mathbf{c}$ and $\widehat{\mathbf{c}}$ are the original RGB color value and the decoded RGB color value, respectively.

However, using the positional index as input and generating fixed and content-agnostic embeddings significantly limit the regression capacity [30]. We design the SCC model following the auto-encoder structure. The process can be described as follows:

$$\alpha^\star, \theta^\star = \arg\min_{\alpha,\theta} \mathcal{L}\left(I_c, \mathbf{D}_\theta(\mathbf{E}_\alpha(I_c))\right), \tag{3}$$

where I_c is the semantically common contents. A trainable encoder $\mathbf{E}_\alpha$ produces a compact content-adaptive embedding $\mathbf{e}$, and the decoder $\mathbf{D}_\theta$ reconstructs the common content from this embedding. Through training this model, we can obtain the optimal weights ($\alpha^\star$ and $\theta^\star$) of the model. Instead of compressing the entirety of weights within the SCC model, we only compress the weight of the decoder $\theta^\star$ plus the embedding $\mathbf{e}$, where $\mathbf{e} = \mathbf{E}_\alpha(I_c)$. The decoding procedure entails restoring these decompressed weights $\widehat{\theta^\star}$ into the network to reconstruct the image from the decoded embedding ($\hat{\mathbf{e}}$):

$$\hat{I}_c = \mathbf{D}_{\hat{\theta}^\star}(\hat{\mathbf{e}}), \tag{4}$$

where $\hat{I}_c$ is the decoded semantically common contents.

Model architecture. The encoder consists of multiple encoding modules $\mathbf{E}_i$, where i denotes the index. More specifically, $\mathbf{E}_i$ contains a Down block and a dilated parallel residual block (DPR block) [33]. In the DPR block, the input feature undergoes a convolutional layer with a 1×1 kernel size to obtain the feature f. Subsequently, we denote the convolutional operation as $\mathbf{C}_{dr}$ for convenience, where dr means the dilation radius ($dr = \{1, 2, 4, 8, 16\}$). $\mathbf{C}_{dr}$ employs the same kernel size (3×3). The operation in the DPR block can be described as follows:

$$\begin{aligned} d_1 &= \mathbf{C}_1(f) + \mathbf{C}_2(f), \\ d_2 &= d_1 + \mathbf{C}_4(f), \\ d_3 &= d_2 + \mathbf{C}_8(f), \\ d_4 &= d_3 + \mathbf{C}_{16}(f), \\ r &= d_1 \cup d_2 \cup d_3 \cup d_4, \\ \hat{f} &= f + r, \end{aligned} \tag{5}$$

where $\cup$ represents the concatenation operation. The DPR block can increase the receptive field for encoding and decoding. The compact content-adaptive embedding $\mathbf{e}$ can be obtained through multiple $\mathbf{E}_i$ as follows:

$$\mathbf{e} = \mathbf{E}_k(..\mathbf{E}_2(\mathbf{E}_1(I_c))), \tag{6}$$

where k is the number of blocks.

The decoder consists of k decoding modules, which can be adjusted according to the image resolution to get a tiny embedding. Each decoding module $\mathbf{D}_i$ contains a DPR block and an Up block. The DPR block has the same structure as those in the encoder. The Up block comprises a convolution layer, a pixel shuffle layer, and an activation layer. Among these layers, only the convolution layer has learnable parameters. To improve convergence speed and enhance the quality of reconstruction, we propose the integration of a shortcut mechanism following the $\mathbf{D}_i$ module.

$$\begin{aligned} d_1 &= \mathbf{D}_1(\mathbf{e}) + \mathbf{U}_1(\mathbf{e}), \\ d_2 &= \mathbf{D}_2(d_1) + \mathbf{U}_2(\mathbf{e}), \\ &\cdots, \\ d_{k-1} &= \mathbf{D}_{k-1}(d_1) + \mathbf{U}_{k-1}(\mathbf{e}), \\ \hat{I}_c &= \mathbf{D}_k(d_{k-1}). \end{aligned} \tag{7}$$

$\mathbf{U}_i$ is the interpolation operator, which interpolates the embedding to match the size of the features before adding them together. Finally, we optimize the entire model via the L_2 loss.

Model compression. Due to the high-efficiency representation of INR, semantically common content compression can be transformed into model compression. By employing the content-adaptive embedding and SCC-D components, semantically common contents can be reconstructed. The key to reduce these two components lies in reducing model redundancy, which primarily exists in a large number of high-precision parameters. Thus, we utilize post-training quantization (PTQ) [34], [30], which enables us to adjust the precision of the weights without the fine-tuning procedure. The formula for quantization is presented below:

$$\theta_i = \text{round}\left(\frac{\theta_i - \theta_{\min}}{S}\right) * S + \theta_{\min}, \tag{8}$$

where

$$S = \frac{\theta_{\max} - \theta_{\min}}{2^{\mathrm{b}} - 1}. \tag{9}$$

Herein, the term "round" refers to the procedure of rounding a given value to the nearest integer. The variable "b" represents the bit length for the quantized model. $\theta_{\max}$ and $\theta_{\min}$ stand for the maximum and minimum values of the parameter tensor θ, respectively. The scaling factor is denoted by the variable S, and each parameter can be assigned a value based on Eqn. (8) and (9). After applying weight quantization, we utilize Huffman Coding [35], a lossless compression method, to compress the quantized weights.

C. Semantically Unique Content Compression

The SUC model aims to eliminate intra-image redundancy within unique contents, such that the unique contents can be effectively compressed. In analogous to numerous image compression methods [2], [1], our model adopts the autoencoder structure. However, conventional autoencoders encounter information loss issues when transforming images into a low-dimensional latent space [20], [3]. To alleviate information loss, we design an invertible module in the SUC model. This module ensures that both the forward and inverse

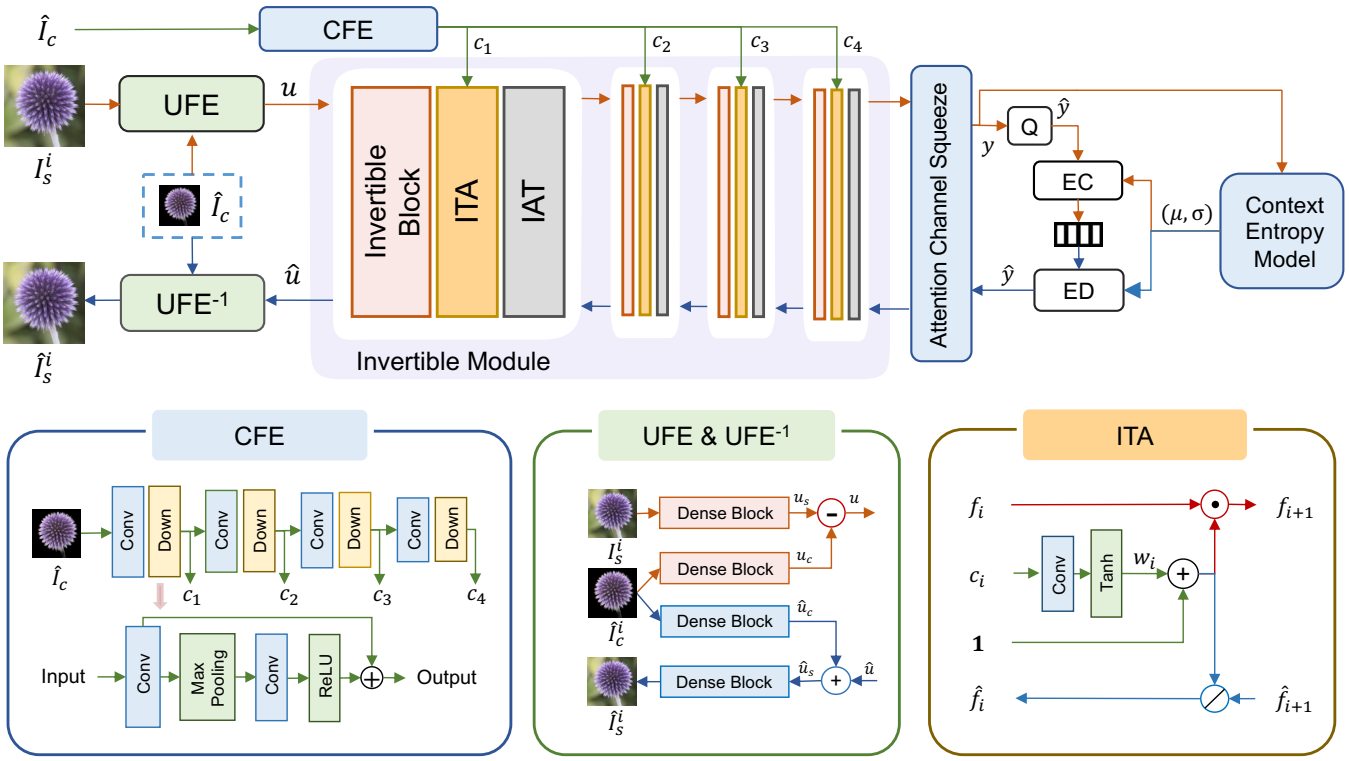

Fig. 3. Illustration of the workflow of the SUC model. The forward processing (encoding) is shown through the red arrows, while the inverse processing (decoding) is represented via the blue arrows. The green arrows represent the process executed in both the forward and inverse processing. In UFE & UFE^{-1}, the red arrows represent the encoding process in the UFE module and the blue arrows represent the decoding process in the UFE^{-1} module. The "Dense Block" shares the same structure in [3].

procedures possess mathematical invertibility, simultaneously enabling adaptive weight adjustments tailored to enhance the fidelity of the semantically unique contents.

Model architecture. As shown in Fig. 3, the unique content u is first extracted from the i-th image I_s^i in the image set via the unique feature extraction module (UFE). Subsequently, the compact representation y of the unique content is obtained through four invertible modules and an attention channel squeeze. This 4-layer structure resembles that of most invertible codecs and learning-based image codecs [3], [20], [1]. In our model, each invertible module consists of an invertible block, an invertible texture attention (ITA) block, and an invertible activation transformation (IAT) block. Next, y is quantized and compressed into a bitstream via a content entropy model [3] and entropy coding (EC). In the decoder, the latent representation $\hat{y}$ is decoded using the entropy model [3]. The invertible modules in the decoder share the same parameters as the encoder, but they operate reversely. Through the four invertible modules, the decoded common content $\hat{u}$ is reconstructed. Finally, the decoded image $\hat{I}_s^i$ is reconstructed via the UFE^{-1} module.

Unique feature extraction module (UFE). To extract the unique content, we design a UFE module, which efficiently extracts the unique content with the assistance of the decoded common contents. In the forward encoding, the input image I_s^i extracts the features $u_s \in \mathbb{R}^{3\times H\times W}$ via the Dense block ($\mathcal{DB}$) [3], while the decoded common content conducts the same operation to obtain the common features $u_c \in \mathbb{R}^{3\times H\times W}$. In this manner, the unique content $u \in \mathbb{R}^{3\times H\times W}$ can be obtained via the subtraction of the features,

$$u_s = \mathcal{DB}(I_s^i), \tag{10}$$

$$u_c = \mathcal{DB}(I_c^i), \tag{11}$$

$$u = u_s - u_c. \tag{12}$$

In the decoding process, the decoded unique features $\hat{u}$ and the common features $\hat{u}_c$ are combined via pixel-wise addition:

$$\hat{u}_c = \mathcal{DB}(I_c^i), \tag{13}$$

$$\hat{u}_s = \hat{u} + \hat{u}_c, \tag{14}$$

$$\hat{I}_s^i = \mathcal{DB}(\hat{u}_s). \tag{15}$$

The images are reconstructed via the Dense block.

Invertible module. The invertible module is composed of three invertible blocks: an invertible block, an ITA block, and an IAT block. The invertible block and IAT block are the same as the structure proposed in [20], [3]. The invertible block serves as the fundamental element in the invertible module, and it performs a transformation on the input features. The ITA block is placed after the invertible block for further processing of the important areas of the transformed features. This block allocates more weight to the texture reconstruction of the important area. Finally, after the ITA block, the IAT block

generates global element-wise activation features based on the input quality level.

The common content feature c_i (i=$\{1, 2, 3, 4\}$) is the input of the ITA block. c_i is extracted by the feature extraction module, which includes convolution (Conv) and down-resolution (Down) layers, as shown in Fig. 3 ("Feature Extraction Module"). The Down layer consists of a convolution operation with a stride of 2 and kernel size of 3, a max pooling operation with the size of 2, a convolution operation with a kernel size of 3 and a stride of 1, an activation function and a shortcut connection. After the Down layer, the common feature resolution decreases to the same resolution as the unique feature. It should be noted that these features c_i are available and shared in the encoder and decoder.

The forward transform of the ITA block is illustrated by red arrows on the top of Fig. 3 ("ITA"). The inputs include the unique features $f_i \in \mathbb{R}^{c\times h\times w}$, the common features $c_i \in \mathbb{R}^{c\times h\times w}$ and the ones features $\mathbf{1} \in \mathbb{R}^{c\times h\times w}$. The common features c_i are used to represent the important area. Specifically, the weight map $w_i \in [-1, 1]$ for the important area is obtained from c_i via a convolution layer and a Tanh activation function. Then, the feature f_i is pixel-wise adjusted to recursively generate the feature f_{i+1} using the following equation:

$$f_{i+1} = f_i \odot (\mathbf{1} + w_i), \tag{16}$$

where $\odot$ denotes the Hadamard product. The inverse transform of the ITA block is illustrated by blue arrows at the bottom of Fig. 3 ("ITA"). This inverse transform is formulated as:

$$\hat{f}_i = \hat{f}_{i+1} \oslash (\mathbf{1} + w_i), \tag{17}$$

where $\oslash$ denotes the element-wise division.

The training loss is defined as,

$$\mathcal{L}_c = R + \lambda \mathcal{L}_{re}(I_s^i, \hat{I}_s^i), \tag{18}$$

where I_s^i and $\hat{I}_s^i$ are the input image and the decoded image, respectively. $R = E_{P_{\hat{y}|\hat{z}}}[\log P_{\hat{y}|\hat{z}}] + E_{P_{\hat{z}}}[-\log P_{\hat{z}}]$ represents the coding rate, where $\hat{y}$ and $\hat{z}$ are the quantized latent representations and side information, respectively. $\mathcal{L}_{re}$ estimates the mean squared error between I_s^i and $\hat{I}_s^i$. λ is positively correlated with the quality level.

IV. Experiments

A. Implementations and Settings

Datasets. Regarding SCC, given the image set for compression, we train a model for this image set. For the training of SUC, we use the Flick 30K dataset [36] with data augmentation through random cropping of 256×256 images. To verify the adaptability of the model to diverse image scenarios, we evaluate the rate-distortion performance on three datasets with different image sizes and numbers: the Oxford flowers dataset (256×256, 6149) [37], the FFHQ dataset (512×512, 1000) [38] and the Chest X-Ray dataset (1024×1024, 234) [39]. Due to the large scale of the FFHQ dataset, we selected the first 1000 images for testing.

Quality evaluation measures. To evaluate the performance of our proposed approach, we employ Peak Signal-to-Noise Ratio (PSNR) and Multi-Scale Structural Similarity Index Measure (MS-SSIM) [40] as the evaluation measures because PSNR and MS-SSIM are commonly used as benchmarks in the field of image compression. To provide a more comprehensive evaluation of our approach, we further use DISTS [41] and LPIPS [42] to evaluate the perceptual quality since these quality measures are specifically designed to quantify the perceptual quality of images. In particular, a lower DISTS/LPIPS value indicates better perceptual quality. Based on these additional measures, we can obtain a more accurate assessment of the visual quality. Furthermore, we extend our evaluation by utilizing FGVC-PIM[1], a classification algorithm, on the Oxford flowers dataset. In addition to the quality evaluation measures, we also consider the coding bitrate, measured in bits per pixel (bpp).

Implementation details. The network is implemented via the PyTorch framework on NVIDIA GeForce RTX 3090 GPUs. The SCC model undergoes training for 350 epochs, using a batch size of 2. These choices of epoch and batch size aim to provide sufficient time for the model to converge and effectively learn the underlying patterns in the data. For the training of SUC, we made a slight modification to the batch size by increasing it to 12 while keeping other settings consistent with the one mentioned in [3].

B. Performance Comparisons

To verify the effectiveness of the proposed scheme, the following image compression schemes are involved for performance comparisons. Herein, we employ state-of-the-art codecs for comparison, including VVC, Ballé *et al.*'s method [2], Minnen *et al.*'s method [4] and Cai *et al.*'s method [3]. We employ the VVC test model (VTM-15.2) [43] under the all intra (AI) setting with quantization parameters QPs = $\{42, 37, 32, 27\}$, and higher QP values correspond to higher compression ratios. The training and testing strategies of Ballé *et al.*'s [2], Minnen *et al.*'s and Cai *et al.*'s [3] methods follow the official codes. All the training models are trained on the Flick 30K dataset to ensure fairness. Due to the special property of the Chest X-Ray dataset, we retrain all models on the Chest X-Ray dataset.

Comparison in terms of the reconstruction quality. We conduct Rate-Distortion (RD) performance comparisons on the testing set by compressing all images with different quality factors. As shown in Fig. 4 and Table I, we compare our model with the state-of-the-art (SOTA) methods on three datasets: the Oxford Flowers dataset, the FFHQ dataset, and the Chest X-Ray dataset, from left to right. We use PSNR, MS-SSIM, LPIPS, and DISTS as the quality evaluation measures. Due to the specificity of the X-Ray dataset, the SCC is applied to compress the whole image. Our results demonstrate that our model achieves excellent performance in terms of both signal quality and perceptual quality on the Oxford Flowers dataset. More specifically, compared with the Balle *et al.*'s model, our proposed model achieves the 42.67%, 46.84%, 57.10% and 48.88% BDBR reduction in terms of PSNR, MS-SSIM, LPIPS and DISTS, respectively. On the FFHQ dataset, our

[1]https://github.com/chou141253/FGVC-PIM

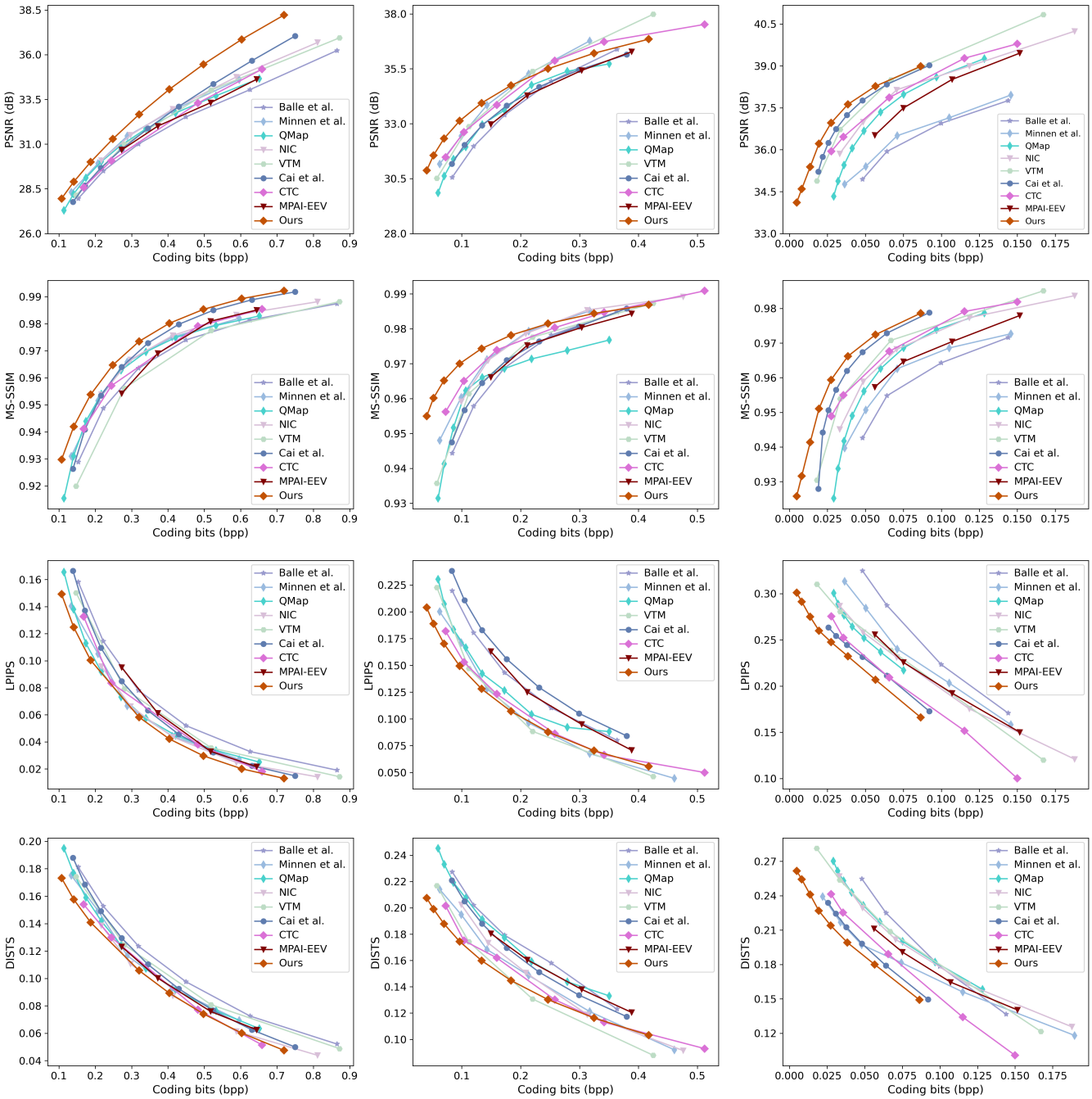

Fig. 4. The comparisons of rate-distortion performance on the Oxford Flowers dataset, the FFHQ dataset, and the Chest X-Ray dataset in terms of PSNR, MS-SSIM, LPIPS, and DISTS.

model outperforms other methods at low-bit rates, including Cai *et al.*'s model. For the Chest X-Ray dataset, our model demonstrates promising performance in terms of both fidelity and perceptual quality, achieving BDBR reductions of 72.31% in PSNR, 64.79% in MS-SSIM, 52.42% in LPIPS, and 45.42% in DISTS against the Balle *et al.*'s model.

Moreover, we show the visual quality comparisons in Fig. 5. In particular, our approach demonstrates effective preservation of texture details, particularly in the foreground of the reconstructed images, as seen in the center of the flower in the first and second rows in Fig. 5. Moreover, when compared to other existing models such as Cai *et al.*'s model and VTM, our model showcases better robustness against variations in content type and resolution. Reconstructed images from Cai *et al.*'s model tend to suffer from blurriness, indicating difficulties in accurately representing finer details. Analogously, the images produced by VTM exhibit blocking artifacts that diminish the overall quality of the decoded images. It is worth noting that the performance of the model is not limited to a specific dataset or category of images. Rather, it demonstrates strong generalization capability across different datasets and produces high-quality reconstructions for various types of images. This makes our approach more versatile and applicable to a wide range of real-world scenarios.

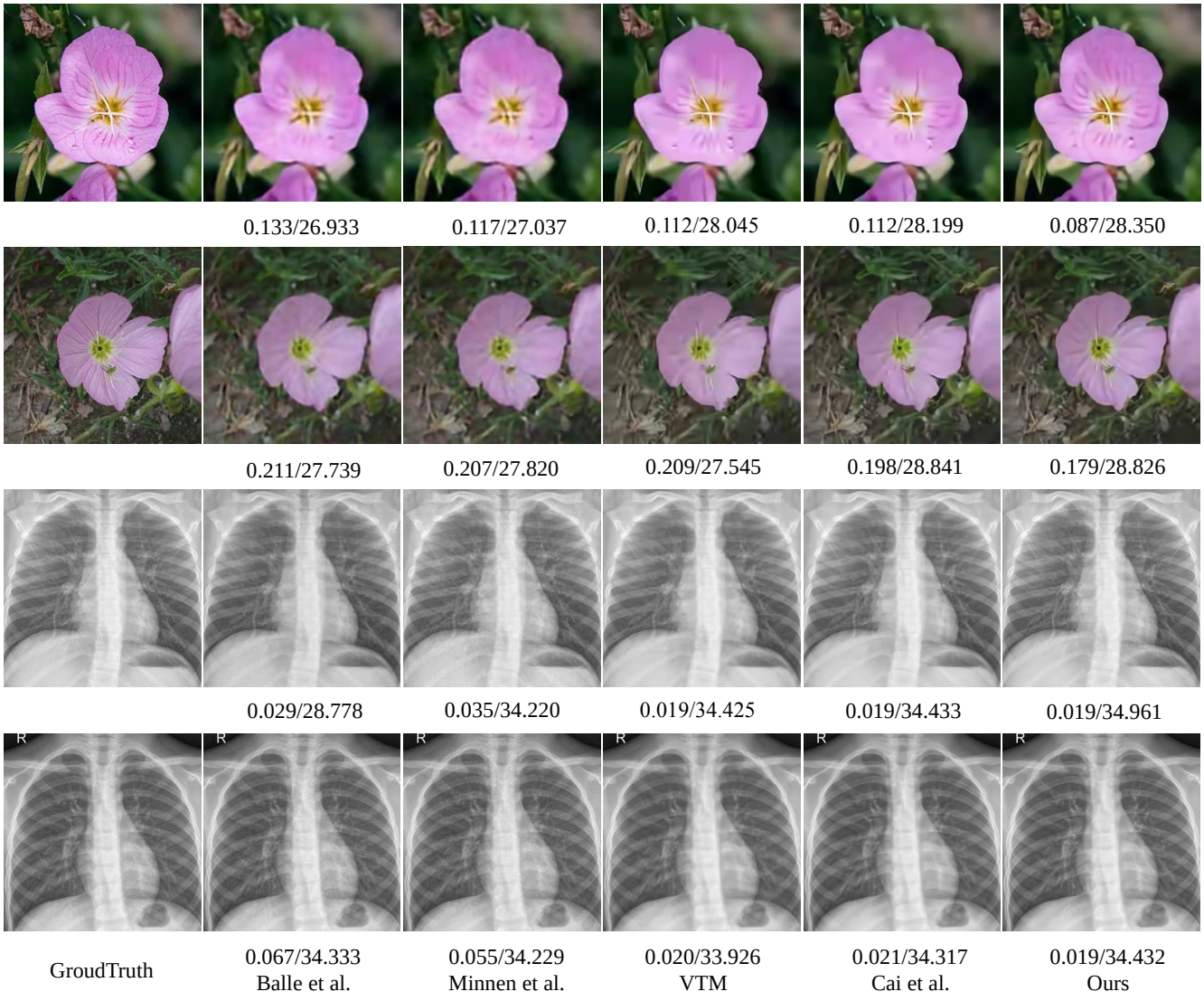

Fig. 5. Visual quality comparison of different methods. The images in both the first and second rows are from the same image set, and similarly, the images in the last two rows also belong to the identical image set. The values below each image are coding bits(bpp)/PSNR(dB) values, where a higher PSNR value represents better signal quality.

Comparisons in terms of the downstream task. To evaluate the effectiveness of our proposed framework for the downstream task, we employ the FGVC-PIM model [44], known for its high accuracy in classifying pristine images on the Oxford flower dataset. We specifically focus our evaluation on the Oxford flower dataset as it is the only dataset designed for classification. The result, shown in Fig. 6, is measured using accuracy (%) as the evaluation measure. Our results show much higher accuracy at the same bitrates, particularly at low bit rates. It indicates that our framework exhibits both robustness and high-quality performance to retain important visual features and information essential for image classification.

C. Ablation Studies

To evaluate the performance of each component, we conduct ablation studies on the SCC and SUC models.

The SCC model. Our proposed SCC employs DPR blocks and a dedicated shortcut to obtain a compact implicit representation of the common content. Therefore, we conduct three ablation studies on this model, including "Ours (w/o shortcut)", "Ours (w/o DPR)" and "Ours (w/o DPR & shortcut)". In "Ours (w/o shortcut)", all shortcut connections are removed. In "Ours (w/o DPR)", all DPR blocks are deleted. In "Ours (w/o DPR & shortcut)", both the DPR blocks and the shortcut connections are detached. The comparison results shown in Table II indicate that the shortcut connection promotes image reconstruction. We assess different methods with BDBR [45]. When we delete all DPR blocks from the model, the results drop dramatically. This indicates the DPR blocks contribute significantly to capturing the underlying structure of the data and improving the performance of the model. The results in "Ours(w/o DPR & shortcut)" suggest that the DPR blocks and shortcuts are necessary for achieving high accuracy.

The SUC model. To verify the efficiency of the UFE module and ITA block, we initially examine the performance of our proposed scheme without the UFE module, referred to as "Ours (w/o UFE)". In this evaluation, we use the

TABLE I
THE BDBR(%) PERFORMANCES IN TERMS OF PSNR, MS-SSIM, LPIPS, AND DISTS ON THE OXFORD FLOWERS DATASET, THE FFHQ DATASET, AND THE CHEST X-RAY DATASET (ANCHOR: BALLE *et al.*'S MODEL).

Datasets	**Oxford flower dataset**				**FFHQ dataset**				**Chest X-Ray dataset**			
Methods	PSNR	MS-SSIM	LPIPS	DISTS	PSNR	MS-SSIM	LPIPS	DISTS	PSNR	MS-SSIM	LPIPS	DISTS
Minnen *et al.*	-14.41	-12.13	-56.43	-38.75	-12.19	-34.92	-82.23	-64.94	-15.24	-21.43	-33.11	-11.80
VTM	-29.18	-22.43	-14.41	-11.64	-80.17	-66.36	-23.40	-16.68	-62.47	-48.51	-20.47	-24.43
Cai *et al.*	-29.95	-41.71	**-57.46**	-39.79	-75.01	-74.82	-73.57	-50.71	-67.00	-55.66	-42.37	-37.05
Ours	**-42.67**	**-46.84**	-57.10	**-48.88**	**-80.37**	**-92.02**	**-98.11**	**-75.63**	**-72.31**	**-64.79**	**-52.42**	**-45.42**

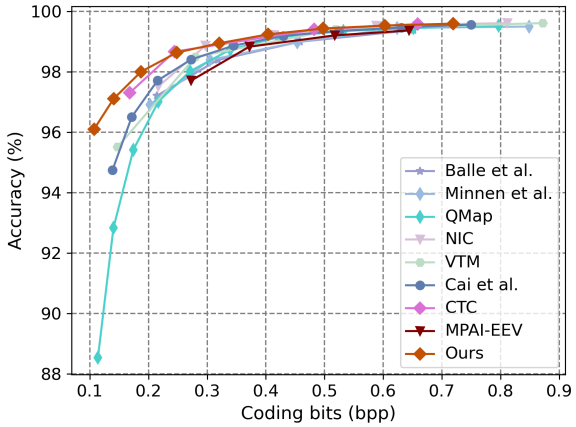

Fig. 6. The comparisons of rate-distortion performance on the Oxford flower dataset in terms of accuracy (%).

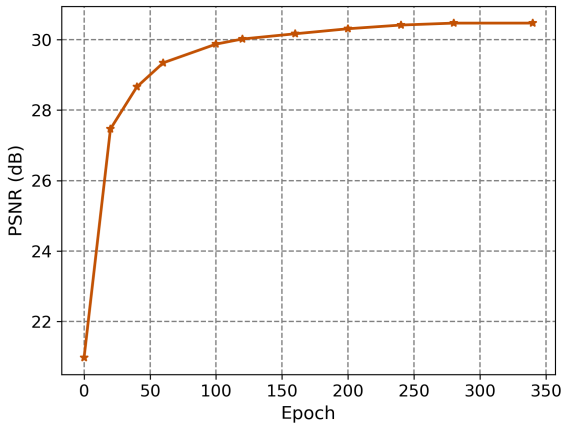

Fig. 7. The influence of the training epoch number on the reconstruction quality. In particular, increasing the epoch number leads to a corresponding increase in the quality of the reconstructed images. When the number of epochs reaches 300, the PSNR growth saturates.

TABLE II
THE RESULTS OF ABLATION STUDIES ON THE MODEL OF SCC AND SUC (ANCHOR: THE PROPOSED MODEL).

Models	BDBR(%)
Ours(w/o shortcut)	1.6
Ours(w/o DPR)	7.6
Ours(w/o DPR & shortcut)	8.2
Ours(w/o UFE)	18.8
Ours(w/o ITA)	5.4
Ours(w/o UFE & ITA)	11.0

TABLE III
THE COMPARISON OF AVERAGE ENCODING AND DECODING TIME (S).

Time (s)	VTM	Cai *et al.*	Ours
Encoder	13.240	0.635	28800
Decoder	0.007	1.455	1.492

Dense block instead of the UFE module, and omit the input of the decoded common content. The results indicate that the absence of these modules significantly deteriorates the compression performance compared to our proposed scheme. Subsequently, we evaluate the performance of our proposed scheme without the ITA block, denoted as "Ours (w/o ITA)". The findings highlight the crucial role played by the ITA block in enhancing the reconstruction of the foreground content. The ITA block can capture the underlying structure of the foreground contents by reweighting the features, enabling the model to prioritize the foreground areas during reconstruction and thereby improving its ability to restore the foreground contents faithfully. Finally, we assess the performance of our proposed scheme without UFE and ITA blocks, labeled as "Ours (w/o UFE & ITA)". Thus, there is no need for additional bit consumption for the common content. In this manner, the result is better than "Ours (w/o ITA)", but does not outperform the proposed scheme. This further demonstrates the importance of the UFE module and the ITA block in achieving better image compression performance.

Encoding and decoding complexity analysis. We further conduct an evaluation of the encoding and decoding complexities of several compression schemes and present the results in Table III. The table compares the average running time of encoders and decoders for each method on an image set with 50 images. The QP setting in VTM is 47. Specifically, the training duration for the SCC model typically spans approximately 8 hours. As our model is conditioned on the decoded common

contents, it exhibits a higher level of decoding complexity than Cai *et al.*'s method. It is worth noting that VTM shows the lowest level of decoding complexity.

Moreover, we conduct experiments to study the influence of iteration on the quality of the reconstructed common content. The results are shown in Fig. 7. We can observe that increasing the epoch number leads to a corresponding increase in the quality of the reconstructed content. However, when the number of epochs reaches 300, the PSNR growth saturates. Different image sets require different iterations for training, such that we trained three image sets for 350 epochs to ensure that the model has sufficient time to converge and learn the underlying patterns.

V. Conclusions

The novelty of this paper lies in a novel hybrid image set representation and compression framework, which has been validated via the compression of several typical image sets. The proposed HNR-ISC addresses the challenge of efficiently eliminating redundancy among inter images via the SCC model and within a single image via the SUC model. The SCC model compactly represents the common contents of the image set using an implicit neural representation, which is then compressed with model compression techniques. The SUC model employs an invertible neural network for unique feature extraction and invertible representation. Through extensive evaluations, our proposed scheme demonstrates superior performance in signal quality, perceptual quality, and high accuracy on the downstream task.

References

[1] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," *International Conference on Learning Representations (ICLR)*, 2016.

[2] 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.

[3] S. Cai, Z. Zhang, L. Chen, L. Yan, S. Zhong, and X. Zou, "High-fidelity variable-rate image compression via invertible activation transformation," *in Proceedings of the ACM International Conference on Multimedia*, p. 2021–2031, 2022.

[4] D. Minnen, J. Ballé, and G. D. Toderici, "Joint Autoregressive and Hierarchical Priors for Learned Image Compression," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[5] K. Karadimitriou, "Set redundancy, the enhanced compression model, and methods for compressing sets of similar images," *Louisiana State University and Agricultural & Mechanical College*, 1996.

[6] K. Karadimitriou and J. M. Tyler, "Min-max compression methods for medical image databases," *ACM SIGMOD Record*, vol. 26, no. 1, pp. 47–52, 1997.

[7] K. Karadimitriou and J. M. Tyler, "The centroid method for compressing sets of similar images," *Pattern Recognition Letters*, vol. 19, no. 7, pp. 585–593, 1998.

[8] C.-H. Yeung, O. C. Au, K. Tang, Z. Yu, E. Luo, Y. Wu, and S. F. Tu, "Compressing similar image sets using low frequency template," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2011.

[9] Z. Shi, X. Sun, and F. Wu, "Feature-based image set compression," *IEEE International Conference on Multimedia and Expo (ICME)*, pp. 1–6, 2013.

[10] Z. Shi, X. Sun, and F. Wu, "Multi-model prediction for image set compression," *Visual Communications and Image Processing (VCIP)*, pp. 1–6, 2013.

[11] X. Zhang, Y. Zhang, W. Lin, S. Ma, and W. Gao, "An inter-image redundancy measure for image set compression," *IEEE International Symposium on Circuits and Systems (ISCAS)*, pp. 1274–1277, 2015.

[12] X. Zhang, W. Lin, Y. Zhang, S. Wang, S. Ma, L. Duan, and W. Gao, "Rate-distortion optimized sparse coding with ordered dictionary for image set compression," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 28, no. 12, pp. 3387–3397, 2017.

[13] J. Wang, Y. Shi, Y. Xing, N. Ling, and B. Yin, "Deep correlated image set compression based on distributed source coding and multi-scale fusion," *Data Compression Conference (DCC)*, pp. 192–201, 2022.

[14] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[15] M. Rabbani and R. Joshi, "An overview of the JPEG 2000 still image compression standard," *Signal processing: Image communication*, vol. 17, no. 1, pp. 3–48, 2002.

[16] G. J. Sullivan, J.-R. 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.

[17] L. Zhu, S. Kwong, Y. Zhang, S. Wang, and X. Wang, "Generative adversarial network-based intra prediction for video coding," *IEEE Transactions on Multimedia*, vol. 22, no. 1, pp. 45–58, 2019.

[18] B. Bross, Y.-K. Wang, Y. Ye, S. Liu, J. Chen, G. J. Sullivan, and J.-R. Ohm, "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 31, no. 10, pp. 3736–3764, 2021.

[19] M. Akbari, J. Liang, J. Han, and C. Tu, "Learned multi-resolution variable-rate image compression with octave-based residual blocks," *IEEE Transactions on Multimedia*, vol. 23, pp. 3013–3021, 2021.

[20] Y. Xie, K. L. Cheng, and Q. Chen, "Enhanced invertible encoding for learned image compression," *in Proceedings of the ACM International Conference on Multimedia*, pp. 162–170, 2021.

[21] P. Zhang, M. Wang, B. Chen, R. Lin, X. Wang, S. Wang, and S. Kwong, "Learning-based compression for noisy images in the wild," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2022.

[22] H. Ma, D. Liu, R. Xiong, and F. Wu, "iwave: Cnn-based wavelet-like transform for image compression," *IEEE Transactions on Multimedia*, vol. 22, no. 7, pp. 1667–1679, 2019.

[23] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Energy compaction-based image compression using convolutional autoencoder," *IEEE Transactions on Multimedia*, vol. 22, no. 4, pp. 860–873, 2019.

[24] Y. Strümpler, J. Postels, R. Yang, L. V. Gool, and F. Tombari, "Implicit neural representations for image compression," *in Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 74–91, 2022.

[25] H. Chen, B. He, H. Wang, Y. Ren, S. N. Lim, and A. Shrivastava, "Nerv: Neural representations for videos," *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, pp. 21 557–21 568, 2021.

[26] E. Dupont, A. Goliński, M. Alizadeh, Y. W. Teh, and A. Doucet, "Coin: Compression with implicit neural representations," *arXiv preprint arXiv:2103.03123*, 2021.

[27] E. Dupont, H. Loya, M. Alizadeh, A. Golinski, Y. W. Teh, and A. Doucet, "COIN++: Neural compression across modalities," *Transactions on Machine Learning Research*, vol. 2022, no. 11, 2022.

[28] Z. Li, B. Ni, T. Li, X. Yang, W. Zhang, and W. Gao, "Residual quantization for low bit-width neural networks," *IEEE Transactions on Multimedia*, vol. 25, pp. 214–227, 2023.

[29] W. Duan, Z. Liu, C. Jia, S. Wang, S. Ma, and W. Gao, "Differential weight quantization for multi-model compression," *IEEE Transactions on Multimedia*, vol. 25, pp. 6397–6410, 2023.

[30] H. Chen, M. Gwilliam, S.-N. Lim, and A. Shrivastava, "Hnerv: A hybrid neural representation for videos," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10 270–10 279, 2023.

[31] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-net: Going deeper with nested u-structure for salient object detection," *Pattern recognition*, vol. 106, p. 107404, 2020.

[32] T. Germer, T. Uelwer, S. Conrad, and S. Harmeling, "Pymatting: A python library for alpha matting," *Journal of Open Source Software*, vol. 5, no. 54, p. 2481, 2020.

[33] S. Mehta, M. Rastegari, A. Caspi, L. Shapiro, and H. Hajishirzi, "Espnet: Efficient spatial pyramid of dilated convolutions for semantic segmentation," *in Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 552–568, 2018.

[34] A. Gholami, S. Kim, Z. Dong, Z. Yao, M. W. Mahoney, and K. Keutzer, "A survey of quantization methods for efficient neural network inference," *Low-Power Computer Vision*, pp. 291–326, 2022.

[35] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural network with pruning, trained quantization and huffman coding," *International Conference on Learning Representations (ICLR)*, 2016.
[36] P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, "From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions," *Transactions of the Association for Computational Linguistics*, vol. 2, pp. 67–78, 2014.
[37] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," *Indian Conference on Computer Vision, Graphics & Image Processing*, pp. 722–729, 2008.
[38] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1867–1874, 2014.
[39] D. Kermany, K. Zhang, M. Goldbaum *et al.*, "Labeled optical coherence tomography (oct) and chest x-ray images for classification," *Mendeley data*, vol. 2, no. 2, p. 651, 2018.
[40] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multiscale structural similarity for image quality assessment," *Asilomar Conference on Signals, Systems & Computers*, vol. 2, pp. 1398–1402, 2003.
[41] K. Ding, K. Ma, S. Wang, and E. Simoncelli, "Image quality assessment: Unifying structure and texture similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2020.
[42] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," *in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 586–595, 2018.
[43] "VVC software vtm-15.2," https://vcgit.hhi.fraunhofer.de/jvet/ VVCSoftware_VTM/-/tags/VTM-15.2, online; accessed February 2022.
[44] P.-Y. Chou, C.-H. Lin, and W.-C. Kao, "A novel plug-in module for fine-grained visual classification," *arXiv preprint arXiv:2202.03822*, 2022.
[45] G. Bjontegaard, "Calculation of average psnr differences between rd-curves," *ITU SG16 Doc. VCEG-M33*, 2001.