Facial Image Compression via Neural Image Manifold Compression

Wenhan Yang, *Member, IEEE*, Haofeng Huang, *Student Member, IEEE*, Jiaying Liu, *Senior Member, IEEE*, and Alex C. Kot, *Life Fellow, IEEE*

***Abstract*— Although the recent learning-based image and video coding techniques achieve rapid development, the signal fidelity-driven target in these methods leads to the divergence to a highly effective and efficient coding framework for both human and machine. In this paper, we aim to address the issue by making use of the power of generative models to bridge the gap between full fidelity (for human vision) and high discrimination (for machine vision). Therefore, relying on existing pretrained generative adversarial networks (GAN), we build a GAN inversion framework that projects the image into a low-dimensional natural image manifold. In this manifold, the feature is highly discriminative and also encodes the appearance information of the image, named as *latent code*. Taking a variational bit-rate constraint with a hyperprior model to model/suppress the entropy of image manifold code, our method is capable of fulfilling the needs of both machine and human visions at very low bit-rates. To improve the visual quality of image reconstruction, we further propose *multiple latent codes* and *scalable inversion*. The former gets several latent codes in the inversion, while the latter additionally compresses and transmits a shallow compact feature to support visual reconstruction. Experimental results demonstrate the superiority of our method in both human vision tasks, *i.e.* image reconstruction, and machine vision tasks, including semantic parsing and attribute prediction.**

***Index Terms*— Video coding for machine, generative adversarial networks, GAN inversion, multiple latent codes, scalable GAN inversion.**

I. Introduction

THE coming of the big data era changes the way we perceive the signal. The massive images/videos collected every day from a large number of applications of smart cities and the Internet of Things (IoT) that will be consumed by machines constitute challenges to existing processing paradigms in terms of latency and accuracy. On one hand, the *Compress then Analyze* video compression and transmission paradigms [1], [2], [3] are achieving continuously improving performance to compress full pixels, especially when recent deep learning techniques have been adopted for optimizing codecs modules or changing the whole framework. However, the full pixel representations are of low density for analytics applications. Compressing and storing all video data inevitably incurs the low quality of reconstructed images/videos and harms the analytics performance severely. On the other hand, the *Analyze then Compress* framework extracts dedicated compact analytics features to support video analytics tasks, by consuming much fewer bit-rates than the compressed video itself. However, this one-by-one solution is also sub-optimal to support diverse analytics tasks along with the collected massive data, due to the existence of multi-task redundancy.

To further develop collaborative compression techniques for human and machine, arising from the emerging MPEG standardization efforts,[1] video coding for machine (VCM) [4] starts the researches to improve the performance of intelligent analytics while reducing the bit-rate cost via multi-task end-to-end optimization. Some works [5], [6], [7] propose to redesign the existing codecs to target machine analysis. However, the design principle of existing codecs and the form of transmitting images/videos constrain the extent of feature compactness and model flexibility. Some works propose to focus on compressing intermediate features [8], [9], [10], [11] extracted from pretrained deep networks. However, as the pretrained deep networks are not designed with bit-rate constraints and only for machine analytics, compressing intermediate features might not achieve desirable performance in terms of compactness and might fail to serve human.

Other works [12], [13], [14], [15], [16] explore collaborative operations between video and feature streams to optimize the coding efficiency from the perspective of both human and machine visions, called *Scalable Coding*. The side information or semantic features are first extracted to support machine

Manuscript received 14 December 2021; revised 28 June 2022 and 5 January 2023; accepted 25 April 2023. Date of publication 10 May 2023; date of current version 7 April 2025. This work was supported in part by the NTU-PKU Joint Research Institute (a Collaboration Between Nanyang Technological University and Peking University, which is sponsored by a donation from the Ng Teng Fong Charitable Foundation); in part by the National Natural Science Foundation of China under Contract 62172020; in part by the Research Achievement of Key Laboratory of Science, Technology, and Standard in Press Industry (Key Laboratory of Intelligent Press Media Technology); in part by the Basic and Frontier Research Project of Peng Cheng Laboratory (PCL); and in part by the Major Key Project of PCL. The work of Wenhan Yang was supported by the Wallenberg-NTU Presidential Post-Doctoral Fellowship. This article was recommended by Associate Editor Z. Li. *(Corresponding author: Jiaying Liu.)*

Wenhan Yang is with the Peng Cheng Laboratory, Shenzhen 518066, China, and also with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: yangwh@pcl.ac.cn).

Haofeng Huang and Jiaying Liu are with the Wangxuan Institute of Computer Technology, Peking University, Beijing 100080, China (e-mail: liujiaying@pku.edu.cn; hhf@pku.edu.cn).

Alex C. Kot is with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: eackot@ntu.edu.sg).

Digital Object Identifier 10.1109/TCSVT.2023.3274739

[1] https://lists.aau.at/mailman/listinfo/mpeg-vcm

analytics, and these features are used to facilitate the full-pixel image/video reconstruction. In this way, a desirable performance might be obtained in terms of compactness, human perception, and machine analytics. However, for this kind of methods, there still are not well-explored critical issues on what features are first extracted and how the scalable mechanism is designed.

Recently, the fast development of generative adversarial network (GAN) techniques brings in new opportunities. On one hand, taking the image generation task as the supervision task, GANs make great processes in learning the intrinsic distributions of real data and generating photo-realistic images. Thus, this "abstract-to-appearance" translation leads to the potential to construct an efficient scalable mapping to convert a semantic code to the full-pixel image with appearances and textures. On the other hand, GANs provide a mapping mechanism, *i.e.* GAN inversion, to project the images into a manifold where we can edit or manipulate the mapped latent codes with more explicit intention. This "appearance-to-abstract" translation brings in the capacity to encode the image's most critical semantic information with a very compact latent code. These two points just meet the critical requirements in building a *Scalable Coding* framework to support both human and machine effectively.

In our work, we propose a scalable GAN inversion mechanism to transmit two bit-streams (including a basic stream and additional stream) to fulfill both human and machine visions at very low bit rates. Based on the GAN inversion, we first derive a very compact basic layer to support both human and machine visions at very low bit rates. An encoder maps the image into several latent codes and the corresponding coefficients for better image reconstruction. The latent codes help generate intermediate features of a pretrained GAN model, and the related coefficients aim to combine those intermediate features for more strong modeling capacity in GAN inversion. The image manifold code is constrained by a variational bit-rate constraint equipped with hyperprior, to support both machine and human visions at very low bit-rates. We also further propose to extend the latent code with *multiple latent codes*, and *scalable inversion* schemes, adopting additional bit-rates for visual reconstruction. Experimental results demonstrate our superiority in term of both human and machine visions.

The contributions of this work are summarized as follows:

- We propose a novel compression framework that combines the power of generative models as well as learned bit-rate constraints and optimization for very low bit-rate image/video compression.
- In our proposed framework, a very compact feature vector is first compressed and transmitted to the decoder side, which can be converted into machine analytics results in an accurate and computationally efficient manner to support the paradigm of collaborative intelligence.
- Beyond the baseline GAN inversion framework, to further improve the human perception of full-pixel reconstruction results, our proposed *multiple latent codes* and *scalable inversion* significantly improve the visual reconstruction quality by bypassing the stream with reasonable bit-rates.

The rest of this paper is organized as follows. In Section II, we review related works in GAN inversion and video coding for machine. Section III defines the image compression problem for both machine and human vision, and gives an overview of the framework of our method. In Section IV, the details about the proposed vision-driven image coding model, Natural Image MAnifold Compression (NIMAC) network, is presented. We validate our method by conducting extensive experiments and thorough analyses in Section IV. Finally, we conclude our work in Section VIII.

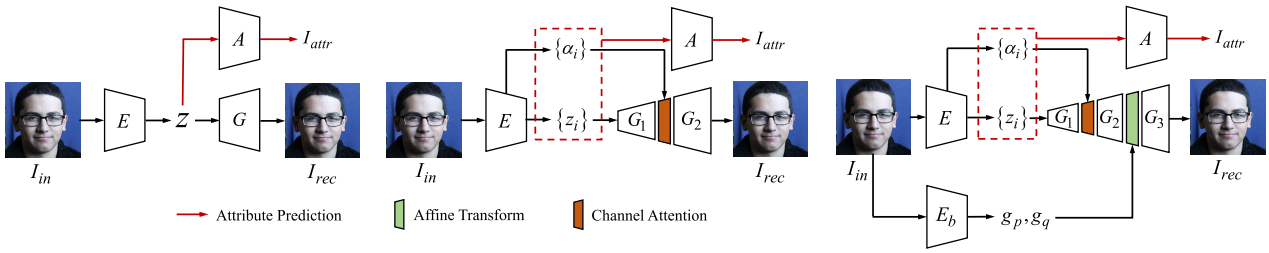
II. Related Works

A. GAN Inversion

GANs provide a mapping mechanism, *i.e.* GAN inversion, to project the images into a manifold where we can edit or manipulate the mapped latent codes with more explicit intention. In [17], an encoder is learned to project a pre-trained GAN's generated images back into their corresponding latent codes. Later works adopt StyleGAN [18], [19] as the baseline generator, considering its extraordinary performance in generated image quality and rich semantics of the latent space. Based on the optimization technique, all methods can be categorized into three classes: 1) *online optimization* [20], [21], [22], [23]: optimizing the latent code at the testing time for each testing image to minimize the distance between the generated image and the input one; 2) *offline optimization* [24], [25], [26], [27]: training a encoder to project the images into the latent space; 3) their combination [28]. Via pretraining on a large-scale dataset to obtain an encoder mapping, or utilizing the online optimization with a substantially longer time, these methods achieve desirable reconstruction performance. However, they do not consider the bit-rate constraint on the codes and fail to support machine vision tasks, which are quite necessary for real applications. Different from previous methods, our encoder generates a compact latent code that can accurately predict the attribute labels and generate the full-pixel images without optimization.

B. Video Coding for Machine

Generally speaking, the new VCM approach have developed along three lines in the last two years. The first branch is built on image/video coding and codecs are redesigned towards machine analysis, called *Machine Vision Targeted Codec* [5], [6], [7]. These methods deliver images/video that is conducive to the analysis and can achieve better analysis performance at low bit rates. The second branch extends the specialized feature compression route to compress deep intermediate features, including *Intermediate Feature Compression* [8], [9], [29] and *Optimization* [10], [11]. The former aims to rebuild pre-trained deep features based on feature fidelity constraints, while the latter directly gets deep intermediate features and compression models involved in optimization according to the joint task-driven analytics losses. The third branch considers more collaborative operations between the streams of videos and features. A sub-branch only optimizes the image/video coding efficiency towards human vision (categorized as *Feature Assisted Coding* [12], [13]). Meanwhile, the other sub-branch



(a) GAN Inversion (b) GAN Inversion with Multiple Latent Codes (c) Scalable GAN Inversion

Fig. 1. Overview of the proposed image coding framework based on GAN inversion. The red line denotes the path related to the attribute prediction.

improves the performance of both reconstruction quality and machine analytics performance for both human and machine (categorized as *Scalable Coding* [14], [15], [16]). Our proposed method belongs to *Scalable Coding*, where an extremely compact tensor is first transmitted for both highly accurate analytics and a high-level semantic preserving full-pixel reconstruction. After that, more bit-rates are transmitted, which helps reconstruct competitive visual results with low bit-rates.

III. Towards Scalable GAN Inversion

In this section, we introduce a novel GAN inversion method progressively. We first introduce the vanilla GAN inversion method. After that, to make the inversion process perceive richer information, a GAN inversion with multiple latent codes is designed. In these two schemes, the transmitted codes take the form of the feature vector, which is highly compact with less information. To enable a reconstruction with relatively higher fidelity, a scalable GAN inversion is introduced by bypassing a compact feature tensor.

A. GAN Inversion

Given the input image I_{in}, as shown in Fig. 1-(a), GAN inversion aims to project I_{in} into the latent space so that the obtained code z can well reconstruct the input image with a pretrained generator, *e.g.* BigGAN or StyleGAN, *etc.* Formally, we have:

$$\begin{aligned} z &= E(I_{in}|\theta_e), \\ I_{rec} &= G(z|\theta_g). \end{aligned} \tag{1}$$

where $E(\cdot)$ is the encoding process to project I_{in} to the latent space, and $G(\cdot)$ is the synthesis process that generates an image from a sampled noise or a given latent code provided by the encoder. θ_e and θ_g are their parameters, respectively. In GAN inversion, $G(\cdot|\theta_g)$ is usually well trained and we hope to train $E(\cdot|\theta_e)$ as follows:

$$\begin{aligned} \arg\min_{\theta_e} & L_{fid}(I_{rec}, I_{in}) + L_{per}(I_{rec}, I_{in}) + L_{task}(I_{rec}, I_{in}) \\ & + L_{reg}(I_{rec}), \end{aligned} \tag{2}$$

where $L_{fid}(\cdot)$ is the fidelity measure, such as L_1 or L_2 loss. $L_{per}(\cdot)$ is the perceptual quality measure, such as perceptual loss [30] or LPIPS [31]. $L_{task}(\cdot)$ is the task-driven metric, such as the constraint to keep two face images having the same identity. $L_{reg}(\cdot)$ is the regularization term imposed on the I_{rec}. For the need of machine analysis, we also introduce an analysis model $A(\cdot|\theta_a)$ with θ_a as its parameter to perform attribute prediction using z as its input:

$$I_{attr} = A(z|\theta_a). \tag{3}$$

where I_{attr} is the predicted attributes, *e.g.* man/women, smile, taking glasses, *etc.*

B. GAN Inversion With Multiple Latent Codes

The latent code is of low dimension and has limited expressiveness. Hence, it is impossible to reconstruct every detail of an arbitrary real image using a single latent code. Then, as shown in Fig. 1-(b), we extend the single-code GAN inversion in Eq. (1) into the case with multiple codes:

$$\begin{aligned} \{z_i, \alpha_i\}_{i=1,\ldots,N} &= E(I_{in}|\theta_e), \\ f_i &= G_1\left(z_i|\theta_{g_1}\right), \quad i = 1, \ldots, N \\ I_{rec} &= G_2\left(\sum_{i=1}^{M} f_i \odot \alpha_i \middle| \theta_{g_2}\right). \end{aligned} \tag{4}$$

Generally, $G(\cdot)$ can be decomposed as a sequential operation of $G_1(\cdot)$ and $G_2(\cdot)$. where $f_i \in \mathbb{R}^{H\times W\times C}$ is the intermediate feature generated from a part of generator $G_1(\cdot)$. θ_{g_1} and θ_{g_2} are the parameters of $G_1(\cdot)$ and $G_2(\cdot)$, respectively. H, W, C are the height, width, and channel of f_i, respectively. $\alpha_i \in \mathbb{R}^{\times C}$ is a vector denoting the channel importance of f_i. $\odot$ is the channel-wise product operation (channel attention) defined as follows,

$$(f_i \odot \alpha_i)_{k,l,m} = (f_i)_{k,l,m} \times (\alpha_i)_m \tag{5}$$

where $k \in [1, \ldots, H]$, $l \in [1, \ldots, W]$, and $m \in [1, \ldots, C]$.

Similarly, the predicted attributes for machine analysis are obtained via:

$$I_{attr} = A(\{z_i, \alpha_i\}|\theta_a). \tag{6}$$

Via generating multiple intermediate features from z_i and then merging them based on the channel attention, rich information is included to boost the performance of machine analytics and semantic guided visual reconstruction.

C. Scalable GAN Inversion

For full-pixel visual reconstruction, the multiple latent codes scheme still does not provide enough information for a

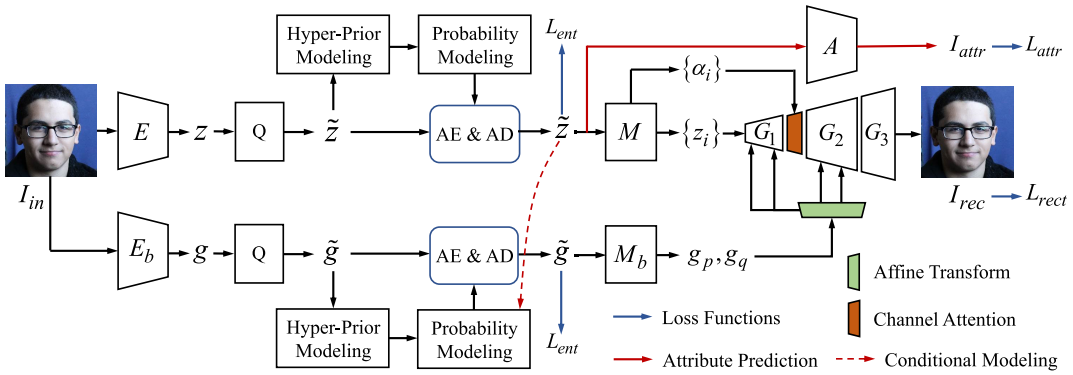


Fig. 2. Overview of our Natural Image MAnifold Compression Network (NIMAC-Net) for video coding for human and machine. The red line denotes the path related to the attribute prediction. The blue lines denote the paths related to the loss calculation.

relatively high-fidelity reconstruction.

$$\begin{aligned} \{z_i, \alpha_i\}_{i=1,\ldots,N} &= E(I_{in}|\theta_e), \\ f_i &= G_1\left(z_i|\theta_{g_1}\right), \quad i = 1, \ldots, N \\ \beta &= G_2\left(\sum_{i=1}^{M} f_i \odot \alpha_i \middle| \theta_{g_2}\right). \end{aligned} \tag{7}$$

Besides the information brought by the multiple latent codes (feature vector), we also introduce a bypass connection that transmits a compact feature tensor to the decoder side from a relatively shallow encoder. After merging the transmitted tensor and the feature tensor generated from the multiple latent codes, we obtain the feature tensor with rich information that helps full-pixel visual reconstruction as follows,

$$\begin{aligned} \{g_p, g_q\} &= E_b\left(I_{in}|\theta_{e_b}\right), \\ \gamma &= \beta \cdot g_q + g_p, \\ I_{rec} &= G_3\left(\gamma|\theta_{g_3}\right), \end{aligned} \tag{8}$$

where $\cdot$ is the element-wise product. This bypass connection can significantly improve the signal fidelity of the visual reconstruction results with more bit-rates.

Similarly, the predicted attributes for machine analysis are obtained via:

$$I_{attr} = A(\{z_i, \alpha_i\}|\theta_a). \tag{9}$$

IV. Natural Image Manifold Compression Network for Human and Machine

In this section, based on the scalable GAN inversion paradigm, we propose a more detailed Natural Image MAnifold Compression Network (NIMAC-Net) for human and machine visions to enforce the bit-rate constraint on the transmitted latent codes and feature tensor. The overview of NIMAC-Net is shown in Fig. 2. We then illustrate its components one by one in the following.

A. Encoder

We first use two convolutional neural networks to extract the compact latent code z and feature tensor g to be transmitted:

- The compact latent code z can be transmitted to support semantic preserving full-pixel visual reconstruction at an extremely low bit-rate;
- The feature tensor g compensate for the information of appearances and textures in the reconstructed images, which is absent in the very compact z.

In formulation, we have:

$$\begin{aligned} z &= E(I_{in}|\theta_e), \\ g &= E_b(I_{in}|\theta_b), \end{aligned} \tag{10}$$

where θ_e and θ_b are the parameters of the corresponding processes $E(\cdot)$ and $E_b(\cdot)$, respectively. Then, z and g are quantized as follows:

$$\begin{aligned} \tilde{z} &= Q(z), \\ \tilde{g} &= Q(g). \end{aligned} \tag{11}$$

In our implementation, $E(\cdot)$ takes the framework of pixel2style2pixel encoder [32] and $E_b(\cdot)$ takes the framework of a 9-layer convolutional neural network injected with squeeze-and-excitation modules.

B. Hyperprior Probability Modeling

After obtaining quantized $\tilde{z}$ and $\tilde{g}$, we adopt the end-to-end image compression method to constrain their entropy and squeeze out their redundancy. Ideally, $\tilde{z}$ and $\tilde{g}$ are compact. Their probability distributions are tractable and can be compressed compactly into bit-stream. Therefore, we estimate and constrain the entropy of the structured representation $\tilde{z}$ and $\tilde{g}$.

The details about the hyperprior and the related probability modeling can be found in [33]. It it noted that, when performing the probability modeling for $\tilde{g}$, $\tilde{z}$ will be included as the additional condition as denoted by the red dash in Fig. 2.

C. Analytics Module

After the quantization and entropy coding, along with the bit-stream transmission, we obtain reconstructed $\tilde{z}$ and $\tilde{g}$ at the decoder side. Then, at first, we can infer the face attribute prediction results via applying the analytics model as follows,

$$I_{attr} = A(\tilde{z}|\theta_a), \tag{12}$$

where $\tilde{z}$ is almost equivalent to $\{z_i, \alpha_i\}$ in Eq. (9). As $A(\cdot|\theta_a)$ is a very lightweight model, *i.e.* several cascaded convolutional layers and fully-connected layers, the response time at the decoder side can be very fast and computation is very light, which is desirable for *collaborative intelligence* [34]. Namely, the most computation can be finished at the encoder side.

D. Mapping Module

After that, the mapping modules will map $\widetilde{z}$ and $\widetilde{g}$ into a continuous variables to generate the full-pixel images:

$$\begin{aligned}\{z_i, \alpha_i\}_{i=1,\ldots,N} &= M\left(\widetilde{z}|\theta_m\right),\\ \left\{g_p, g_q\right\} &= M_b\left(\widetilde{g}|\theta_{m_b}\right),\end{aligned} \tag{13}$$

where $M\left(\cdot|\theta_m\right)$ and $M_b\left(\cdot|\theta_{m_b}\right)$ are the mapping modules and θ_m and θ_{m_b} are their corresponding parameters, respectively.

E. Generator Module

With $\{z_i, \alpha_i\}_{i=1,\ldots,N}$ and $\left\{g_p, g_q\right\}$, we then map them into the full-pixel visual reconstruction. Slightly different from Sec. III-C, the $\left\{g_p, g_q\right\}$ layers are split into several variables, which are then injected into intermediate layers of $G_1(\cdot)$ and $G_2(\cdot)$ in the way of spatial feature transform [35].

More specifically, $G_1(\cdot)$, $G_2(\cdot)$ and $G_3(\cdot)$ can be decomposed into a sequential network:

$$\begin{aligned}G_1(\cdot) &= \widehat{G}_6 \circ \widehat{G}_5 \circ \widehat{G}_4 \circ \widehat{G}_3 \circ \widehat{G}_2 \circ \widehat{G}_1 \circ \widehat{G}_0(\cdot),\\ G_2(\cdot) &= \widehat{G}_{10} \circ \widehat{G}_9 \circ \widehat{G}_8 \circ \widehat{G}_7(\cdot),\\ G_3(\cdot) &= \widehat{G}_{16} \circ \widehat{G}_{15} \circ \widehat{G}_{14} \circ \widehat{G}_{13} \circ \widehat{G}_{12} \circ \widehat{G}_{11}(\cdot),\end{aligned} \tag{14}$$

where each $\widehat{G}_i(\cdot)$ is a style convolution defined in StyleGAN [19].

g_p and g_q are then split into several variable along their channel dimension as follows:

$$\begin{aligned}g_p &= \left\{g_p^2, g_p^4, g_p^6, g_p^8, g_p^{10}\right\},\\ g_q &= \left\{g_q^2, g_q^4, g_q^6, g_q^8, g_q^{10}\right\}.\end{aligned} \tag{15}$$

Then, we inject them into $G_1(\cdot)$ and $G_2(\cdot)$ via:

$$\begin{aligned}G_1(\cdot) &= \widetilde{G}_6 \circ \widehat{G}_5 \circ \widetilde{G}_4 \circ \widehat{G}_3 \circ \widetilde{G}_2 \circ \widehat{G}_1 \circ \widehat{G}_0(\cdot),\\ G_2(\cdot) &= \widetilde{G}_{10} \circ \widehat{G}_9 \circ \widetilde{G}_8 \circ \widehat{G}_7(\cdot),\\ \widetilde{G}_i(x) &= \widehat{G}_i(x) \cdot \widehat{g}_p^i + \widehat{g}_q^i, i = 2, 4, \ldots, 10,\end{aligned} \tag{16}$$

where $\widehat{g}_p^i$ and $\widehat{g}_q^i$ are the resized version of g_p^i and g_q^i based on the size of $\widehat{G}_i(x)$. Following [19], $\widehat{G}_i(x)$ is implemented by cascaded two-layer convolutions, where the first convolution up-samples the feature maps by the 2-time scale. In this way, the information brought by the additional bit-stream $\widetilde{g}$ leads to the hierarchical manipulation on the different layers of the generator following Eq. (16).

F. Loss Functions

To regularize the training of our NIMAC-Net, we adopt three composite losses, attribute loss L_{attr}, reconstruction loss L_{rect}, and entropy loss L_{ent}, in the training phase, as shown in Fig. 2.

1) Attribute Loss: Assume I_{attr}^t and l_{attr}^t are the prediction results and labels of the t-th sample, the attribute loss is defined as follows,

$$L_{attr} = \lambda_{attr} \sum_{t=1}^{M} L_{sm}\left(I_{attr}^t, l_{attr}^t\right), \tag{17}$$

where $L_{sm}(\cdot)$ is the softmax loss, M is the number of all samples, and λ_{attr} is the parameter that balances the importance of this term and other terms. This loss helps the network own the capacity to predict the attribute labels of the input image with little computation complexity at the decoder side.

2) Reconstruction Loss: The reconstruction loss makes our network own the capacity to reconstruct the input images. Our adopted reconstruction loss consists of five parts as follows:

$$\begin{aligned}L_{rect} = \lambda_{id} L_{id} + \lambda_{lpips\_1024} L_{lpips\_1024} + \lambda_{lpips\_256} L_{lpips\_256}\\ + \lambda_{l_2\_1024} L_{l_2\_1024} + \lambda_{l_2\_256} L_{l_2\_256},\end{aligned} \tag{18}$$

where L_{id} aims to preserve the input identity when encoding the facial images via dedicated recognition loss measuring the cosine similarity between the generated result and the label image:

$$L_{id} = \sum_{t=1}^{M} 1 - \left\langle R(I_{rec}^t), R(I_{in}^t)\right\rangle, \tag{19}$$

where $R(\cdot)$ is a pretrained ArcFace network [36] and $\langle\cdot,\cdot\rangle$ measures the cosine similarity.

L_{lpips_1024} and L_{lpips_256} measure the perceptual similarities at the resolution of 1024×1024 and 256×256, respectively, as follows,

$$L_{lpips\_1024} = \sum_{t=1}^{M} \left\| F\left(I_{rec}^t\right) - F\left(I_{in}^t\right)\right\|_2^2, \tag{20}$$

$$L_{lpips\_256} = \sum_{t=1}^{M} \left\| F\left(D_{\frac{1}{4}}(I_{rec}^t)\right) - F\left(D_{\frac{1}{4}}(I_{in}^t)\right)\right\|_2^2, \tag{21}$$

where $F(\cdot)$ is the perceptual feature extractor, *i.e.* AlexNet. $D_{\frac{1}{4}}(\cdot)$ is the a down-sampler that down-samples the input image into its 1/4 resolution.

The pixel-wise l_2 loss is adopted to maintain the basic signal fidelity as follows,

$$L_{l_2\_1024} = \sum_{t=1}^{M} \left\| I_{rec}^t - I_{in}^t\right\|_2^2, \tag{22}$$

$$L_{l_2\_256} = \sum_{t=1}^{M} \left\| D_{\frac{1}{4}}(I_{rec}^t) - D_{\frac{1}{4}}(I_{in}^t)\right\|_2^2. \tag{23}$$

λ_{id}, λ_{lpips_1024}, λ_{lpips_256}, $\lambda_{l_2\_1024}$ and $\lambda_{l_2\_256}$ are the weighting parameters of each term to balance their importance in the training.

3) Bit-Rate Loss: We also estimate the entropy of $\widetilde{z}$ and $\widetilde{g}$ based on Sec-IV-B as follows,

$$L_{ent} = \lambda_{ent} \sum_{t=1}^{M} \left(EN(\widetilde{z}^t) + EN(\widetilde{g}^t)\right), \tag{24}$$

where $EN(\cdot)$ is the process to estimate the entropy of the input variable/feature. More details about the bit-rate/entropy estimation can be found in [33]. λ_{ent} is the weighting parameter to balance the importance of the term during the training.

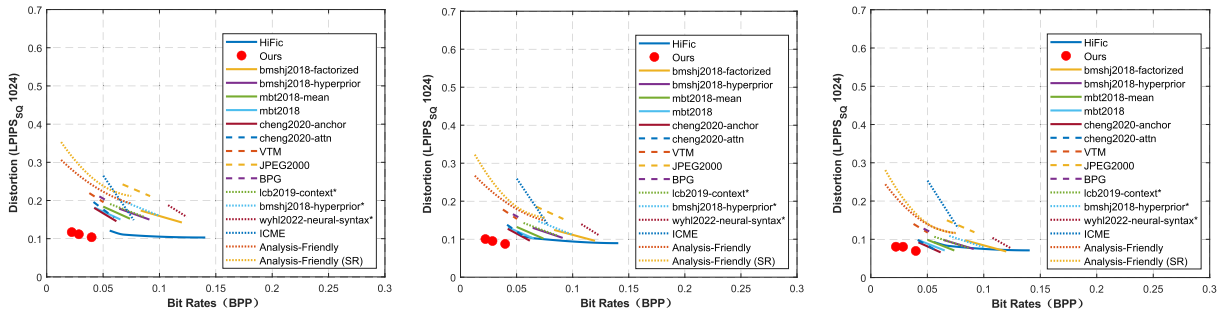
4) Whole Training Loss: We finally combine all losses into an integrated loss function for training:

$$L_{All} = L_{attr} + L_{rect} + L_{ent}. \tag{25}$$

TABLE I

QUANTITATIVE RESULTS OF DIFFERENT METHODS FOR IMAGE RECONSTRUCTION. BPP DENOTES BITS-PER-PIXEL. LPIPS_{SQ} (1024), LPIPS_{SQ} (768) AND LPIPS_{SQ} (512) CALCULATE THE LPIPS AS THE DISTORTION MEASURE BASED ON THE FEATURES EXTRACTED FROM THE SQUEEZENET AT THE RESOLUTION OF 1024×1024, 768×768, AND 512×512, RESPECTIVELY. THE BEST AND SECOND BEST RESULTS ARE DENOTED IN RED AND BLUE, RESPECTIVELY

Reconstruction	BPP	LPIPS_{SQ} (1024)	LPIPS_{SQ} (768)	LPIPS_{SQ} (512)	Setting
HiFiC	0.1407	0.1031	0.0895	0.0711	Target_0.03_Lambda_2**2
HiFiC	0.0559	0.1211	0.1073	0.0928	Target_0.01_Lambda_2**2
HiFiC	0.0687	0.1106	0.1014	0.0828	Target_0.002_Lambda_2**2
Ours	0.0222	0.1172	0.1003	0.0806	
Ours	0.0285	0.1116	0.0951	0.0805	
Ours	0.0398	0.1038	0.0875	0.0696	
bmshj2018-factorized	0.0834	0.1736	0.1244	0.0947	q=1
bmshj2018-factorized	0.1199	0.1424	0.0957	0.0682	q=2
bmshj2018-hyperprior	0.0644	0.1786	0.1299	0.0982	q=1
bmshj2018-hyperprior	0.0911	0.1501	0.1034	0.0747	q=2
mbt2018-mean	0.0501	0.1840	0.1322	0.0987	q=1
mbt2018-mean	0.0739	0.1523	0.1016	0.0708	q=2
mbt2018	0.0423	0.1816	0.1310	0.0976	q=1
mbt2018	0.0654	0.1502	0.1015	0.0708	q=2
cheng2020-anchor	0.0421	0.1807	0.1273	0.0922	q=1
cheng2020-anchor	0.0618	0.1457	0.0960	0.0658	q=2
cheng2020-attn	0.0415	0.1960	0.1379	0.0984	q=1
cheng2020-attn	0.0596	0.1554	0.1009	0.0679	q=2
lcb2019-context*	0.0562	0.1907	0.1427	0.1066	$\lambda = 0.0008$
lcb2019-context*	0.0870	0.1565	0.1086	0.0772	$\lambda = 0.0015$
bmshj2018-hyperprior*	0.0694	0.1960	0.1472	0.1090	$\lambda = 0.0008$
bmshj2018-hyperprior*	0.1000	0.1600	0.1117	0.0791	$\lambda = 0.0015$
wyhl2022-neural-syntax*	0.1077	0.1877	0.1395	0.1039	$\lambda = 0.0008$
wyhl2022-neural-syntax*	0.1241	0.1572	0.1091	0.0763	$\lambda = 0.0015$
VTM	0.0504	0.1955	0.1531	0.1174	QP=40
VTM	0.0378	0.2193	0.1780	0.1401	QP=42
JPEG2000	0.0673	0.2423	0.1860	0.1497	q=32
JPEG2000	0.0917	0.2115	0.1531	0.1185	q=33
BPG	0.0540	0.1984	0.1539	0.1171	q=42
BPG	0.0467	0.2107	0.1658	0.1280	q=43
ICME (HV)	0.0772	0.1487	0.1411	0.1295	
ICME (MV)	0.0502	0.2649	0.2593	0.2535	
Analysis-Friendly	0.0128	0.3062	0.2670	0.2438	
Analysis-Friendly (SR)	0.0128	0.3530	0.3225	0.2809	
Analysis-Friendly	0.0750	0.1922	0.1490	0.1161	
Analysis-Friendly (SR)	0.0750	0.2118	0.1572	0.1170	
Analysis-Friendly	0.0606	0.2075	0.1637	0.1257	
Analysis-Friendly (SR)	0.0606	0.2228	0.1682	0.1256	



(a) LPIPS_{SQ} result at the resolution 1024×1024 (b) LPIPS_{SQ} result at the resolution 768×768 (c) LPIPS_{SQ} result at the resolution 512×512

Fig. 3. RD-curves of different methods for image reconstruction taking LPIPS_{SQ} as the distortion measure at the resolutions of 1024×1024, 768×768, and 512×512.

V. EXPERIMENTAL RESULTS

A. Datasets

We use FFHQ-Aging [37] for evaluation. FFHQ-Aging is a Dataset of human faces designed for benchmarking age transformation algorithms as well as many other possible vision tasks. This dataset is an extension of the NVIDIA FFHQ dataset [19]. On top of the 70,000 original FFHQ images, it also contains the following information for each image: 1) Gender information (male/female with confidence score); 2) Age group information (10 classes with confidence score); 3) Head pose (pitch, roll & yaw); 4) Glasses type (none, normal or dark); 5) Eye occlusion score (0-100, different score for each eye); 6) Full semantic map (19 classes, based on CelebAMask-HQ labels). We split the whole FFHQ into the training set (ID: 00000-59999) and the testing set (ID: 60000-60099). There are in total 60,000 training and 100 testing images. We use the gender, age, and glass attributes as the labels to evaluate the performance of different methods in attribute prediction. We also adopt the full semantic map as the semantic label to evaluate the performance of different methods in semantic segmentation.

B. Compared Methods

The compared methods covers learning-based compression methods including bmshj2018-factorized [38], bmshj2018-hyperprior [38], mbt2018-mean [39], mbt2018 [39], cheng-2020-anchor [40], cheng2020-attn [40], lcb2019-context

TABLE II

QUANTITATIVE RESULTS OF DIFFERENT METHODS FOR IMAGE RECONSTRUCTION. BPP DENOTES BITS-PER-PIXEL. $\text{LPIPS}_{\text{ALEX}}$ (1024), $\text{LPIPS}_{\text{ALEX}}$ (768) AND $\text{LPIPS}_{\text{ALEX}}$ (512) CALCULATE THE LPIPS AS THE DISTORTION MEASURE BASED ON THE FEATURES EXTRACTED FROM THE ALEXNET AT THE RESOLUTION OF 1024×1024, 768×768, AND 512×512, RESPECTIVELY. THE BEST AND SECOND BEST RESULTS ARE DENOTED IN RED AND BLUE, RESPECTIVELY

Reconstruction	BPP	$\text{LPIPS}_{\text{Alex}}$ (1024)	$\text{LPIPS}_{\text{Alex}}$ (768)	$\text{LPIPS}_{\text{Alex}}$ (512)	Setting
HiFiC	0.1407	0.1076	0.1082	0.0900	Target_0.03_Lambda_2**2
HiFiC	0.0559	0.1367	0.1430	0.1257	Target_0.01_Lambda_2**2
HiFiC	0.0687	0.1301	0.1287	0.1082	Target_0.002_Lambda_2**2
Ours	0.0222	0.1302	0.1222	0.0968	
Ours	0.0285	0.1287	0.1194	0.0953	
Ours	0.0398	0.1124	0.1032	0.0810	
bmshj2018-factorized	0.0834	0.1865	0.1379	0.0888	q=1
bmshj2018-factorized	0.1199	0.1546	0.1046	0.0577	q=2
bmshj2018-hyperprior	0.0644	0.1936	0.1446	0.0948	q=1
bmshj2018-hyperprior	0.0911	0.1616	0.1113	0.0661	q=2
mbt2018-mean	0.0501	0.1929	0.1424	0.0911	q=1
mbt2018-mean	0.0739	0.1603	0.1084	0.0616	q=2
mbt2018	0.0423	0.1877	0.1374	0.0880	q=1
mbt2018	0.0654	0.1534	0.1033	0.0590	q=2
cheng2020-anchor	0.0421	0.1867	0.1363	0.0889	q=1
cheng2020-anchor	0.0618	0.1512	0.1012	0.0574	q=2
cheng2020-attn	0.0415	0.2058	0.1524	0.1010	q=1
cheng2020-attn	0.0596	0.1622	0.1098	0.0629	q=2
lcb2019-context*	0.0562	0.1918	0.1408	0.0940	$\lambda = 0.0008$
lcb2019-context*	0.0870	0.1567	0.1069	0.0657	$\lambda = 0.0015$
bmshj2018-hyperprior*	0.0694	0.1982	0.1467	0.0970	$\lambda = 0.0008$
bmshj2018-hyperprior*	0.1000	0.1602	0.1096	0.0667	$\lambda = 0.0015$
wyhl2022-neural-syntax*	0.1077	0.1780	0.1306	0.0889	$\lambda = 0.0008$
wyhl2022-neural-syntax*	0.1241	0.1459	0.0996	0.0623	$\lambda = 0.0015$
VTM	0.0504	0.1991	0.1523	0.1066	QP=40
VTM	0.0378	0.2272	0.1807	0.1317	QP=42
JPEG2000	0.0673	0.2692	0.2151	0.1538	q=32
JPEG2000	0.0917	0.2350	0.1786	0.1198	q=33
BPG	0.0540	0.2155	0.1680	0.1187	q=42
BPG	0.0467	0.2285	0.1810	0.1312	q=43
ICME (HV)	0.0772	0.2164	0.1945	0.1701	
ICME (MV)	0.0502	0.3621	0.3480	0.3316	
Analysis-Friendly	0.0128	0.3804	0.3428	0.3194	
Analysis-Friendly (SR)	0.0128	0.3721	0.3405	0.2920	
Analysis-Friendly	0.0750	0.2608	0.2232	0.1829	
Analysis-Friendly (SR)	0.0750	0.2317	0.1841	0.1345	
Analysis-Friendly	0.0606	0.2699	0.2359	0.1929	
Analysis-Friendly (SR)	0.0606	0.2398	0.1920	0.1414	

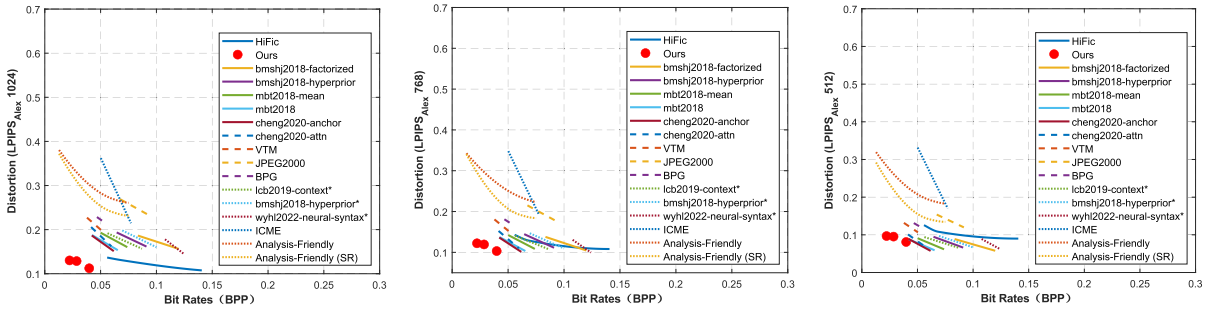
[41], wyhl2022-neural-syntax [42], HiFiC [43], ICME [44], Analysis-Friendly [45], Analysis-Friendly (SR) [45], and traditional codecs including JPEG (implemented by MATLAB), JPEG2000[2] [46], HEVC-based BPG[3] [2], and VVC-based VTM [47]. The former eight learning-based methods are signal fidelity-driven methods while HiFiC targets optimizing human visual perception trained with generative adversarial networks. ICME and Analysis-Friendly are methods designed specially for facial image compression. The technique in [45] can only support the compression of the image with a resolution 256×256. In our paper, we focus on the compression of facial images at a high resolution 1024×1024. To compare with [45] fairly, we first use the method to compress the down-sampled images with a resolution of 256×256, and then adopt the state-of-the-art super-resolution method ESRGAN [48] to up-sample the reconstructed images to the resolution 1024×1024 as the final compression results. These results are denoted as *Analysis-Friendly*. At the same time, the results of [45] are further enhanced by a retrained post-processing super-resolution network [49], which narrows down the resolution gaps between our method and [45]. The related results are denoted as *Analysis-Friendly (SR)*. For the implementation of the former eight learning-based methods, we select the implementation of CompressAI[4] in the evaluation. VTM/VVC is the state-of-the-art traditional image/video codec. For attribute prediction, we also compare our method with a feature compression method Chen et al. [8].

C. Evaluation Measures

Different distortion measures are adopted in evaluating different tasks.

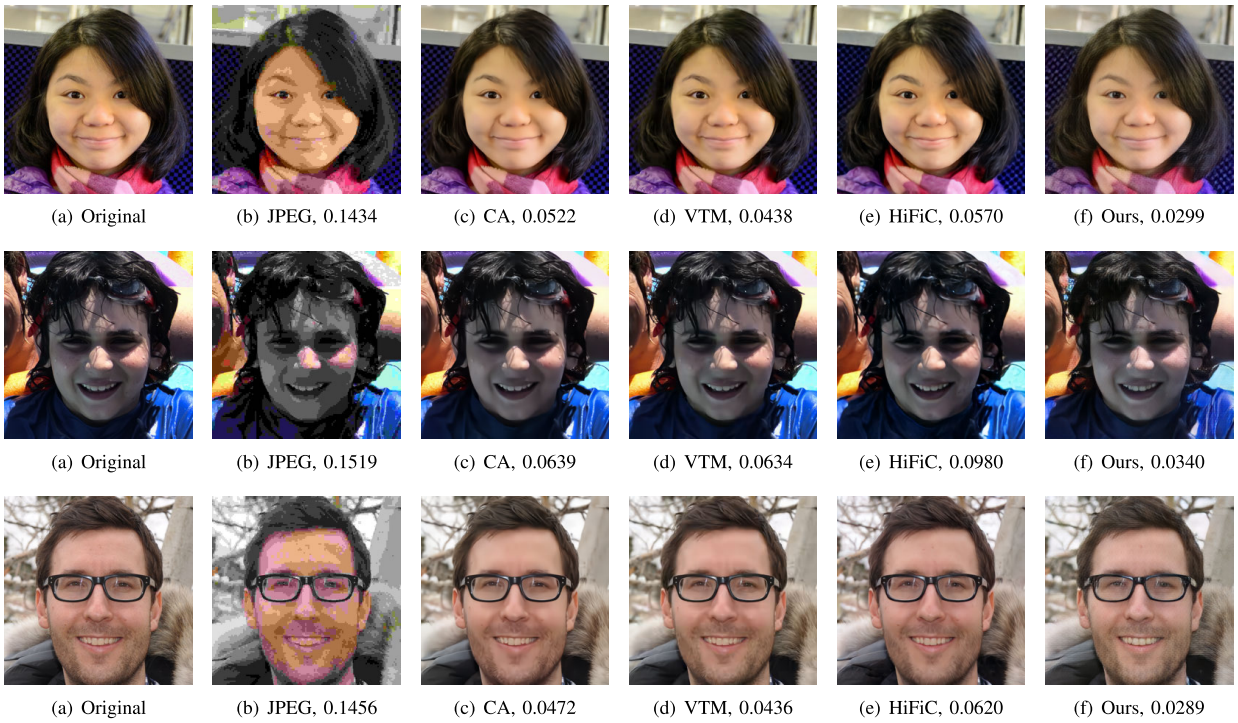
- For *image reconstruction*, we use LPIPS [31] at the resolution of 1024×1024, 768×768, and 512×512 as the quality measures. The features used to calculate the measure are extracted from the pretrained AlexNet and SqueezeNet. A lower LPIPS score demonstrates better visual quality.
- For *face parsing*, the metrics of common semantic segmentation and scene parsing evaluations are delivered. Those are variations [50] on pixel accuracy and region intersection over union (IU): pixel accuracy, mean accuracy, mean IU, and frequency weighted IU. We use the BiSeNet[5] as the parsing network. The models are retrained.

[2]https://github.com/uclouvain/openjpeg/releases/tag/v2.4.0
[3]https://bellard.org/bpg/
[4]https://github.com/InterDigitalInc/CompressAI
[5]https://github.com/zllrunning/face-parsing.PyTorch



(a) LPIPS$_{Alex}$ result at the resolution 1024×1024 (b) LPIPS$_{Alex}$ result at the resolution 768×768 (c) LPIPS$_{Alex}$ result at the resolution 512×512

Fig. 4. RD-curves of different methods for image reconstruction taking LPIPS$_{Alex}$ as the distortion measure at the resolutions of 1024 × 1024, 768 × 768, and 512 × 512.

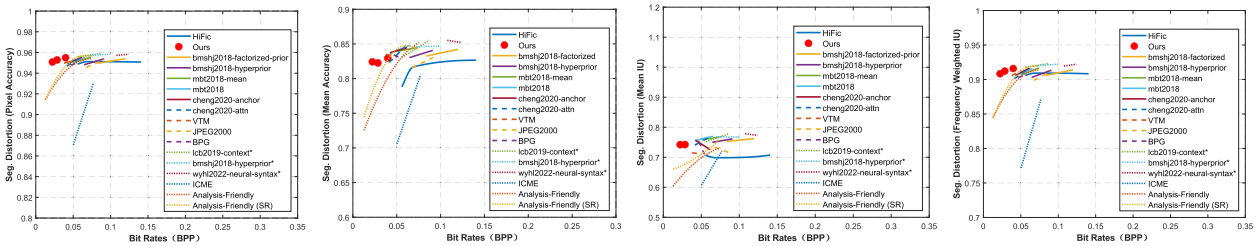


(a) Original (b) JPEG, 0.1434 (c) CA, 0.0522 (d) VTM, 0.0438 (e) HiFiC, 0.0570 (f) Ours, 0.0299

(a) Original (b) JPEG, 0.1519 (c) CA, 0.0639 (d) VTM, 0.0634 (e) HiFiC, 0.0980 (f) Ours, 0.0340

(a) Original (b) JPEG, 0.1456 (c) CA, 0.0472 (d) VTM, 0.0436 (e) HiFiC, 0.0620 (f) Ours, 0.0289

Fig. 5. Visual comparisons of different methods for image reconstruction. The followed number denote the BPP of each method for the given image. CA denotes cheng2020-attn. Zoom-in for better visualization of the detailed reconstructed textures.



(a) Pixel Accuracy (b) Mean Accuracy (c) Mean IU (d) Frequency Weighted IU

Fig. 6. RD-curves of different methods for semantic segmentation.

- For *attribute prediction*, the accuracy in age, gender and glass predictions are adopted for evaluation. We use the ResNet50[6] as the attribute prediction network.

We also adopt bits-per-pixel (BPP) to measure the compactness of features/images for all tasks.

D. Implementation Details

In our implementation, we set $\lambda_{attr} = 10$, $\lambda_{id} = 0.1$, $\lambda_{lpips\_1024} = 80$, $\lambda_{lpips\_256} = 8$, $\lambda_{l_2\_1024} = 10$, $\lambda_{l_2\_256} = 1$, and set $\lambda_{ent} = 5, 2, 0.1$ in three versions of our method. The initial learning rate is set to 0.0001 and Adam is adopted as the optimizer. The training stops after 1,500,000 iterations.

[6]https://github.com/d-li14/face-attribute-prediction

E. Evaluation for Image Reconstruction

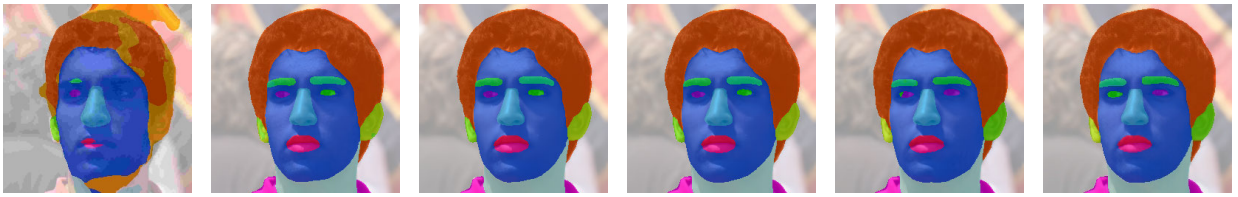
The quantitative results of different methods for image reconstruction are shown in Table II and Fig. 4. Several observations are obtained:

- For evaluating in LPIPS at both the resolutions of 1024 × 1024, 768 × 768, and 512 × 512, the

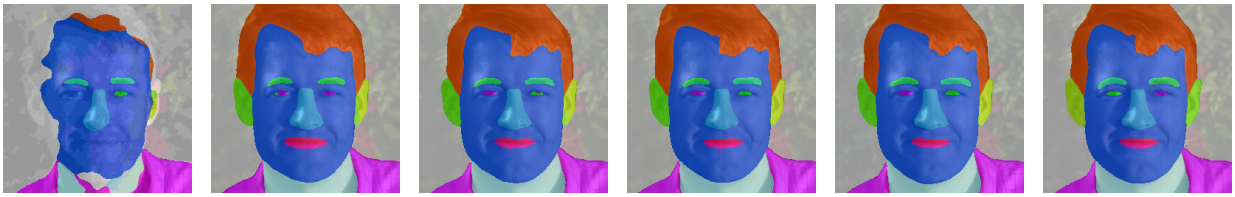
TABLE III

QUANTITATIVE RESULTS OF DIFFERENT METHODS FOR SEMANTIC SEGMENTATION. BPP DENOTES BITS-PER-PIXEL. THE BEST AND SECOND BEST RESULTS, EXCEPT FOR THE GROUND TRUTH, ARE DENOTED IN RED AND BLUE, RESPECTIVELY

Segmentation	BPP	Pixel Accuracy	Mean Accuracy	Mean IU	Frequency Weighted IU	Setting
HiFiC	0.1407	0.9507	0.8265	0.7072	0.9083	Target_0.03_Lambda_2**2
HiFiC	0.0559	0.9488	0.7881	0.7061	0.9059	Target_0.01_Lambda_2**2
HiFiC	0.0687	0.9511	0.8167	0.6985	0.9092	Target_0.002_Lambda_2**2
Ours	0.0222	0.9508	0.8242	0.7421	0.9084	
Ours	0.0285	0.9528	0.8226	0.7422	0.9122	
Ours	0.0398	0.9549	0.8295	0.7501	0.9159	
GT	-	0.9639	0.8788	0.8065	0.9317	
bmshj2018-factorized	0.0834	0.9496	0.8325	0.7526	0.9059	q=1
bmshj2018-factorized	0.1199	0.9540	0.8421	0.7621	0.9140	q=2
bmshj2018-hyperprior	0.0644	0.9481	0.8302	0.7499	0.9030	q=1
bmshj2018-hyperprior	0.0911	0.9538	0.8407	0.7613	0.9136	q=2
mbt2018-mean	0.0501	0.9501	0.8391	0.7581	0.9066	q=1
mbt2018-mean	0.0739	0.9556	0.8449	0.7671	0.9167	q=2
mbt2018	0.0423	0.9493	0.8361	0.7558	0.9052	q=1
mbt2018	0.0654	0.9552	0.8483	0.7702	0.9159	q=2
cheng2020-anchor	0.0421	0.9497	0.8371	0.7570	0.9060	q=1
cheng2020-anchor	0.0618	0.9553	0.8445	0.7250	0.9164	q=2
cheng2020-attn	0.0415	0.9468	0.8234	0.7403	0.9011	q=1
cheng2020-attn	0.0596	0.9547	0.8460	0.7660	0.9152	q=2
lcb2019-context*	0.0562	0.9559	0.8304	0.7521	0.9179	$\lambda = 0.0008$
lcb2019-context*	0.0870	0.9589	0.8552	0.7785	0.9228	$\lambda = 0.0015$
bmshj2018-hyperprior*	0.0694	0.9571	0.8458	0.7680	0.9198	$\lambda = 0.0008$
bmshj2018-hyperprior*	0.1000	0.9583	0.8468	0.7672	0.9222	$\lambda = 0.0015$
wyhl2022-neural-syntax*	0.1077	0.9572	0.8553	0.7784	0.9196	$\lambda = 0.0008$
wyhl2022-neural-syntax*	0.1241	0.9583	0.8518	0.7729	0.9220	$\lambda = 0.0015$
VTM	0.0504	0.9532	0.8312	0.7104	0.9128	QP=40
VTM	0.0378	0.9495	0.8350	0.7150	0.9057	QP=42
JPEG2000	0.0673	0.9455	0.8155	0.7336	0.8987	q=32
JPEG2000	0.0917	0.9532	0.8305	0.7104	0.9127	q=33
BPG	0.0540	0.9497	0.8300	0.7102	0.9063	q=42
BPG	0.0467	0.9478	0.8341	0.7528	0.9026	q=43
ICME (HV)	0.0772	0.9309	0.8055	0.7212	0.8727	
ICME (MV)	0.0502	0.8707	0.7058	0.6068	0.7717	
Analysis-Friendly	0.0128	0.9144	0.7253	0.6036	0.8449	
Analysis-Friendly (SR)	0.0128	0.9146	0.7446	0.6599	0.8438	
Analysis-Friendly	0.0750	0.9541	0.8515	0.7318	0.9139	
Analysis-Friendly (SR)	0.0750	0.9569	0.8407	0.7609	0.9196	
Analysis-Friendly	0.0606	0.9520	0.8375	0.7153	0.9105	
Analysis-Friendly (SR)	0.0606	0.9563	0.8475	0.7289	0.9182	



(a) JPEG (b) JPEG2000 (c) CA (d) VTM (e) HiFiC (f) Ours

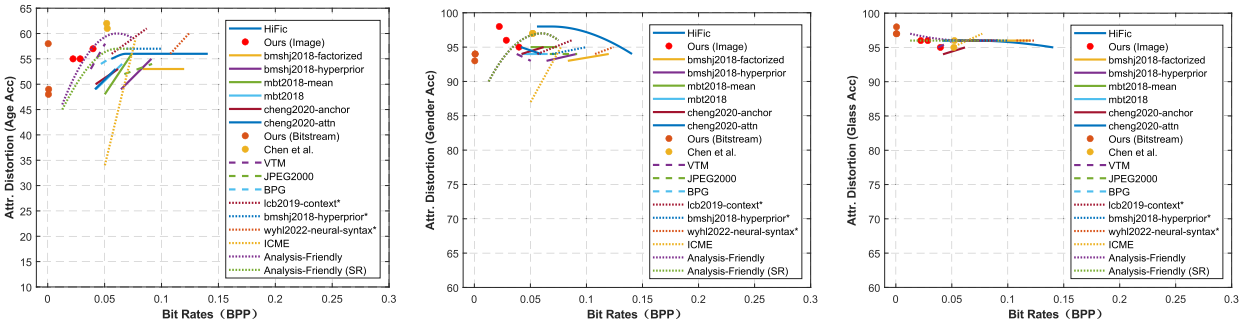


(a) JPEG (b) JPEG2000 (c) CA (d) VTM (e) HiFiC (f) Ours

(a) JPEG (b) JPEG2000 (c) CA (d) VTM (e) HiFiC (f) Ours

Fig. 7. Visual comparisons of different methods for semantic segmentation. CA denotes cheng2020-attn. It is noted that, in the ground truth, the left and right eyes are denoted in green and purple, respectively.

proposed method occupies the most left-bottom location. It shows that, our method obtains less distortion with fewer bit-rates, which clearly demonstrates our superiority.



(a) Age Accuracy (b) Gender Accuracy (c) Glass Accuracy

Fig. 8. RD-curves of different methods for attribute prediction.

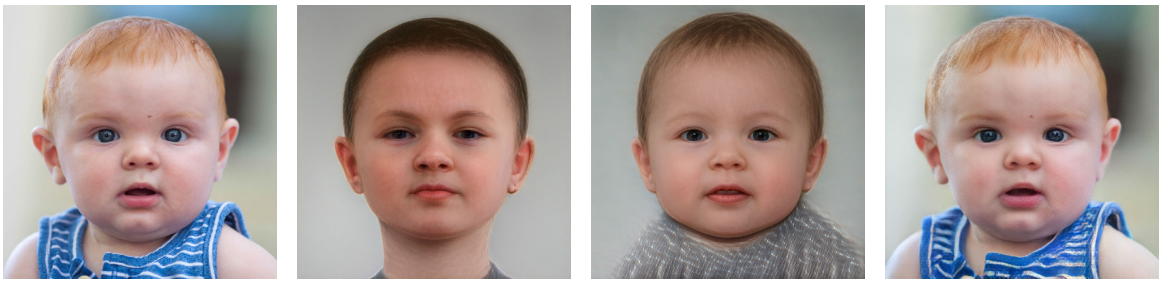
TABLE IV

QUANTITATIVE RESULTS OF DIFFERENT METHODS FOR ATTRIBUTE PREDICTION. BPP DENOTES BITS-PER-PIXEL. THE BEST AND SECOND BEST RESULTS, EXCEPT FOR THE GROUND TRUTH, ARE DENOTED IN RED AND BLUE, RESPECTIVELY

Method	BPP	Age Accuracy (%)	Gender Accuracy (%)	Glass Accuracy (%)	Setting
HiFiC	0.1407	56	94	95	Target_0.03_Lambda_2**2
HiFiC	0.0559	55	98	96	Target_0.01_Lambda_2**2
HiFiC	0.0687	56	98	96	Target_0.002_Lambda_2**2
Ours (Bitstream)	0.0006	48	94	98	
Ours (Bitstream)	0.0004	58	93	97	
Ours (Bitstream)	0.0008	49	94	97	
Ours (Image)	0.0222	55	98	96	
Ours (Image)	0.0285	55	96	96	
Ours (Image)	0.0398	57	95	95	
GT	-	60	97	96	
bmshj2018-factorized	0.0834	53	93	96	q=1
bmshj2018-factorized	0.1199	53	94	96	q=2
bmshj2018-hyperprior	0.0644	49	93	96	q=1
bmshj2018-hyperprior	0.0911	55	94	96	q=2
mbt2018-mean	0.0501	48	95	96	q=1
mbt2018-mean	0.0739	56	95	96	q=2
mbt2018	0.0423	49	94	96	q=1
mbt2018	0.0654	54	94	96	q=2
cheng2020-anchor	0.0421	50	94	96	q=1
cheng2020-anchor	0.0618	53	95	96	q=2
cheng2020-attn	0.0415	49	95	96	q=1
cheng2020-attn	0.0596	53	94	96	q=2
Icb2019-context*	0.0562	56	94	96	$\lambda = 0.0008$
Icb2019-context*	0.0870	61	96	96	$\lambda = 0.0015$
bmshj2018-hyperprior*	0.0694	57	94	96	$\lambda = 0.0008$
bmshj2018-hyperprior*	0.1000	57	95	96	$\lambda = 0.0015$
wyhl2022-neural-syntax*	0.1077	56	94	96	$\lambda = 0.0008$
wyhl2022-neural-syntax*	0.1241	60	95	96	$\lambda = 0.0015$
VTM	0.0504	58	93	96	QP=40
VTM	0.0378	53	94	95	QP=42
JPEG2000	0.0673	52	94	96	q=32
JPEG2000	0.0917	54	94	96	q=33
BPG	0.0540	55	94	96	q=42
BPG	0.0467	54	94	96	q=43
Chen et al. (qp42)	0.0523	61	97	96	QP=42
Chen et al. (qp51)	0.0518	62	97	95	QP=51
ICME (HV)	0.0772	59	95	97	
ICME (MV)	0.0502	34	87	95	
Analysis-Friendly	0.0128	46	90	97	
Analysis-Friendly (SR)	0.0128	45	90	96	
Analysis-Friendly	0.0750	59	96	96	
Analysis-Friendly (SR)	0.0750	57	96	96	
Analysis-Friendly	0.0606	60	97	96	
Analysis-Friendly (SR)	0.0606	57	97	96	

- For LPIPS at the resolution 1024×1024, HiFiC significantly outperforms other previous methods, while for LPIPS at the resolution 512×512, HiFiC's results are mixed with other learned-based methods in locations.
- VTM achieves the best results for both LPIPS at all resolutions among all traditional codecs.
- For learning-based fidelity-driven methods, cheng2020-anchor and cheng2020-attn achieve the best results.

The visual comparisons of different methods are shown in Fig. 5. It is observed that, JPEG's results suffer from severe blockiness and visual artifacts even consuming much more bit-rates. HiFiC, bmshj2018-factorized, and cheng2020-attn's results include fewer artifacts. However, HiFiC sometime includes severe artifacts in the background, *e.g.* the third row of Fig. 5 (e). Overall, HiFiC, VTM, and cheng2020-attn's results are smoother in texture regions, while our results contain richer and more vivid texture appearance, *e.g.* in scarf and hair regions in the first row, hair regions in the second row, and face wrinkles in the third raw in Fig. 5.



(a) Original (b) Single Latent Code, 0.00091 (c) Multiple Latent Codes, 0.00091 (d) Scalable Inversion, 0.0379

(e) Original (f) Single Latent Code, 0.00079 (g) Multiple Latent Codes, 0.00079 (h) Scalable Inversion, 0.0381

Fig. 9. Visual comparisons of different versions of our method for image compression. The followed number denotes the BPP of each method for the given image.

TABLE V
THE TOTAL ENCODING AND DECODING TIME OF DIFFERENT METHODS

Method	Running Time
HiFiC	6.51
Ours (Image)	1.91
bmshj2018-factorized	0.30
bmshj2018-hyperprior	0.33
mbt2018-mean	0.33
mbt2018	29.87
cheng2020-anchor	27.72
cheng2020-attn	31.24
VTM	21.11
JPEG2000	0.64
BPG	2.63
ICME	3.64
Analysis-Friendly	19.35

F. Evaluation for Semantic Segmentation

The quantitative results of different methods for semantic segmentation are shown in Table III and Fig. 6. Several conclusions are reached:

- For pixel accuracy and frequency weighted IU, our method, which locates at the top-left of the figures, wins all other compared methods, achieving higher performance when consuming the same bit-rate.
- For mean accuracy and mean IU, our method locates at the left of the figures compared to other methods. The results show that, our method achieves almost the competitive performance at the very low bit-rates cost compared to other methods.
- It seems that, HiFiC, ICME, and Analysis-Friendly achieve the worst results among all methods, considering all figures in Fig. 6.

The visual comparisons of different methods are shown in Fig. 7. It is noted that, in the ground truth, left eyes are denoted in green while right eyes are denoted in purple. It is observed that, our method generates the most correct semantic predictions for left and right eyes.

G. Evaluation for Attribute Prediction

The quantitative results of different methods for attribute prediction are shown in Table IV and Fig. 8. We obtain several interesting observations:

- All methods provide enough good results if an analysis process is performed on reconstructed images.
- If there is no expectation on much additional complexity for attribute prediction, our method with bit-stream (taking I_{attr} as the final analysis result) and Chen et al. also provide good attribute prediction results.
- Ours (bit-stream) leads to both bit-rate and complexity saving, which shows the superiority of our design in attribute prediction.

H. Evaluation for Running Time Complexity

We compare the total encoding and decoding time of different methods in Table V. All methods are compared on a machine with an AMD Ryzen Threadripper PRO 3955WX 16-Cores CPU, 128G Memory, and NVIDIA RTX A5000 GPU. JPEG2000, BPG, VTM, and Analysis-Friendly, are tested on CPU, while other methods are tested on GPUs. bmshj2018-factorized, bmshj2018-hyperprior, mbt2018-mean, mbt2018, cheng2020-anchor, cheng2020-attn are implemented by CompressAI[4]. Others are tested based on the author provided or public available implementations. It is observed that, our method achieves competitive results, only slower than JPEG2000 and much faster than other learning-based methods.

I. Ablation Studies on Different Architectures

We compare the image reconstruction results with different architectures shown in Fig. 1. The results are shown in Fig. 9.

It is observed that, GAN inversion with a single latent code can roughly reconstruct a face that looks similar to the input image at very low bit-rates. With multiple latent codes, our method is capable of reconstructing faces with a much more similar identity to the original ones, which can provide useful information to human at very low bit-rates. With the scalable GAN inversion, we obtain compression results with high fidelity and perceptual quality, also with much more bit-rate use.

VI. Conclusion

In this paper, we develop a GAN inversion-based compression framework that pursues a compact low-dimensional natural image manifold for facial image compression. With the merit of GAN training, the latent code in this space is not only highly discriminative but also capable of encoding the appearance information of the image. After applying a bit-rate constraint, a compact code is obtained to perform the multiple tasks from the view of both machine and human visions at very low bit-rates. We further extend the latent code form with the proposed *multiple latent codes*, and *scalable inversion* schemes to additionally compress and transmit a shallow compact feature to support visual reconstruction for better visual quality. Experimental results demonstrate the superiority of our method in both human vision and machine vision tasks.

References

[1] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

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

[3] S. Ma, T. Huang, C. Reader, and W. Gao, "AVS2? Making video coding smarter [standards in a nutshell]," *IEEE Signal Process. Mag.*, vol. 32, no. 2, pp. 172–183, Mar. 2015.

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

[5] S. Suzuki, M. Takagi, K. Hayase, T. Onishi, and A. Shimizu, "Image pre-transformation for recognition-aware image compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2019, pp. 2686–2690.

[6] Z. Yang et al., "Discernible image compression," in *Proc. 28th ACM Int. Conf. Multimedia*, 2020, pp. 1561–1569.

[7] Y. Hou, L. Zheng, and S. Gould, "Learning to structure an image with few colors," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10113–10122.

[8] Z. Chen, K. Fan, S. Wang, L.-Y. Duan, W. Lin, and A. Kot, "Lossy intermediate deep learning feature compression and evaluation," in *Proc. 27th ACM Int. Conf. Multimedia*, 2019, pp. 2414–2422.

[9] Z. Chen, L.-Y. Duan, S. Wang, W. Lin, and A. C. Kot, "Data representation in hybrid coding framework for feature maps compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3094–3098.

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

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

[12] D. Chen, Q. Chen, and F. Zhu, "Pixel-level texture segmentation based AV1 video compression," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2019, pp. 1622–1626.

[13] H. Li, Y. Guo, Z. Wang, S. Xia, and W. Zhu, "Adacompress: Adaptive compression for online computer vision services," in *Proc. 27th ACM Int. Conf. Multimedia*, New York, NY, USA, 2019, pp. 2440–2448.

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

[15] T. M. Hoang, J. Zhou, and Y. Fan, "Image compression with encoder–decoder matched semantic segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Oct. 2020, pp. 619–623.

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

[17] J.-Y. Zhu, P. Krähenbühl, E. Shechtman, and A. Alexei Efros, "Generative visual manipulation on the natural image manifold," in *Proc. Eur. Conf. Computer Vis.*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds., Cham, Switzerland: Springer, 2016, pp. 597–613.

[18] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 33, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, Eds. Red Hook, NY, USA: Curran Associates, 2020, pp. 12104–12114.

[19] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4396–4405.

[20] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN: How to embed images into the StyleGAN latent space?" in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4431–4440.

[21] R. Abdal, Y. Qin, and P. Wonka, "Image2StyleGAN++: How to edit the embedded images?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8293–8302.

[22] A. Creswell and A. A. Bharath, "Inverting the generator of a generative adversarial network," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 1967–1974, Jul. 2019.

[23] Z. C. Lipton and S. Tripathi, "Precise recovery of latent vectors from generative adversarial networks," 2017, *arXiv:1702.04782*.

[24] G. Perarnau, J. V. D. Weijer, B. Raducanu, and M. J. Álvarez, "Invertible conditional GANs for image editing," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2016, pp. 1–9.

[25] S. Pidhorskyi, D. A. Adjeroh, and G. Doretto, "Adversarial latent autoencoders," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14092–14101.

[26] Y. Nitzan, A. Bermano, Y. Li, and D. Cohen-Or, "Face identity disentanglement via latent space mapping," *ACM Trans. Graph.*, vol. 39, no. 6, pp. 1–14, Nov. 2020.

[27] S. Guan, Y. Tai, B. Ni, F. Zhu, F. Huang, and X. Yang, "Collaborative learning for faster StyleGAN embedding," 2020, *arXiv:2007.01758*.

[28] J. Zhu, Y. Shen, D. Zhao, and B. Zhou, "In-domain GAN inversion for real image editing," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 592–608.

[29] S. Suzuki, M. Takagi, S. Takeda, R. Tanida, and H. Kimata, "Deep feature compression with spatio-temporal arranging for collaborative intelligence," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3099–3103.

[30] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Computer Vis.*, 2016, pp. 1–9.

[31] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.

[32] E. Richardson et al., "Encoding in style: A styleGAN encoder for image-to-image translation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2287–2296.

[33] Y. Hu, W. Yang, Z. Ma, and J. Liu, "Learning end-to-end lossy image compression: A benchmark," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 8, pp. 4194–4211, Mar. 2021.

[34] R. A. Cohen, H. Choi, and I. V. Bajic, "Lightweight compression of intermediate neural network features for collaborative intelligence," *IEEE Open J. Circuits Syst.*, vol. 2, pp. 350–362, 2021.

[35] X. Wang, K. Yu, C. Dong, and C. Change Loy, "Recovering realistic texture in image super-resolution by deep spatial feature transform," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 606–615.

[36] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019.

[37] R. Or-El, S. Sengupta, O. Fried, E. Shechtman, and I. Kemelmacher-Shlizerman, "Lifespan age transformation synthesis," in *Proc. Eur. Conf. Comput. Vis.*, vol. 12351, 2020, pp. 739–755.
[38] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–11.
[39] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 31, S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, Eds., 2018, pp. 1–10.
[40] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learned image compression with discretized Gaussian mixture likelihoods and attention modules," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 7939–7948.
[41] J. Lee, S. Cho, and S.-K. Beack, "Context-adaptive entropy model for end-to-end optimized image compression," in *Proc. Int. Conf. Learn. Represent.*, May 2019, pp. 1–20.
[42] D. Wang, W. Yang, Y. Hu, and J. Liu, "Neural data-dependent transform for learned image compression," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 17379–17388.
[43] F. Mentzer, G. D. Toderici, M. Tschannen, and E. Agustsson, "High-fidelity generative image compression," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 11913–11924.
[44] Y. Hu, S. Yang, W. Yang, L.-Y. Duan, and J. Liu, "Towards coding for human and machine vision: A scalable image coding approach," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2020, pp. 1–6.
[45] S. Wang et al., "Towards analysis-friendly face representation with scalable feature and texture compression," *IEEE Trans. Multimedia*, vol. 24, pp. 3169–3181, 2021.
[46] G. K. Wallace, "Overview of the JPEG (ISO/CCITT) still image compression standard," *Proc. SPIE*, vol. 1244, pp. 220–233, Jun. 1990.
[47] B. Bross et al., "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.
[48] X. Wang et al., "ESRGAN: Enhanced super-resolution generative adversarial networks," in *Proc. Eur. Conf. Comput. Vis. Worksop*, Sep. 2018, pp. 1–16.
[49] Y. Zhang, Y. Tian, Y. Kong, B. Zhong, and Y. Fu, "Residual dense network for image super-resolution," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 2472–2481.
[50] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 3431–3440.

Wenhan Yang (Member, IEEE) received the B.S. and Ph.D. (Hons.) degrees in computer science from Peking University, Beijing, China, in 2012 and 2018, respectively. He is currently an Associate Researcher with the Peng Cheng Laboratory, Shenzhen, Guangdong, China. From 2021 to 2022, he was with the Rapid-Rich Object Search Laboratory, School of EEE, Nanyang Technological University, Singapore, as a Wallenberg-NTU Presidential Post-Doctoral Researcher. He has authored over 50 technical articles in refereed journals and proceedings and holds nine granted patents. His current research interests include image/video processing/restoration, bad weather restoration, and human–machine collaborative coding. He received the IEEE ICME-2020 Best Paper Award, the IFTC 2017 Best Paper Award, the IEEE CVPR-2018 UG2 Challenge First Runner-Up Award, and the MSA-TC Best Paper Award of ISCAS 2022. He was the Candidate of CSIG Best Doctoral Dissertation Award in 2019.

Haofeng Huang (Student Member, IEEE) received the B.S. degree in computer science from Peking University, Beijing, China, in 2021, where he is currently pursuing the Ph.D. degree with the Wangxuan Institute of Computer Technology. His current research interests include deep-learning based image/video processing, intelligent visual enhancement, and image/video coding for machines.

Jiaying Liu (Senior Member, IEEE) received the Ph.D. degree (Hons.) in computer science from Peking University, Beijing, China, in 2010. She is currently an Associate Professor and the Boya Young Fellow with the Wangxuan Institute of Computer Technology, Peking University. From 2007 to 2008, she was a Visiting Scholar with the University of Southern California, Los Angeles, CA, USA. She was a Visiting Researcher with the Microsoft Research Asia in 2015, supported by the Star Track Young Faculties Award. She has authored more than 100 technical articles in refereed journals and proceedings and holds 60 granted patents. Her current research interests include multimedia signal processing, compression, and computer vision. She is a Senior Member of CSIG and CCF. She was a member of the Multimedia Systems and Applications Technical Committee (MSA TC) and the Visual Signal Processing and Communications Technical Committee (VSPC TC) in IEEE Circuits and Systems Society. She is also a member of the Image, Video, and Multimedia (IVM) and Signal and Information Processing Theory and Methods (SIPTM) Technical Committee in APSIPA. She was a recipient of the IEEE ICME-2020 Best Paper Awards and IEEE MMSP 2015 Top10% Paper Awards. She was the Technical Program Chair of IEEE ICME-2021 and ACM ICMR-2021, the Area Chair of CVPR-2021/ECCV-2020/ICCV-2019, and the CAS Representative at the ICME Steering Committee. She was an Associate Editor of IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, and Elsevier *Journal of Visual Communication and Image Representation*. From 2016 to 2017, she was a APSIPA Distinguished Lecturer.

Alex C. Kot (Life Fellow, IEEE) has been with Nanyang Technological University, Singapore, since 1991. He was the Head of the Division of Information Engineering and the Vice Dean of research with the School of Electrical and Electronic Engineering. Subsequently, he served as an Associate Dean for the College of Engineering, for eight years. He is currently a Professor and the Director of the Rapid-Rich Object SEarch (ROSE) Laboratory and the NTU-PKU Joint Research Institute. He has published extensively in the areas of signal processing, biometrics, image forensics and security, computer vision, and machine learning. He is a fellow of the Academy of Engineering, Singapore. He received the Best Teacher of the Year Award and the coauthor for several Best Paper Awards, including ICPR, IEEE WIFS, IWDW, CVPR Precognition Workshop, and VCIP. He served on the IEEE SP Society in various capacities, such as the General Co-Chair for the 2004 IEEE International Conference on Image Processing and the Vice-President for the IEEE Signal Processing Society. He served as an Associate Editor for more than ten journals, mostly for IEEE TRANSACTIONS. He was elected as the IEEE Distinguished Lecturer for the Signal Processing Society and the Circuits and Systems Society.