Preprocessing Enhanced Image Compression for Machine Vision

Guo Lu†, Member, IEEE, Xingtong Ge‡, Student Member, IEEE, Tianxiong Zhong‡, Qiang Hu‡, and Jing Geng‡

Abstract—Recently, more and more images are compressed and sent to the back-end devices for machine analysis tasks (e.g., object detection) instead of being purely watched by humans. However, most traditional or learned image codecs are designed to minimize the distortion of the human visual system without considering the increased demand from machine vision systems. In this work, we propose a preprocessing enhanced image compression method for machine vision tasks to address this challenge. Instead of relying on the learned image codecs for end-to-end optimization, our framework is built upon the traditional non-differential codecs, which means it is standard compatible and can be easily deployed in practical applications. Specifically, we propose a neural preprocessing module before the encoder to maintain the useful semantic information for the downstream tasks and suppress the irrelevant information for bitrate saving. Furthermore, our neural preprocessing module is quantization adaptive and can be used in different compression ratios. More importantly, to jointly optimize the preprocessing module with the downstream machine vision tasks, we introduce the proxy network for the traditional non-differential codecs in the back-propagation stage. We provide extensive experiments by evaluating our compression method for several representative downstream tasks with different backbone networks. Experimental results show our method achieves a better trade-off between the coding bitrate and the performance of the downstream machine vision tasks by saving about 20% bitrate.

Index Terms—Image compression, machine vision, preprocessing, deep learning.

I. INTRODUCTION

WITH the successful applications of deep neural networks, the machine vision tasks such as detection and classification have made a lot of progress in recent years [1], [2], [3], [4], [5], [6], [7]. Therefore, more and more images are captured by the front-end devices (e.g., cameras) and sent to the back-end (e.g., cloud servers) for machine analysis.

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCSVT.2024.3441049.

Manuscript received 26 December 2023; revised 25 May 2024 and 14 July 2024; accepted 31 July 2024. Date of publication 9 August 2024; date of current version 23 December 2024. This work was supported by the National Natural Science Fund of China under Project 62102024, Project 62331014, and Project 42201461. This article was recommended by Associate Editor C. Herglotz. (Corresponding authors: Qiang Hu; Jing Geng.)

Guo Lu is with the School of Electronics Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: luguo2014@sjtu.edu.cn).

Xingtong Ge was with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China. He is now with SenseTime Technology, Beijing 100080, China (e-mail: gexingtong@sensetime.com).

Tianxiong Zhong and Jing Geng are with the School of Computer Science, Beijing Institute of Technology, Beijing 100081, China (e-mail: inkoizhong@gmail.com; janecheng@bit.edu.cn).

Qiang Hu is with the School of Electronics Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: qiang.hu@sjtu.edu.cn).

According to the report from Cisco [8] and Sandvine [9], the percentage of the connections from this machine-to-machine scenario will be up to 50% in the future, and the complexity of downstream application scenarios is also increasing. Therefore, how to reduce the transmission bitrate while maintaining performance for the downstream vision tasks has become a challenge for the image compression field.

Unfortunately, although several traditional image compression standards, such as JPEG [10] and BPG [11], have been proposed in the past decades, they are designed to minimize the compression distortion for the human visual system (e.g., PSNR) instead of the machine vision tasks (see Fig. 1(a)). More importantly, most compression standards are non-differential, which cannot be jointly optimized with the neural network based machine analysis methods. Therefore, the existing compression-then-analysis pipeline with the traditional codecs may not be optimal when we mainly focus on the performance of the downstream machine analysis. Recently, learned image compression methods [12], [13], [14] start to gain a lot of attention. Several approaches [15], [16], [17], [18], [19] also try to jointly optimize the learned compression methods with the downstream analysis tasks. For instance, Sun et al. [18] proposed a learning-based semantically structured image compression method, which divided the bitstream into different semantic parts to support different machine vision tasks, such as image classification. Wang et al. [19] pro-

Fig. 1. (a) Image compression method for human visual system. (b) Our proposed preprocessing enhanced image compression for machine vision tasks. (c) Image classification results for the image from the BPG codec and ours(NPP+BPG).

Authorized licensed use limited to the terms of the applicable license agreement with IEEE. Restrictions apply.
posed an joint compression and analysis framework, achieving variable bitrate coding and rate-accuracy optimization for detection and segmentation tasks. Kim et al. [20] proposed an end-to-end learnable multi-scale feature compression method, optimizing machine vision performance with a novel fusion and encoding process. However, the computational complexity for the learned image codec is usually high, and the standardization is not finalized; therefore, the massive deployment of learned compression approaches is unlikely to happen soon, which means these approaches [15], [16], [17], [18] may not be feasible in practical applications.

To address these challenges, we propose a preprocessing enhanced image compression framework for machine vision as shown in Fig. 1(b). Our framework builds upon the traditional standard compatible image codecs and can be easily applied to the practical compression-then-analysis systems. Specifically, we propose a neural preprocessing (NPP) module before the traditional codec and the input image will be filtered before encoding. After that, the decoded image is used for the downstream tasks, like detection or classification. To enable the end-to-end optimization, we further introduce the proxy network for the traditional non-differential image codecs (e.g., BPG) in the training stage, where the gradients of the proxy network are propagated to the neural preprocessing module. Therefore, the proposed preprocessing module will be optimized to maintain the meaningful semantic information and reduce the irrelevant information for machine vision tasks, which leads to a better trade-off between the coding bitrate and machine analysis performances (see Fig. 1(c)).

Furthermore, the proposed neural preprocessing module is quantization adaptive and can be integrated into traditional codecs with different compression ratios. To demonstrate the superiority of our preprocessing enhanced image compression method, we perform extensive experiments on several representative machine vision tasks (e.g., object detection and image classification) with different downstream backbone networks. Experiments show that compared with the existing traditional codecs like BPG [11], the proposed approach can save about 20% bitrate for the downstream tasks with the same accuracy.

The main contributions of our work are summarized as follows,

- Building upon the traditional codec, we propose a neural preprocessing module to generate the filtered images, which can be effectively compressed by the traditional codecs with high machine perception performance.
- To enable an end-to-end optimization for a better trade-off between coding bitrate and machine perception performance, we introduce the learned proxy network to approximate the traditional codecs for the back-propagation in the training stage.
- Experimental results show our approach is general and the optimized NPP model for one specific scenario can be used for other codecs, downstream backbones, or even other tasks.

II. RELATED WORKS

A. Image Compression

Many traditional image compression algorithms [10], [11], [21], [22], [23] have been proposed in the past decades. These methods are based on hand-craft techniques (e.g., Discrete Cosine Transform) to reduce spatial redundancy. Recently, the learned image compression methods [12], [13], [14], [24], [25], [26], [27], [28], [29] have become popular. The mainstream methods [12], [13], [14], [26] adopt an auto-encoder style network to convert the images to the latent representations, which are further encoded by entropy coding. For example, Ballé et al. [12] proposed using a convolutional neural network (CNN) to learn non-linear transformations and additionally introduced a hyper-prior network to model the probability distribution of the latent representations [13]. Minnen et al. [14] introduce an autoregressive module for a more accurate entropy model. Latest works [26], [27], [28], [29], [30], [31], [32] also propose to use more powerful transform networks, such as residual blocks [27], nonlocal layers [28], invertible layers [29] and transformer [26]. However, although these learning-based compression methods have achieved better compression performance, there is no unified coding standard for them, which makes large-scale deployment of practical applications more difficult.

B. Image Compression for Machine Vision

Most existing image compression methods [10], [11], [12], [13] aim to reduce reconstruction distortion in terms of the human visual system and are optimized based on pixel field metrics such as PSNR. With the development of deep learning, some studies began to explore connections between compression and downstream tasks [18], [33], [34], [35], some studies [15], [16], [17], [20], [36], [37], [38], [39], [40], [41] also focus on joint optimization of the image compression and the downstream machine vision tasks. These innovations can be broadly categorized into following distinct approaches.

Firstly, several studies emphasize using compressed features to directly support vision tasks. For instance, Torfason et al. [15] proposed to utilize the compressed representations produced by learning-based image codec to perform image understanding tasks, such as classification and segmentation. Following this, Choi and Bajić [40] introduced a unified scalable image coding framework for both humans and machines, which divides the coding bitstream into different layers and takes advantage of feature space scalability to achieve obviously superior compression efficiency for tasks like object detection and segmentation. Recently, a self-supervised learning scheme [42] is proposed to constrain the intermediate-layer features to be semantics-complete and achieved high performances in different downstream vision tasks.

Besides, there are efforts concentrating on using the decoded images to assist in various vision tasks. For example, Fischer et al. [39] introduced the feature loss and latent space masking to optimize the image compression network, enhancing the balance between bitrate cost and analysis accuracy. Wang et al. [19] proposed an integrated compression and machine analysis optimization framework, utilizing variable bitrate coding and rate-accuracy optimization for better object detection and segmentation in machine vision systems. Liu et al. [43] proposed a partitioning and aggregation method for learning base image codecs and optimized the partitioned bitstreams for machine vision tasks such as...
object detection and image classification, etc. Additionally, there’re also some task-specific optimizations proposed to address unique challenges in different domains. This includes works [44], [45], [46] learning analysis-friendly representation for face compression, and prior-guided contrastive image compression specifically designed for underwater machine vision by Fang et al. [47].

Despite the innovations, most existing works have to rely on the learning based codecs to enable the end-to-end optimization, which may not be feasible in the practical application considering the mainstream codecs are traditional ones. In contrast, our framework is built upon the traditional codecs and can also be end-to-end optimized through the proxy network.

C. Preprocessing

In the past decades, several methods [48], [49], [50] have been proposed to use the preprocessing technique to improve the performance of the image and video compression algorithms. Most of these methods are based on the Just Noticeable Distortion (JND) technique [51] and try to improve the perceptual quality of reconstructed frames. For example, Xiang et al. [48] proposed adaptive perceptual preprocessing by removing information that is not perceptible to the human visual system. Vidal et al. [49] combined several adaptive filters to denoise the image for bitrate saving.

In recent years, several learning-based preprocessing methods have been proposed [52], [53], [54], [55]. Chadha and Andreopoulos [52] proposed a rate-aware perceptual pre-processing module for video coding. Guleryuz et al. [53] proposed neural network based preprocessing and postprocessing modules to improve the compression performance of the traditional codecs. Talebi et al. [54] designed a pre-editing neural network on the JPEG method to improve the visual quality of reconstructed images. Klopp et al. [56] proposed to borrow the rate estimator of learning-based codec to boost the perceptual quality of traditional codecs such as JPEG. We propose using the neural network based preprocessing method to improve the compression performance in machine vision instead of the human visual system.

III. PROPOSED METHOD

A. Overview

The overall architecture of our preprocessing enhanced image compression framework for machine vision is shown in Fig. 2. The whole system aims to achieve a better trade-off between coding bitrate and the performance of the machine analysis task. Specifically, we first feed the input image $X$ to the neural preprocessing module (NPP) for non-linear transform and generate the filtered image $\bar{X}$, which is expected to maintain the critical semantic information. Then, $\bar{X}$ is encoded and reconstructed by a traditional codec, like BPG [11]. Finally, the decoded $\hat{X}$ is input to machine analysis networks, such as FCOS [1].

Since the traditional codecs may not be differential, the proposed preprocessing module cannot enjoy the benefits of the joint end-to-end optimization with the downstream machine analysis tasks. To solve this problem, we additionally introduce a learned image compression network as the proxy network for the traditional codec in the training stage and the gradients of the proxy network are propagated to the preprocessing module (see Section III-C for more details). Here, we use BPG [11] as the traditional codec in our implementation.

Then the framework is optimized by using the following loss function,

$$
\mathcal{L} = R_t + \lambda D_m + \beta D_{\text{pre}}
$$

where $D_m$ and $R_t$ represent the loss of the downstream machine vision task based on reconstructed image $\hat{X}$ and coding bitrate from the traditional codec, respectively. $\lambda$ is a hyper-parameter used to control the trade-off. In addition, to stabilize the training process, we also consider the distortion between the input image $X$ and the enhanced image $\bar{X}$, which is denoted as $D_{\text{pre}}$. $\beta$ is the constant weight parameter.

B. Neural Preprocessing Network

As shown in Fig. 3, we provide the network architecture of our neural preprocessing module. Specifically, the original image $X$ is input into two parallel branches, where the first branch uses $1 \times 1$ convolutional layers to learn non-linear pixel-level transforms, and the second branch uses a U-Net [57] style network to extract the semantic information. The outputs of two branches are added together as the final filtered image $\bar{X}$, which preserves the useful texture and semantical information through both shallow and deep transforms.

Furthermore, considering the traditional codecs have different compression ratios (i.e., quantization parameter), the neural preprocessing module is required to generate the optimal filtered image $\bar{X}$ for each compression ratio. Here, we propose
a quantization adaptation layer for the neural preprocessing module, which leads to an adaptive preprocessing based on the quantization parameters in the codec. As shown in Fig. 3, we integrate the quantization adaptive layer into the NPP module and scale the intermediate features for adaptive filtering. Specifically, based on the given quantization parameter (QP) in the traditional codec, we use a 2-layer MLP network to generate the scale vector $s$ and the output feature $f'$ is the channel-wise multiplication product between input feature $f$ and the generated scale vector $s$, i.e., $f' = f \odot s$. Based on this strategy, the intermediate features in the preprocessing module will be modulated based on the quantization parameter; therefore, our module will generate the optimal filtered image $\bar{X}$ for the given QP in the BPG codec and achieve a better rate-accuracy trade-off.

Here we give an example in Fig. 4 to show the effectiveness of our preprocessing module. Fig. 4(a) and (b) represent the original image and output from the NPP module, respectively. Moreover, the corresponding compressed file sizes using the BPG [11] ($QP = 37$) codec are 63.7kb and 47.0kb. At the same time, Fig. 4(c) shows that the information discarded by the preprocessing module is mainly distributed in the background region. In contrast, based on the GradCAM method [58], the classification network [5] focuses on the foreground Dingo in the image as shown in Fig. 4(d,e). Simultaneously, Fig. (d) and Fig. (e) reveal that the classification model focuses more extensively on the expected region of the Dingo in the preprocessed image, potentially leading to a more accurate classification outcome. These results prove that the preprocessing module can preserve more critical semantic information for the downstream analysis tasks and reduce the irrelevant information for bitrate saving. We’ll present more illustrations in the section of experimental results.

## Table I

| The QP-$\lambda_p$ Pairs Used in Our Framework |
|---|---|---|---|---|---|---|
| QP  | 44 | 41 | 37 | 34 | 31 | 28 |
| $\lambda_p$ | 128 | 256 | 512 | 1024 | 2048 | 4096 |

## C. Proxy Network

In our framework, to enable an end-to-end optimization for the whole system, a learned image compression network is introduced as the proxy network to replace the traditional codec during the backward propagation stage. Here, we use Minnen’s approach [14] as our proxy network.

To make sure that the proxy network can well approximate the traditional codec, the reconstruction quality of the BPG and Minnen’s approach should be similar. The learned image compression approach [14] is optimized based on Rate-Distortion (R-D) loss $R + \lambda_p D$ and the quality of the reconstructed image depends on the hyper-parameter $\lambda_p$. We start by selecting a pre-trained image compression model, which, after being optimized with the R-D distortion loss and a suitable $\lambda_p$ parameter, has a performance comparable to the BPG. The paired hyper-parameter $\lambda_p$ and $QP$ are provided in Table I. To make it even more aligned with BPG, we then fine-tune the proxy network using the following approach,

$$L_p = R_p + \lambda_p D_p = R_p + \lambda_p d(\hat{X}, \hat{Y})$$

(2)
where \( d(\hat{X}, \hat{Y}) \) denotes the distortion between the reconstructed image \( \hat{X} \) from the BPG codec and the reconstructed image \( \hat{Y} \) from the proxy network (see Fig. 2). \( R_p \) represents the corresponding bitrate from the proxy network. After that, we obtain an optimized proxy codec to mimic the BPG codec.

**Algorithm 1 Training Procedure**

Require: training dataset \( D = \{(X_k)_k\}_{k=1}^m \), trade-off parameter \( \lambda, \beta = 0.5 \), learning rate \( \eta \)

\[ 1: \text{for all } (X) \in D \text{ do} \]
\[ 2: \hat{X} \leftarrow \text{Preprocessor}(X|\theta_{\text{pre}}) \quad \triangleright \text{\( \theta_{\text{pre}} \) represents the parameters of preprocessing module} \]
\[ 3: \hat{X}, R_t \leftarrow \text{BPG}(\hat{X}) \]
\[ 4: \hat{Y}, R_p \leftarrow \text{Proxy Network}(\hat{X}) \]
\[ 5: \hat{Y}.\text{data} \leftarrow \hat{X}.\text{data} \]
\[ 6: R_p.\text{data} \leftarrow R_t.\text{data} \]
\[ 7: \hat{O} \leftarrow \text{detector}(\hat{Y}) \]
\[ 8: L \leftarrow R_p + \lambda \cdot \text{loss}(\hat{O}) + \beta \cdot \text{mse}(X, \hat{X}) \quad \triangleright \text{calculate loss} \]
\[ 9: g_{\text{task}} \leftarrow \frac{\partial L}{\partial \hat{O}} \cdot \frac{\partial \hat{O}}{\partial \hat{Y}} \quad \triangleright \text{calculate gradients of detector} \]
\[ 10: g_p \leftarrow g_{\text{task}} \cdot \frac{\partial \hat{Y}}{\partial X} + \frac{\partial L}{\partial R_p} \cdot \frac{\partial R_p}{\partial X} \quad \triangleright \text{calculate gradients of Proxy Network} \]
\[ 11: g_{\text{pre}} \leftarrow g_p \cdot \frac{\partial \hat{Y}}{\partial X} + \frac{\partial \beta \cdot \text{mse}(X, \hat{X})}{\partial X} \cdot \frac{\partial \hat{X}}{\partial X} \quad \triangleright \text{calculate gradients of Preprocessor} \]
\[ 12: \theta_{\text{pre}} \leftarrow \theta_{\text{pre}} - \eta \cdot g_{\text{pre}} \quad \triangleright \text{optimize parameters of Preprocessor} \]
\[ 13: \text{end for} \]

We provide more implementation details of the end-to-end training procedure in Algorithm 1. In forward propagation, we can get the processed image \( \hat{X} \) based on the input image \( X \), where \( \theta_{\text{pre}} \) represents the parameters of preprocessing module. Then the processed image is compressed by BPG codec and BPG will calculate the bitrate \( R_t \) and produce the reconstructed image \( \hat{X} \). At the same time, we also generate the corresponding reconstructed image \( \hat{Y} \) and bitrate \( R_p \) based on the proxy network. Here, the values of \( \hat{Y} \) and \( R_p \) will be reassigned to \( \hat{X} \) and \( R_t \) from the BPG codec as shown in Line 5 and 6 in Algorithm 1. Then the reassigned \( \hat{Y} \) is input to the analysis models (e.g., an object detection module) and used to calculate the machine perception loss \( P_m \). After that, we can calculate the loss function as shown in Line 8. Based on this operation, we can use the bitrate and reconstructed image from the BPG codec in forward propagation and calculate the value of loss function while using the gradients of the proxy network in backward propagation. Finally, we perform the backward propagation, and optimize the weights of the neural preprocessing module, as shown in Line 9-12 in Algorithm 1. In the backward propagation, the gradients of the machine vision task model, proxy network, and preprocessing module, denoted as \( g_{\text{task}}, g_p, \) and \( g_{\text{pre}} \), will be calculated sequentially. And the weight of preprocessing module will be optimized while the weights of other modules are fixed.

**IV. EXPERIMENTS**

**A. Experimental Setup**

1) Datasets, Backbone Models and Evaluation Metrics:

For the proxy network, we adopted the Flicker [59] Dataset for training. We randomly cropped the images from Flicker dataset into size of 256 \( \times \) 256 and obtained a total of about 800k images, which constitute our training set.

For the object detection task, our framework is trained on the COCO 2017 training dataset [60]. The mean average precision (mAP) results are reported by evaluating the proposed framework on the COCO 2017 validation set which contains 5k images. Our empirical analysis incorporates three object detection baselines: FCOS [1], Faster RCNN [2], and RetinaNet [3] for comprehensive evaluation.

In the context of image classification, we utilize the ImageNet dataset [61], encompassing 1.28 million training images and 50,000 validation images across 1,000 classes. Here, Top-1 accuracy metric is employed to gauge performance. To validate the efficacy of our method, we also use three classification models: ResNet [5], Swin Transformer [62] and Vision Transformer [63] for evaluation.

To showcase the inter-task generalization ability of our proposed NPP module, we also evaluate our approach for the semantic segmentation task and pose estimation tasks and the corresponding backbone networks are DeepLabv3 [64] and Deeppose [65]. To further demonstrate its generalization capability across different codecs, we extend our evaluation to traditional codecs, including JPEG via the libjpeg-turbo [66] and Versatile Video Coding (VVC) via the VVC codec [67].

Additionally, we assess the compression performance relative to the human visual system by employing perceptual metrics like LPIPS [68] on the Kodak dataset [25]. Throughout all experiments, bits-per-pixel (bpp) is the standard metric for quantifying coding costs associated with the compression processes.

2) Implementation Details: Our whole framework is implemented on PyTorch [69] with CUDA support and trained on one RTX 3090 GPU card. We use BPG [11] as the default traditional codec with different \( Q P \) values (\( QP = \{28, 31, 34, 37, 41\} \)) and the corresponding \( \lambda \)s in Eq. 1 are set as \{0.5, 1, 2, 4, 8\}. The trade-off parameter \( \beta \) is set as 0.5. The weights of the downstream networks, like FCOS [1], remain fixed throughout the entire training process unless stated otherwise.

The whole training process has the following stages: First, based on the finetuning procedure in Section III-C, we can obtain several proxy networks that mimic the BPG codec with different quantization parameters. Then we end-to-end optimize the neural preprocessing module without the quantization adaptive layers according to the loss function in Eq. 1 and set the \( QP \) of the BPG codec to a fixed value, such as \( QP = 34 \). Finally, we add the quantization adaptive layers into the preprocessing module and further train the preprocessing module by randomly sampling \( QP \) values. Specifically, we use the Adam optimizer [70] and the initial learning rate is set as 1e \(-4\). The framework is optimized for 400k, 120k and 100k steps during the three training stages. And the learning rate is
Fig. 5. (a) Rate-accuracy (mAP) results of different compression methods on the object detection task. (b) Rate-accuracy (Top-1 accuracy) results of different compression methods on the image classification task.

### TABLE II

| Anchor    | FCOS BDBR (%) | Faster RCNN BDBR (%) | RetinaNet BDBR (%) |
|-----------|---------------|----------------------|--------------------|
| BPG       | -20.2         | -18.2                | -17.3              |
| Minnen    | -19.5         | -20.5                | -19.0              |

### TABLE III

| Anchor    | ResNet50 BDBR (%) | Swin Transformer BDBR (%) | Vision Transformer BDBR (%) |
|-----------|-------------------|---------------------------|-----------------------------|
| BPG       | -22.5             | -19.6                     | -16.2                       |
| Minnen    | -21.2             | -18.4                     | -18.6                       |

B. Main Results

We compare our preprocessing enhanced image compression method with existing traditional codec BPG [11] and neural network based compression model [14]. BDBR [71] is used to measure compression performance in terms of the accuracy of the downstream tasks and the negative value represents the bitrate saving at the same accuracy. We use FCOS [1] and ResNet50 [5] as the default backbone networks for object detection and image classification and train the corresponding NPP modules, respectively.

1) Object Detection: Fig. 5(a) shows the rate-accuracy curve from the different backbone networks and compression approaches on the COCO dataset. It is obvious that our preprocessing enhanced image compression method shows a much better rate-accuracy trade-off than the baseline approaches on the downstream object detection task. Specifically, compared with the existing BPG codec and learned compression model, our neural preprocessing enhanced codec saves 20.3% and 19.5% bitrate at the same mAP value when evaluated on FCOS, respectively. We further provide the visualization results in the following sections.

2) Image Classification: We also compare our method with the traditional and learning based codecs on the image classification task. Fig. 5(b) shows the rate-accuracy (Top-1) curves from different compression methods on the ImageNet dataset [61]. It is noted that our approach still achieves better rate-accuracy performance and saves about 22.5% bitrate when compared with traditional codec BPG [11] by evaluating it on the ResNet50 [5] model.

C. Generalization Ability

To illustrate the generalization capacity of the optimized neural preprocessing (NPP) module, we employed the pre-trained module across various downstream backbone networks, tasks, and traditional codecs. It is observed that our NPP module shows strong generalization ability.

1) Different Backbones Networks: For the object detection task, we extend previous experiments by applying the...
Fig. 6. (a) Rate-accuracy (mAP) results of fine-tuning FCOS using compressed images for the object detection task. (b) Rate-accuracy (Top-1 accuracy) results of fine-tuning ResNet50 using compressed images for the image classification task.

TABLE IV

| Anchor Task (Backbone) | Detection (FCOS) | Classification (ResNet50) | Segmentation (DeepLabV3) | Pose estimation (DeepPose) |
|------------------------|------------------|---------------------------|--------------------------|---------------------------|
| Detection (FCOS)       |                  | -21.7                     | -12.3                    | -25.6                     |
| Classification (ResNet50) | -13.9           | -12.6                     | -14.4                    |                           |

FCOS-optimized NPP module to various backbone networks, including RetinaNet. Fig. 5(a) demonstrates that our compression method significantly surpasses baseline techniques, yielding bitrate reductions of 19.5% and 18.8% compared with BPG for the downstream Faster RCNN and RetinaNet models, respectively. Analogous outcomes are evident in Fig. 5(b) where the NPP module, initially trained on ResNet50, is adapted to alternative architectures such as Swin Transformer and Vision Transformer. Detailed BD-Rate metrics for these adaptations are presented in Tables II and III.

2) Different Tasks: Furthermore, we adapted the pre-trained Neural Processing Pipeline (NPP) module, initially optimized for a single specific task, to three additional vision tasks without the need for fine-tuning. The outcomes, denoted as “Ours(Trans.)” curves, are depicted in Figures 7(d)-(i). It was observed that our method maintains efficacy and realizes substantial bitrate savings across diverse tasks. The corresponding BD-RATE metrics are documented in Table IV. For instance, applying the detection-optimized NPP module to the pose estimation task resulted in a 24.6% bitrate reduction.

3) Different Codecs: To ascertain the generalizability of our approach across various codecs, we deployed the NPP module, originally optimized for the BPG codec, to JPEG and VVC codecs without necessitating any fine-tuning. Experimental findings presented in Fig. 7(b) demonstrate that our enhanced JPEG preprocessing achieves bitrate reductions exceeding 8.5% and 10.0% over the standard JPEG codec when applied to FCOS and Faster RCNN backbone networks, respectively. Similarly, as depicted in Fig. 7(c), our enhanced VVC preprocessing facilitates more than 15.6% and 12.9% bitrate reductions compared to the conventional VVC codec, when assessed on ResNet and ResNeSt networks, respectively.

D. Ablation Study and Model Analysis

1) Analysis of the Generalization Ability: Tasks like detection and classification often focus on similar regions of interest within images, typically including salient features of objects—such as textures or distinctive shapes that are crucial for identifying and classifying objects. Our NPP module is designed to enhance and preserve the information from these critical areas. By doing so, it ensures that key visual cues are retained, which are pivotal for the success of downstream tasks. This design philosophy supports the notion that, despite variations in specific task objectives, the underlying visual processing requirements share significant overlap.

To empirically validate the adaptability of our NPP module across different vision tasks, we employed GradCAM to visualize focus areas across different models. As shown in Fig. 10, an image from the COCO dataset (ID: 00000001323), underwent object detection using Faster RCNN, targeting the detection of a “cat”. The detected bounding box area and its corresponding heatmap are illustrated in Fig. 10 Left and Middle, respectively. Subsequently, this cropped image was processed through ResNet, which also produced a heatmap for the specified area, shown in Fig. 10 Right. The heatmaps vividly illustrate that both models, despite their distinct task-specific objectives, concentrate on largely overlapping regions within the images. Although our NPP module is optimized for
one of the tasks, the overlap in focus areas underscores our NPP module’s ability to generalize across tasks, reinforcing its utility and effectiveness in a multi-task environment. The consistent heatmap patterns across tasks not only demonstrate the network’s robustness but also affirm its potential to facilitate various machine vision applications without the need for task-specific tuning.

2) Analysis of End-to-end Optimization: In the proposed method, BPG is employed to generate reconstructed images during forward propagation, while backward propagation utilizes the gradients of the proxy network to update the preprocessing module’s parameters. We additionally present results from exclusively using the proxy network for both forward and backward propagation; nonetheless, BPG codec continues to be employed during the inference phase.

Experimental findings indicate that this alternative approach (NPP+Minnen) effectively optimizes the preprocessing module and enhances rate-accuracy performance. As illustrated in Fig. 7(a), this approach yields approximately a 14.6% bitrate reduction compared to the original BPG codec at an equivalent mAP value; however, our original training strategy proves more effective, achieving a 20.3% bitrate reduction.
This superiority is attributed to our method’s use of BPG in the forward pass, aligning closely with the real inference scenario.

3) Analysis of Quantization Adaptation Strategy: Our proposed NPP module is quantization-adaptive and compatible with varying QPs for the BPG codec. We also explore an alternate methodology, namely Ours(Multiple), wherein the quantization adaptive layers are omitted, and distinct NPP modules are developed for various QPs within BPG. Experimental results indicate that while this approach yields slight enhancements in high bitrate scenarios (refer to Fig. 7(a)), the overall performance is largely equivalent to the quantization adaptive implementation. However, this method necessitates the training and maintenance of multiple NPP models, thereby increasing the storage requirements on the encoder side.

4) Analysis of the NPP module’s impact on bitrate: Under identical Quantization Parameter (QP) settings, it is observed that our NPP module significantly reduces the bitrates required for encoding pre-processed images. Using the QP settings 31, 34, 37, 41 as examples, we compare the bitrates for encoding both original and NPP pre-processed images from the COCO dataset, as detailed in Table V. For instance, at a QP of 34, the NPP module decreased the bpp from 0.2735 to 0.1760.

5) Experimental Results for Fine-tuned Downstream Networks: In the scenarios outlined above, our focus is predominantly on the image compression component. We’ve

---

**TABLE V**

| QP | 41 | 37 | 34 | 31 |
|----|----|----|----|----|
| BPG | 0.1116 | 0.1851 | 0.2735 | 0.3859 |
| Ours | 0.0851 | 0.1247 | 0.1760 | 0.2494 |
operated under the assumption that the downstream visual task model remains static, without any fine-tuning or optimization for images that have undergone lossy compression. On the other hand, in some scenarios, the downstream task models can also be fine-tuned to further boost the performance for compressed images. To verify the effectiveness of our method in this case, we fine-tuned the target detection and classification network using BPG decoded images and trained our method for these new fine-tuned backbones. The experimental results in Fig. 6(a)(b) show our method still achieves 17.31% and 21.27% bitrate savings.

6) Quantization parameter settings for different codecs: Given the variation in Quantization Parameter (QP) ranges across different codecs, we selected QP settings that yield approximately equivalent reconstruction quality, measured by PSNR. For instance, a QP of 44 for the BPG codec results in a PSNR of approximately 27.63 on the Kodak dataset. Correspondingly, a QP of 43 for VVC yields a PSNR of approximately 27.38 on the same dataset. The detailed correspondence relationship is shown in Tab. VI. The experimental results in Fig. 7(b) and (c) demonstrate that this approach can achieve a significant improvement in the “Bpp-mAP” and “Bpp-Top1” performance trade-offs.

7) Compression Performance in terms of Human Visual System: We also analyze the compression performance of our preprocessing enhanced image compression approach in terms of the human visual system. Since our compression framework is optimized for machine vision tasks, the compression performance in terms of PSNR drops, which is no surprise. However, when we use more perceptual related metrics like LPIPS [68], MS-SSIM [72] and FSIM [73], we found the gap is narrowing and our approach consumes an additional 8.5%, 8.8% and 7.3% bitrate when compared with the traditional baseline codec BPG.

8) Effectiveness of the Proxy Network: To demonstrate the approximate ability of the proposed proxy network, we compare the rate-distortion performance between the BPG codec and the corresponding proxy network. The experimental results show that the proxy network exhibits similar performance to BPG (a -1.4% BDBR gap), indicating the feasibility of substituting the BPG codec with the proxy network during the training stage.

9) Visualization of Downstream Results: We provide the visualization results in Fig. 8 and it is evident that our neural preprocessing module is beneficial for the downstream tasks. For example, the reconstructed images produced by our method in the first and second row can be correctly classified while the corresponding result from BPG is wrong. At the same time, the proposed method also consumes less bitrate compared with BPG (0.42 vs. 0.49). We have a similar observation for the object detection task in the third and fourth row. The small objects can be recognized in our compressed results with less bitrate while they’re missed in the results of BPG compressed images.
10) Analyzing the effectiveness of NPP module using Grad-CAM: We provide additional figures and employ Grad-CAM to evaluate the efficacy of the NPP module as shown in Fig. 9. In the first row, the classification model’s attention is more expansive over the animal in the filtered image, evidenced by a pronounced red region in the heatmap, which may potentially benefit the model in making correct prediction. Consequently, after compression by BPG, the ResNet50 model accurately identifies the compressed filtered image as “Koala”, whereas it misclassifies the compressed original image as “Monkey” at a comparable bpp of 1.03. A similar observation is noted in the second row. The ResNet50 classifier correctly identifies the compressed filtered image as “Mongoose” at bpp 0.30, but misclassifies the compressed original image as “Banded Gecko” at a slightly higher bpp of 0.37.

11) Running Time and Complexity: The number of parameters of our NPP module is 9.42M. For the input image with a size of 224 × 224, the inference time of our NPP module is only 4.23ms on a single RTX3090 GPU, suggesting that it brings little computational complexity to the existing pipeline.

V. CONCLUSION

In this work, based on traditional image compression algorithms, we propose a preprocessing enhanced image compression framework for downstream machine vision tasks. We introduce the neural preprocessing module to achieve a better trade-off between coding bitrate and the performance of machine vision tasks. Furthermore, we propose to use the proxy network to deal with the non-differentiable problem of the traditional codecs, which ensures that the gradients can be back-propagated to the neural preprocessing module and achieves the end-to-end optimization. Experiments show that our framework outperforms existing image codecs in several downstream tasks. More importantly, our approach shows strong generalization ability for different codecs, backbones, and tasks.

REFERENCES

[1] Z. Tian, C. Shen, H. Chen, and T. He, “FCOS: Fully convolutional one-stage object detection,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 9627–9636.
[2] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN: Towards real-time object detection with region proposal networks,” in Proc. Adv. Neural Inf. Process. Syst., vol. 28, 2015, pp. 91–99.
[3] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 2998–3006.
[4] J. Redmon and A. Farhadi, “YOLOv3: An incremental improvement,” 2018, arXiv:1804.02767.
[5] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2016, pp. 770–778.
[6] H. Zhang et al., “ResNet: Split-attention networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW), Jul. 2022, pp. 2735–2745.
[7] S. Xie, R. Girshick, P. Dollár, Z. Tu, and K. He, “Aggregated residual transformations for deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jul. 2017, pp. 1492–1500.
[8] CISCO. Cisco Annual Internet Report (2018–2023) White Paper. Accessed: May 5, 2022. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html
[9] SANDVINE. (2024) 2024 Global Internet Phenomena Report. [Online]. Available: https://www.sandvine.com/phenomena
[10] G. K. Wallace, “The JPEG still picture compression standard.” IEEE Trans. Consum. Electron., vol. 38, no. 1, pp. 18–34, Feb. 1992.
[11] F. Bellard. (2014). BPG Image Format. Accessed: Oct. 30, 2018. [Online]. Available: http://bellard.org/bpg/
[12] J. Ballé, V. Laparra, and E. P. Simoncelli, “End-to-end optimized image compression,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2017.
[13] D. Minnen, D. Ballé, S. Singh, S. H. Phong, and N. Johnston, “Variational image compression with a scale hyperprior,” in Proc. 6th Int. Conf. Learn. Represent. (ICLR), 2018, pp. 1–47.
[14] D. Minnen, J. Ballé, and G. D. Toderici, “Joint autoregressive and hierarchical priors for learned image compression,” in Proc. Adv. Neural Inf. Process. Syst., 2018, pp. 10771–10780.
[15] R. Torfason, F. Mentzer, E. Agustsson, M. Tschannen, R. Timofte, and L. Van Gool, “Towards image understanding from deep compression without decoding,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2018, pp. 1–17.
[16] M. Akbari, J. Liang, and J. Han, “DSSLIC: Deep semantic segmentation-based layered image compression,” in Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP), May 2019, pp. 2042–2046.
[17] 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. 2980–2988.
[18] S. Sun, T. He, and Z. Chen, “Semantic structured image coding framework for multiple intelligent applications,” IEEE Trans. Circuits Syst. Video Technol., vol. 31, no. 9, pp. 3631–3642, Sep. 2021.
[19] S. Wang, Z. Wang, S. Wang, and Y. Ye, “Deep image compression towards machine vision: A unified optimization framework,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 6, pp. 2979–2989, Dec. 2022.
[20] Y. Kim et al., “End-to-end learnable multi-scale feature compression for VCM,” IEEE Trans. Circuits Syst. Video Technol., vol. 34, no. 5, pp. 3156–3167, May 2024.
[21] A. Skodras, C. Christopoulos, and T. Ebrahimi, “The JPEG 2000 still image compression standard,” IEEE Signal Process. Mag., vol. 18, no. 5, pp. 36–58, Sep. 2001.
[22] 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.
[23] 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.
[24] N. Johnston et al., “Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit., Jun. 2018, pp. 4385–4393.
[25] G. Toderici et al., “Full resolution image compression with recurrent neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jul. 2017, pp. 5435–5443.
[26] Y. Zhu, Y. Yang, and T. Cohen, “Transformer-based transform coding,” in Proc. Int. Conf. Learn. Represent., 2021, pp. 1–35.
[27] 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.
[28] T. Chen, H. Liu, Z. Ma, Q. Shen, X. Cao, and Y. Wang, “End-to-end learned image compression via non-local attention optimization and improved context modeling.” IEEE Trans. Image Process., vol. 30, pp. 3179–3191, 2021.
[29] Y. Xie, K. L. Cheng, and Q. Chen, “Enhanced invertible encoding for learned image compression,” in Proc. 29th ACM Int. Conf. Multimedia, Oct. 2021, pp. 162–170.
[30] D. He, Y. Zheng, B. Sun, Y. Wang, and H. Qin, “Checkerboard context model for efficient learned image compression,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 14771–14780.
[31] D. He, Z. Yang, W. Peng, R. Ma, H. Qin, and Y. Wang, “ELIC: Efficient learned image compression with unevenly grouped space-channel contextual adaptive coding,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2022, pp. 5718–5727.
[32] S. Ma et al., “Overview of intelligent video coding: From model-based to learning-based approaches,” Vis. Intell., vol. 1, no. 1, p. 15, Aug. 2023.
[33] L. Gueguen, A. Sergeev, B. Kadlec, R. Liu, and J. Yosinski, “Faster neural networks straight from JPEG,” in Proc. Adv. Neural Inf. Process. Syst., vol. 31, 2018, pp. 1–12.
M. Ehrlich and L. Davis, “Deep residual learning in the JPEG transform domain,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2019, pp. 3484–3493.

B. Du, C. Du, and L. Yu, “MEGF-Net: Multi-exposure generation and fusion network for vehicle detection under dim light conditions,” Vis. Intell., vol. 1, no. 1, p. 28, Nov. 2023.

P. Zhang, S. Wang, M. Wang, J. Li, X. Wang, and S. Kwong, “Rethinking semantic image compression: Scalable representation with cross-modality transfer,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 8, pp. 4441–4454, Jan. 2023.

X. Li, J. Shi, and Z. Chen, “Task-driven semantic coding via reinforcement learning,” IEEE Trans. Image Process., vol. 30, pp. 6307–6320, 2021.

N. Yan, C. Gao, D. Liu, H. Li, L. Li, and F. Wu, “SSSC: Semantics-to-image scalable image coding with learned structural representations,” IEEE Trans. Image Process., vol. 30, pp. 8939–8954, 2021.

K. Fischer, F. Brand, and A. Kaup, “Boosting neural image compression for machines using latent space masking,” IEEE Trans. Circuits Syst. Video Technol., early access, Aug. 1, 2022, doi: 10.1109/TCSVT.2022.3195322.

H. Choi and I. V. Bajic, “Scalable image coding for humans and machines,” Image Process., vol. 31, pp. 2739–2754, 2022.

E. Özyiλkan, M. Ulhaq, H. Choi, and F. Raciòpe, “Learned disentangled latent representations for scalable image coding for humans and machines,” in Proc. Data Compress. Conf. (DCC), Mar. 2023, pp. 42–51.

R. Feng et al., “Image coding for machines with omnipotent feature learning,” in Proc. Eur. Conf. Comput. Vis. (ECCV), S. Avidan, G. Brostow, M. Cissé, G. M. Farinella, and T. Hassner, Eds., Cham, Switzerland: Springer, Oct. 2022, pp. 510–526.

L. Liu, Z. Hu, Z. Chen, and D. Xu, “ICMNet: Neural image compression towards both machine vision and human vision,” in Proc. 31st ACM Int. Conf. Multimedia, Oct. 2023, pp. 8047–8056.

X. Fang, Y. Duan, Q. Du, X. Tao, and F. Li, “Sketch assisted face image coding for human and machine vision: A joint training approach,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 10, pp. 6086–6100, Mar. 2023.

B. Chen, Z. Wang, B. Li, S. Wang, and Y. Ye, “Compact temporal trajectory representation for talking face video compression,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 11, pp. 7009–7023, Apr. 2023.

S. Wang et al., “Towards analysis-friendly face representation with scalable feature and texture compression,” IEEE Trans. Multimedia, vol. 24, pp. 3169–3181, 2021.

Z. Fang, L. Shen, M. Li, Z. Wang, and Y. Jin, “Prior-guided contrastive image compression for underwater machine vision,” IEEE Trans. Circuits Syst. Video Technol., vol. 33, no. 6, pp. 2950–2961, Dec. 2022.

G. Xiang, H. Jia, J. Liu, B. Cai, Y. Li, and X. Xie, “Adaptive perceptual preprocessing for video coding,” in Proc. IEEE Int. Symp. Circuits Syst. (ISCAS), May 2016, pp. 2535–2538.

E. Vidal, N. Sturmel, C. Guillemot, P. Corlay, and F.-X. Coutoud, “New adaptive filters as perceptual preprocessing for rate-quality performance optimization of video coding,” Signal Process., Image Commun., vol. 52, pp. 124–137, Mar. 2017.

C. Doutré and P. Nasipoulou, “Color correction preprocessing for multiview video coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 9, pp. 1400–1406, Sep. 2009.

X. K. Yang, W. S. Ling, Z. K. Lu, E. P. Ong, and S. S. Yao, “Just noticeable distortion model and its applications in video coding,” Signal Process., Image Commun., vol. 20, no. 7, pp. 662–680, Aug. 2005.

A. Chadha and Y. Andreopoulos, “Deep perceptual preprocessing for video coding,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 14852–14861.

O. G. Gulerayzu et al., “Sandwiched image compression: Wrapping neural networks around a standard codec,” in Proc. IEEE Int. Conf. Image Process. (ICIP), Sep. 2021, pp. 3757–3761.

H. Talebi et al., “Better compression with deep pre-editing,” IEEE Trans. Image Process., vol. 30, pp. 6673–6685, 2021.

H. Son, T. Kim, H. Lee, and S. Lee, “Enhanced standard compatible image compression framework based on auxiliary codec networks,” IEEE Trans. Image Process., vol. 31, pp. 664–677, 2021.

J. P. Kloppe, K.-C. Liu, L.-G. Chen, and S.-Y. Chien, “How to exploit the transferability of learned image compression to conventional codecs,” in Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2021, pp. 16165–16174.

O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional networks for biomedical image segmentation,” in Proc. 18th Int. Conf. Med. Image Comput. Comput. Assist. Intervent. (MICCAI), Munich, Germany: Springer, Oct. 2015, pp. 234–241.

R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in Proc. IEEE Int. Conf. Comput. Vis. (ICCV), Oct. 2017, pp. 618–626.

P. Young, A. Lai, M. Hodosh, and J. Hockenmaier, “From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions,” Trans. Assoc. Comput. Linguistics, vol. 2, pp. 67–78, Dec. 2014.

T. Lin et al., “Microsoft COCO: Common objects in context,” in Proc. Eur. Conf. Comput. Vis. (ECCV), 2014, pp. 740–755.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2009, pp. 248–255.

Z. Liu et al., “Swin transformer: Hierarchical vision transformer using shifted windows,” in Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV), Oct. 2021, pp. 10012–10022.

A. Dosovitskiy et al., “An image is worth 16 × 16 words: Transformers for image recognition at scale,” in Proc. Int. Conf. Learn. Represent. (ICLR), 2021, pp. 1–21.

L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking atrous convolution for semantic image segmentation,” 2017, arXiv:1706.05587.

A. Toshev and C. Szegedy, “DeepPose: Human pose estimation via deep neural networks,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., Jun. 2014, pp. 1563–1566.

(2024). Libjpeg-Turbo. Accessed: Jul. 14, 2024, [Online]. Available: https://libjpeg-turbo.org.

A. Wieckowski et al., “Venc: An open and optimized VVC encoder implementation,” in Proc. IEEE Int. Conf. Multimedia Expo. Workshops (ICMWC), Jul. 2021, pp. 1–2.

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. 586–593.

A. Paszke et al., “PyTorch: An imperative style, high-performance deep learning library,” in Proc. Adv. Neural Inf. Process. Syst., vol. 32, 2019, pp. 1–12.

D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” 2014, arXiv:1412.6980.

G. Bjontegaard, Calculation of Average PSNR, document VCEG-M33, 2001.

Z. Wang, E. P. Simoncelli, and A. C. Bovik, “Multiscale structural similarity for image quality assessment,” in Proc. 37th Asilomar Conf. Signals, Syst., Comput., vol. 2, Nov. 2003, pp. 1398–1402.

L. Zhang, L. Zhang, X. Mou, and D. Zhang, “FSIM: A feature similarity index for image quality assessment,” IEEE Trans. Image Process., vol. 20, no. 8, pp. 2378–2386, Aug. 2011.
Xingtong Ge (Student Member, IEEE) received the B.S. degree in computer science and technology and the M.S. degree in software engineering from Beijing Institute of Technology, Beijing, China, in 2022 and 2024, respectively. He is currently an Algorithm Researcher with SenseTime Technology. His research interests include learned video compression and generative AI.

Tianxiong Zhong received the B.S. degree in computer science and technology from Beijing Institute of Technology, Beijing, China, in 2023, where he is currently pursuing the M.S. degree in computer science and technology. His research interests include video database, video compression, and deep learning.

Qiang Hu received the B.E. degree from the University of Electronic Science and Technology of China in 2013 and the Ph.D. degree in information and communication engineering from Shanghai Jiao Tong University in 2019. He was a Post-Doctoral Researcher with ShanghaiTech University from 2019 to 2021. He was an Assistant Researcher with ShanghaiTech University from 2021 to 2023. He is currently an Assistant Researcher with the Cooperative Medianet Innovation Center, Shanghai Jiao Tong University. His research interests include video compression, neural rendering, and free-viewpoint video streaming.

Jing Geng received the B.E. degree in software engineering from the University of Electronic Science and Technology of China in 2009 and the Ph.D. degree in photogrammetry and remote sensing from Wuhan University in 2015. She is currently an Assistant Professor with the Department of Computer Science, Beijing Institute of Technology, China. Her research interests include computer vision, remote sensing, data mining, big data, knowledge service in GIS, and e-government.