
Preprocessing Enhanced Image Compression for Machine Vision

Guo Lu

Beijing Institute of Technology, China
sdluguo@gmail.com

Xingtong Ge

Beijing Institute of Technology, China
xingtong.ge@gmail.com

Tianxiong Zhong

Beijing Institute of Technology, China
inkosizhong@gmail.com

Jing Geng

Beijing Institute of Technology, China
janegeng@bit.edu.cn

Qiang Hu

ShanghaiTech University, China
huqiang@shanghaitech.edu.cn

Abstract

Recently, more and more images are compressed and sent to the back-end devices for the 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 two 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.

1 Introduction

Due to 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 [11, 19, 24, 28, 29, 35, 43, 47]. Therefore, more and more images are captured at the front-end devices (*e.g.*, cameras) and sent to the back-end (*e.g.*, cloud servers) for machine analysis. According to the report from Cisco [1], the percentage of the connections from this machine-to-machine scenario (*e.g.*, video surveillance) will be up to 50% in the future. Therefore, how to reduce the transmission bitrate while maintaining the performance for the downstream machine vision tasks has become a challenge for the image compression field.

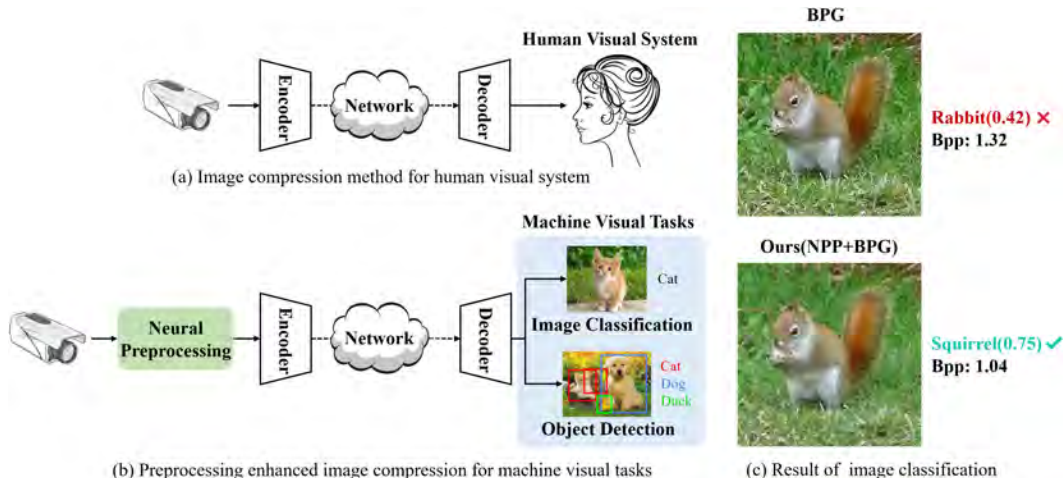


Figure 1: (a) Image compression method for human visual system. (b) Our proposed preprocessing enhanced image compression for machine vision tasks. (c) Image classification results for the image from the BPG codec and ours(NPP+BPG).

Unfortunately, although several traditional image compression standards, such as JPEG [39] and BPG [3], 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 [8, 9, 26] start to gain a lot of attention. Several approaches [7, 20, 37] also have tried to jointly optimize the learned compression methods with the downstream analysis tasks. However, the computational complexity for the learned image compression 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 [7, 20, 37] 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 vision 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 two representative machine vision tasks (object detection and image classification) with different downstream backbone networks. Experiments show that compared with the existing traditional codec like BPG [3], the proposed approach can save about 20% bitrate for object detection and image classification tasks with the same accuracy.

The main contributions of our work are summarised as follows,

- Building upon the traditional codec, we propose a neural preprocessing module to generate the filtered images, which the traditional codecs can effectively compress 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 the other tasks.

2 Related Works

2.1 Image Compression

Many traditional image compression algorithms [3, 32, 39] 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 [8, 9, 13, 14, 21, 26, 36, 44, 49] have become popular. The mainstream methods [8, 9, 26, 49] 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.* [8] proposed to use a convolutional neural network (CNN) to learn non-linear transformation and additionally introduce a hyper-prior network to model the probability distribution of the latent representations [9]. Latest works [13, 14, 44, 49] also propose to use more powerful transform networks, such as residual blocks [14], nonlocal layers [13], invertible layers [44] and transformer [49]. Although these approaches have achieved better compression performance, they are computationally expensive. More importantly, there is no coding standard for these learned compression methods, which cannot be massively deployed to practical applications.

2.2 Compression for Machine Vision

Most existing image compression methods [3, 8, 9, 39] aim to reduce reconstruction distortion in terms of the human visual system and are optimized based on the pixel field metrics such as PSNR or MS-SSIM [41]. With the development of deep learning, some studies [7, 17, 20, 23, 37, 40, 45] also focus on the joint optimization of the image compression and the downstream machine vision tasks. For example, Torfason *et al.* [37] proposed to directly perform image understanding tasks, such as classification and segmentation, on the compressed representations produced by the learning-based image compression methods. Fischer *et al.* [17] introduced the feature loss to optimize the image compression network and achieved a better trade-off between bitrate cost and analysis accuracy. Wang *et al.* [40] proposed a scalable coding based compression for both face reconstruction and face analysis, where the base layer encodes the valuable features for analysis. In contrast, the enhancement layers encode the texture information for reconstruction.

However, 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.

2.3 Preprocessing

In the past decades, several methods [16, 38, 42] 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 [46] and try to improve the perceptual quality of reconstructed frames. For example, Xiang *et al.* [42] proposed adaptive perceptual preprocessing by removing the information that is not perceptible to the human visual system. Vidala *et al.* [38] combined several adaptive filters to denoise the image for bitrate saving.

In recent years, several learning-based preprocessing methods have been proposed [12, 18, 33, 34]. Chadha *et al.* [12] proposed a rate-aware perceptual preprocessing module for video coding. Onur *et al.* [18] proposed neural network based preprocessing and postprocessing modules to improve the compression performance of the traditional codecs. Talebi *et al.* [34] designed a pre-editing neural network on the JPEG method to improve the visual quality of reconstructed images. In contrast, we propose using the neural network based preprocessing method to improve the compression performance in machine vision instead of the human visual system.

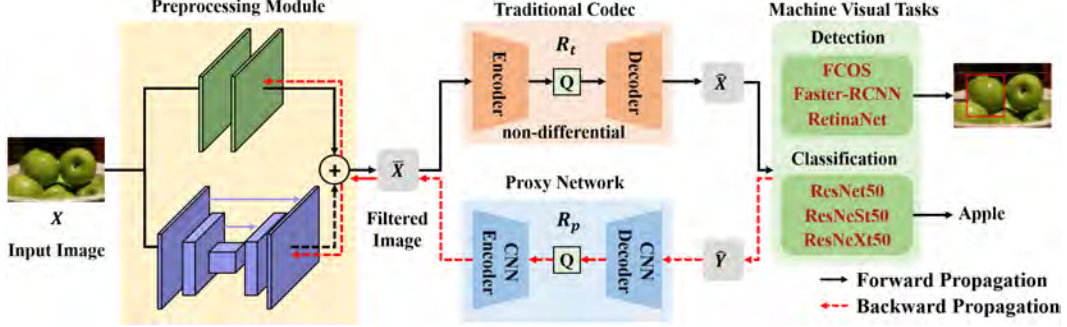


Figure 2: Overview of our preprocessing enhanced image compression for machine vision.

3 Proposed Method

3.1 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 \hat{X} to the neural preprocessing module (NPP) for non-linear transform and generate the filtered image \hat{X} , which is expected to maintain the critical semantic information. Then, \hat{X} is encoded and reconstructed by a traditional codec, like BPG [3]. Finally, the decoded \hat{X} is input to machine analysis networks, such as FCOS [35].

Since the traditional codecs maybe 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 preprocessing module (see Section 3.3 for more details). Here, we use BPG [3] as the traditional codec in our implementation.

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

$$\mathcal{L} = R_t + \lambda \mathcal{D}_m + \beta \mathcal{D}_{pre} \quad (1)$$

where \mathcal{D}_m and R_t represents the loss of the downstream machine vision task based on reconstructed image \hat{X} and coding bitrate from traditional codec, respectively. λ 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 \hat{X} , which is denoted as \mathcal{D}_{pre} . β is the constant weight parameter.

3.2 Neural Preprocessing

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 1×1 convolutional layers to learn non-linear pixel-level transform, and the second branch uses a U-Net [30] style network to extract the semantic information. The outputs of two branches are added together as the final filtered image \hat{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), therefore, the neural preprocessing module is required to generate the optimal filtered image \hat{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 generated scale vector s , *i.e.*, $f' = f \odot s$. Based on this strategy, the intermediate features in the

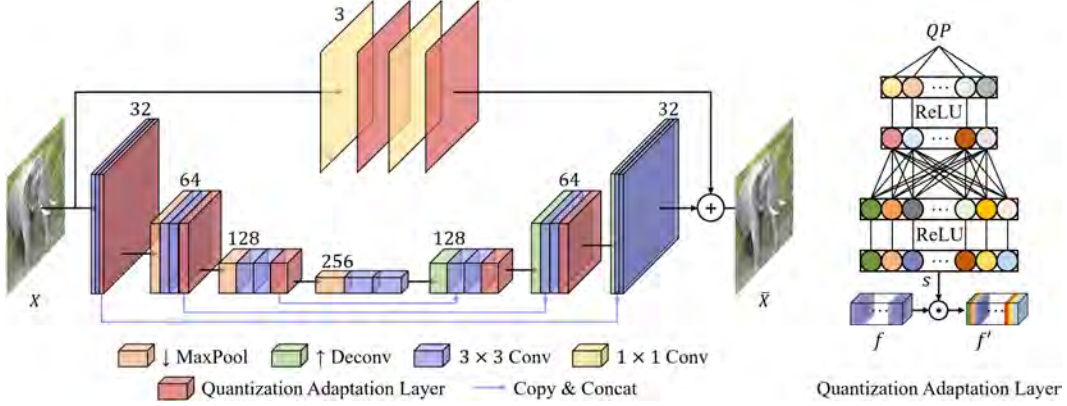


Figure 3: The implementation of our neural preprocessing module. The numbers represent the numbers of output channel for different operations.

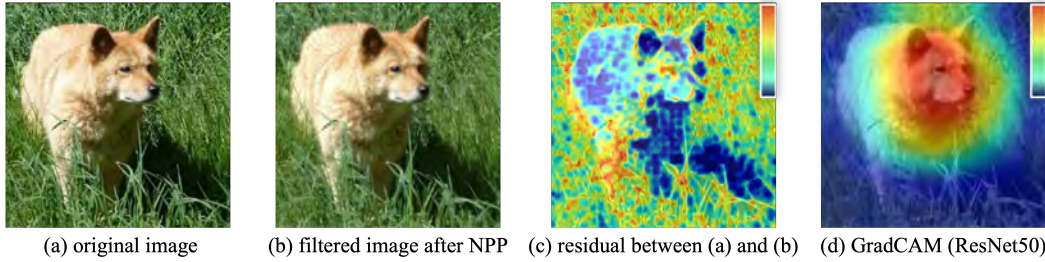


Figure 4: Visualization results of the neural preprocessing module. The color in (c) and (d) represents the values of residual and localized class-discriminative regions form GradCAM [31], where larger values are denoted by red color.

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 [3] ($QP = 37$) 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 [31], the classification network [19] focuses on the foreground *Dingo* in the image as shown in Fig. 4(d). 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.

3.3 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 [26] as our proxy network.

To make sure that the proxy network can well approximate the traditional codec, the reconstruction quality of BPG and Minnen’s approach should be similar. Since the learned image compression approach [26] is optimized based on R-D distortion loss $R + \lambda_p D$ and the quality of the reconstructed image depends on the hyper-parameter λ_p . Therefore, we first choose the learned image compression model with a suitable λ_p parameter to approximate BPG codec and then finetune the proxy network [26] in the following way,

$$\mathcal{L}_p = R_p + \lambda_p D = R_p + \lambda_p d(\hat{X}, \hat{Y}) \quad (2)$$

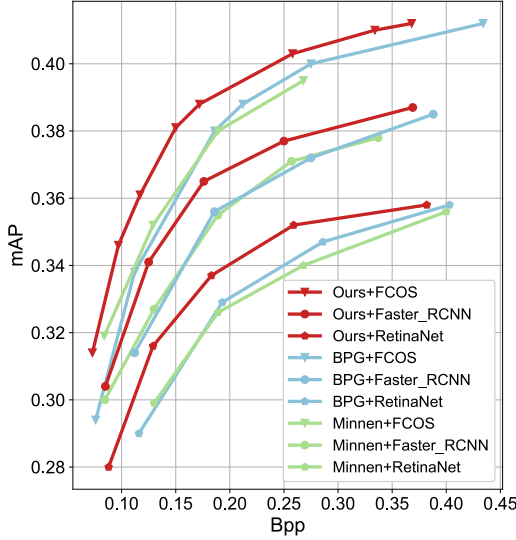


Figure 5: Rate-accuracy(mAP) curves from different compression methods for the object detection tasks on the COCO dataset.

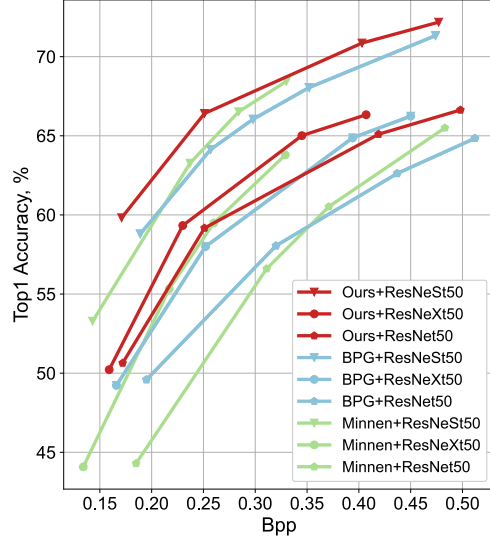


Figure 6: Rate-accuracy(top-1 accuracy) curves from different compression methods for the image classification task on the ImageNet dataset.

where $d(\hat{X}, \hat{Y})$ denotes the distortion between the reconstructed image \hat{X} from BPG 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 can obtain an optimized proxy codec to mimic BPG codec.

For the forward stage in the training procedure, we first use the BPG codec to generate the reconstructed image \hat{X} , which is input to the analysis models and used to calculate the machine perception loss \mathcal{D}_m . Then we can obtain the loss function in Eq. 1 based on the bitrate R_t from BPG codec. Finally, as shown in Fig. 2, the gradients are calculated and propagated through the paired proxy network to update our preprocessing module in the backward stage. To end-to-end optimize the preprocessing module based on Eq. 1, the values of reconstructed image \hat{Y} and the corresponding bitrate R_p from the proxy network are assigned to \hat{X} and R_t from the BPG codec in the back-propagation, respectively. More implementation details are given in supplementary material.

4 Experiments

4.1 Experimental Setup

Datasets and Backbone Models. For the object detection task, we train our framework on the COCO dataset [25]. We use the COCO trainval135k split (union of 80k images from train and a random 35k subset of images from the 40k image val split) as our training dataset [2]. The mean average precise (mAP) results are reported by evaluating the proposed framework on the minival split (the remaining 5k images from val) with different compression settings. In our experiments, three popular object detection baselines FCOS [35], Faster-RCNN [29] and RetinaNet [24] are used for evaluation [6].

For the image classification task, we use the ImageNet dataset [15], which contains 1.28M training images and 50K validation images from 1,000 classes [4]. The top-1 accuracy is reported in our experiments. To demonstrate the effectiveness of our approach, we use ResNet [19], ResNeSt [47] and ResNeXt [43] for the performance evaluation [5].

We further evaluate the compression performance in terms of the human visual system by using the perceptual metrics LPIPS [48] on the Kodak dataset [36]. We use the bits-per-pixel(bpp) in all experiments to measure the coding cost during the compression procedure.

Implementation Details. Our whole framework is implemented based on PyTorch [27] with CUDA support and trained on one RTX 3090 GPU card. We use BPG [3] as the traditional codec with different QP values ($QP = 28, 31, 34, 37, 41$) in our experiments and the corresponding λ s in Eq. 1 are set as 0.5, 1, 2, 4, 8, empirically. The trade-off parameter β is set as 0.5. The weights of the downstream networks such as FCOS [35] are fixed during the whole training process.

Table 1: BDBR (%) results between our approach and anchor methods for different backbone networks in object detection task.

Anchor	Backbones		
	FCOS	Faster-RCNN	RetinaNet
BPG	-20.3	-19.5	-18.8
Minnen	-19.5	-22.6	-21.7

Table 2: BDBR (%) results between our approach and anchor methods for different backbone networks in image classification task.

Anchor	Backbones		
	ResNet50	ResNeSt50	ResNeXt50
BPG	-22.0	-16.4	-15.7
Minnen	-24.1	-12.8	-14.9

The whole training process has the following stages. First, based on the finetuning procedure in Section 3.3, we can obtain several proxy networks that mimic 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 [22] 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 reduced to $1e-5$ after 320k, 80k and 60k steps when the loss becomes stable. The whole training process takes about five days.

4.2 Experimental Results

We compare our preprocessing enhanced image compression method with the existing traditional codec BPG [3] and neural network based compression model [26]. In addition, BD-Rate [10] (BDBR) is used to measure the percentage of saved bitrate with the same accuracy. We use FCOS [35] and ResNet50 [19] as the default backbone networks for object detection and image classification and train the corresponding NPP modules, respectively.

Object Detection Fig. 5 shows the rate-accuracy curve from different backbone networks and compression approaches on the COCO dataset [25]. It is obvious that our preprocessing enhanced image compression method shows a much better rate-accuracy trade-off than the baseline approaches on 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 evaluating on FCOS, respectively.

To further verify the generalization ability of our proposed neural preprocessing module, we perform new experiments by directly applying the NPP optimized for FCOS to other backbone networks like RetinaNet [24]. Experiments in Fig. 5 show that our compression approach can still outperform the baseline methods and reduces 19.5% and 18.8% bitrate when compared with BPG on the downstream Faster-RCNN [29] and RetinaNet [24] models, respectively. These results demonstrate that the proposed solution can be used for downstream networks with different architectures, which is beneficial in practical applications. The corresponding BD-Rate results are provided in Table 1.

Image Classification We also compare our method with the traditional and learning based codecs on the image classification task. Fig. 6 shows the rate-accuracy (top-1) curves from different compression methods on the ImageNet dataset [15]. It is noted that our approach still achieves better rate-accuracy performance and saves more than 22.0% bitrate when compared with traditional codec BPG [3] by evaluating on the ResNet50 [19] model.

We further perform new experiments on other mainstream image classification networks by directly using the NPP module optimized for ResNet50. Compared with BPG codec, our framework has 16.4% and 15.7% bitrate reduction when the downstream classification networks are ResNeSt [47] and ResNeXt [43], respectively. The corresponding BD-Rate results are provided in Table 2.

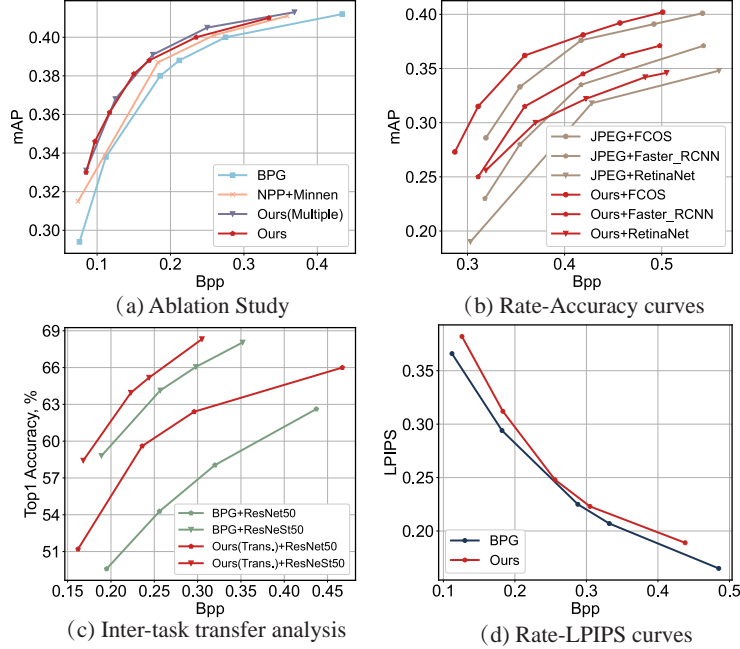


Figure 7: (a) Ablation study. (b) Rate-accuracy curves from our NPP enhanced JPEG codec and native JPEG codecs. (c) Inter-task transfer of our NPP from object detection to image classification. (d) Rate-LPIPS curves from ours approach and BPG codec.

4.3 Ablation Study and Model Analysis

Analysis of End-to-end Optimization. In the proposed approach, we use BPG to generate reconstructed images in the forward propagation, while the gradients of the proxy network are used in the backward propagation to update the parameters of the preprocessing module. Here, we also provide the result of directly using the proxy network in both forward and backward propagation, denoted as NPP+Proxy; however, we still use the BPG codec in the inference stage.

Experimental results show that this alternative solution (NPP+Minnen) can also effectively optimize the preprocessing module and improve the rate-accuracy performance. As shown in Fig. 7(a), compared with the original BPG codec, it saves about 14.6% bitrate at the same mAP value; however, our proposed training strategy is more effective and achieves 20.3% bitrate saving. The reason is that our approach uses BPG to generate the reconstructed images in the forward pass, which is consistent with the actual inference stage.

Analysis of Quantization Adaptation Strategy. Our proposed NPP module is quantization adaptive and can be used for BPG codec with different QPs . Here, we provide another alternative solution, *i.e.*, Ours(Multiple), where the quantization adaptive layers are removed and we train different NPP modules for different QPs in BPG. Experiments show that it has marginal improvements compared to our quantization adaptive implementation (See Fig. 7(a)). However, it needs to train and store multiple NPP models, which brings more storage burden to the encoder side.

Preprocessing Module for JPEG Compression. We also provide more experimental results for JPEG compression. Here, we apply the NPP module optimized for the BPG [3] codec to the JPEG [39] without any finetuning. Experimental results in Fig. 7(b) show that our proposed preprocessing enhanced JPEG compression achieves more than 8.5% bitrate savings than the original JPEG codec when evaluating on the FCOS backbone networks.

Inter-task Transfer of the Preprocessing Model. To demonstrate the generalization ability of the preprocessing module, we apply the preprocessing model optimized for the object detection task to the image classification task. The results in Fig. 7(c) show that our transfer method (Ours(Trans.)) is still useful and achieves more than 10% bitrate reduction. We have the same observation when we

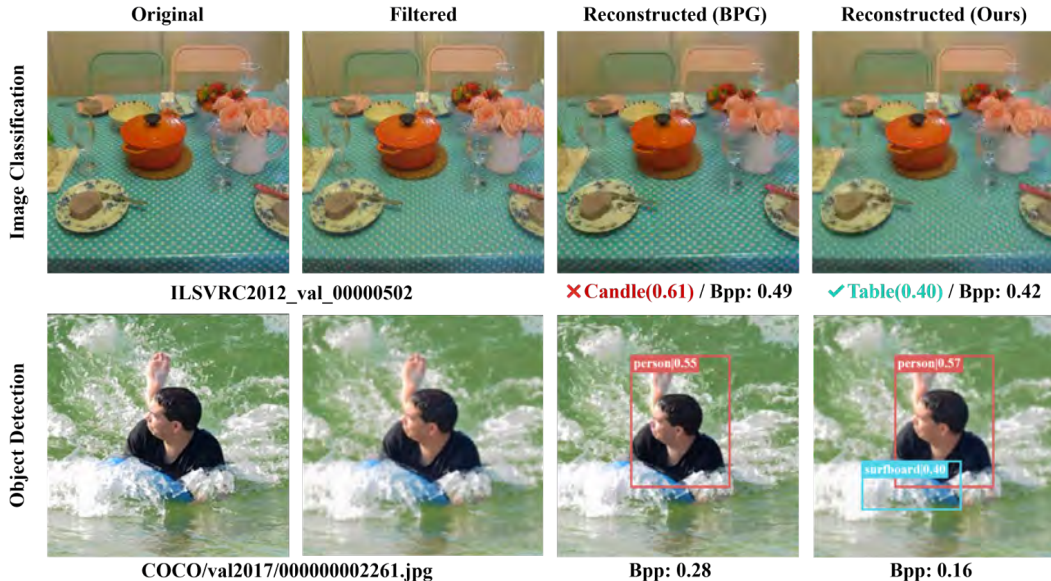


Figure 8: Visualization results of the downstream tasks.

apply the preprocessing model optimized for the image classification task to the object detection task. And more detailed results will be provided in the supplementary material.

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 or MS-SSIM drops, which is no surprise. However, when we use more perceptual related metrics like LPIPS [48], we found the gap is narrowing and our approach consumes an additional 8% bitrate when compared with traditional baseline codec BPG in Fig. 7(d).

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 row can be correctly classified while the corresponding result from BPG is wrong. At the same time, the proposed also consumes fewer bitrate compared with BPG (0.42 vs. 0.49). We have a similar observation for the object detection task in the second row. The small object can be recognized in our compressed results with fewer bitrates while it is missed for the BPG compressed image.

Running Time and Complexity. The number of parameters of our preprocessing module is 9.42M. For the input image with the size of 224×224 , the inference time of our neural preprocessing module is only 5.17ms, which means it brings little computational complexity to the existing pipeline.

5 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 traditional codec, which ensures that the gradients can be back-propagated to the neural preprocessing module. Experiments show that our framework outperforms existing image codecs in object detection and classification tasks. More importantly, our approach shows strong generalization ability for different codecs, backbones, and even for different tasks.

Limitations. We only select two representative machine vision tasks (*i.e.*, image classification and object detection) and two codecs (BPG and JPEG) to demonstrate the effectiveness of our approach. Our approach may not apply to other tasks or codecs in practical applications. However, evaluating

more machine vision tasks or codecs will significantly increase the workload, which is beyond the scope of this paper. We will try to include more extensive experiments in the future.

Potential Negative Societal Impacts. Our approach is proposed to improve the performance of the downstream tasks. At the same time, the proposed solution can also be used to attack or mislead the downstream tasks by modifying the loss function.

References

- [1] Cisco annual internet report (2018–2023). <https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html>. Accessed: 2022-5-10.
- [2] Coco dataset, license: Cc-by-4.0. <https://cocodataset.org/#download>. Accessed: 2021-05-17.
- [3] F. bellard, bpg image format. <http://bellard.org/bpg/>. Accessed: 2018-10-30.
- [4] Imagenet dataset. <https://www.image-net.org/download.php>. Accessed: 2021-05-17.
- [5] Mmclassification, license: Apache 2.0. <https://github.com/open-mmlab/mclassification>. Accessed: 2021-05-17.
- [6] Mmdetection, license: Apache 2.0. <https://github.com/open-mmlab/mmdetection>. Accessed: 2021-05-17.
- [7] Mohammad Akbari, Jie Liang, and Jingning Han. Dsslic: Deep semantic segmentation-based layered image compression. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2042–2046. IEEE, 2019.
- [8] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. End-to-end optimized image compression. In *5th International Conference on Learning Representations, ICLR*, 2017.
- [9] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *6th International Conference on Learning Representations, ICLR*, 2018.
- [10] Gisle Bjontegaard. Calculation of average psnr differences between rd-curves. *VCEG-M33*, 2001.
- [11] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, 2020.
- [12] Aaron Chadha and Yiannis Andreopoulos. Deep perceptual preprocessing for video coding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14852–14861, June 2021.
- [13] Tong Chen, Haojie Liu, Zhan Ma, Qiu Shen, Xun Cao, and Yao Wang. End-to-end learnt image compression via non-local attention optimization and improved context modeling. *IEEE Transactions on Image Processing*, 30:3179–3191, 2021.
- [14] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7939–7948, 2020.
- [15] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on Computer Vision and Pattern Recognition*, pages 248–255. Ieee, 2009.
- [16] Colin Doutre and Panos Nasiopoulos. Color correction preprocessing for multiview video coding. *IEEE Transactions on Circuits and Systems for Video Technology*, 19(9):1400–1406, 2009.
- [17] Kristian Fischer, Fabian Brand, and André Kaup. Boosting neural image compression for machines using latent space masking. *arXiv preprint arXiv:2112.08168*, 2021.
- [18] Onur G Guleryuz, Philip A Chou, Hugues Hoppe, Danhang Tang, Ruofei Du, Philip Davidson, and Sean Fanello. Sandwiched image compression: Wrapping neural networks around a standard codec. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3757–3761. IEEE, 2021.
- [19] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.
- [20] Yueyu Hu, Shuai Yang, Wenhan Yang, Ling-Yu Duan, and Jiaying Liu. Towards coding for human and machine vision: A scalable image coding approach. In *2020 IEEE International Conference on Multimedia and Expo (ICME)*, pages 1–6. IEEE, 2020.

- [21] Nick Johnston, Damien Vincent, David Minnen, Michele Covell, Saurabh Singh, Troy Chinen, Sung Jin Hwang, Joel Shor, and George Toderici. Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In *CVPR*, June 2018.
- [22] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
- [23] Xin Li, Jun Shi, and Zhibo Chen. Task-driven semantic coding via reinforcement learning. *IEEE Transactions on Image Processing*, 30:6307–6320, 2021.
- [24] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2980–2988, 2017.
- [25] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.
- [26] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, pages 10771–10780, 2018.
- [27] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32, 2019.
- [28] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018.
- [29] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015.
- [30] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-assisted Intervention*, pages 234–241. Springer, 2015.
- [31] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.
- [32] Athanassios Skodras, Charilaos Christopoulos, and Touradj Ebrahimi. The jpeg 2000 still image compression standard. *IEEE Signal Processing Magazine*, 18(5):36–58, 2001.
- [33] Hanbin Son, Taeoh Kim, Hyeongmin Lee, and Sangyoun Lee. Enhanced standard compatible image compression framework based on auxiliary codec networks. *IEEE Transactions on Image Processing*, 31:664–677, 2021.
- [34] Hossein Talebi, Damien Kelly, Xiyang Luo, Ignacio Garcia Dorado, Feng Yang, Peyman Milanfar, and Michael Elad. Better compression with deep pre-editing. *IEEE Transactions on Image Processing*, 30:6673–6685, 2021.
- [35] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9627–9636, 2019.
- [36] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. Full resolution image compression with recurrent neural networks. In *CVPR*, pages 5435–5443, 2017.
- [37] Robert Torfason, Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Towards image understanding from deep compression without decoding. *ICLR*, 2018.
- [38] Eloïse Vidal, Nicolas Sturmel, Christine Guillemot, Patrick Corlay, and François-Xavier Coudoux. New adaptive filters as perceptual preprocessing for rate-quality performance optimization of video coding. *Signal Processing: Image Communication*, 52:124–137, 2017.
- [39] Gregory K Wallace. The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics*, 38(1):xviii–xxxiv, 1992.
- [40] Shurun Wang, Shiqi Wang, Wenhan Yang, Xinfeng Zhang, Shanshe Wang, Siwei Ma, and Wen Gao. Towards analysis-friendly face representation with scalable feature and texture compression. *IEEE Transactions on Multimedia*, 2021.
- [41] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003.

- [42] Guoqing Xiang, Huizhu Jia, Jie Liu, Binbin Cai, Yuan Li, and Xiaodong Xie. Adaptive perceptual preprocessing for video coding. In *2016 IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2535–2538. IEEE, 2016.
- [43] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1492–1500, 2017.
- [44] Yueqi Xie, Ka Leong Cheng, and Qifeng Chen. Enhanced invertible encoding for learned image compression. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 162–170, 2021.
- [45] Ning Yan, Changsheng Gao, Dong Liu, Houqiang Li, Li Li, and Feng Wu. Ssic: Semantics-to-signal scalable image coding with learned structural representations. *IEEE Transactions on Image Processing*, 30:8939–8954, 2021.
- [46] Xiaokang Yang, Weisi Lin, Zhongkang Lu, Ee Ping Ong, and Susu Yao. Just noticeable distortion model and its applications in video coding. *Signal processing: Image communication*, 20(7):662–680, 2005.
- [47] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Haibin Lin, Zhi Zhang, Yue Sun, Tong He, Jonas Mueller, R Manmatha, et al. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020.
- [48] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 586–595, 2018.
- [49] Yin hao Zhu, Yang Yang, and Taco Cohen. Transformer-based transform coding. In *International Conference on Learning Representations*, 2021.