Redundancy Removal Module for Reducing the Bitrates of Image Coding for Machines

Zhongpeng Zhang and Ying Liu
Department of Computer Science and Engineering
Santa Clara University
Santa Clara, CA 95053, USA
{zzhang13, yliu15}@scu.edu

***Abstract*—With the popularity of the Internet of Things (IoT) and surveillance, the amount of image and video information collected by internet has exploded. On the one hand, transmitting the massive information requires plenty of bandwidth. On the other hand, processing these data requires tremendous manpower. In order to solve these two problems at the same time, image coding for machines (ICM) came into being. At present, most ICM technologies combine traditional codecs such as BPG or machine learning codecs such as the hyperprior codec and the coarse-to-fine codec with task networks such as image classification and semantic segmentation. This process requires the complete restoration of the image, which greatly increases the bitrates. Moreover, the restored images are for human eyes and contain a large amount of redundant information, which is not required by the task network. In order to solve this problem, side information driven image coding for machines (SIIC) was proposed, which only needs to input hyperprior information to the image classification network, which greatly reduces the scale of data transmission. Now, we propose a redundancy removal module that can further reduce the usage of bitrates for SIIC. Through this module, the new codec uses 3% to 4% less bitrates than the original SIIC when the bitrates are under 0.06 bpp, and can save up to 10% of bitrates when the bitrates are over 0.1 bpp.**

Keywords—bitrate, image classification, image coding for machines, redundancy removal, transpose convolution

I. Introduction

As stated in [1], with the rapid development of the Internet of Things, humans are becoming less and less important in processing images or visual tasks such as image recognition. On the one hand, this is because these images are processed more by machine learning task networks such as ResNet-50 [2] and Yolov3 [3], which save manpower and are faster and more accurate than human labor. On the other hand, a large amount of image data are captured by terminals such as surveillance equipment, drones and autonomous vehicles, and then transmitted on the network, which results in massive bandwidth resources being occupied. Therefore, these images need to be compressed before transmission to save bandwidth resources. Image coding for machines was born to solve the problem of image transmission efficiency and visual tasks at the same time. It generally contains two parts. The first part is the codec responsible for image compression and transmission, which can be traditional image compression tools such as BPG [4]. However, with the development of machine learning, more learned codecs based on the convolutional neural network (CNN) are developed, such as Coarse-to-fine codec [5] and Hyperprior codec [6]. ICM not only compresses images to save bandwidth, but also uses task networks to handle visual recognition tasks.

Most of the current ICM frameworks are obtained by splicing codec and task network. The image needs to be restored first and then processed through the task network. The quality of the restored image is judged by the metrics of human vision. These metrics include PSNR and MS-SSIM. However, it should be noted that the information required by human vision and the information required by the subsequent task network may be far apart. This will cause the reconstructed images to contain redundancy and also lack key information for the subsequent task.

SIIC [7] only transmits hyperprior information to save bitrates. It does not generate intermediate images, but directly inputs the side information to the image classification network ViT [8]. Its metric is the top-1 classification accuracy, so the transmitted information does not carry too much redundant information irrelevant to the task.

In this article, we proposed a redundancy removal module to further reduce the usage of bitrates in SIIC. In addition, this module is fast to train, requiring only 4 epochs of additional training, and is easy to insert into the pipeline. Ultimately, it enables SIIC to save 3% to 10% of bandwidth without degrading SIIC's performance in image classification.

The following is the structure of this article: Section II introduces related ICM frameworks, Section III describes our proposed method in details, Section IV presents the performance of redundancy removal module through experiments, and Section V concludes the paper.

II. Related Works

As mentioned in the Section I, most ICM pipelines are directly spliced together the codec and task network, such as J-FT T-FT [9], transformed images [10], compressed representation [11], SPIC-Q [12], Post-SA [13], RNN-C + ResNet-50 [14]. Taking J-FT T-FT as an example, the image is first compressed by the learned codec, restored to $\hat{x}$, and then assigned to different tasks such as image classification and object detection.

This work was supported in part by the National Science Foundation under Grant ECCS-2138635 and in part by the NVIDIA Academic Hardware Grant.

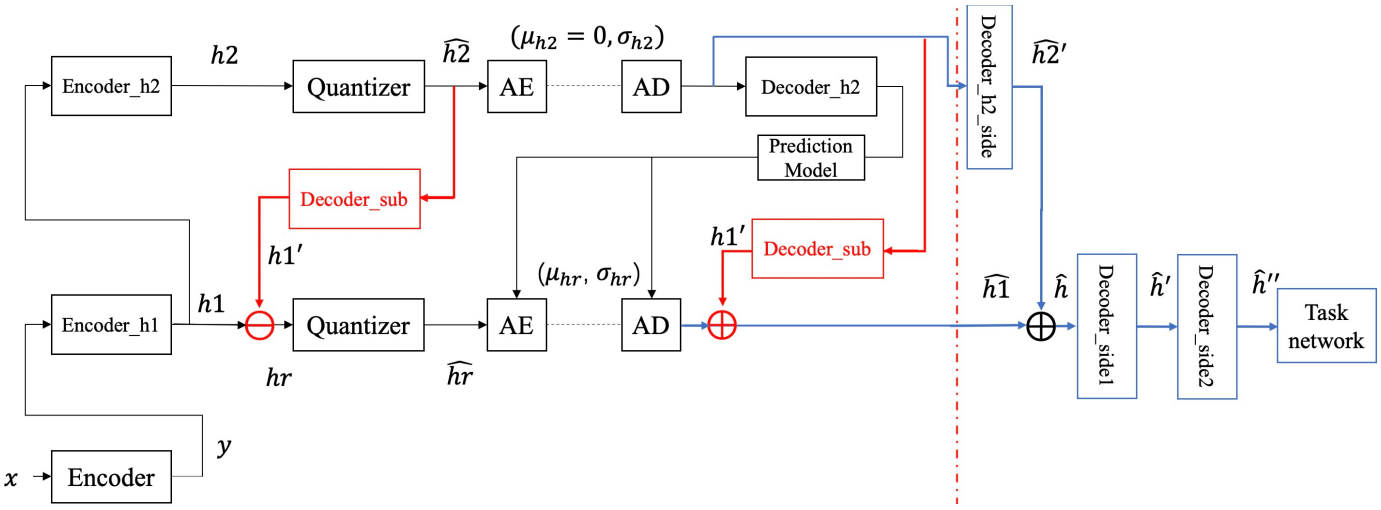

Fig. 1. The structure of SIIC with the proposed redundancy removal module.

The other type is scaled feature pipeline, such as HMI-IC [15]. It contains a base layer and an enhancement layer. The base layer transmits a small part of the image information and performs relatively simple tasks such as image classification and thumbnail generation. The enhancement layer transmits the remaining information of the image, which is added to the information of the above-mentioned base layer to complete more complex tasks such as image reconstruction.

The original SIIC [7] belongs to the first category, but it does not need to restore images, so it saves much bandwidth. It uses side information, also known as hyperpriors, which can effectively reduce the redundancy.

The redundancy removal module we proposed can act on both encoder and decoder ends of the codec in SIIC, thereby further reducing the usage of bitrates.

III. Proposed Method

Based on the original SIIC [7], we added redundancy removal modules named Decoder_sub on both the encoder and decoder sides. The overall process is shown in Fig. 1. Image x passes through the Encoder to produce latent y, y enters Encoder_h1 to produce the first layer hyperprior $h1$, and $h1$ enters Encoder_h2 to produce the second layer hyperprior $h2$. $h2$ is quantized and become $\widehat{h2}$. $\widehat{h2}$ enters redundancy removal module Decoder_sub to produce $h1'$, then we subtract $h1'$ from $h1$ to obtain residue hr. According to [5], $\widehat{h2}$ and $\widehat{hr}$ conform to the normal distribution. In particular, the parameters (μ_{hr}, σ_{hr}) of $\widehat{hr}$ are calculated by $\widehat{h2}$ through Decoder_h2 and the prediction model. After $\widehat{h2}$ and $\widehat{hr}$ arrive at the decoder end, $\widehat{h2}$ passes through Decoder_sub again to generate $h1'$, then $\widehat{hr}$ and $h1'$ are added to obtain $\widehat{h1}$. Finally, we put $\widehat{h2}$ and $\widehat{h1}$ into the decoders and the task network ViT to produce the final classification results. The formulas needed are as follows:

$$y = Encoder\ (x), \tag{1}$$

$$h1 = Encoder\_h1\ (y), \tag{2}$$

$$h2 = Encoder\_h2\ (h1), \tag{3}$$

$$h1' = Decoder\_sub\ (\widehat{h2}), \tag{4}$$

$$hr = h1 - h1', \tag{5}$$

$$\widehat{h1} = \widehat{hr} + h1', \tag{6}$$

$$\widehat{h2}' = Decoder\_h2\_side\ (\widehat{h2}), \tag{7}$$

$$\hat{h} = \widehat{h2}' + \widehat{h1}, \tag{8}$$

$$\hat{h}' = Decoder\_side1\ (\hat{h}), \tag{9}$$

$$\hat{h}'' = Decoder\_side2\ (\hat{h}'). \tag{10}$$

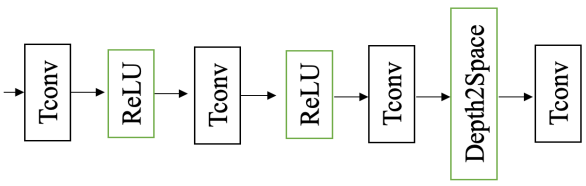

Fig. 2. The specific structure of Decoder_sub, where Depth2Space [5] doubles the tensor's height and width and downsizes the channel by a factor of 4.

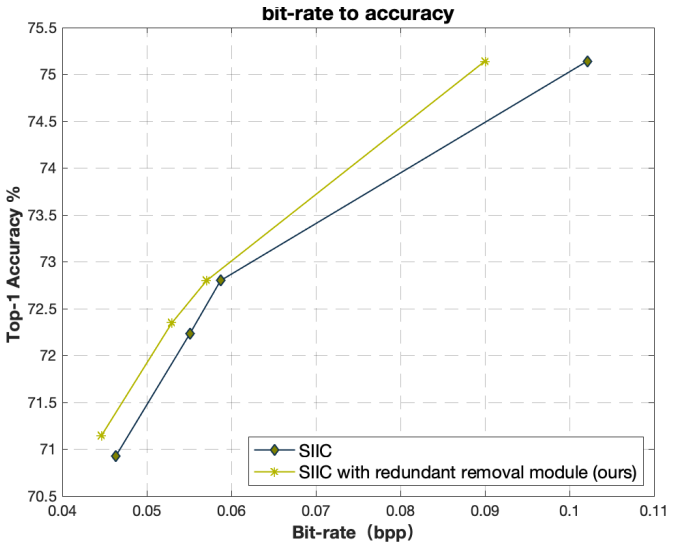

Fig. 3. Comparison of the top-1 accuracy of SIIC with redundancy removal module (ours) and original SIIC [7] on the ImageNet1K [16] verification set.

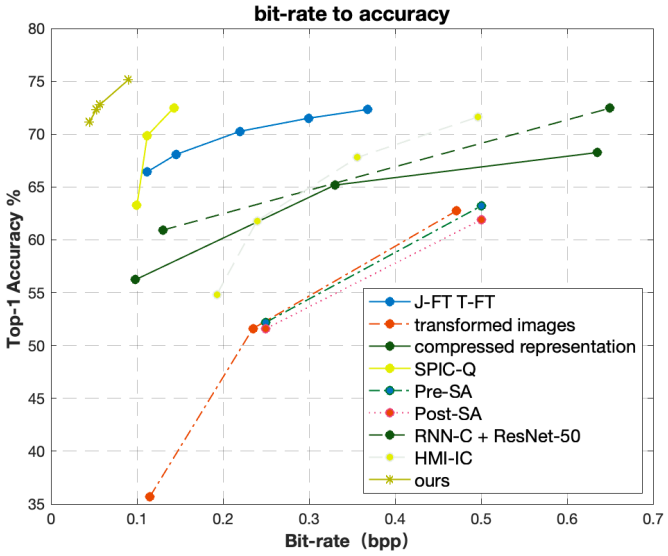

Fig. 4. Comparison of the top-1 accuracy of SIIC with redundancy removal module (ours) and J-FT T-FT [9], transformed images [10], compressed representation [11], SPIC-Q [12], Post-SA [13], RNN-C + ResNet-50 [14], HMI-IC [15] on the ImageNet1K [16] verification set.

SIIC [7] selected ViT [8] as the network for image classification. It uses transformers, which can well extract global information of the image and help produce better accuracy.

The specific structure of Decoder_sub is shown in Fig. 2. We use Decoder_h2_side structure from [7] containing four transpose convolution layers (Tconv) and one Depth2Space [5] layer to build Decoder_sub, which will amplify the shape of $\widehat{h2}$ to imitate $h1$. The original shape of $\widehat{h2}$ is (n, c, h, w). The first Tconv layer increases the channel number c of $\widehat{h2}$ from 128 to 256, which is 2c. The second and third layers of Tconv maintain the shape $(n, 2c, h, w)$ to extract and keep the important information that the following layers need. Then the Depth2Space [5] decreases the channels of $\widehat{h2}$ to 1/4 of the original, which is 0.5c = 64, and at the same time doubles h and w to 2h and 2w. Finally, the Tconv layer increases 0.5c to 2c and changes the shape of $\widehat{h2}$ from $(n, 0.5c, 2h, 2w)$ into $(n, 2c, 2h, 2w)$, which is the shape of $h1$ and $h1'$.

As shown in Fig. 1, Decoder_sub reversely predicts $h1'$ through $\widehat{h2}$, and the generated $h1'$ can be offset with $h1$, so that we can achieve the purpose of removing the redundant information in $h1$.

IV. Experiments and Results

We use the ImageNet1K [16] dataset as the training and validation dataset for the classification task. ImageNet1K has 1,000 categories of images. The training set contains 1.28M images, and the validation set has 50,000 images. Most images are larger than 256×256.

A. The First Training Stage

The purpose of this stage is to make $h1'$ as close as possible to $h1$. At this time, the batchsize is 64, and the learning rate is set to 1e-5 for the first epoch and 1e-6 for the second epoch. The optimizer is Adam [17]. The required loss function is as follows:

$$loss = MSE\ (h1', h1). \tag{11}$$

At this time, only the parameters of Decoder_sub are trained.

B. The Second Training Stage

The purpose of this stage is to transmit $\widehat{hr}$ with as low bitrates as possible. At this time, the batchsize is 64 and the learning rate is set to 1e-6 for 2 epochs. The optimizer is Adam [17]. The loss function is as follows:

$$loss = R_{\widehat{hr}}. \tag{12}$$

$R_{\widehat{hr}}$ is the bitrate required to transmit $\widehat{hr}$. The parameters trained at this time are from Decoder_sub, Decoder_h2 and Prediction Model.

C. Results and Analysis

Next are comparisons on the top-1 classification accuracy between our proposed method and other ICM pipelines on the ImageNet1K [16] validation set. The first is the comparison between the proposed SIIC with redundancy removal module and the original SIIC [7], as shown in Fig. 3. It can be found that in the range of 0~0.06 bpp, at the same accuracy level, our method requires 3%~4% less bitrates than the original SIIC [7]. Our method can also achieve more than 75% accuracy using 0.09 bpp with the help of the redundancy removal module, which is 10% less than the bitrates of the original SIIC [7]. In addition, we also compared SIIC with the proposed redundancy removal module with J-FT T-FT [9], HMI-IC [15], etc., as shown in Fig. 4. It can be seen that our proposed method can consume fewer bitrates but achieve higher top-1 classification accuracy.

V. Conclusion

In this article, we proposed a redundancy removal module to further reduce the bitrates used in SIIC. The proposed network module can be trained quickly and achieve improved results. After SIIC used the redundancy removal module, it not only improved the classification accuracy, but also further saved bitrates. Next, we plan to use the redundancy removal module in more ICM pipelines to achieve the goal of saving their bitrates.

References

[1] A. Al-Kaff, D. Martin, F. Garcia, A. de la Escalera, and J. M. Armingol, "Survey of computer vision algorithms and applications for unmanned aerial vehicles," Expert Syst. Appl., vol. 92, pp. 447–463, Feb. 2018.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[3] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018.

[4] F. Bellard. "The BPG image format," [Online]. Available: http://bellard.org/bpg/, accessed on Jul. 12, 2022.

[5] Y. Hu, W. Yang, and J. Liu, "Coarse-to-fine hyper-prior modeling for learned image compression," in Proc. AAAI Conf. Artificial Intelligence, New York, NY, USA, Feb. 2020, pp. 11013-11020.

[6] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in Proc. Int. Conf. Learn. Represent. (ICLR), Vancouver, BC, Canada, May 2018.

[7] Z. Zhang and Y. Liu, "Side information driven image coding for machines," in 2022 Picture Coding Symposium (PCS). IEEE, 2022, pp. 193–197.

[8] A. Dosovitskiy et al., "An image is worth 16x16 words: transformers for image recognition at scale," in Proc. Int. Conf. Learn. Represent. (ICLR), Virtual Event, Austria, May 2021.

[9] L. D. Chamain, F. Racapé, J. Bégaint, A. Pushparaja, and S. Feltman, "End-to-end optimized image compression for multiple machine tasks," arXiv preprint arXiv:2103.04178, Mar. 2021.

[10] N. Le, H. Zhang, F. Cricri, R. Ghaznavi-Youvalari, and E. Rahtu, "Image coding for machines: an end-to-end learned approach," in Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, Jun. 2021, pp. 1590-1594.

[11] 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), Vancouver, BC, Canada, May 2018.

[12] N. Patwa, N. Ahuja, S. Somayazulu, O. Tickoo, S. Varadarajan, and S. Koolagudi, "Semantic-preserving image compression," in Proc. IEEE International Conference on Image Processing (ICIP), Abu Dhabi, United Arab Emirates, Oct. 2020, pp. 1281-1285.

[13] S. Luo, Y. Yang, and M. Song, "DeepSIC: deep semantic image compression," in Proc. International Conference on Neural Information Processing (ICONIP), Siem Reap, Cambodia, Dec. 2018, pp. 96-106.

[14] M. Weber, C. Renggli, H. Grabner, and C. Zhang, "Observer dependent lossy image compression," in Proc. DAGM German Conf. Pattern Recognit., Bingen, Germany, Sept. 2020, pp. 130-144.

[15] Z. Wang, F. Li, J. Xu and P. C. Cosman, "Human–machine interaction-oriented image coding for resource-constrained visual monitoring in IoT," IEEE Internet of Things Journal, vol. 9, no. 17, pp. 16181-16195, Sept. 2022.

[16] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," International journal of computer vision, vol. 115, no. 3, pp. 211-252, Apr. 2015.

[17] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," in Proc. Int. Conf. Learn. Represent. (ICLR), San Diego, CA, USA, May 2015.