Side Information Driven 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 continuous improvement of computer vision technology, more and more image information is consumed by machines rather than humans. Image coding for machines (ICM) is to compress image data such that they can be more efficiently sent to the receiver side for machines to conduct visual analysis. A typical deep learning-based ICM structure contains one codec network which compresses and transmits images through the Internet and one semantic analysis task network such as image classification and object recognition. In the codec part, the side information is the hyper-prior or hierarchical layers of hyper-priors for the compression of image latent representations. In this paper, we propose a Side Information Driven Image Coding (SIIC) framework based on deep learning. It only compresses and transmits the side information to the receiver for image classification tasks. We obtain a top-1 accuracy of 70.38% on the ImageNet1K dataset with 0.046 bits per pixel.**

***Index Terms*—side information, hyper-prior, image classification, image coding for machines, transformer**

I. Introduction

The rapid development of smart cities and Internet-of-Things (IoT) [4] has greatly accelerated the progress of deep learning-based image compression and visual recognition. Deep learning-based image compression uses deep neural networks such as convolutional networks to compress images. According to [6], deep learning-based image compression significantly improves the rate-distortion performance compared to conventional image compression techniques such as JPEG [23] and BPG [24]. While traditional image compression approaches are designed for human vision, nowadays more and more images are generated by end users and transmitted to cloud servers to perform visual recognition tasks such as image classification, object detection and instance segmentation, etc. The massive amount of images transmitted to the cloud serves consumes a large Internet bandwidth. Therefore, image coding for machines (ICM) has emerged as a new coding paradigm to extract and compress image features more useful for visual recognition tasks at cloud servers.

One type of ICM frameworks is to directly extract and transmit the latent representation from the encoder side to the decoder side. For example, images are directly fed into a pre-trained Mask-RCNN networks in [5] to extract the instance segmentation map which is further compressed as a 16-bit gray-scale profile and transmitted to perform an object detection task. This method directly transmits the latent representation which is the feature tensor produced by the encoder layers. Other methods first perform special processing on the image and then transmit it, such as MAGIC [7]. It is used in mountain fire recognition and building crack detection. MAGIC transforms the original image into triangulation, and the sparse points and colors in the triangulation will cost lower bit-rate. However, MAGIC needs to learn the knowledge of how to build the triangulation for each training set. SSSIC [8] inserts backward prediction modules to remove the redundancy. Because the latent representation of 37-class classification task is encoded from the latent representation of 200-class classification task, these two representations contain redundancy. SSSIC predicts the second representation through the first one and only encodes the difference between the predicted representation and the second representation for the 200-class classification task.

Another type of existing ICM frameworks directly concatenate an image codec [1] and a visual recognition task network in an end-to-end manner [3], [9], [10], [12], [13], [14], as shown in Fig. 1. The codec usually consists of a main encoder-decoder pair and a hyper encoder-decoder pair. While the main encoder encodes the image x into latent representation y, followed by quantization, arithmetic encoder (AE), and arithmetic decoder (AD), the hyper encoder further processes y to generate side information to estimate the probability distribution parameters for the mainstream AE and AD. Then, the framework uses the output of the codec, that is, the decoded image $\hat{x}$, as the input of the task network to perform visual recognition tasks. However, the decoded image $\hat{x}$ contains redundant information for task inference, which will increase the transmission burden and is not conducive to the visual recognition task accuracy.

Compared to the main stream information y generated by the main encoder, the side information generated by the hyper encoder not only consumes less bit rates, but also contains more abstract semantics, which can be further processed to perform high-level visual recognition tasks.

In this work, for the first time in the literature, we propose an ICM framework, Side Information Driven Image Coding (SIIC), which only compresses and sends the side information generated by the hyperpriors of a learned image codec for image classification. The extremely low bit rates of the side information can greatly relieve the transmission bandwidth pressure and the proposed SIIC can still achieve highly reliable image recognition results.

We use the coarse-to-fine learned image compression framework [2] as our codec, Vision Transformer (ViT) [11] as

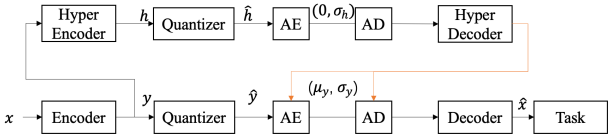

Fig. 1. The bmshj2018-hyperprior image codec [1] connected with machine task network, AE and AD are arithmetic encoding and decoding.

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

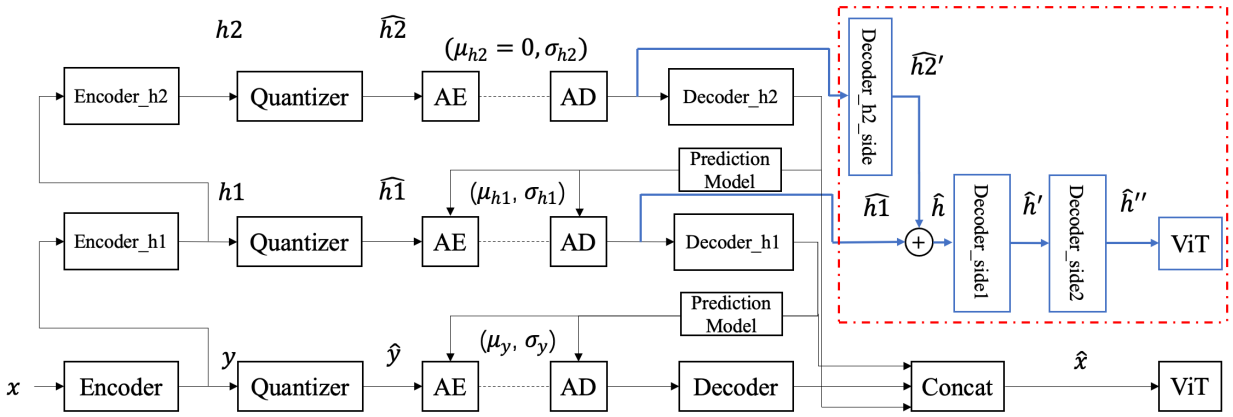

Fig. 2. The red dotted line zone is our SIIC framework for second stage training. The Concat part in the first stage is to concatenate all decoder outputs and form $\hat{x}$.

the task network, and focus on the combination of side information and ViT. The coarse-to-fine framework is improved on the basis of [1] which uses convolution layers to do the encoding and decoding process. It includes two layers of hyperpriors, which can further reduce transmission redundancy. ViT divides an image into small patches, uses the attention mechanism to calculate the degree of association among different patches, and finally obtains the image classification results. It has been proved to perform well on the ImageNet1K dataset [16].

The remaining of this paper is organized as follows: Section II introduces the related ICM frameworks, Section III elaborates our proposed method SIIC in details, Section IV presents the performance of SIIC through experiments, and Section V concludes the paper.

II. Related Work

The human-machine interaction-oriented image coding framework (HMI-IC) [4] compresses and transmits the images to complete the task of classifying images. The stream of transmitted information is also capable of generating smaller-resolution preview images. This method not only provides an early warning mechanism, but also saves the bandwidth.

The Semantics-Preserving Image Compression framework (SPIC) [18] and the compressed representation method [20] both directly feed the quantized feature into task networks to perform visual recognition, which omits the step of converting the latent representation to a recovered image.

Image pre-transformation method [19] achieves high image classification accuracy and low bit-rate by a deep encoder-decoder network with a bypass structure. In [21], while the pre-semantic DeepSIC places the semantic analysis at the encoder side, the post-semantic DeepSIC performs semantic analysis at the decoder side. The RNN-C + ResNet-50 model [22] trains a recurrent neural network (RNN) as the codec, and incorporates with a ResNet-50 network in the classification task. By this way the compressed image preserves features relevant for classification.

In [13], a hyperprior codec [1] is linked with a segmentation task. Only the first 128 channels out of the main stream information are used for the visual task. Although not using all the main stream information can objectively reduce the bit-rate required by the task, more than half of the channels may still cause data redundancy.

All the above methods directly use all or part of the mainstream latent representation, which will cause relatively large redundant information.

Compared with [3], our proposed method uses the coarse-to-fine framework in [2] as our codec, and the ViT in [11] as the task network for image classification. Compared with [13], we directly utilize the side information instead of layering the main stream information, which further reduces the bit-rate.

III. Proposed Method

A. Concatenation of the codec and the task network

The first-stage algorithmic development is to concatenate an image compression codec and an image classification task network to form an end-to-end ICM network. As shown in Fig. 2, we take the coarse-to-fine pipeline [2] as the image compression codec, and the vision transformer (ViT) as the image classification task network. The output decoded image $\hat{x}$ of the codec serves as the input of ViT.

The coarse-to-fine image codec has a main encoder-decoder pair and two hyper encoder-decoder pairs. The image x is compressed by the main encoder into a latent representation y as shown in the following equation

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

which is then quantized as $\hat{y}$, entropy encoded by an arithmetic encoder (AE) into a bitstream and transmitted to the decoder. The decoder then performs arithmetic decoding (AD), followed by the main Decoder. The first-layer hyperprior $h1$ and the second-layer hyperprior $h2$ are obtained by the following equations

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

$$h2 = Encoder\_h2(h1;\ \theta_{encoder\_h2}). \tag{3}$$

To perform entropy coding of the latent vector $\hat{y}$ and convert it to a bitstream, the arithmetic encoder and decoder need to know the probability distribution of $\hat{y}$ [2]. Here, we assume that $\hat{y}$ has a Normal distribution, and we use hyperprior $\widehat{h1}$ to estimate the parameters (μ_y, σ_y) through the Prediction Model. Similarly, $\widehat{h1}$ is $\mathcal{N}(\mu_{h1}, \sigma_{h1})$ distributed whose parameters are estimated by $\widehat{h2}$ and $\widehat{h2}$ is assumed to

be $\mathcal{N}(\mu_{h2} = 0, \sigma_{h2} = random)$ distributed. After $\widehat{h1}$, $\widehat{h2}$ and $\hat{y}$ are entropy encoded and transmitted to the cloud server end, they are parsed by the corresponding decoders: Decoder, Decoder_h1, and Decoder_h2, spliced along the channel direction and further processed by the Concat layers to obtain the decoded image $\hat{x}$.

The aforementioned coarse-to-fine image codec has two layers of hyper-priors, one more hyperprior than [1], which can further remove the redundancy contained in $h1$.

Denote the estimated distributions of $\hat{y}$, $\widehat{h1}$ and $\widehat{h2}$ as $\hat{P}(\hat{y}|\widehat{h1})$, $\hat{P}(\widehat{h1}|\widehat{h2})$ and $\hat{P}(\widehat{h2})$, then the bit-rate required to transmit the encoded image $\hat{y}$, the two encoded hyperpriors $\widehat{h1}$ and $\widehat{h2}$ can be approximated by their entropy as shown in eq (4)-(6).

$$R_{main} = E_{P(\hat{y}|\widehat{h1})}[-\log(\hat{P}(\hat{y}|\widehat{h1}))] \quad (4)$$

$$R_{\widehat{h1}} = E_{P(\widehat{h1}|\widehat{h2})}[-\log(\hat{P}(\widehat{h1}|\widehat{h2}))] \quad (5)$$

$$R_{\widehat{h2}} = E_{P(\widehat{h2})}[-\log(\hat{P}(\widehat{h2}))] \quad (6)$$

The total bit-rate R_{all} in (7) is the sum of all three bit streams, while the side information bit rate R_{side} refers to the bitstreams of the two hyperpriors.

$$R_{all} = R_{main} + R_{\widehat{h1}} + R_{\widehat{h2}} \quad (7)$$

$$R_{side} = R_{\widehat{h1}} + R_{\widehat{h2}} \quad (8)$$

We choose ViT [11] as the task network to perform image classification. It utilizes the self-attention mechanism and Transformers [15] originally applied to natural language processing (NLP) to extract image semantic information. ViT takes $\hat{x}$ as input for the classification task.

We adopt the trained coarse-to-fine model [2] and the trained ViT model [11] to initialize the concatenated network, then finetune the weights of the entire network on the ImageNet1K training set.

In order to balance the bit-rates required for transmission and the image classification accuracy, the loss function we use at this stage is:

$$Loss_{1st} = \lambda \times R_{all} + Loss_{mse} + Loss_{vit}, \quad (9)$$

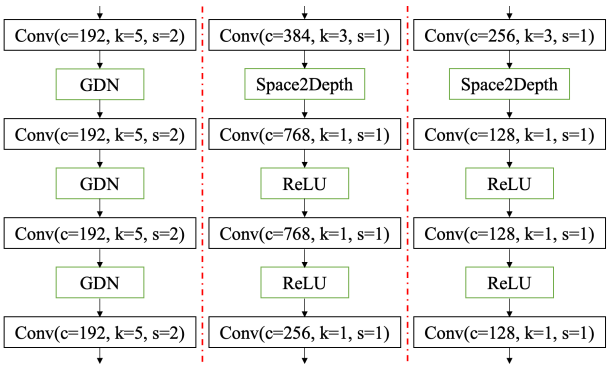

Fig. 3. The encoder structures: Encoder, Encoder_h1 and Encoder_h2 (left to right). c is the number of output channels, k is the kernel size, s is the stride. Space2Depth [2] doubles the tensor's channel and downsizes the height and width by a factor of 2.

where λ is the hyperparameter to trade off the bit-rates and the image decoding and classification fidelity, $Loss_{mse}$ is the mean squared error (MSE) between x and $\hat{x}$, and $Loss_{vit}$ is the cross-entropy loss between the ViT predicted image class label and the ground-truth class label.

B. Side information network for image classification

While existing ICM architectures use the main latent representation $\hat{y}$ or the decoded image $\hat{x}$ to conduct the classification task, in this work, we propose to use only the hyperpriors $\widehat{h1}$ and $\widehat{h2}$ at the decoder side to perform classification. This avoids the transmission of $\hat{y}$ to the decoder, hence saving the bit rates.

The second stage of algorithmic development is our proposed side information driven image coding (SIIC) network, as shown in the red dashed box in Fig. 2. In order to maximize the use of hyperprior information and let $\widehat{h2}$ have the same shape as $\widehat{h1}$, the proposed SIIC enlarges $\widehat{h2}$ by Decoder_h2_side to get $\widehat{h2}'$, as shown in eq (10).

$$\widehat{h2}' = Decoder\_h2\_side(\widehat{h2}; \theta_{decoder\_h2\_side}) \quad (10)$$

Then, SIIC adds $\widehat{h2}'$ to $\widehat{h1}$ to get $\hat{h}$. $\hat{h}$ now contains information from both $\widehat{h1}$ and $\widehat{h2}$. In order to be used as the input of ViT, $\hat{h}$ needs to be enlarged by Decoder_side1 and Decoder_side2 to get $\hat{h}''$. The details are depicted by eq (11) and (12).

$$\hat{h}' = Decoder\_side1(\hat{h}; \theta_{decoder\_side1}) \quad (11)$$

$$\hat{h}'' = Decoder\_side2(\hat{h}'; \theta_{decoder\_side2}) \quad (12)$$

Besides, the detailed structures of the proposed new network components are depicted in Fig. 3 (Encoder, Encoder_h1 and Encoder_h2) and Fig. 4 (Decoder, Decoder_side2, Decoder_h1, Decoder_side1, Decoder_h2 and Decoder_h2_side).

To train the proposed SIIC network components which include Decoder_h2_side, Decoder_side1, Decoder_side2, and the ViT parameters, the codec parameters after the first stage training need to be fixed.

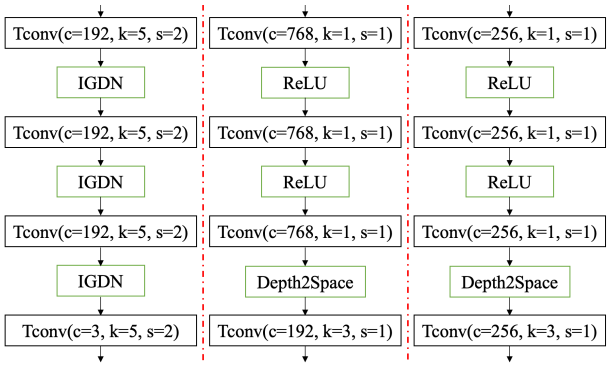

Fig. 4. The decoder structures: Decoder (or Decoder_side2), Decoder_h1 (or Decoder_side1) and Decoder_h2 (or Decoder_h2_side) from left to right. c is the number of output channels, k is the kernel size, s is the stride size. Depth2Space [2] doubles the tensor's height and width and downsizes the channel by a factor of 2.

The side information $\widehat{h1}$ and $\widehat{h2}$ are connected to the SIIC, and then its output $\hat{h}''$ is fed into the ViT trained in the first stage. The loss function used in the second stage is simply the cross-entropy loss that captures image classification errors:

$$Loss_{2nd} = Loss_{vit}, \tag{13}$$

the weights of the codec do not participate in training, so the bit-rate remains unchanged, and it is R_{side} in (8).

IV. Experiments and Results

We use ImageNet1K [16] as the dataset for experimental studies. The training set has 1.28M images, the validation set has 50,000 images, and all images belong to 1,000 categories. The images need to be resized to 256×256 before being input into the codec.

A. Two-stage training

We trained the proposed ICM network in two stages. In the first stage, we concatenate the coarse-to-fine image codec and the ViT image classifier. The coarse-to-fine framework [2] provides 7 models pre-trained on the DIV2K dataset [25] for different bit-rates. We use the first three of them, and the corresponding hyperparameter λ that controls the bit-rate is 1/0.0012, 1/0.0015, 1/0.008, respectively. We adopt the trained model of ViT provided by [11], which was pre-trained on ImageNet21k and then fine-tuned on ImageNet1k. Afterwards, we finetune the concatenated end-to-end ICM network on the ImageNet1k dataset. In the first-stage training, the batch size is 32, the learning rate is set to 1e-6, and the optimizer is the Adam optimizer [17]. We trained the network for 2 epochs.

In the second-stage training, we fix the concatenated ICM network parameters trained in the first stage, extract $\widehat{h1}$ and $\widehat{h2}$ and connect them to the proposed SIIC components. Then, we trained the parameters of the side decoders with the ViT parameters finetuned. The batch size is set as 64， and the network was trained for 7 epochs.

B. Comparison with other DNN based ICM frameworks

Fig. 5 shows the comparison of classification top-1 accuracy between different methods on the ImageNet1K validation set, with bit-rate on the horizontal axis and top-1 accuracy on the vertical axis. Compared to other methods which achieve the same classification accuracy, our method requires much less bit rates. For example, to achieve an accuracy of about 72.5%, our method only needs 0.0587 bpp, but SPIC-Q [18], which is the closest to our method, requires at least 0.143 bpp, which is 2.43 times that of our method. A detailed comparison is provided in Table I.

TABLE I. Bit-rate required to achieve around 72.5% Image Classification Top-1 accuracy

Method	Bit-rate(bpp)	Accuracy (%)
Ours	0.0587	72.84
SPIC-Q[18]	0.143	72.51
HMI-IC[4]	0.847	72.72
J-FT T-FT[3]	0.368	72.34
RNN-C + ResNet-50[22]	1.0	73.16

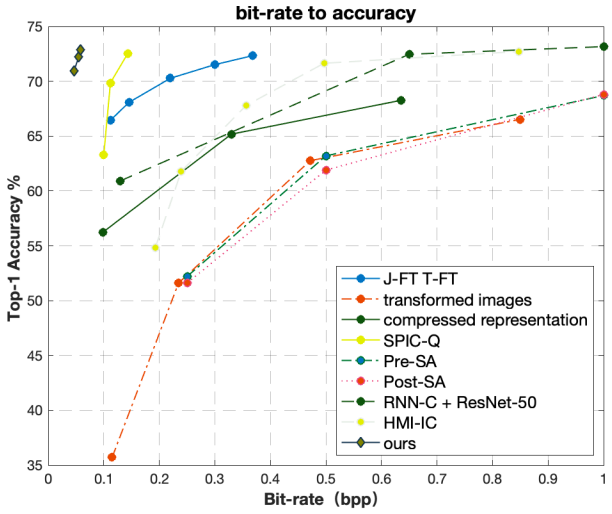

Fig. 5. Comparison of the results of different methods on the ImageNet1K validation set, including J-FT T-FT[3], transformed images[19], compressed representation[20], SPIC-Q[18], Pre-SA[21], Post-SA[21], RNN-C + ResNet-50[22], HMI-IC[4], and ours.

V. Conclusion

We propose a deep learning-based image compression network for classification tasks. For the first time in the literature, we combine the side information that assists the main stream information encoding with the image classification network to form the SIIC framework. Since the hyperprior has more abstract semantic information and consumes less bit-rates, the proposed method can save the transmission bandwidth while maintaining a high level of image classification accuracy. In future work, we will continue to explore the role of side information in other visual recognition tasks.

References

[1] 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.

[2] 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.

[3] 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.

[4] 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.

[5] S. Chen et al., "A new image codec paradigm for human and machine uses," *arXiv preprint arXiv:2112.10071*, 2021.

[6] 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 Conference on Computer Vision and Pattern Recognition*, New Orleans, LA, USA, Jun. 2022, pp. 5718-5727.

[7] P. Chakraborty, J. Cruz, and S. Bhunia, "MAGIC: machine-learning-guided image compression for vision applications in Internet of Things," *IEEE Internet of Things Journal*, vol. 8, no. 9, pp. 7303-7315, Sept. 2021.

[8] N. Yan, C. Gao, D. Liu, H. Li, L. Li, and F. Wu, "SSSIC: semantics-to-signal scalable image coding with learned structural representations," *IEEE Trans. on Image Process.*, vol. 30, pp. 8939-8954, Nov. 2021.

[9] L. D. Chamain, F. Racape, J. Begaint, A. Pushparaja, and S. Feltman, "End-to-end optimized image compression for machines, a study," in

Proc. Data Compression Conference (DCC), Snowbird, UT, USA, Mar. 2021, pp. 163-172.

[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] 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.

[12] M. Kawawa-Beaudan, R. Roggenkemper, and A. Zakhor, "Recognition-aware learned image compression," *arXiv preprint arXiv:2202.00198*, 2022.

[13] H. Choi and I. V. Bajic, "Scalable image coding for humans and machines," *IEEE Trans. on Image Process.*, vol. 31, pp. 2739-2754, Mar. 2022.

[14] S. Wang, Z. Wang, S. Wang, and Y. Ye, "End-to-end compression towards machine vision: network architecture design and optimization," *IEEE Open Journal of Circuits and Systems*, vol. 2, pp. 675-685, Nov. 2021.

[15] A. Vaswani et al., "Attention is all you need," in *Proc. 31st Annu. Conf. Neural Inf. Process. Syst. (NIPS)*, vol. 30, Long Beach, CA, USA, Dec. 2017, pp. 1-11.

[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.

[18] 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.

[19] S. Suzuki, M. Takagi, K. Hayase, T. Onishi, and A. Shimizu, "Image pre-transformation for recognition-aware image compression," in *Proc. IEEE International Conference on Image Processing (ICIP)*, Taipei, Taiwan, Sept. 2019, pp. 2686-2690.

[20] 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.

[21] 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.

[22] 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.

[23] G. K. Wallace, "The JPEG still picture compression standard," *Commun. ACM*, vol. 34, no. 4, pp. 30-44, Feb. 1991.

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

[25] E. Agustsson and R. Timofte, "NTIRE 2017 challenge on single image super-resolution: dataset and study," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Honolulu, HI, USA, Jul. 2017, pp. 1122-1131.