Learned Image Compression with Super-Resolution Residual Modules and DISTS Optimization

Akifumi Suzuki* Hiroaki Akutsu* Takahiro Naruko*
Hitachi, Ltd., Japan
akifumi.suzuki.nu@hitachi.com

Koki Tsubota Kiyoharu Aizawa
The University of Tokyo, Japan

Abstract

Neural network-based image compressors have the ability to optimize various perceptual image quality metrics. We propose improved methods that is based on selective-detail decoding, which uses two decoders (a main decoder and selective-detail decoder) optimized for different image-quality metrics and applies the output result of a suitable decoder for each part of an image. The following three improvements are obtained with the proposed method. (1) Inspired by the super-resolution task, we add a super-resolution residual module to the main decoder, which is trained to up-sample an image to a resolution beyond the source image, aiming to output a visually clearer image. (2) To improve the perceptual image quality of the main decoder, we use an image quality metric based on Deep Image Structure and Texture Similarity (DISTS), the similarity of which is close to that of human senses with respect to texture. (3) To improve the mask accuracy for decoder selection, cross entropy loss is used for comparing predicted masks and ground truth masks. We also use the weighted mean squared error to improve the visual quality of the text part of an image.

1. Introduction

Research on learned image compressors using end-to-end neural networks has recently been conducted [9], [10], [12]. A neural network-based learned image compressor generally consists of an encoder that converts an image into a feature map, quantizer that quantizes the feature map, and decoder that generates an image from the quantized feature map. It also consists of an entropy estimator that predicts the probability of each value in the quantized feature map, and an adaptive arithmetic coder that uses the prediction of the probability to reduce the amount of data in the quantized feature map.

Among the components of the image compressor, the encoder, decoder, and entropy estimator are composed of neural network to acquire compression capability through learning. The advantage of such a learning image compressor over traditional compressors is that any differentiable image quality metric can be used as the loss function in learning, and various functions of neural networks (e.g., super-resolution, Generative Adversarial Networks (GANs) [13], identification), which have recently been studied, can be added as components to the image compressor. For example, by using a neural network that estimates the score of human perceptual image quality [17], [5], [14] as the loss function of an image compressor [11], it is possible to generate images that do not make people feel uncomfortable, even at low bit rates. In addition, research on the application of GANs to image compression [1] has shown that it is possible to generate images that are close to the distribution of the training data set at low bit rates.

We propose a method that is based on selective-detail decoding [2], which uses two decoders and automatically selects a suitable decoder for each part of an image to improve human perception of image quality at low bit rates. The following three improvements are obtained with the proposed method. (1) Inspired by the super-resolution [6] [15] task, we add a super-resolution residual module (S) to the main decoder (G_m), which is trained to scale the image to a resolution beyond the source image to output a visually clearer image. (2) The image compressor is optimized using a polynomial equation that combines the results of the discriminator (D) and Deep Image Structure and Texture Similarity (DISTS) [5], the similarity of which is close to human senses with respect to texture, to improve the perceptual quality of the G_m. (3) Cross entropy loss is used to improve the mask accuracy for decoder selection. We also use the weighted Mean Squared Error (MSE) to improve the visual quality of the text part.

*The first three authors contributed equally

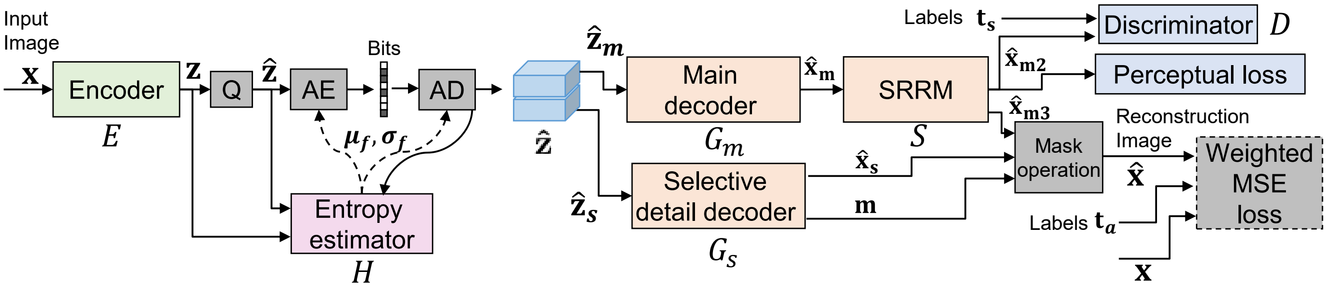


Figure 1. Network architecture overview of the proposed method. AE and AD are arithmetic encoder and decoder.

2. Proposed Method

2.1. Architectural Overview of the Proposed Method

The encoder E, main decoder G_m, selective-detail decoder G_s, entropy estimator H, discriminator D, super-resolution residual module S are configured in convolutional neural networks and have learnable parameters. We use round based quantizer [3] as a quantizer Q. We obtain quantized feature maps $\hat{\boldsymbol{z}}$ of an image $\boldsymbol{x}$ by $\hat{\boldsymbol{z}} = Q(E(\boldsymbol{x}))$. H outputs the parameters of the probability density distribution function of $\hat{\boldsymbol{z}}$ required for arithmetic coding and decoding.

2.2. Main Decoder and Selective-Detail Decoder

The proposed method consists of two types of decoders based on those developed by Akutsu et al. [2]. The G_m is responsible for the output of the entire image and emphasizes expressions such as textures. The G_s is responsible for a specific part such as texts and faces and outputs high-quality images. The G_s outputs the mask $\boldsymbol{m}$ and $\hat{\boldsymbol{x}}_{\boldsymbol{s}}$ of the image. Mask $\boldsymbol{m}$ represents the region output by the G_s. The G_s learns to generate a mask properly for the ground truth label (the weighted MSE loss and cross entoropy loss of the G_s are described later).

2.3. Super-Resolution Residual Module

Figure 2 is an overview of Super-Resolution Residual Module (SRRM). The SRRM receive the output image ($\hat{\boldsymbol{x}}_{\boldsymbol{m}}$) of the G_m as input, and outputs a high-resolution output ($\hat{\boldsymbol{x}}_{\boldsymbol{m2}}$), obtained by adding a residual image ($\hat{\boldsymbol{x}}_{\boldsymbol{sr}}$) and a simply twice up sampled image of $\hat{\boldsymbol{x}}_{\boldsymbol{m}}$ by bilinear. After pre-training E, G_m, G_s, H, and D, the SRRM S is added and all the networks are fine-tuned. In this fine-tune training, 1/2 downsampled image ($\boldsymbol{x}$) of ground truth high-resolution image ($\boldsymbol{x}_{\boldsymbol{hr}}$) is used for input to the E. Note that additional dataset for training super resolution residual module is not required because the original and the downsampled images of training dataset are used as x_{hr} and x respectively during training. We train the networks with perceptual loss and discriminator loss using $\hat{\boldsymbol{x}}_{\boldsymbol{m2}}$ and $\boldsymbol{x}_{\boldsymbol{hr}}$.

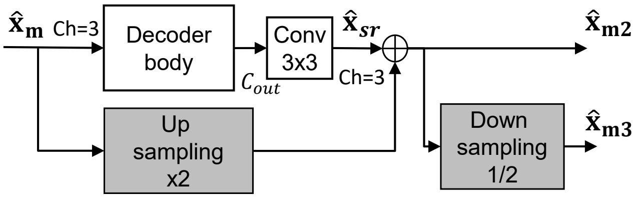


Figure 2. Super-resolution residual module.

Finally we obtain output images for reconstruction ($\hat{\boldsymbol{x}}_{\boldsymbol{m3}}$) by simply 1/2 downsampling of $\hat{\boldsymbol{x}}_{\boldsymbol{m2}}$.

2.4. Entropy Estimator

The entropy estimator uses the context estimator with the addition of a casual convolution module and hyperprior predictions as [2]. This estimator predicts the probability of symbols of each element of feature maps used for arithmetic coding by Gaussian distribution. The entorpy estimator outputs two parameters of the Gaussian distribution for each element of the feature map.

2.5. Loss Functions

2.5.1 Distortion Loss

With our proposed method, we use loss function L_{dm} using DISTS [5] with Multi-Scale Structural SIMilarity (MS-SSIM) [16] to estimate the score of human perceptual image quality in the training of the G_m.

$$\begin{aligned} \mathcal{L}_{dm} = \lambda_{mp}\mathbb{E}\left[DISTS(\boldsymbol{x}_{\boldsymbol{hr}}, \hat{\boldsymbol{x}}_{\boldsymbol{m2}})\right] \\ +\lambda_{ms}\mathbb{E}\left[1 - MSSSIM(\boldsymbol{x}_{\boldsymbol{hr}}, \hat{\boldsymbol{x}}_{\boldsymbol{m2}})\right]. \end{aligned} \tag{1}$$

We also use the weighted MSE for the training of the G_s. The weighted MSE calculates the MSE between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ in the region specified in the given annotation information $\boldsymbol{t}_{\boldsymbol{a}}$.

$$\mathcal{L}_{ds} = \mathbb{E}\left[wMSE(\boldsymbol{x}, \hat{\boldsymbol{x}}, \boldsymbol{t}_{\boldsymbol{a}})\right]. \tag{2}$$

2.5.2 Mask Loss

To improve the accuracy of the mask output of the G_s, the proposed method uses a new mask loss that calculates the

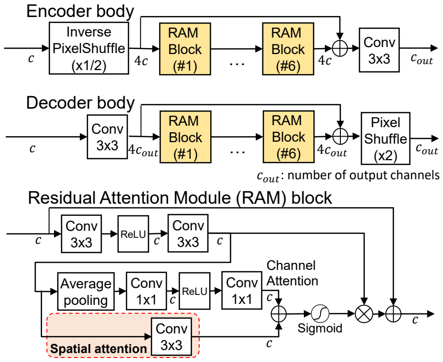


Figure 3. Network building blocks.

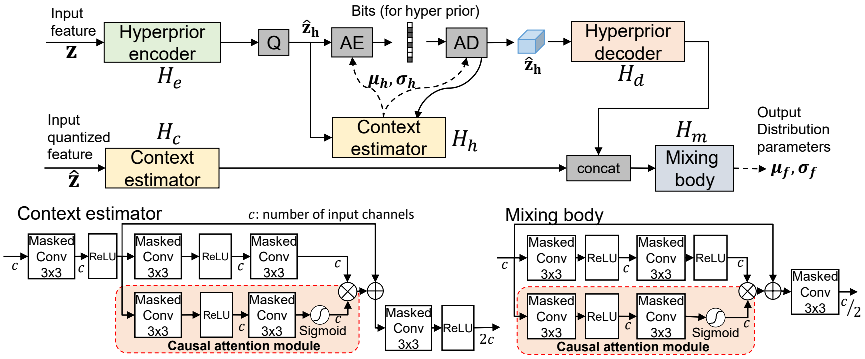

Input feature $\mathbf{z}$
Hyperprior encoder
H_e
Q
$\hat{\mathbf{z}}_\mathbf{h}$
AE
Bits (for hyper prior)
AD
$\hat{\mathbf{z}}_\mathbf{h}$
Hyperprior decoder
H_d
$\boldsymbol{\mu_h}, \boldsymbol{\sigma_h}$
Context estimator
H_h
Input quantized feature $\hat{\mathbf{z}}$
H_c
Context estimator
concat
H_m
Mixing body
Output Distribution parameters
$\boldsymbol{\mu_f}, \boldsymbol{\sigma_f}$
Context estimator
c: number of input channels
Masked Conv 3x3
ReLU
Sigmoid
Causal attention module
$2c$
Mixing body
$c/2$

Figure 4. Entropy estimator.

cross entropy of the mask output $\boldsymbol{m}$ against the the ground truth annotation information $\boldsymbol{t_a}$. To avoid missing the mask area, we weight the cross entropy by multiplying the annotation area by a factor of 2.

$$\mathcal{L}_{mask} = -\mathbb{E}\left[2\boldsymbol{t_a}\ \log(\boldsymbol{m}) + (1-\boldsymbol{t_a})\log(1-\boldsymbol{m})\right]. \quad (3)$$

2.5.3 Entropy Loss

The entropy loss of the feature map of the proposed method is

$$\mathcal{L}_e = -\mathbb{E}\left[I(\hat{\boldsymbol{z}}, \boldsymbol{\mu_f}, \boldsymbol{\sigma_f})\right]. \quad (4)$$

$$I_f(\hat{\mathbf{z}}) = -\sum \log(\frac{1}{2}\text{erf}(\frac{\hat{\mathbf{z}} - \boldsymbol{\mu_f} + 0.5}{\sqrt{2}\boldsymbol{\sigma_f}}) - \frac{1}{2}\text{erf}(\frac{\hat{\mathbf{z}} - \boldsymbol{\mu_f} - 0.5}{\sqrt{2}\boldsymbol{\sigma_f}})). \quad (5)$$

This is the same for hyper prior entropy loss $\mathcal{L}_h$.

2.5.4 Adversarial Loss

In addition to $\mathcal{L}_{dm}$ and $\mathcal{L}_{pm}$, discriminator loss is also used to learn the G_m. The discriminator loss is defined by

$$\mathcal{L}_g = \mathbb{E}\left[\log(1+e^{D(\mathbf{x_{hr}},\mathbf{t_s})})\right] + \mathbb{E}\left[\log(1+e^{rD(\hat{\mathbf{x}}_{\mathbf{m2}},\mathbf{t_s})})\right]. \quad (6)$$

where r is defined as $r = -1$ at discriminator phase and as $r = 1$ at generator phase. The D uses semantic segmentation labels $\boldsymbol{t_s}$ with images as inputs, similar to conditional GANs. With our method, the labels are not input to the generator side for practical use as introduced in [2].

2.5.5 Total Loss

The final loss function is the following:

$$\min_{\theta_{G_{m,s},E,H,S}} \min_{\theta_D} V(\theta_{G_{m,s},E,H,S}, \theta_D) = \mathcal{L}_{dm} + \lambda_s\mathcal{L}_{ds} + \lambda_g\mathcal{L}_g + \lambda_{mask}\mathcal{L}_{mask} + \lambda_e\mathcal{L}_e. \quad (7)$$

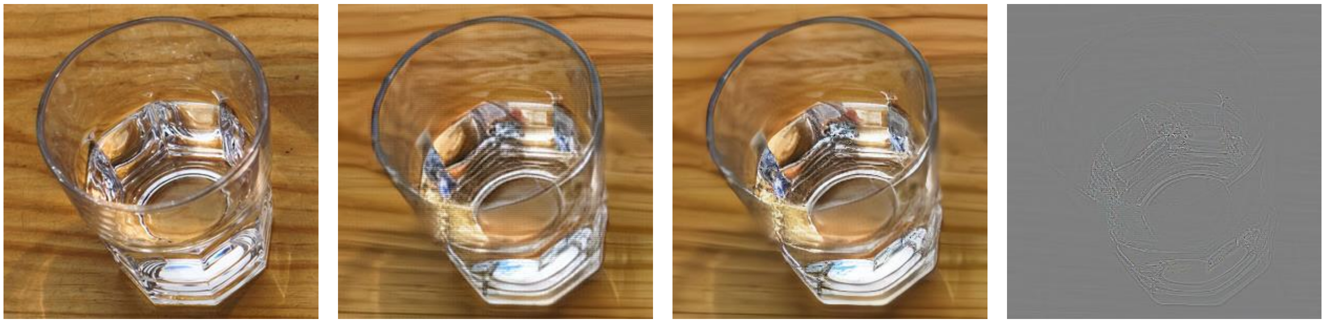
3. Experimental Results

3.1. Experimental Conditions

The E was composed of four encoder bodies, and a 3x3 conv layer was added at the end. The numbers of channels between these components were 3, 32, 64, 128, 192, and 64. The G_m was composed of four encoder bodies, and a 3x3 conv layer was inserted first. The numbers of channels among these components were 32, 192, 128, 64, 32, and 3. The G_s was similarly configured, but the numbers of channels among the components were 32, 192, 96, 64, 32, and 4. Hyper encoder (H_e) was composed of two encoder bodies, and a 3x3 conv layer was inserted first. The numbers of channels among these components were 64, 32, 32, and 32. Hyper decoder (H_d) was composed of two decoder bodies, and a 3x3 conv layer was inserted last. The numbers of channels among these components were 32, 32, 32, and 128. The configuration of the other components of the entropy estimator is as illustrated in Figure 3.

The D was composed of four encoder bodies, and a 3x3 conv layer was added at the end. The numbers of channels among these components were 8, 32, 64, 128, 192, and 1. The input of the images was three channels, and the remaining five channels were used for the input of label $\mathbf{t_s}$ in D. The input labels were one-hot expressions, and additional 1x1 convolution networks with a final output of five channels were added to reduce the label dimension.

We used images from the Open Images Challenge 2018 dataset [7] for training. For those images, semantic segmentation for $\mathbf{t_s}$ was machine generated, and annotations of faces and text parts for $\mathbf{t_a}$ were also machine generated, and used for training. We ran 800,000 training iterations using ADAM Optimizer [8] as pre-training before adding the SRRM. We then added the SRRM and ran 400,000 training iterations with a learning rate of 4e-5 and batch size of 4. The hyperparameters in each target bit-per-pixel (bpp) are listed in Table 2. The values in Table 2 were determined based on bpp and human observation of images from mul-



(a) Ground truth

(b) Main decoder output ($\hat{x}_m$)

(c) SRRM output ($\hat{x}_{m3}$)

(d) Residual image ($\hat{x}_{sr}$)

Figure 5. Experimental results using CLIC2021 validation dataset (configuration: Target bpp 0.075)

	Target bpp								
	0.075			0.15			0.30		
	Ours	JPEG	BPG [4]	Ours	JPEG	BPG	Ours	JPEG	BPG
Peak Signal-to-Noise Ratio (PSNR)	26.1	22.5	28.4	27.3	26.5	30.7	29.0	30.0	33.1
MS-SSIM	0.938	0.744	0.915	0.957	0.860	0.947	0.972	0.939	0.967
DISTS	0.185	0.409	0.231	0.138	0.306	0.186	0.120	0.196	0.145
bpp	0.0686	0.0727	0.0652	0.134	0.141	0.141	0.299	0.299	0.282

Table 1. Evaluation results using CLIC2021 validation dataset. The chroma format of JPEG and BPG were set to 4:2:0.

	Target bpp		
	0.075	0.15	0.30
λ_{mp}	250	250	250
λ_{ms}	5000	5000	5000
λ_g	50	50	50
λ_s	5.00	5.00	5.00
λ_e	4000	1500	250
λ_{mask}	2667	1000	167

Table 2. Hyperparameters for each target bpp

tiple compressors with different hyperparameters.

3.2. Results

Figure 5 shows the evaluation results using the validation dataset of CLIC2021. Figure 5 (b) is the image output from the G_m when the ground-truth image (a) is input to the compressor. Figure 5 (d) is the residual image using (b) as input of the SRRM, normalized, and reduced to the same size as (a). Figure 5 (d) shows that the residual image draws the edges of the glass in the image. The output image of the SRRM is shown in (c), which is a clearer image than (b) with reduced artifacts in areas where intensity changes are steeper. The evaluation results for each target bpp of the CLIC2021 validation dataset are listed in Table 1. Table 1 shows that the proposed method provides higher perceptual image quality (DISTS) compared to conventional methods in each bitrate conditions.

4. Conclusion

We proposed a method that is based on selective detail decoding [2], which involves using two decoders and automatically selects the suitable decoder for each part of an image to improve human perception of image quality at low bit rates. We proposed the SRRM module for image compression that trains to output high resolution image and downsample it to the original size. We also proposed a image compression method optimized with the image quality index based on DISTS and the adversarial loss, so that the similarity of the texture is close to the human sense. For the selective-detail decoder responsible for specific parts of an image and outputs high-quality images, we used a mask loss function to improve the results of automatically generating masks. We believe the proposed method will improve perceptual image quality at ultra-low bitrates.

Acknowledgement

Computational resource of AI Bridging Cloud Infrastructure (ABCI) provided by National Institute of Advanced Industrial Science and Technology (AIST) was used.

References

[1] Eirikur Agustsson, Michael Tschannen, Fabian Mentzer, Radu Timofte, and Luc Van Gool. Generative adversarial networks for extreme learned image compression. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 221–231, 2019. 1

[2] Hiroaki Akutsu, Akifumi Suzuki, Zhisheng Zhong, and Kiyoharu Aizawa. Ultra low bitrate learned image compression by selective detail decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 118–119, 2020. 1, 2, 3, 4

[3] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. *arXiv preprint arXiv:1611.01704*, 2016. 2

[4] Fabrice Bellard. Bpg image format. https://bellard.org/bpg/. 4

[5] Keyan Ding, Kede Ma, Shiqi Wang, and Eero P Simoncelli. Image quality assessment: Unifying structure and texture similarity. *arXiv preprint arXiv:2004.07728*, 2020. 1, 2

[6] Chao Dong, Chen Change Loy, Kaiming He, and Xiaoou Tang. Image super-resolution using deep convolutional networks. *IEEE transactions on pattern analysis and machine intelligence*, 38(2):295–307, 2015. 1

[7] Google. Overview of the open images challenge 2018. https://storage.googleapis.com/openimages/web/challenge.html, 2018. 3

[8] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 3

[9] Mu Li, Wangmeng Zuo, Shuhang Gu, Debin Zhao, and David Zhang. Learning convolutional networks for content-weighted image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3214–3223, 2018. 1

[10] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. Conditional probability models for deep image compression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4394–4402, 2018. 1

[11] Fabian Mentzer, George Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *arXiv preprint arXiv:2006.09965*, 2020. 1

[12] David Minnen, Johannes Ballé, and George Toderici. Joint autoregressive and hierarchical priors for learned image compression. *arXiv preprint arXiv:1809.02736*, 2018. 1

[13] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. 1

[14] Ekta Prashnani, Hong Cai, Yasamin Mostofi, and Pradeep Sen. Pieapp: Perceptual image-error assessment through pairwise preference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2018. 1

[15] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883, 2016. 1

[16] Zhou Wang, Eero P Simoncelli, and Alan C Bovik. Multiscale structural similarity for image quality assessment. In *The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402. Ieee, 2003. 2

[17] 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. 1