Image Coding Method with a Super-Resolution Convolutional Neural Network

Yutaka Nagasaki, Tomo Miyazaki, Yoshihiro Sugaya, Shinichiro Omachi
Department of Communications Engineering, Graduate School of Engineering, Tohoku University
6-6-05, Aoba Aramaki, Aoba-ku, Sendai, 980-8579 Japan
{nagad,tomo,sugaya}@iic.ecei.tohoku.ac.jp, machi@ecei.tohoku.ac.jp

***Abstract*—To improve the quality of image coding, a combination strategy of image coding and deep neural network is proposed. The input image is passed through a network before encoding. This network is designed using convolutional neural network and is trained so that the image quality after decoding is improved. To achieve an effective training, we propose a network that simulates JPEG encoding and propose a strategy of training these networks for image coding. The experimental results show that the image quality is improved by the proposed method.**

***Keywords*—Image coding, JPEG, convolutional neural network, super-resolution**

I. Introduction

Image and video coding methods have been actively studied, and higher compression ratio and higher quality coding methods have been developed. However, due to various factors, these methods are not always widespread. For example, JPEG is still the de facto standard for image coding [8]. This is mainly because its encoding-decoding process is simple and huge computational power is not necessary for decoding. Although JPEG is a widely used image coding method, there is room for improvement. Specifically, JPEG is based on hand crafted encoder and decoder. Hence, it is not optimal coding method for all image contents.

In the field of image processing, technologies for solving various problems with machine learning have been developed. Inspired by the success of image processing using machine learning technologies, we use convolutional neural network to modify images so that the images can be more suitable for the JPEG.

In this paper, we propose a method to improve the quality of encoded images by using machine learning technology without changing the decoding algorithm. Our goal is to realize higher quality image coding using the existing image coding standard. The method is to turn the image into the one suitable for encoding. The advantage of this method is that existing decoding devices can decode the image in the same way as the normally encoded images. Our proposed method is extremely effective when the bandwidth of communication is limited such as using mobile networks. The proposed method can improve the image quality by modifying the image before communication instead of increasing the quality parameter. In addition, our proposed approach is not limited to JPEG but can be applied to any coding method.

II. Related Work

When enlarging an image, bilinear, bicubic and Lanczos-N methods are used. However, these methods deteriorate the image quality by blurring and jaggies. Since the interpolation algorithm uses the pixels included in the original image, such defects will inevitably occur. It is necessary to predict the high-frequency component lost from the original image. Super-resolution is a method to convert a low-resolution image into a high-resolution one by predicting the high-frequency component. SRCNN [1], DRCN [10], FSRCNN [2], SRGAN [3] and RCAN [4] are deep learning-based super-resolution networks. In this work, we use VDSR [5].

VDSR is a deep learning-based super-resolution approach. This is a very deep neural network that has 20 convolution-ReLU layers. Once image details are predicted by this network, they are added to the input image and it is the final output. In other words, the output of the network is a residual image. Super-resolution mainly interpolates higher frequency information and it is a slight change from a low-resolution image. Calculating the residual image is more effective than directly make the output image. Thus, the super-resolution network emphasizes high-frequency components.

ViSTRA [6] is a network that combine VDSR and video coding. It is about coding the whole video, but there is also an approach for still images. It is a method based on HEVC [9], but the image is not directly encoded at the time of encoding, but the one with height and width reduced to half is encoded with higher image quality. After decoding, it is up sampled to its original size and input to VDSR to obtain the final output.

Inspired by these methods, we considered using a convolutional neural network before JPEG encoding so that we can modify images into more suitable ones. We stress that JPEG decoding devices do not need neural networks or extra calculation. Our approach is to improve the image quality after JPEG encoding and decoding by emphasizing high frequency components by a convolutional neural network so that it is not lost by JPEG encoder-decoder.

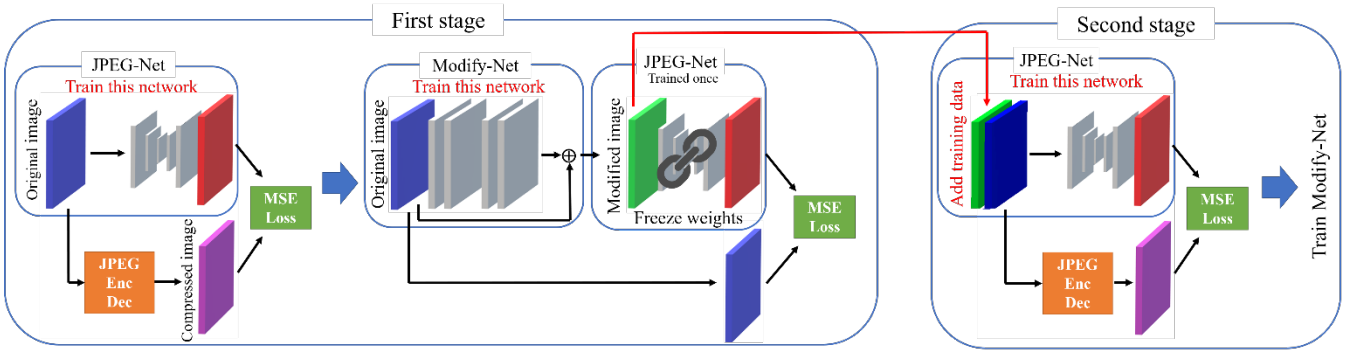


Figure 2: **The training stages.** In the first stage, JPEG-Net is trained. Subsequently, we train Modify-Net by fixing JPEG-Net. In the second stage, both of modified images and original images are used.

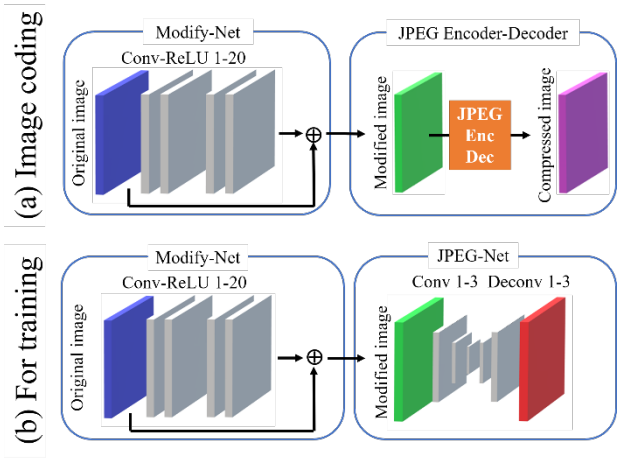


Figure 1: **The overview of the proposed method.**

III. Method

We show the proposed method in Figure 1. In image coding, we modify the image suitable for JPEG encoding by Modify-Net. Then we give the modified image into the JPEG encoder to produce a compressed image. In order to improve the quality of the compressed image, it is ideal to use the mean square error (MSE) between the input image and the JPEG compressed image as the loss function of the Modify-Net. However, it is difficult because we cannot propagate the loss since the JPEG encoder is not differentiable. Therefore, we introduce another convolutional neural network called JPEG-NET to train the Modify-Net. The JPEG-Net emulates the encoding process of JPEG and produces images that are similar to JPEG output. The aim of the JPEG-Net is to train the Modify-Net. Consequently, we can train the Modify-Net.

The advantage of the proposed method is that additional calculation is not required for decoder. Only the encoder needs computational resources to perform the proposed method, and there is no problem using low-performance computers on the decoding side.

3.1. The Network Archtecture

Inspired by VDSR [5], we design the Modify-Net. We use 20 convolution-ReLU layers. Each convolution layer except the first and the last one has 64 filters of the size 3x3x64. The first one has 64 filters of the size 3x3x1 and the last one is a single 3x3x64 filter. These filters are applied to zero-padded images, so the size of images does not change from the input to the output. We adopted three channels, Y, Cr, and Cb, for the input and output images.

JPEG-Net has three convolution layers and three deconvolution layers. Its input size is 3-channel 8x8 and output is the same. JPEG encoder quantizes an image with 8x8 blocks, and it does not use the data out of the block. This network imitates the structure of the JPEG encoder. The size of the output from the Modify-Net is three channels 64x64, so it is cropped to three channels 8x8 blocks.

3.2. Training

There are two stages for training as shown in Figure 2.

In the first stage, we train the JPEG-Net using original images and JPEG compressed images. We calculate MSE as loss function over compressed images produced by JPEG encoder and JPEG-Net. We fix the quality parameter of JPEG. Note that image quality needs to be fixed for the JPEG-Net. We use OpenCV as a JPEG encoder and decoder. After the JPEG-Net is trained, we train the Modify-Net using trained JPEG-Net. We fixed the parameters of the JPEG-Net while training the Modify-Net. We calculate MSE as loss function over the compressed images and original images so that the compressed image by the proposed method can be closer to the original images, i.e., high quality images.

The problem of the first stage is to input the original image into the JPEG-Net. Whereas, modified images are provided to the JPEG-Net when image coding, see Figure 1 (a). Therefore, we propose the second stage to train the JPEG-Net using modified images, and subsequently, we train the Modify-Net by fixing the parameters of JPEG-Net.

In the second stage, we use both of the images modified by the Modify-Net and the original images. This makes the JPEG-Net more appropriate to the Modiry-Net output. We can repeat the second stage, result in the third stage.

We use Momentum SGD [7] as an optimizer in the whole training. The number of training epoch is 50, and the learning rate is 0.1 and we change it to 1/10 in every 10 epochs. Then the momentum is 0.9. All losses are calculated by the MSE.

IV. Results

We verified the effectiveness of the proposed method through the experiments. We used the images in Kodak Image Dataset for the experiments. This dataset consists of a total of 24 images of size 768x512. We used 21 images for training, and the rests for test. We crop them to 64x64 for Modify-Net training. In addition, we extracted 8x8 images from the 64x64 images compressed by JPEG, and the extracted 8x8 images are used to train JPEG-Net. Inversion extension is applied only for Modify-Net training. The total number of images is 8064 for training and 288 for testing. To train this network, we changed the quality parameter of OpenCV JPEG encoder.

4.1. Quantitative comparisons

The average PSNR and average SSIM are shown in Table 1. Note that these results are calculated by using the original images and the images compressed by the proposed method that is shown as Figure 1 (a). The results show that the average PSNRs of the proposed method on all situations are lower than that of the JPEG images. However, the average SSIMs at the second and third stages are better than JPEG. In addition, the networks trained the second or the third stages are higher than the first stage at PSNR and SSIM. These results show the effectiveness of the additional training.

The PSNRs of JPEG-Net are shown in Table 2. These results were obtained by comparing the difference between outputs of the JPEG-Net and the JPEG. The PSNRs on quality 15 are better. It is because the difference between the original images to JPEG compressed images on quality 15 is smaller than that of quality 5, and the JPEG-Net does not have to change the image much on quality 15. Moreover, PSNRs are getting lower as training is repeated. On this testing we used only original images for input. Since the JPEG-Net that has been repeatedly trained is learned at the output of the Modify-Net, the reproducibility of the JPEG for the original image is low.

4.2. Visual comparisons

We show the output images of the proposed method in Figure 3, 4, and 5. The size of these images is 256x256, and

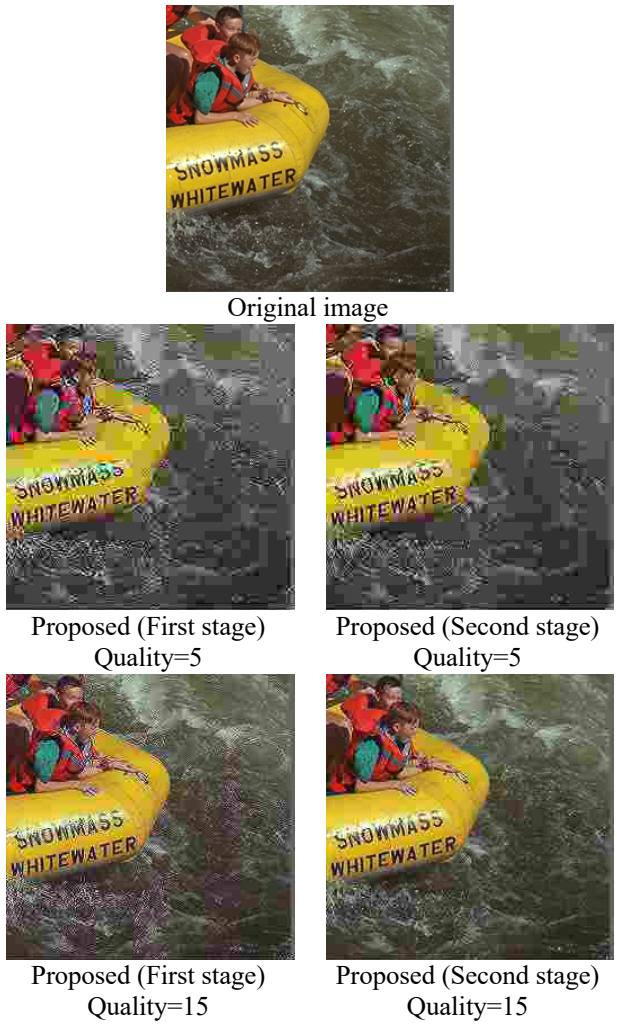
Table 1: **PSNRs and average SSIMs of the proposed method**

Average PSNR	Proposed			JPEG
	First stage	Second stage	Third stage	
Quality=5	23.4266	25.4661	25.7309	26.8417
Quality=15	25.4329	30.0170	29.7987	31.7482

Average SSIM	Proposed			JPEG
	First stage	Second stage	Third stage	
Quality=5	0.4272	0.4498	0.4545	0.4380
Quality=15	0.5439	0.6197	0.6148	0.6280

Table 2: **Results of JPEG-Net PSNR**

Average PSNR	JPEG-Net		
	First stage	Second stage	Third stage
Quality=5	30.249	29.934	29.9577
Quality=15	35.8391	35.3654	35.3914



Original image

Proposed (First stage) Quality=5

Proposed (Second stage) Quality=5

Proposed (First stage) Quality=15

Proposed (Second stage) Quality=15

Figure 3: **The results of the proposed method changing the number of training.**

Figure 4: **The results of the proposed method on every quality.** Upper images are the whole image and lowers are the partially expanded image of the upper ones.

they are made from four 64x64 images that are the output of the proposed method.

Figure 3 shows the effect of training multiple stages. The proposed image trained once is overemphasized with high-frequency components, resulting in an unnatural image. It can be seen that the outputs at the third stage are more natural.

Figure 4 compares the proposed method with JPEG. In quality 5, the image obtained by the proposed method is edgy compared to the JPEG image. This makes the text in the image very readable. These characteristics are also seen in other images. In quality 15, the image obtained by the proposed method is sharper than the JPEG image as quality 5. However, since the block noise was increased due to the increase of high-frequency components, the proposed method with quality 15 generated more unnatural than the normal JPEG one.

Figure 5 shows the images generated by the Modify-Net. The Modify-Net has been trained to enhance the high-frequency components.

4.3. Consideration

The whole results show that the proposed method is more effective in low-quality JPEG encoding. It is thought that the high-frequency component is easily lost in low-quality JPEG encoding, so the enhancement of the high-frequency component is directly connected to the fact that the high-frequency information tends to remain in the encoding. However, in high-quality JPEG encoding, the enhancement of the high-frequency component remains in the image after JPEG compression. In high quality coding, the high-frequency information of the original image can be left almost as it is, so even if the modified image is encoded, it will only be encoding an image different from the original image. One reason is that JPEG-Net cannot completely simulate the JPEG. The PSNR of the JPEG-Net is quite low. It is because the hue reduction caused by JPEG is not completely represented by the proposed network. The output of the JPEG-Net is rich in color. If the result of this network is better, the Modify-Net will be able to output better images.

Original

Quality = 5

Quality = 15

Figure 5: **The output image by Modify-Net**

V. CONCLUSION

We proposed a method to realize higher quality image coding without changing the decoding algorithm. We designed and combined two neural networks: Modify-Net and JPEG-Net. We also proposed an effective training strategy for these networks. Although each network was not completely tuned for image coding, experimental results showed the effectiveness of the proposed method. Important future work is to realize higher quality coding by applying the proposed strategy.

[1] C. Dong, C. C. Loy, K. He, X. Tang, "Image Super-Resolution Using Deep Convolutional Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 38, pp. 295-307, 2016

[2] Chao Dong, Chen Change Loy, Xiaoou Tang, "Accelerating the Super-Resolution Convolutional Neural Network," The European Conference on Computer Vision (ECCV), pp. 391-407, 2016

[3] C. Ledig, L. Theis, F. Huszar, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, W. Shi, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4681-4690, 2017

[4] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, Y. Fu, "Image Super-Resolution Using Very Deep Residual Channel Attention Networks," The European Conference on Computer Vision (ECCV), pp. 286-301 , 2018

[5] J. Kim, J. K. Lee, K. M. Lee, "Accurate Image Super-Resolution Using Very Deep Convolutional Networks," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1646-1654, 2016

[6] M. Afonso , F. Zhang , D. R. Bull, "Video Compression Based on Spatio-Temporal Resolution Adaptation," IEEE Transactions on Circuits and Systems for Video Technology, vol.29, pp.275-280, 2019

[7] D. E. Rumelhart, G. E. Hinton, R. J. Williams, "Learning representation by back-propagating errors," Nature, vol. 323, pp. 533–536, 1986

[8] G. Hudson, A. Léger, B. Niss, I. Sebestyén, "JPEG at 25: Still Going Strong," IEEE MultiMedia, vol. 24, pp. 96-103, 2017

[9] G. J. Sullivan, J. Ohm, W. Han, Member, T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Transactions on Circuits and Systems for Video Technology, vol. 22, pp. 1649-1668, 2012

[10] J. Kim, J. K. Lee, K. M. Lee, "Deeply-recursive convolutional network for image super-resolution," The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1637-1645, 2016