Sandwiched Image Compression: Increasing the resolution and dynamic range of standard codecs

Onur G. Guleryuz, Philip A. Chou, Hugues Hoppe, Danhang Tang, Ruofei Du, Philip Davidson, Sean Fanello
{oguleryuz, philchou, danhangtang, ruofei, pdavidson, seanfa}@google.com, hhoppe@gmail.com
Google Research, USA

***Abstract*—Given a standard image codec, we compress images that may have higher resolution and/or higher bit depth than allowed in the codec's specifications, by sandwiching the standard codec between a neural pre-processor (before the standard encoder) and a neural post-processor (after the standard decoder). Using a differentiable proxy for the the standard codec, we design the neural pre- and post-processors to transport the high resolution (super-resolution, SR) or high bit depth (high dynamic range, HDR) images as lower resolution and lower bit depth images. The neural processors accomplish this with spatially coded modulation, which acts as watermarks to preserve the important image detail during compression. Experiments show that compared to conventional methods of transmitting high resolution or high bit depth through lower resolution or lower bit depth codecs, our sandwich architecture gains ∼9 dB for SR images and ∼3 dB for HDR images at the same rate over large test sets. We also observe significant gains in visual quality.**

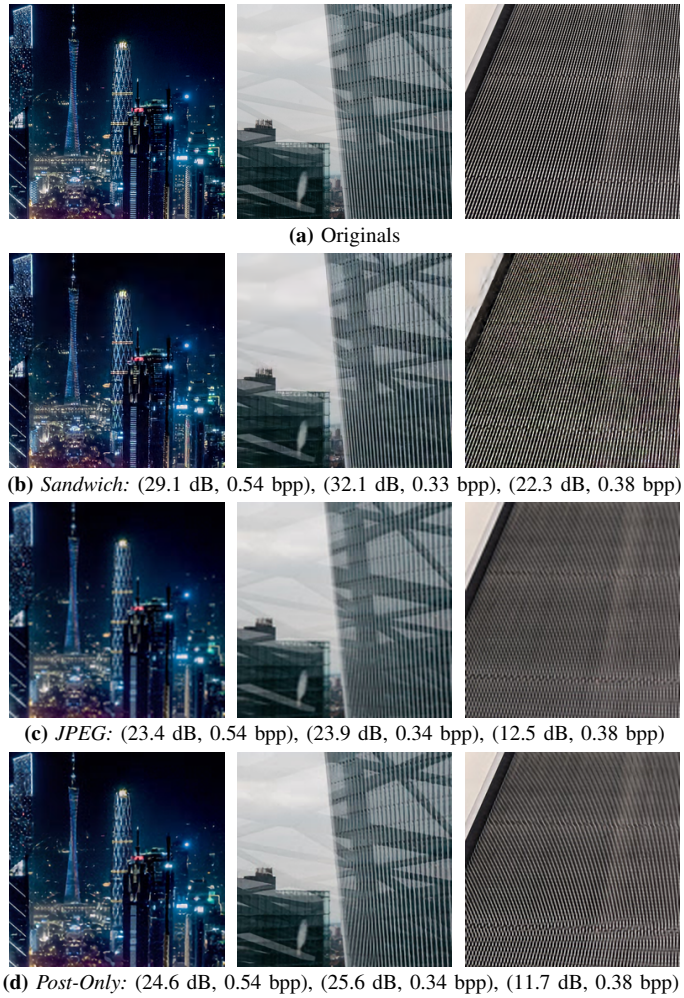
***Index Terms*—deep learning, image compression, nonlinear transform coding, high dynamic range, super-resolution**

I. Introduction

In this paper, we continue our study of the *sandwich architecture* [1], in which a standard image codec is sandwiched between a neural pre-processor and a neural post-processor. In particular, we apply the sandwich architecture to compression of super-resolution and/or high dynamic range images using a standard codec with limited spatial resolution and/or bit depth. In our previous work [1], which introduced the sandwich architecture, we applied the sandwich architecture to compressing 3-channel color images using a 1-channel grayscale codec, and to compressing 3-channel normal map images with nonlinear channel dependencies. We further outlined a research agenda including application of the sandwich architecture to other image types such as HDR. This paper furthers that agenda.

Works prior to [1] either pair a neural pre-processor with a standard codec (where the pre-processor performs, e.g., denoising [2]–[4]) or pair a standard codec with a neural post-processor (where the post-processor performs deblocking or other enhancements [5]–[7]). However, few works prior to ours sandwich a standard codec between two neural processors. Those that do (e.g., [8]–[11]), like existing non-neural solutions (e.g., the "frame super-resolution" coding tool within VP9/11, or [12]), do so in such a way that the pre- and post-processors may be used independently — thus do not take full advantage of the communication available between pre- and post-processors — or require side information [13], [14].

The advantage to having both a neural pre-processor and a neural post-processor is that they can work in tandem to convert source images to and from images of latent codes. The images of latent codes can be better suited than the source images themselves for surviving compression with the standard codec, in a rate-distortion sense, especially if the



(a) Originals

(b) *Sandwich:* (29.1 dB, 0.54 bpp), (32.1 dB, 0.33 bpp), (22.3 dB, 0.38 bpp)

(c) *JPEG:* (23.4 dB, 0.54 bpp), (23.9 dB, 0.34 bpp), (12.5 dB, 0.38 bpp)

(d) *Post-Only:* (24.6 dB, 0.54 bpp), (25.6 dB, 0.34 bpp), (11.7 dB, 0.38 bpp)

Fig. 1. Super-resolution sandwich of a low-res JPEG codec: Original 256×256 source images and reconstructions by sandwich, JPEG with linear upsampling, and JPEG with neural post-processing respectively. Observe the substantial improvements obtained by the sandwiched codec over JPEG and neural post-processing: Detail is retained in the city visage, aliasing is reduced on the building facade and the texture. All with substantial dB improvements (+4.5 dB, +6.5 dB, +10.6 dB over neural post-processing) at the same rate. Note in particular for the last column that while the sandwiched codec transports the detail accurately, neural post-processing produces a picture even less accurate than JPEG by guessing the wrong details. The sandwiched codec is clearly a superior architecture.

standard codec is not designed for the source image format or type. Figures 1 and 2 illustrate the results of a scenario where a low-resolution codec that transports 128×128 images is sandwiched using a jointly trained neural pre-processor and neural post-processor pair. The goal is to obtain high quality 256×256 reconstructions. As illustrated, this codec performs substantially better in a rate-distortion sense not only compared to the low-resolution codec equipped with a linear upsampler but also to one equipped with a neural post-processor. This is because the sandwich architecture transports images watermarked with spatial modulation patterns (Figure 3) such that the modulation patterns are efficiently compressible with the standards codec, and such that the decompressed modulation patterns can be decoded by the post-processor into a high-quality picture.

In the present paper, using a methodology similar to that of [1], we apply the sandwich architecture to squeeze super resolution (SR) content through codecs at a standard or lower resolution (LR) and to squeeze 16-bit high dynamic range (HDR) content through codecs with 8-bit standard or low dynamic range (LDR). In both cases, the neural pre-/post-processors learn to map/unmap the source images to/from latent images containing neural codes that best preserve (in a rate-distortion sense) the source image details when compressed with the given codec.

Of course, it is possible to eliminate the standard codec altogether, and replace it by simple uniform scalar quantization and entropy coding of the latent codes at the bottleneck of a neural network in an autoencoder configuration. This is the essence of nonlinear transform coding (NTC), which is the state of the art in end-to-end learned image and video compression [15]–[20]. Presumably, end-to-end learned systems can be trained to compress classes of images with arbitrary numbers of channels, spatial resolution, bit depth, distribution, and loss. However, to our knowledge only a few end-to-end learned systems have been able to outperform the best standard codecs in PSNR at a given bit rate, and these systems are computationally complex [21]. Hence a key motivation for building around existing standard codecs is to leverage the existing compression ecosystem, particularly existing hardware and existing compression-aware networking/routing, which may be able to perform the heavy lifting.

Given the desire to sandwich a standard codec between neural pre- and post-processors, the crucial problem is to differentiate through the standard codec when training the neural pre- and post-processors using gradient descent to minimize the loss. Thus a primary problem is to develop a differentiable approximation to the standard codec, called a *proxy* for the codec. As in [1], we use a proxy modeled after JPEG, though in this paper we show that this relatively simply proxy is sufficient for training pre- and post-processors that can be used with more complicated codecs such as HEIC.

At the highest quality levels where the standard codecs saturate, our results show that to compress a large variety of high resolution images using a low resolution HEIC or JPEG codec, the sandwich architecture has $\sim$9 dB gain over bicubic filtering and downsampling as the pre-processor, and Lanczos upsampling as the post-processor. If neural processing is used as the post-processor, the gain is still $\sim$7 dB (Figure 7). Furthermore, our results show that to compress a large variety of 16-bit HDR images with 8-bit HEIC (JPEG), the sandwich architecture has $\sim$5 dB ($\sim$6 dB) gain over nearest-neighbor bit truncation as the pre-processor, and midpoint reconstruction as the post-processor. If a neural post-processor is used, the gain is still up to $\sim$4 dB (Figure 8). These gains are made possible because the neural pre-processor is able to construct neural codes to robustly transmit the needed image detail, which the neural post-processor can reconstruct, given sufficient training.

(a) Originals

(b) *Sandwich:* (31.4 dB, 0.58 bpp), (27.3 dB, 0.70 bpp), (28.2 dB, 0.57 bpp)

(c) *JPEG:* (24.7 dB, 0.60 bpp), (22.4 dB, 0.71 bpp), (21.7 dB, 0.58 bpp)

(d) *Post-Only:* (26.3 dB, 0.60 bpp), (23.2 dB, 0.71 bpp), (23.1 dB, 0.58 bpp)

Fig. 2. Super-resolution sandwich: Original 256×256 source images and reconstructions by sandwich, JPEG with linear upsampling, and JPEG enhanced with neural post-processing respectively. With the sandwich visually relevant ornaments/textures are preserved, images are sharper in a way that matches the originals, and text in the scene is easier to read. Beyond significantly improved visual quality the sandwich obtains substantial dB improvements (+5.1 dB, +4.1 dB, +5.1 dB over neural post-processing) at the same rate.

Section II reviews the sandwich architecture, including the differentiable approximation, and shows how to apply the sandwich architecture to SR and HDR imagery. Section III presents experimental results. Section IV concludes the paper.

Fig. 3. 128×128 reconstructed bottleneck images for the super-resolution sandwich results in Figures 1 and 2 [enlarged for clarity]. Observe that while the bottlenecks appear aliased, noisy etc., the sandwich post-processor has correctly demodulated this noise in the final pictures.

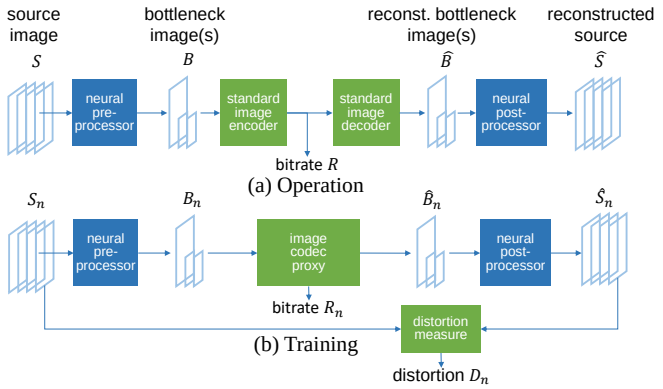


Fig. 4. Sandwich architecture in (a) operation and (b) training.

II. The Sandwich Architecture for SR and HDR

The generic sandwich architecture in *operation* is shown in Fig. 4(a). An *original source image* S with one or more full-resolution channels is mapped by a neural preprocessor into one or more channels of latent codes. Each channel of latent codes may be full resolution or reduced resolution. The channels of latent codes are grouped into one or more *bottleneck images* B suitable for consumption by a standard image codec. The bottleneck images are compressed by the standard image encoder into a bit string of length R bits. The bit string is decompressed by the corresponding decoder into *reconstructed bottleneck images* $\hat{B}$, incurring distortion $d(B, \hat{B})$. The channels of the reconstructed bottleneck images are then mapped by a neural postprocessor into a *reconstructed source image* $\hat{S}$.

The neural pre- and post-processors are shown in Fig. 5. In our work, each is an MLP in parallel with a U-Net [22]. Both branches operate at full resolution but are resampled as necessary to meet the resolution requirements of the codec. The MLPs and U-Nets have the same structure in both the pre- and post-processors. We refer the reader to [1] for the specific hyper-parameters of our networks.

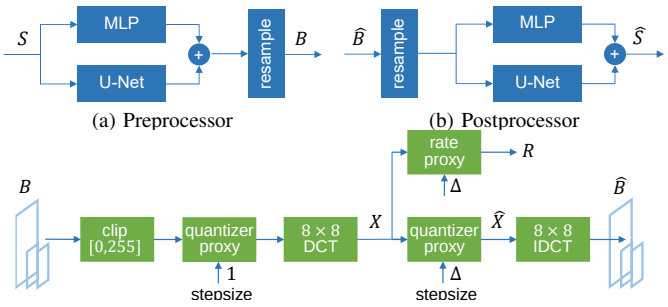


Fig. 6. Image codec proxy.

The generic sandwich architecture in *training* is shown in Fig. 4(b). On a training set of full-resolution images $\{S_n\}_{n=1}^N$, the parameters of the neural pre- and post-processors minimize the loss function $L = D + \lambda R$, where $D = (1/N)\sum_n d(S_n, \hat{S}_n)$ is the average distortion, $R = (1/N)\sum_n R_n$ is the average rate, and $\lambda > 0$ is a Lagrange multiplier chosen to balance rate and distortion. Minimization of L is performed by back-propagating the gradient of L with respect to the parameters. For the purpose of computing these gradients, the standard codec must be replaced by a *codec proxy* that is differentiable.

The differentiable codec proxy is shown in Fig. 6. The proxy is modeled after JPEG, but as we show in this paper suffices to represent more complex codecs such as HEIC in our experiments. The codec proxy clips all values in the real-valued bottleneck images to a fixed dynamic range, such as $[0, 255]$; quantizes them to integers; performs the DCT on each 8×8 block; quantizes the DCT coefficients to learnable stepsize Δ; estimates the bit rate of the quantized coefficients; and performs the inverse DCT on each block.

Within the codec proxy, the quantizer is the differentiable *quantizer proxy*, $Q(X) = X + W$, where $W = \text{stop\_gradient}(\text{round}(X/\Delta) - (X/\Delta))\Delta$ is the true quantization error and $\text{stop\_gradient}(\cdot)$ is the identity but stops the gradient of its output from being back-propagated to its argument [23]. Further, the bit rate is estimated by a differentiable *rate proxy*, where the number of bits to compress bottleneck image B to stepsize Δ is estimated to be

$$R(B) = a \sum_{k,i} \log\left(1 + \left|x_i^{(k)}\right| / \Delta\right), \tag{1}$$

where $x_i^{(k)}$ is the ith coefficient of the kth block of DCT coefficients, and a is chosen such that $R(B)$ is the rate at which JPEG codes the image B with uniform stepsize Δ.

The generic sandwich architecture is applied to the super resolution (SR) and high dynamic range (HDR) problems as follows:

In the SR problem, the RGB $H \times W \times 3$ source images have source bit depth $d = 8$. Thus they have the standard dynamic range, $[0, 255]$. However, the bottleneck images have lower spatial resolution, $H/2 \times W/2 \times 3$. In our work, the resampler in the pre-processor comprises bicubic filtering and 2x downsampling; the resampler in the post-processor comprises Lanczos3 interpolation of the half-resolution images back to full-resolution.

In the HDR problem, the source images have dynamic range $[0, 2^d - 1]$, where d is the source bit depth. The bottleneck images have dimensions that match the source images: $H \times W \times 3$. Thus no resampling is required. However, the bottleneck images are restricted to the standard dynamic range $[0, 255]$. Since the codec proxy does not pass any information outside of this range, the pre-processor produces images in this range.

In both the SR and HDR problems, the sandwiched codecs operate in 4:4:4 mode without a color transform. However, in the following experimental results, the baseline (non-sandwiched) codecs that we compare to use the RGB $\leftrightarrow$ YUV transform when it is beneficial for them in an R-D sense: In the SR scenario they use the color transform; in HDR they encode RGB directly.

III. Experimental Results

For all results, we report the MSE distortion between S and $\hat{S}$ using the RGB PSNR,

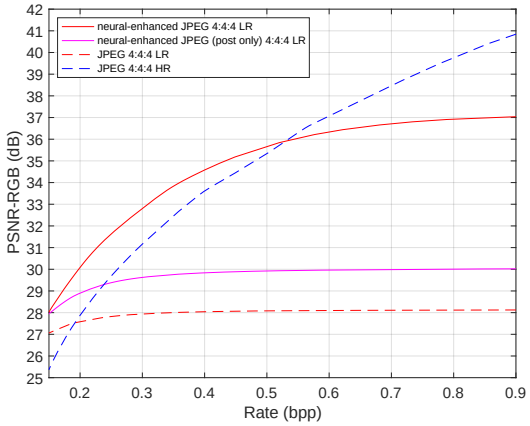
$$\text{RGB PSNR} = 10 \log_{10} \left(\left(2^d - 1\right)^2 (3HW) / \left\| S - \hat{S} \right\|^2 \right). \quad (2)$$

While it is possible to combine the HDR and SR problems, here we study them separately.

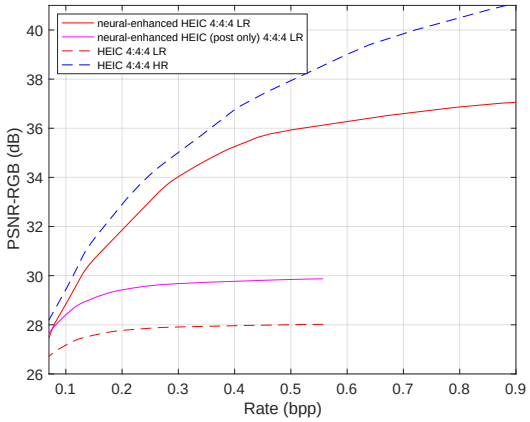
A. Super-Resolution

We used different subsets of the CLIC dataset [24] to train and evaluate the networks. Shown results are over 500 evaluation images (256×256) randomly cropped from the eval subset of the dataset. Figures 1, 2, and 3 show qualitative and objective results on a set of images. We compare with a post-processor-only network consisting of a U-Net identical to the sandwich neural post-processor but trained for post-processing only. The substantial improvement obtained by the sandwich over the post-processor only network clearly points to the importance of the neural pre-processor and the joint training of the networks. Figure 7 shows the combined rate-distortion performance over the entire eval set using (a) JPEG and (b) HEIC as the underlying codec. The networks are identical between codecs, with no retraining for HEIC. The substantial improvements of the sandwiched architecture are clearly observed.

Table I compares our SR sandwich to the closely related but independently developed solution of [9], in which their CNN-RD also surrounds a standard codec with neural pre- and post-processors using 2x down- and up-sampling. However their networks' formulation and training regiment prohibits them from learning to communicate the neural codes needed to carry good SR information. (Their post-processor is trained first to super-resolve a low-pass image; then their pre-processor is trained to minimize $D + \lambda R$ with the fixed post-processor. This misses the main advantage of having neural pre- and post-processors.) Table I shows that we have significantly higher gains in PSNR-Y (dB) relative to the same standard codec (JPEG) on the Div2k validation image 0873 [25]. Indeed, though not shown in the table, their solution saturates and



(a) JPEG



(b) HEIC

Fig. 7. RD performance of the super-resolution sandwich.

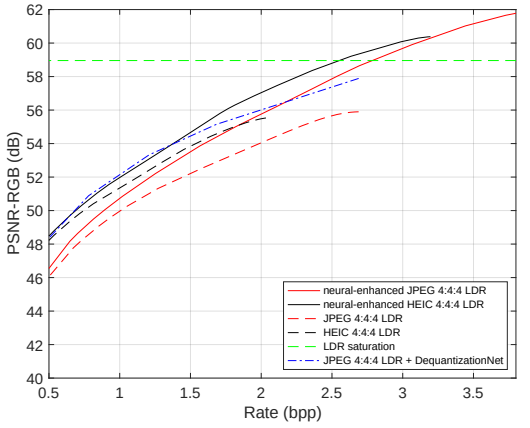


Fig. 8. RD performance of the HDR sandwich.

begins to under-perform the standard codec above 0.8 bpp ($\sim$ 30 dB); ours out-performs the standard codec until about 2.0 bpp ($\sim$ 37 dB).

bpp	0.2	0.3	0.4	0.5	0.6	0.7	0.8
CNN-RD [9]	1.58	1.09	0.67	0.55	0.33	0.18	0.15
SR sandwich	**1.59**	**1.49**	**1.42**	**1.49**	**1.49**	**1.46**	**1.69**

TABLE I
GAIN IN PSNR-Y (DB) OVER JPEG ON DIV2K VALIDATION IMAGE 0873.

B. High Dynamic Range

For HDR simulations, we use the HDR+ dataset [26]. Original images are 16-bit, standard codecs are 8-bit. Figure 8 illustrates the performance of the sandwich architecture in comparison to standard codecs as well as to JPEG post-processed with the state-of-the-art Dequantization-Net [27] (trained on the same dataset). The maximum PSNR one can obtain by losslessly encoding the most significant 8-bits is illustrated as LDR saturation. The standard codecs alone, or with only a post processor [27] all saturate at that level. Observe that the sandwiched codecs rise above the saturation line, highlighting the importance of the preprocessor. Unfortunately the software implementing the standard codecs precluded the transmission of higher rates. Neither our JPEG nor HEIC implementation was able to go beyond $\sim$3 bpp on average. For all R-D curves the highest rate point is where the software cuts off. Using codec implementations accomplishing higher rates, the gains of the sandwich are expected to increase further.

IV. Conclusion

The proposed sandwich architecture extends the use of standard codecs to resolutions and bit-depths beyond regimes allowed by the specification of the standard codec. As the results of this paper show, the sandwich architecture has the promise of leveraging standard hardware and software codec implementations while generating significant quality improvements. Future work will study complexity reductions of the pre- and post-processors, and will also explore applications of the sandwich architecture to adapting standard codecs to alternative distortion measures and non-standard image types such as multispectral images in remote sensing, material maps in graphics, medical images, depth images, motion fields, and multiview images. These applications are made possible by the pre- and post-processors learning to communicate with each other by sending neural codes as images that can survive heavy compression by an ordinary image codec.

Acknowledgment

The authors thank Jonathan Taylor for helpful discussions.

References

[1] O. G. Guleryuz, P. A. Chou, H. Hoppe, D. Tang, R. Du, P. Davidson, and S. Fanello, "Sandwiched Image Compression: Wrapping Neural Networks Around a Standard Codec," in *2021 IEEE Int'l Conf. Image Processing (ICIP)*, 2021.

[2] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a Gaussian Denoiser: Residual Learning of Deep CNN for Image Denoising," *IEEE Trans. Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[3] C. Tian, L. Fei, W. Zheng, Y. Xu, W. Zuo, and C.-W. Lin, "Deep Learning on Image Denoising: an Overview," *Neural Networks*, vol. 131, pp. 251 – 275, 2020.

[4] Huy Vu, Gene Cheung, and Yonina C. Eldar, "Unrolling of Deep Graph Total Variation for Image Denoising," in *IEEE Int'l Conf. Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 2050–2054.

[5] P. Svoboda, Michal Hradis, David Barina, and P. Zemcík, "Compression Artifacts Removal Using Convolutional Neural Networks," *ArXiv*, vol. abs/1605.00366, 2016.

[6] T. Kim, H. Lee, H. Son, and S. Lee, "SF-CNN: a Fast Compression Artifacts Removal Via Spatial-to-Frequency Convolutional Neural Networks," in *IEEE Int'l Conf. Image Processing (ICIP)*, 2019, pp. 3606–3610.

[7] J. Niu, "End-to-End JPEG Decoding and Artifacts Suppression Using Heterogeneous Residual Convolutional Neural Network," *Int'l Joint Conf. Neural Networks (IJCNN)*, pp. 1–8, 2020.

[8] Y. Li, D. Liu, H. Li, L. Li, Z. Li, and F. Wu, "Learning a Convolutional Neural Network for Image Compact-Resolution," *IEEE Trans. Image Processing*, vol. 28, no. 3, pp. 1092–1107, 2019.

[9] P. Eusébio, J. Ascenso, and F. Pereira, "Optimizing an Image Coding Framework With Deep Learning-Based Pre- and Post-Processing," in *European Signal Processing Conf. (EUSIPCO)*, 2021, pp. 506–510.

[10] K. Qiu, L. Yu, and D. Li, "Codec-Simulation Network for Joint Optimization of Video Coding With Pre- and Post-Processing," *IEEE Open J. Circuits and Systems*, vol. 2, pp. 648–659, 2021.

[11] Y. Andreopoulos, "Neural Pre and Post-Processing for Video Encoding With AVC, VP9, and AV1," in *AOM Research Symp*, 2022.

[12] C.A. Segall and A.K. Katsaggelos, "Pre- and Post-Processing Algorithms for Compressed Video Enhancement," in *Asilomar Conf. Signals, Systems and Computers*, 2000, vol. 2, pp. 1369–1373 vol.2.

[13] S. Battista, G. Meardi, S. Ferrara, L. Ciccarelli, F. Maurer, M. Conti, and S. Orcioni, "Overview of the Low Complexity Enhancement Video Coding (LCEVC) Standard," *IEEE Trans. Circuits and Systems for Video Technology*, pp. 1–1, 2022.

[14] A. Artusi, R. Mantiuk, T. Richter, P. Hanhart, P. Korshunov, M. Agostinelli, A. Ten, and T. Ebrahimi, "Overview and Evaluation of the JPEG XT HDR Image Compression Standard," *J. Real-Time Image Processing*, vol. 16, 04 2019.

[15] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-End Optimized Image Compression," in *Int'l Conf. Learning Representations (ICLR)*, 2017.

[16] Johannes Ballé, "Efficient Nonlinear Transforms for Lossy Image Compression," in *Picture Coding Symp. (PCS)*, 2018, pp. 248–252.

[17] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational Image Compression with a Scale Hyperprior," in *Int'l Conf. Learning Representations (ICLR)*, 2018.

[18] D. Minnen, J. Ballé, and G. Toderici, "Joint Autoregressive and Hierarchical Priors for Learned Image Compression," in *Advances in Neural Information Processing Systems 31*, 2018.

[19] J. Balle, P. A. Chou, D. Minnen, S. Singh, N. Johnston, E. Agustsson, S. J. Hwang, and G. Toderici, "Nonlinear Transform Coding," *IEEE J. Selected Topics in Signal Processing*, pp. 1–1, 2020.

[20] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson, "High-Fidelity Generative Image Compression," in *Advances in Neural Information Processing Systems*, 2020, vol. 33, pp. 11913–11924.

[21] Z. Guo, Z. Zhang, R. Feng, and Z. Chen, "Causal Contextual Prediction for Learned Image Compression," *IEEE Trans. Circuits and Systems for Video Technology*, pp. 1–1, 2021.

[22] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015*, Cham, 2015, pp. 234–241.

[23] "TensorFlow API: tf.stopgradient," https://www.tensorflow.org/api_docs/python/tf/stop_gradient, 2022.

[24] "Dataset for the Challenge on Learned Image Compression 2020," http://www.tensorflow.org/datasets/catalog/clic, 2022.

[25] E. Agustsson and R. Timofte, "NTIRE 2017 Challenge on Single Image Super-Resolution: Dataset and Study," in *IEEE Conf. Computer Vision and Pattern Recognition (CVPR) Workshops*, July 2017.

[26] S. W. Hasinoff, D. Sharlet, R. Geiss, A. Adams, J. Barron, F. Kainz, J. Chen, and M. Levoy, "Burst Photography for High Dynamic Range and Low-Light Imaging on Mobile Cameras," *ACM Trans. Graphics*, Nov. 2016.

[27] Y.-L. Liu, W.-S. Lai, Y.-S. Chen, Y.-L. Kao, M.-H. Yang, Y.-Y. Chuang, and J.-B. Huang, "Single-Image HDR Reconstruction by Learning to Reverse the Camera Pipeline," in *IEEE Conf. Computer Vision and Pattern Recognition*, 2020.