# 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*

Google Research

{oguleryuz, philchou, hopp, danhangtang, ruofei, pdavidson, seanfa}@google.com

## ABSTRACT

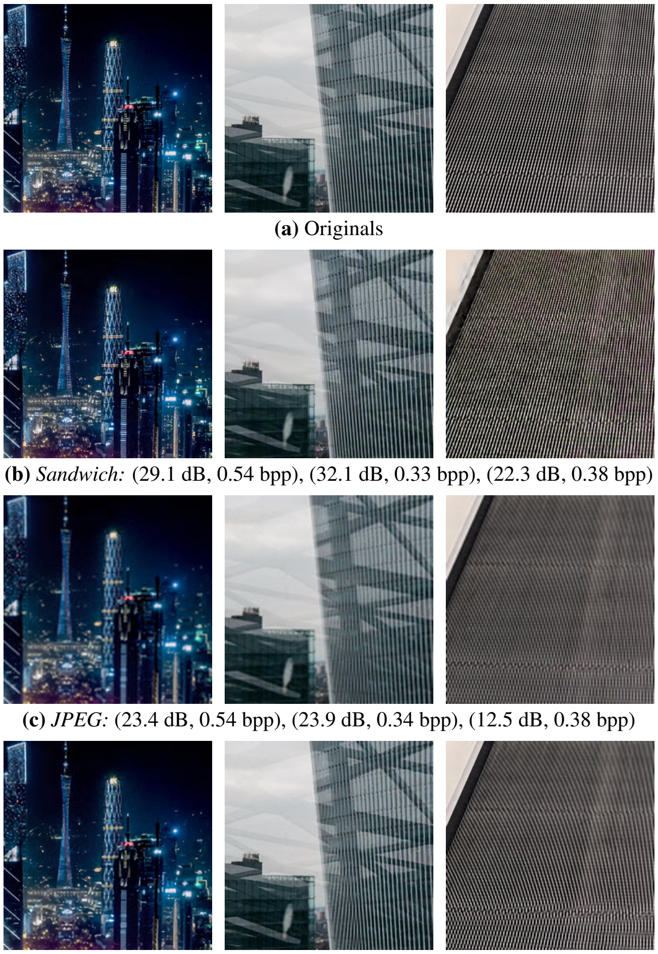
Given a standard image codec, we compress images that may have higher resolution and/or higher bit depth than allowed in the codec's specifications, by sandwiching the standard codec between a neural pre-processor (before the standard encoder) and a neural post-processor (after the standard decoder). 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

## 1. 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.

Works prior to [1] have either paired a neural pre-processor with a standard codec (where the pre-processor for e.g., performs denoising [2, 3, 4]) or paired a standard codec with a neural post-processor (where the post-processor performs deblocking or other enhancements [5, 6, 7]). However,



**(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 $256 \times 256$ 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. The sandwiched codec is clearly a superior architecture.

The authors thank Jonathan Taylor for helpful discussions.

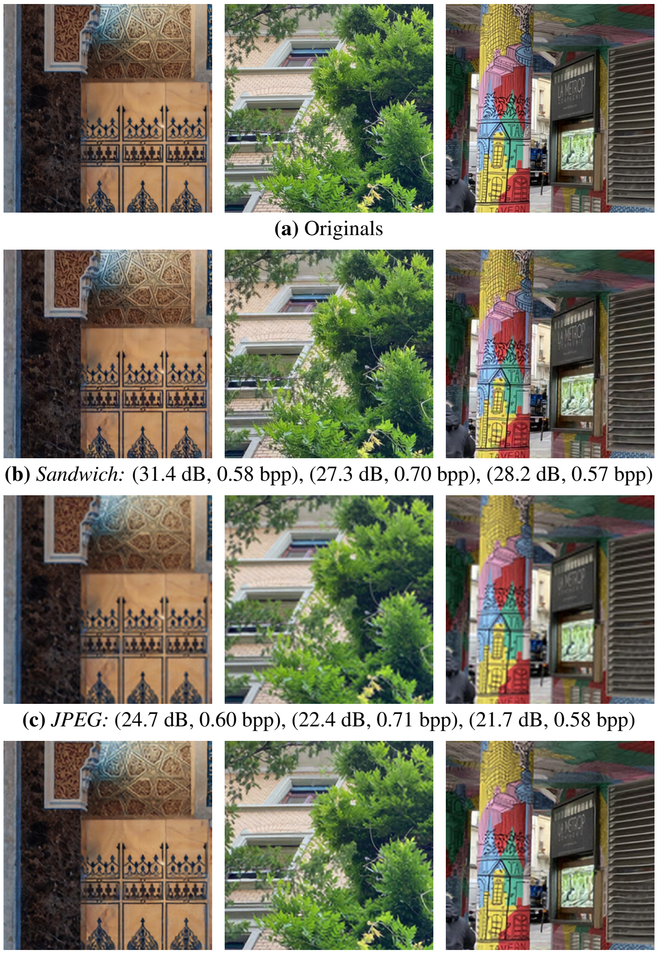
to our knowledge no works prior to ours have sandwiched a standard codec between two neural processors.

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 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 $256 \times 256$ 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 [8]–[16]. 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 [17]. 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 standard codecs such as HEIC.


**(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.

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 ∼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 ∼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 ∼5 dB (∼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 ∼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 is able to reconstruct, given sufficient training.

Section 2 reviews the sandwich architecture, including the differentiable approximation. Section 3 shows how to apply the sandwich architecture to SR and HDR imagery and discusses associated results. Section 4 concludes the paper.

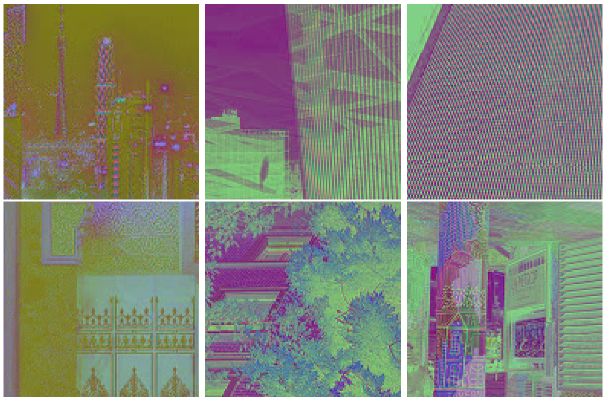


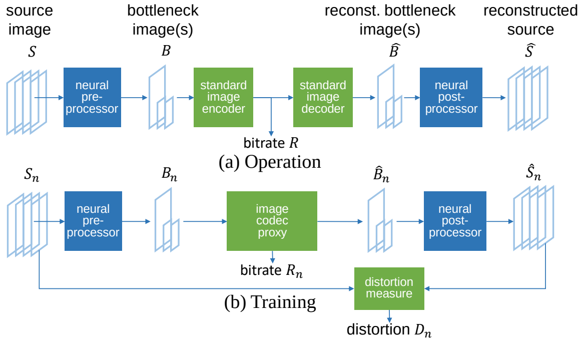
**Fig. 3:** $128 \times 128$ reconstructed bottleneck images for the super-resolution sandwich results in Figures 1 and 2 [enlarged for clarity]. Observe that while the bottlenecks appear aliased, noisy etc., the sandwich post-processor has correctly demodulated this noise in the final pictures.

## 2. THE SANDWICH ARCHITECTURE

The 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 [18]. Both branches operate at full resolution but are resampled as necessary to meet the resolution requirements of the codec.

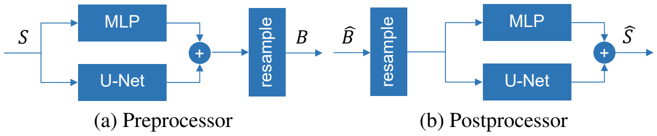
The sandwich architecture in *training* is shown in Fig.4(b). On a training set of full-resolution images $\{S_n\}_{n=1}^N$, the parameters of the neural pre- and post-processors minimize the loss function $L = D + \lambda R$, where $D = (1/N)\sum_n d(S_n, \hat{S}_n)$ is the average distortion, $R = (1/N)\sum_n R_n$ 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.



(a) Operation

(b) Training

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



(a) Preprocessor

(b) Postprocessor

**Fig. 5:** Neural preprocessor and postprocessor.

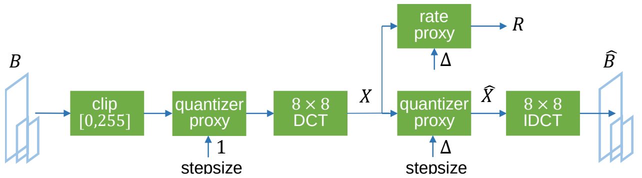


**Fig. 6:** Image codec proxy.

The differentiable codec proxy is shown in Fig. 6. The proxy is modeled after JPEG, but 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 $8 \times 8$ block; quantizes the DCT coefficients to learnable stepsize $\Delta$; 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*, $Q(X) = X + \text{stop\_gradient}(W)$, where $W = \Delta\text{round}(X/\Delta) - X$ is the true quantization error and $\text{stop\_gradient}(W)$ equals $W$ but stops the gradient of $W$ from being back-propagated [19]. Further, the bit rate is estimated by a differentiable *rate proxy*, where the number of bits to compress bottleneck image $B$ to stepsize $\Delta$ 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 $i$th coefficient of the $k$th 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 $\Delta$.

## 3. EXPERIMENTAL RESULTS

We now apply the sandwich architecture to super resolution (SR) and high dynamic range (HDR)[1]. The source images are RGB and have dimensions of $H \times W \times 3$. The standard codecs operate in 4:4:4 mode. The sandwiched codec does not use a color transform. The compared to codecs use the RGB $\leftrightarrow$ YUV transform when it is beneficial for them in an R-D sense: In the SR scenario standard codecs use the color transform, in HDR they encode RGB directly.

In the SR problem, the source images have source bit depth $d = 8$. Bottleneck images have lower spatial resolution, $H/2 \times W/2 \times 3$. In the HDR problem, the source images have dynamic range $\left[0, 2^d - 1\right]$, where $d$ is the source bit depth. The bottleneck images have dimensions that match the source images: $H \times W \times 3$. However, the dynamic range of the bottleneck images is $[0, 255]$, since the codec proxy does not pass any information outside of this range and hence encourages the pre-processor to produce images in this range.

While it is possible to combine the HDR and SR problems, here we study them separately. We measure the distortion between $S$ and $\hat{S}$ as the RGB PSNR,

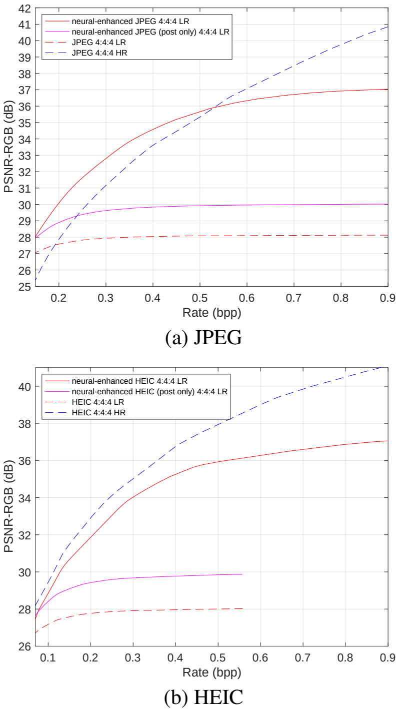
$$\text{PSNR} = 10 \log_{10} \left( \left(2^d - 1\right)^2 (3HW) / \left\| S - \hat{S} \right\|^2 \right). \quad (2)$$

### 3.1. Super-Resolution

We used the CLIC dataset [20] to train and evaluate the networks. Shown results are over 500 evaluation images $(256 \times 256)$ randomly cropped from the eval portion 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 JPEG and 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.

### 3.2. High Dynamic Range

For HDR simulations, we use the HDR+ dataset [21]. 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 [22] (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 [22] all saturate at that level.



(a) JPEG

(b) HEIC

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

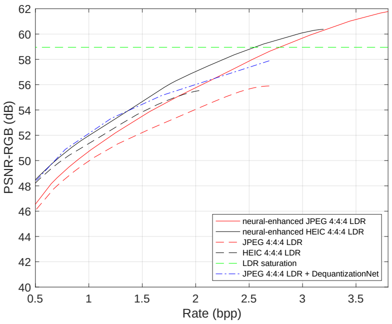


**Fig. 8:** RD performance of the HDR sandwich.

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[2].

## 4. 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 architecture retains standard, hardware, and network layer compatibility while generating significant quality improvements.

[1] The reader is referred to [1] for utilized network parameters. Source code will be released with the presentation version of this paper.

[2] For the presentation version of this paper, we will look for codec implementations that allow higher rate points.

## 5. 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 International Conference on 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 Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[3] Chunwei Tian, Lunke Fei, Wenxian Zheng, Yong Xu, Wangmeng Zuo, and Chia-Wen 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 *2021 IEEE International Conference on 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 *2019 IEEE International Conference on Image Processing (ICIP)*, 2019, pp. 3606–3610.

[7] Jun Niu, "End-to-End JPEG Decoding and Artifacts Suppression Using Heterogeneous Residual Convolutional Neural Network," *2020 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020.

[8] G. Toderici, S. M. O'Malley, S. J. Hwang, D. Vincent, D. Minnen, S. Baluja, M. Covell, and R. Sukthankar, "Variable Rate Image Compression With Recurrent Neural Networks," in *4th Int. Conf. on Learning Representations (ICLR)*, 2016.

[9] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-End Optimization of Nonlinear Transform Codes for Perceptual Quality," in *2016 Picture Coding Symposium (PCS)*, 2016, pp. 1–5.

[10] G. Toderici, D. Vincent, N. Johnston, S. J. Hwang, D. Minnen, J. Shor, and M. Covell, "Full Resolution Image Compression With Recurrent Neural Networks," in *2017 IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[11] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-End Optimized Image Compression," in *5th Int. Conf. on Learning Representations (ICLR)*, 2017.

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

[13] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational Image Compression With a Scale Hyperprior," in *6th Int. Conf. on Learning Representations (ICLR)*, 2018.

[14] 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.

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

[16] 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.

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

[18] 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.

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

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

[21] Samuel W. Hasinoff, Dillon Sharlet, Ryan Geiss, Andrew Adams, Jonathan Barron, Florian Kainz, Jiawen Chen, and Marc Levoy, "Burst Photography for High Dynamic Range and Low-Light Imaging on Mobile Cameras," *ACM Transactions on Graphics*, Nov. 2016.

[22] Yu-Lun Liu, Wei-Sheng Lai, Yu-Sheng Chen, Yi-Lung Kao, Ming-Hsuan Yang, Yung-Yu Chuang, and Jia-Bin Huang, "Single-Image HDR Reconstruction by Learning to Reverse the Camera Pipeline," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2020.