End-to-end optimization of nonlinear transform codes for perceptual quality

Johannes Ballé, Valero Laparra, Eero P. Simoncelli
Center for Neural Science and Courant Institute of Mathematical Sciences
New York University, New York, NY, USA
{johannes.balle,valero,eero.simoncelli}@nyu.edu

***Abstract*—We introduce a general framework for end-to-end optimization of the rate–distortion performance of nonlinear transform codes assuming scalar quantization. The proposed framework can be used to optimize any differentiable pair of analysis and synthesis transforms in combination with any differentiable perceptual metric. As an example, we optimize a code built from a linear transform followed by a form of multi-dimensional gain control. Distortion is measured with a state-of-the-art perceptual metric. The code, optimized over a large database of images, offers substantial improvements in bitrate and perceptual appearance over fixed (DCT) codes, as well as over linear transform codes optimized for mean squared error.**

I. Introduction

Transform coding [1] is one of the most successful concepts of signal processing. Virtually all modern image and video compression standards operate by applying an invertible transformation to the signal, quantizing the transformed data to achieve a compressed representation, and inverting the transform to recover an approximation of the original signal.

Generally these transforms have been linear. Non-Gaussian/non-linear aspects of signal statistics are typically handled by augmenting the linear system with carefully selected non-linearities (for example, non-uniform quantization via companding, prediction in the case of hybrid compression, etc.). Deciding which combination of these operations, also known as "coding tools," are ultimately useful is a cumbersome process. The operations are generally studied and optimized individually, with different objectives, and each possible combination of coding tools must then be empirically validated in terms of average code rate and distortion.

This is reminiscent of the state of affairs in the field of object and pattern recognition about a decade ago. As in the compression community, most solutions were built by manually combining a sequence of individually designed and optimized processing stages. In recent years, that field has seen remarkable performance gains [2], which have arisen primarily because of end-to-end system optimization. Specifically, researchers have chosen architectures that consist of a cascade of transformations that are differentiable with respect to their parameters, and then used modern optimization tools to jointly optimize the system over large databases of images.

Here, we take a step toward using such end-to-end optimization in the context of compression. We develop an optimization framework for nonlinear transform coding (fig. 1), which generalizes the traditional transform coding paradigm. The image vector $\boldsymbol{x}$ is transformed to a code domain vector using an arbitrary differentiable function $\boldsymbol{y} = g_a(\boldsymbol{x}; \boldsymbol{\theta})$ (the analysis transform), parameterized by a vector $\boldsymbol{\theta}$ (containing linear filter coefficients, for example). The transformed $\boldsymbol{y}$ is subjected to scalar quantization, yielding the vector of integer quantization indices $\boldsymbol{q}$ and a reconstructed vector $\hat{\boldsymbol{y}}$. The latter is then non-linearly transformed back to the signal domain to obtain the reconstructed image $\hat{\boldsymbol{x}} = g_s(\hat{\boldsymbol{y}}; \boldsymbol{\phi})$, where the synthesis transform g_s is parameterized by vector $\boldsymbol{\phi}$.

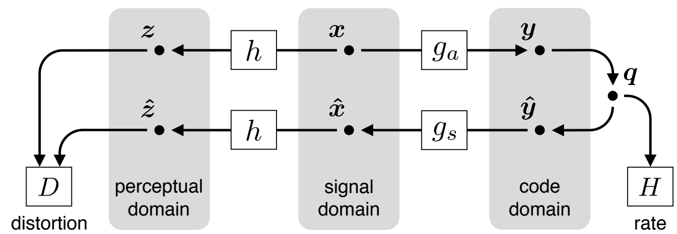


Fig. 1. Nonlinear transform coding optimization framework. See text.

The code rate is assessed by measuring the entropy, H, of the discrete probability distribution $P_{\boldsymbol{q}}$ of the quantization indices over an ensemble of images. Traditionally, the distortion is assessed directly in the image domain by taking the squared Euclidean norm of the difference between $\boldsymbol{x}$ and $\hat{\boldsymbol{x}}$ (or equivalently, the peak signal-to-noise ratio, PSNR). However, it is well known that PSNR is not well-aligned with human perception [3]. To alleviate this problem, we allow an additional "perceptual" transform of both vectors $\boldsymbol{z} = h(\boldsymbol{x})$ and $\hat{\boldsymbol{z}} = h(\hat{\boldsymbol{x}})$, on which we then compute distortion using a norm. With an appropriate transform h, this can provide a better approximation of subjective visual distortion [4].

II. Optimization framework

In the transform coding framework given above, we seek to adjust the analysis and synthesis transforms g_a and g_s so as to minimize the rate–distortion functional:

$$L[g_a, g_s] = H[P_{\boldsymbol{q}}] + \lambda \, \mathbb{E} \, \|\boldsymbol{z} - \hat{\boldsymbol{z}}\|. \tag{1}$$

The first term denotes the discrete entropy of the vector of quantization indices $\boldsymbol{q}$. The second term measures the distortion between the reference image $\boldsymbol{z}$ and its reconstruction $\hat{\boldsymbol{z}}$ in a perceptual representation. Note that both terms are expectations taken over an ensemble of images.

We wish to minimize this objective. Standard optimization methods (e.g., gradient descent) require differentiable functions, but both terms in the functional depend on the quantized values in $\boldsymbol{q}$, and the derivative of a quantizer only takes zero or infinite values. In what follows, we introduce dithered quantization to make the objective differentiable with respect to the parameters of g_a and g_s.

A uniform scalar quantizer applies a piecewise constant function to each of the elements of $\boldsymbol{y}$: $\hat{y}_i = \mathrm{round}(y_i)$.[1] In a dithered quantizer, the quantization bins are shifted randomly with an i.i.d. noise source $\boldsymbol{u}$:

$$\hat{y}_i = \mathrm{round}(y_i + u_i) - u_i. \tag{2}$$

With an appropriate noise source, the "quantization noise" $\Delta y_i = \hat{y}_i - y_i$ can be made independent of y_i [5], which allows us to rewrite the quantization as an addition of independent random variables (here, with uniform noise):

$$\hat{y}_i = y_i + \Delta y_i \quad \text{with } \Delta y_i \sim \mathrm{rect}, \tag{3}$$

where rect denotes the uniform density between $-\frac{1}{2}$ and $\frac{1}{2}$. With dithering, the discrete entropy of a quantization index q_i is equal to the differential entropy of the corresponding element in $\hat{\boldsymbol{y}}$:

$$\begin{aligned} H[P_{q_i}] &= -\mathbb{E}\log_2 \int_{\hat{y}_i - \frac{1}{2}}^{\hat{y}_i + \frac{1}{2}} p_{y_i}(y_i)\,\mathrm{d}y_i \\ &= -\mathbb{E}\log_2 \big(p_{y_i} * \mathrm{rect}\big)(\hat{y}_i) = h[p_{\hat{y}_i}], \end{aligned} \tag{4}$$

where p_{y_i} and $p_{\hat{y}_i}$ are the marginal densities of one element of the code vector y_i and its dithered version $\hat{y}_i$, respectively. Note that the probability distribution P_{q_i}, *obtained without dithering*, is identical to the $p_{\hat{y}_i}$ defined above, sampled at all integers $q_i \in \mathbb{Z}$: $P_{q_i}(q_i) = p_{\hat{y}_i}(q_i)$. Between these points, $p_{\hat{y}_i}$ is a smooth interpolator, as it arises from a convolution with rect, a moving average kernel. Dithering can thus be seen as providing a *continuous relaxation* of the discrete optimization problem in (1). Altogether, the rate–distortion functional can be rewritten as a function of the transform parameters:

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\phi}) = \mathbb{E}_{\boldsymbol{x},\Delta\boldsymbol{y}} \Big(& -\log_2 p_{\hat{\boldsymbol{y}}}(g_a(\boldsymbol{x};\boldsymbol{\theta}) + \Delta\boldsymbol{y}) \\ &+ \lambda \, \big\| h\big(g_s(g_a(\boldsymbol{x};\boldsymbol{\theta}) + \Delta\boldsymbol{y};\boldsymbol{\phi})\big) - h(\boldsymbol{x}) \big\| \Big), \end{aligned} \tag{5}$$

where $p_{\hat{\boldsymbol{y}}}(\hat{\boldsymbol{y}}) = \prod_i p_{\hat{y}_i}(\hat{y}_i)$. We can now take derivatives of L with respect to $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$, and use stochastic gradient descent to minimize it. Note that finding the global optimum is not guaranteed. However, optimization problems of this type are known to behave very well in practice.

III. Review of parametric transforms

In a traditional transform code, both analysis and synthesis transforms are linear, and exact inverses of each other. In general, this need not be the case, so long as the system minimizes the rate–distortion functional. We have previously shown that a linear transform followed by a particular form of joint normalization (generalized divisive normalization, GDN) is well-matched to the local probability structure of photographic images [6]. This suggests that jointly normalized representations might also prove useful for compression. To demonstrate the use of our optimization framework, we examine GDN as a candidate analysis transform, and introduce an approximate inverse as the corresponding synthesis transform. For the perceptual transform, we use the normalized Laplacian pyramid [4] (NLP), which mimics the local luminance and contrast behaviors of the human visual system.

A. Generalized divisive normalization (GDN)

The GDN transform consists of a linear decomposition $\boldsymbol{H}$ followed by a joint nonlinearity, which divides each linear filter output by a measure of overall filter activity:

$$\begin{aligned} \boldsymbol{y} = g_a(\boldsymbol{x};\boldsymbol{\theta}) \qquad \text{s.t.} \qquad & y_i = \frac{v_i}{\big(\beta_i + \sum_j \gamma_{ij} |v_j|^{\alpha_{ij}}\big)^{\varepsilon_i}} \\ \text{and} \qquad & \boldsymbol{v} = \boldsymbol{H}\boldsymbol{x}, \end{aligned} \tag{6}$$

with parameter vector $\boldsymbol{\theta} = \{\boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}, \boldsymbol{\varepsilon}, \boldsymbol{H}\}$.

B. Approximate inverse of GDN

The approximate inverse we introduce here is based on the fixed point iteration for inversion of GDN introduced in [6]. It is similar in spirit to the LISTA algorithm [7], in that it uses the parametric form of the inversion iteration, but unties its parameters from their original values for faster convergence. We find that for purposes of image compression, one iteration is sufficient:

$$\begin{aligned} \hat{\boldsymbol{x}} = g_s(\hat{\boldsymbol{y}};\boldsymbol{\theta}) \quad \text{s.t.} \quad & \hat{\boldsymbol{x}} = \boldsymbol{H}'\boldsymbol{w} \\ \text{and} \quad & w_i = \hat{y}_i \cdot \big(\beta'_i + \sum_j \gamma'_{ij} |\hat{y}_j|^{\alpha'_{ij}}\big)^{\varepsilon'_i}, \end{aligned} \tag{7}$$

where the parameter vector consists of a distinct set of parameters: $\boldsymbol{\phi} = \{\boldsymbol{\alpha}', \boldsymbol{\beta}', \boldsymbol{\gamma}', \boldsymbol{\varepsilon}', \boldsymbol{H}'\}$.

C. Normalized Laplacian pyramid (NLP)

The NLP imitates the transformations associated with the early visual system: local luminance subtraction and local gain control [4]. Images are decomposed using a Laplacian pyramid [8], which subtracts a local estimate of the mean luminance at multiple scales. Each pyramid coefficient is then divided by a local estimate of amplitude (a constant plus the weighted sum of absolute values of neighbors). Perceptual quality is assessed by evaluating the norm of the difference between reference and reconstruction in this normalized domain. The parameters (constant and weights used for amplitudes) are optimized to best fit perceptual data in the TID2008 database [9], which includes images corrupted by artifacts arising from compression with block transforms. This simple distortion measure provides a near-linear fit to the human perceptual judgments in the database, outperforming the widely-used SSIM [10] and MS-SSIM [11] quality metrics [4]. Examples and code are available at http://www.cns.nyu.edu/~lcv/NLPyr.

[1] Without loss of generality, we assume that the quantization bin size is 1, since we can always modify the analysis/synthesis transforms to include a rescaling. Further, we can implement non-uniform quantization by using nonlinear transforms (as in companding).

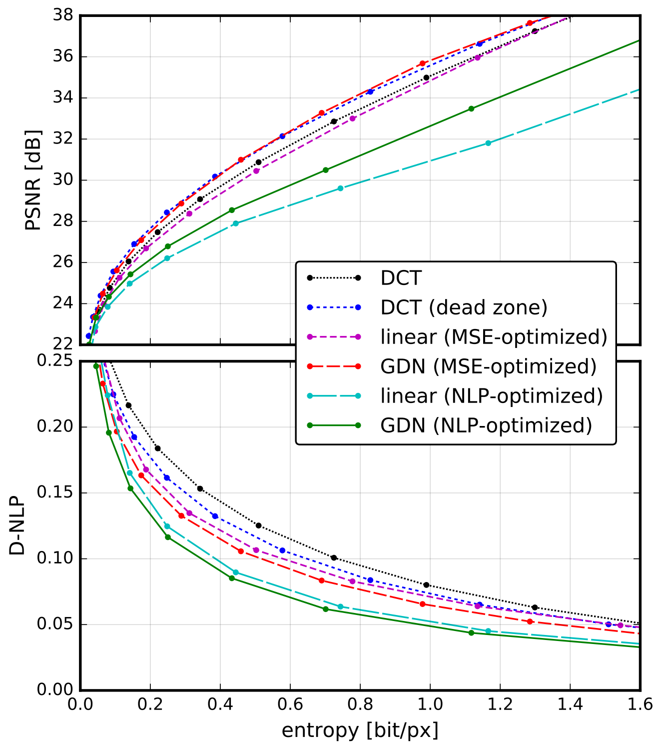
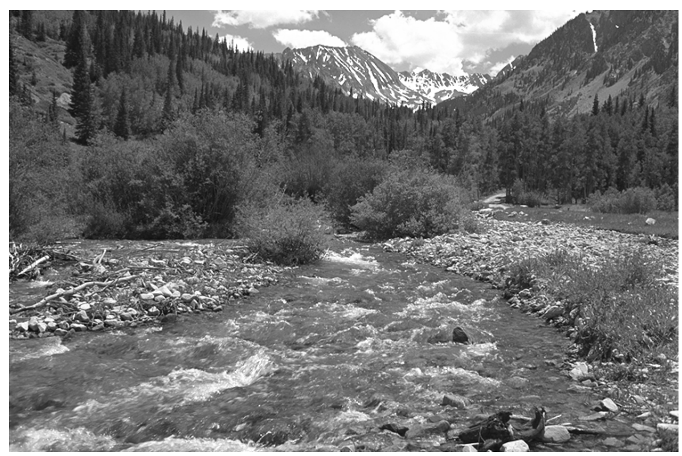


Fig. 2. Rate–distortion results averaged over the entire Kodak image set (24 images, 752×496 pixels each). Reported rates are average discrete entropy estimates. Reported distortion is average PSNR (top) and distance in the normalized Laplacian pyramid domain (bottom – see text).

NLP-GDN, 0.190 bit/px. PSNR: 20.95 D-NLP: 0.21 MS-SSIM: 0.868

DCT (dead z.), 0.204 bit/px. PSNR: 21.81 D-NLP: 0.28 MS-SSIM: 0.827



Original, 8 bit/px. PSNR: ∞ D-NLP: 0 MS-SSIM: 1

Fig. 3. Example image from Kodak set (bottom), compressed with DCT and hand-optimized (for MSE) dead-zone quantization (middle), and GDN with uniform quantization optimized in the NLP domain (top).

IV. Experimental results

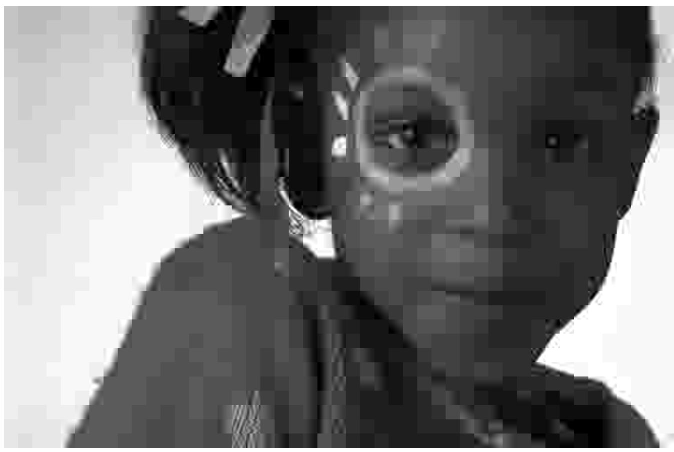
The proposed framework can be used to optimize any differentiable pair of analysis and synthesis transforms in combination with any differentiable perceptual metric. Here, we optimized two types of transforms: a set of linear analysis and synthesis transforms operating on 16×16 pixel blocks (in this case, $\boldsymbol{\theta}$ and $\boldsymbol{\phi}$ consist only of the filter coefficients) and a 16×16 block GDN transform with the approximate inverse defined above. We optimized for two types of distortion metrics, mean squared error (MSE) and distance in the NLP domain [4]. Each combination of transform and distortion metric was optimized for different values of λ.

We also include a fixed linear transform, the 16×16 discrete cosine transform (DCT), which serves as a baseline. For DCT results, we also include a dead-zone quantizer (i.e., uniform, except for a different-size bin around zero), which is commonly used to improve rate–distortion performance. All other codes (i.e., those optimized in our framework) are constrained to use uniform quantization.

The marginal densities $p_{\hat{y}_i}$ were nonparametrically modeled using piecewise linear functions. We used the Adam algorithm [12], a variation of stochastic gradient descent, and a large collection of images from the ImageNet database [2] to perform all optimizations. For each step of the optimization, we used a randomly selected mini-batch of 4 images of 128×128 pixels. Note that, for each value of λ, we optimize a separate transform. Each optimization takes two to three days to complete on a current GPU board. This optimization is entirely offline: once the parameters are determined, no iterative procedure is required to use the transforms for compression.

Dithering was used only during optimization, not for coding the images for evaluation. To estimate code rate, we collected histograms of the quantization index vector $\boldsymbol{q} \in \mathbb{Z}^{256}$ for

NLP-GDN, 0.044 bit/px. PSNR: 26.37 D-NLP: 0.21 MS-SSIM: 0.881



DCT (dead z.), 0.044 bit/px. PSNR: 26.93 D-NLP: 0.24 MS-SSIM: 0.857

Original, 8 bit/px. PSNR: ∞ D-NLP: 0 MS-SSIM: 1

Fig. 4. A second example image from the Kodak set (see caption for fig. 3).

each λ, over all images in the test set, and computed the discrete entropy of these histograms. To prevent overfitting to the training database, we performed all evaluations on a gray-scale version of the Kodak image set.[2]

[2]Downloaded from http://www.cipr.rpi.edu/resource/stills/kodak.html. We computed linear luminance values assuming sRGB primaries and sRGB gamma, and re-applied the sRGB gamma to get the gray-scale images. We also discarded 8 pixels from each side to eliminate boundary artifacts.

For evaluation of the distortion, we first computed the mean squared error (MSE) over the entire set for each λ, and then converted these values into PSNRs (fig. 2, top panel). In terms of PSNR, the optimized linear transform is slightly worse than the DCT. The reason is that the statistics of the ImageNet database are slightly different from the Kodak set (if we validate on a held-out test set of images from ImageNet, the two transforms perform equally well). For comparison, the DCT with dead-zone quantization is better, but it doesn't outperform the MSE-optimized GDN transform, which uses only uniform quantization. The NLP-optimized transforms don't perform well in terms of PSNR.

The situation is reversed, however, when we examine performance in terms of perceptual distortion (fig. 2, bottom). Here, we evaluated the norm in the NLP domain (D-NLP) for each image in the set, and then averaged across images. Note that this norm is almost (inversely) proportional to the subjective mean opinion score (MOS) across several image databases [4]. Overall, the combination of NLP and GDN achieves an impressive rate savings at similar quality when compared with MSE-optimized methods, and with the DCT (both uniform and dead-zone quantizers).

It is also interesting to note that in terms of NLP distance, the optimized linear transform with uniform quantization outperforms both versions of the DCT. This may be due to the fact that the optimized filters tend to be spatially localized (as well as oriented and bandpass), which may lead to visually less disturbing artifacts (not shown).

For visual evaluation, we also show results on two example images (figs. 3 and 4). Results for the entire test set are available at http://www.cns.nyu.edu/~balle/nlpgdn. The figures serve to illustrate the main effect of using a perceptual metric that is aware of local, *relative* contrast. Traditional, linear systems optimized for MSE give too much preference to high-contrast regions (e.g., the snow-covered mountains in the background, or the pebbles/debris in the foreground; fig. 3, center image). By performing joint normalization before quantization, the NLP-optimized GDN transform allocates more bits to represent detail in low-contrast regions (such as the forest in the depicted scene; top image). Overall, the rate allocation is perceptually more balanced, which leads to a more pleasing visual appearance.

V. Discussion

We have introduced a framework for end-to-end optimization of nonlinear transform codes, which can be applied to any set of parametric, differentiable analysis and synthesis transforms. In optimizing a nonlinear transform for a perceptual metric over a database of photographs, we obtained a nonlinear code that respects the perception of local luminance and contrast errors, allowing for significant rate savings.

The earliest instance of a linear transform optimized for signal properties may be the Karhunen–Loève transform (KLT), or principal components analysis (PCA). The DCT was originally introduced as an efficient approximation to the KLT for a separable autoregressive process of order 1 [13].

Other studies have optimized transform parameters specifically for perceptual compression (e.g., [14, 15]), but these were generally limited to optimizing weighting matrices for the DCT. Models that use "matched" non-linear transformations as a means of converting to/from a more desirable representation of the data are known in the machine learning literature as *autoencoders* [16]. However, we are unaware of any work that directly aims to optimize discrete entropy.

We assume uniform quantization in the transform domain, and use dithering to relax the discrete optimization problem into a differentiable one. Dithering was used in some early image coders [17], but generally does not improve rate–distortion performance. To our knowledge, it has not been used as a form of continuous relaxation for optimization purposes. While uniform quantization has been shown to be asymptotically optimal [18], it is well known that dead-zone quantization generally performs better for linear transform coding of images. Here, we demonstrate empirically that the use of nonlinear transforms with uniform quantization allows equivalent or better solutions, and our framework provides a means of finding these transforms.

Divisive normalization has previously been used in DCT-based image compression, e.g., [19, 20]. These approaches use the normalized representation both for coding and distortion estimation, reasoning that this domain is both perceptually and statistically uniform, and thus well-suited for both. The framework introduced here offers more flexibility, by allowing the perceptual domain and the code domain to be distinct (fig. 1). Further, previous methods required the decoder to invert the normalization transform, either by solving an iterative set of linear equations for every block [19], estimating the multipliers (i.e., the values of the denominators) from neighboring blocks [20], or embedding the multipliers into the code as side information. Our framework eliminates this problem by introducing a highly efficient approximate inverse transform, which is jointly optimized along with the normalization transform.

There are several directions in which to proceed with this work. Well-known techniques to improve performance of linear transform codes, such as run-length encoding, adaptive entropy coding, and signal-adaptive techniques in general, should be investigated in the context of nonlinear transform coding. Furthermore, our framework offers a means for exploring much more sophisticated nonlinear analysis/synthesis transforms as well as perceptual metrics, since it is built on the highly successful paradigm of end-to-end optimization over training data.

Acknowledgement

JB and EPS are supported by the Howard Hughes Medical Institute. VL is supported by the APOSTD/2014/095 Generalitat Valenciana grant (Spain).

References

[1] V. K. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing Magazine*, vol. 18, no. 5, 2001. DOI: 10.1109/79.952802.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: a large-scale hierarchical image database," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2009. DOI: 10.1109/CVPR.2009.5206848.

[3] B. Girod, "What's wrong with mean squared error?," in *Digital Images and Human Vision*. M.I.T. Press, 1993, ISBN: 0-262-23171-9.

[4] V. Laparra, J. Ballé, A. Berardino, and E. P. Simoncelli, "Perceptual image quality assessment using a normalized Laplacian pyramid," in *Proceedings of SPIE, Human Vision and Electronic Imaging XXI*, 2016.

[5] L. Schuchman, "Dither signals and their effect on quantization noise," *IEEE Transactions on Communication Technology*, vol. 12, no. 4, 1964. DOI: 10.1109/TCOM.1964.1088973.

[6] J. Ballé, V. Laparra, and E. P. Simoncelli, "Density modeling of images using a generalized normalization transformation," *arXiv e-prints*, 2015, Presented at the 4th Int. Conf. for Learning Representations, 2016. arXiv:1511.06281.

[7] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning*, 2010.

[8] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Transactions on Communications*, vol. 31, no. 4, Apr. 1983. DOI: 10.1109/TCOM.1983.1095851.

[9] N. Ponomarenko, V. Lukin, A. Zelensky, K. Egiazarian, M. Carli, and F. Battisti, "TID2008 – a database for evaluation of full-reference visual quality assessment metrics," *Advances of Modern Radioelectronics*, vol. 10, pp. 30–45, 2009.

[10] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multi-scale structural similarity for image quality assessment," in *Conf. Rec. of the 37th Asilomar Conf. on Signals, Systems and Computers, 2004*, 2003. DOI: 10.1109/ACSSC.2003.1292216.

[11] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Transactions on Image Processing*, vol. 13, no. 4, Apr. 2004. DOI: 10.1109/TIP.2003.819861.

[12] D. P. Kingma and J. L. Ba, "Adam: a method for stochastic optimization," *arXiv e-prints*, 2014, Presented at the 3rd Int. Conf. for Learning Representations, 2015. arXiv:1412.6980.

[13] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, 1974. DOI: 10.1109/T-C.1974.223784.

[14] A. J. Ahumada and H. A. Peterson, "Luminance-model-based DCT quantization for color image compression," in *Proc. SPIE 1666, Human Vision, Visual Processing, and Digital Display III*, 1992. DOI: 10.1117/12.135982.

[15] A. B. Watson, "DCTune: a technique for visual optimization of DCT quantization matrices for individual images," *Society for Information Display Digest of Technical Papers*, vol. 24, pp. 946–949, 1993.

[16] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, 2006. DOI: 10.1126/science.1127647.

[17] L. G. Roberts, "Picture coding using pseudo-random noise," *IRE Transactions on Information Theory*, vol. 8, no. 2, 1962. DOI: 10.1109/TIT.1962.1057702.

[18] H. Gish and J. N. Pierce, "Asymptotically efficient quantizing," *IEEE Transactions on Information Theory*, vol. 14, no. 5, 1968. DOI: 10.1109/TIT.1968.1054193.

[19] J. Malo, I. Epifanio, R. Navarro, and E. P. Simoncelli, "Nonlinear image representation for efficient perceptual coding," *IEEE Transactions on Image Processing*, vol. 15, no. 1, Jan. 2006. DOI: 10.1109/TIP.2005.860325.

[20] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao, "Perceptual video coding based on SSIM-inspired divisive normalization," *IEEE Transactions on Image Processing*, vol. 22, no. 4, 2013. DOI: 10.1109/TIP.2012.2231090.