

ProxIQA: A Proxy Approach to Perceptual Optimization of Learned Image Compression

Li-Heng Chen¹, Graduate Student Member, IEEE, Christos G. Bampis², Member, IEEE,
Zhi Li, Andrey Norkin, Member, IEEE, and Alan C. Bovik, Fellow, IEEE

Abstract—The use of ℓ_p ($p = 1, 2$) norms has largely dominated the measurement of loss in neural networks due to their simplicity and analytical properties. However, when used to assess the loss of visual information, these simple norms are not very consistent with human perception. Here, we describe a different “proximal” approach to optimize image analysis networks against quantitative perceptual models. Specifically, we construct a proxy network, broadly termed ProxIQA, which mimics the perceptual model while serving as a loss layer of the network. We experimentally demonstrate how this optimization framework can be applied to train an end-to-end optimized image compression network. By building on top of an existing deep image compression model, we are able to demonstrate a bitrate reduction of as much as 31% over MSE optimization, given a specified perceptual quality (VMAF) level.

Index Terms—Perceptual optimization, perceptual image/video quality, convolutional neural networks, deep compression.

I. INTRODUCTION

RECENTLY deep neural networks have been successfully and ubiquitously applied on diverse image processing and computer vision tasks, such as semantic segmentation [1], object recognition [2], video encoding [3], [4], and optical flow [5]. Many classic image transformation problems can be approached using a deep generative network, which learns to reconstruct high-quality output images from degraded input image(s). Explicitly, the generative network is trained in a supervised manner with a loss function, which is used to measure the fidelity between the output and a ground-truth image. For instance, the denoising task aims to reconstruct a noise-free image from a noisy image, and Convolutional Neural Networks (CNNs) have been shown to provide good noisy-to-pristine mapping functions [6], [7]. Similar tasks where retaining image fidelity is important include deep image compression, super-resolution [8], [9], frame interpolation [10], and so on.

Manuscript received October 25, 2019; revised July 22, 2020; accepted October 26, 2020. Date of publication November 16, 2020; date of current version November 20, 2020. The research was supported by a grant from Netflix, Inc. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Joao M. Ascenso. (Corresponding author: Li-Heng Chen.)

Li-Heng Chen and Alan C. Bovik are with the Department of Electrical and Computer Engineering, The University of Texas at Austin, Austin, TX 78712 USA (e-mail: lhchen@utexas.edu; bovik@ece.utexas.edu).

Christos G. Bampis, Zhi Li, and Andrey Norkin are with Netflix Inc., Los Gatos, CA 95032 USA (e-mail: christosb@netflix.com; zli@netflix.com; anorkin@netflix.com).

Digital Object Identifier 10.1109/TIP.2020.3036752

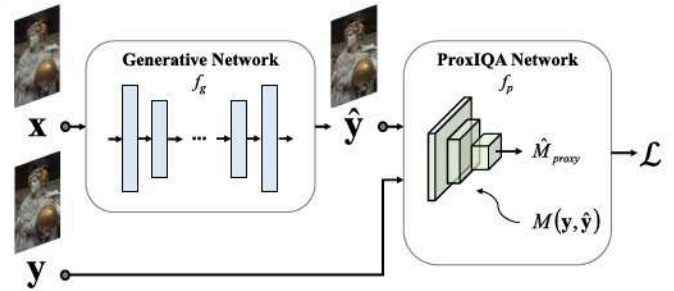


Fig. 1. General framework of perceptual optimization using a ProxIQA network: A generative network takes an image x as input and outputs a reconstructed image $\hat{y} = f_g(x; \theta_g)$. Note that θ_g, θ_p are the parameters of f_g, f_p , respectively, while y is the ground-truth image. Given an image quality measurement M , the ProxIQA network is learned as its proxy, where the output \hat{M}_{proxy} represents $M(y, \hat{y})$.

Although a significant amount of research has been applied on deep learning image transformation problems, most of this work has focused on investigating network architectures or improving convergence speed. The selection of an appropriate loss function, however, has not been studied as much. The choice of the loss functions used to guide model training has been largely limited to the ℓ_p norm family, in particular the MSE (squared ℓ_2 norm), the ℓ_1 norm, and variants of these [11]. The structural similarity quality index (SSIM) [12] and its multi-scale version (MS-SSIM) [13] have also been adopted as loss functions for several image reconstruction tasks [14], [15], owing to their perceptual relevance and good analytic properties, such as differentiability.

Perceptual image quality assessment has been a long-standing research problem. Although numerous powerful perceptual models have been proposed to predict the perceived quality of a distorted picture, most other image quality indexes have never been adopted as deep network loss functions, because they are generally *non-differentiable* and functionally complex.

Towards bridging the gap between modern perceptual quality models and deep generative networks, we explore the potential of adapting more powerful and sophisticated perceptual image quality models as loss functions in deep neural network for addressing the aforementioned problems, by simulating the measurements made by a perceptual model by a proxy network ProxIQA. As shown in Fig. 1, the main idea is to optimize the hyper parameters of the generative network

θ_g , using a ProxIQA network as a perceptual loss function

$$\begin{aligned}\mathcal{L}(\theta_g) &= f_p(y, \hat{y}; \theta_p) \\ &= f_p(y, f_g(x; \theta_g); \theta_p),\end{aligned}\quad (1)$$

where x, \hat{y} are the input and output of the generative network f_g and y is the ground-truth image. In the image compression problem, x is an uncompressed image and we anticipate the fidelity of the compressed image. Thus, this is a special case where $x = y$. The parameters θ_p of the ProxIQA network are optimized so that it mimics $M(y, \hat{y})$, a perceptual image quality measurement between y and \hat{y} .

The outline of this paper is as follows: Section II reviews the relevant literature of image quality assessment, perceptual optimization, and deep image compression. Section III describes the ProxIQA framework, while Section IV provides analysis and experimental results. Finally, Section V concludes the paper.

II. RELATED WORK

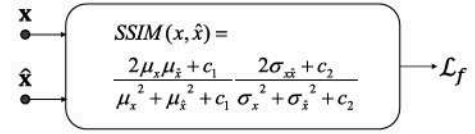
In this section, we provide a literature review of studies that are closely related to our work. The relevant topics of objective image quality assessment, perceptual optimization and deep compression are briefly reviewed.

A. Perceptual Image Quality Metrics

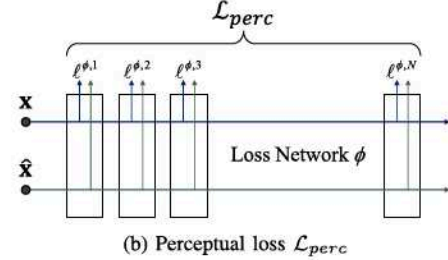
Over the past decade there has been a remarkably increasing interest in developing objective image quality assessment (IQA) methods. Objective IQA models are commonly classified as full-reference, reduced-reference [16], and no-reference [17], based on the amount of information they assess from a reference image of ostensibly pristine quality. Here we only need to consider the full-reference (FR) scenario, since it may be assumed that ground-truth data is available, hence we only review FR IQA models.

Beyond the well-known structural similarity index and other SSIM-type methods, a wide variety of perception-based FR models have been designed, including the visual signal-to-noise ratio index (VSNR) [18], the visual information fidelity (VIF) index [19], the MAD model [20], the feature similarity index (FSIM) and its extension FSIM_c [21], and the Visual Saliency-Induced index (VSI) [22].

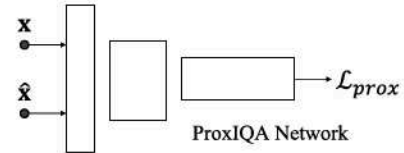
With the rapid development of machine learning, important data-driven models have also begun to emerge. Models using “handcrafted” features with different regressors [23]–[26] can produce accurate quality predictions on existing datasets. However, they cannot be used for end-to-end training if any component is not differentiable, motivating us to pursue a systematic and holistic approach. For example, Random Forest regression as used by [24] is non-differentiable at the split nodes. Other DNN-based IQA models [27]–[29] have also shown promising results. A particularly successful example is Netflix’s announcement of an open-source FR video quality engine called Video Multimethod Assessment Fusion (VMAF) [30]. VMAF combines multiple quality features to train a Support Vector Regressor (SVR) to predict subjective judgments of video quality. When it is applied to still pictures, VMAF treats the data as a video frame having zero motion.



(a) Determined function loss \mathcal{L}_{func}



(b) Perceptual loss \mathcal{L}_{perc}



(c) ProxIQA loss \mathcal{L}_{prox}

Fig. 2. Comparison of different perceptual loss layers for generative neural networks. Given a target patch x and a reconstructed patch \hat{x} , (a) determined function based approaches typically use a differentiable function having a certain degree of convexity, such as SSIM and MS-SSIM. (b) Perceptual-loss based approaches define the loss from the features extracted from intermediate layers of a trained network (such as VGG) (c) Our method uses the output of a proxy network that approximates an IQA model as the loss function.

The study in [31] showed that VMAF correlates quite well against human judgements of still picture quality. Indeed, most of the VMAF features are those of VIF [19], which is among the most powerful and effective image quality models. Like SSIM, VMAF is used to perceptually optimize tremendous volumes of internet picture and video traffic [30].

Generally, more advanced, high-performance quality prediction models such as these are difficult to adopt as loss functions for end-to-end optimization networks.

B. Perceptual Optimization

As tractable tools for perceptual optimization, SSIM and MS-SSIM have been widely adopted because of the simple analytical form of their gradients and computational ease. Moreover, their convexity properties [32] makes them feasible targets for optimization. For example, two recent studies adopted structural similarity functions as loss layers of image generation models, obtaining improved results as validated by conducting a human subjective study [14] and by objective evaluation against several other perceptual models [15]. The recent manuscript [33] systematically evaluated IQA models as loss functions for perceptually optimizing different tasks. It was found that many IQA methods failed due to non-convexity of the objective function.

Rather than optimizing a mathematical function, another approach uses a deep neural network to guide the training. Recent experimental studies suggest that the features extracted

from a well-trained image classification network have the capability to capture information useful for other perceptual tasks [34]. As illustrated in Fig. 2(b), the perceptual loss is defined as

$$\begin{aligned}\mathcal{L}_{perc} &= \sum_i \ell^{\phi,i} \\ &= \sum_i N_i^{-1} \|\phi_i(\mathbf{x}) - \phi_i(\hat{\mathbf{x}})\|_2^2,\end{aligned}\quad (2)$$

where ϕ_i denotes the output feature map of the i -th layer with N_i elements of a pre-trained network ϕ .

In practice, the loss computed from the high-level features extracted from a pre-trained VGG classification network [35], also called VGG loss, has been commonly adopted for diverse computer vision tasks. The VGG loss has been applied to such diverse tasks as style transfer [36], [37], superresolution [36], [38]–[40], and image inpainting [41].

C. End-to-End Optimized Lossy Image Compression

Recently, lossy image compression models have been realized using deep neural network architectures. Most of these have deployed deep auto-encoders. For example, Ballé *et al.* [42] proposed a general infrastructure for optimizing image compression in an end-to-end manner. Unlike other methods, the bitrate is estimated and considered during training. In [43], this model is improved by incorporating a scale hyperprior into the compression framework. The authors use an additional network to estimate the standard deviation of the quantized coefficients to further improve coding efficiency. Later, Minnen *et al.* [44] exploit a PixelCNN layer, which they combine with an autoregressive hyperprior. Beyond these early efforts, other recent approaches have adopted more complex network architectures such as recurrent neural networks (RNNs) [45]–[47] and generative adversarial networks (GANs) [48], [49]. Some works have also been done to extend these ideas to the deep video compression problem [50]–[52].

Unsurprisingly, the idea of optimizing a conventional codec such as H.264/AVC against perceptual models like SSIM, VIF, and VMAF have been deeply studied [53]–[56] and implemented in widespread practice [30]. We seek to extend this concept in similar manner to learn an end-to-end perceptually optimized compression model.

III. PROPOSED PERCEPTUAL OPTIMIZATION FRAMEWORK

Our approach to training an image compression model in a perceptually optimized way is depicted in Fig. 3. This framework involves optimizing two networks: an image compression network f_c , and a sub-network f_p , which is a proxy of an IQA model, which we will refer to as ProxIQA. A source image \mathbf{x} is input to a compression network, which produces a reconstructed image:

$$\hat{\mathbf{x}} = f_c(\mathbf{x}). \quad (3)$$

Separately, the ProxIQA network maps the image pair $(\mathbf{x}, \hat{\mathbf{x}})$ into a proxy of an image quality model M :

$$\hat{M} = f_p(\mathbf{x}, \hat{\mathbf{x}}). \quad (4)$$

In each training iteration, the two networks are alternately updated as follows:

1) *Deep Compression Model Updating*: To integrate f_p into the update of f_c given a mini-batch \mathbf{x} , the model parameters of f_p are fixed during training. In order to minimize distortion, the output of f_p becomes part of the objective in the optimization of f_c :

$$\hat{M}(\mathbf{x}, \hat{\mathbf{x}}) = f_p(\mathbf{x}, \hat{\mathbf{x}}) = f_p(\mathbf{x}, f_c(\mathbf{x})). \quad (5)$$

By back-propagating through the forward model, the loss derivative is used to drive f_c . As we will see later in section III-B, the entropy model is also updated at this stage.

2) *ProxIQA Network Updating*: Given a mini-batch pair \mathbf{x} and $\hat{\mathbf{x}}$ collected from the most recent update of the compression network, the quality scores $M(\mathbf{x}, \hat{\mathbf{x}})$ are calculated. The ProxIQA network is updated such that its output \hat{M} optimally fit M given the input $\{\mathbf{x}, \hat{\mathbf{x}}\}$. Note that the compression network is not needed in this part of the training.

As may be seen, the auxiliary sub-network ProxIQA is incorporated into the training of the compression network. However, it is important to understand that the ProxIQA network is not present during the testing (image compression/decompression) phase. In addition, it could be argued that the training strategy employed by ProxIQA shares some similarities with that of Generative Adversarial Networks (GANs) [57]. However, we have identified an important conceptual difference between the two: the ProxIQA network learns to fit a quality model and does not include the image compression network during its update, which is different from the discriminator in GANs.

A. Network Architecture

1) *ProxIQA Network*: The goal is to learn a nonlinear regressor via a CNN. The network takes a reference patch \mathbf{x} and a distorted patch $\hat{\mathbf{x}}$ as input, where both have $W \times H$ pixels. They are then concatenated into a 6-channel signal, where a $W \times H \times 6$ raw input $\{\mathbf{x}, \hat{\mathbf{x}}\}$ is fed into the network and reduced to a predicted quality score. As depicted in Fig. 4, the ProxIQA network may be as simple as a shallow CNN consisting of three stages of convolution, ReLU as an activation function, and maxpooling. The spatial size is reduced by a factor of 2 after each stage via 2×2 max pooling layers. Finally, $64 W/8 \times H/8$ feature maps are flattened and fed to a fully connected layer which yields the output. The parameterization of each layer is detailed in the figure.

2) *Compression Network*: We build on the deep image compression model [42]. As shown in Fig. 3, the image compression network comprises an analysis transform (g_a) at the encoder side, and a synthesis transform (g_s) at the decoder side. Both transforms are composed of three consecutive layers of convolution-down(up) sampling-activation. Instead of utilizing ReLU, a generalized divisive normalization (GDN) transform is adopted as the activation function [58], similar to normalization of the visual signal by the human visual system (HVS).

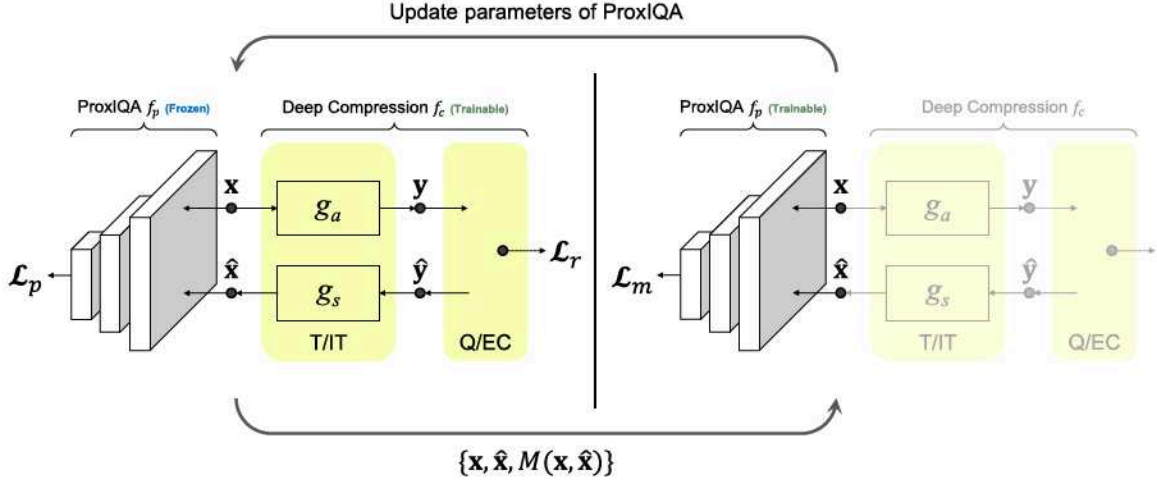


Fig. 3. Detailed framework of the proposed optimization strategy. Perceptually training a deep image compression model involves alternating optimization of the compression network (left side of the figure) and the ProxIQ network (right side of the figure). Thin arrows indicate the flow of data in the network, while bold arrows represent the information being delivered to update the complementary network.

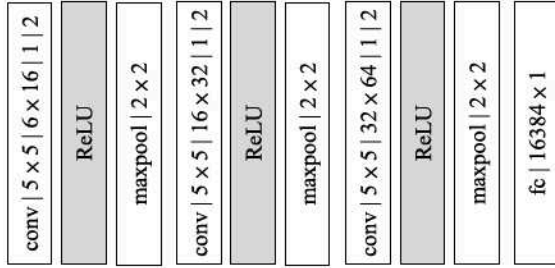


Fig. 4. Architecture of the ProxIQ network. The convolutional parameters are denoted as: height \times width | input channel \times output channel | stride | padding; The max pooling layers are denoted as: vertical pooling size \times horizontal pooling size.

B. Loss Functions

As illustrated in Fig. 3, let \mathbf{x} , \mathbf{y} , $\hat{\mathbf{y}}$, and $\hat{\mathbf{x}}$ be the source batch, latent presentation, quantized latent presentation, and reconstructed batch, respectively. The model parameters in the analysis and synthesis transforms are collectively denoted by $\theta = (\theta_a, \theta_s)$. The ProxIQ network has model parameters ϕ . Given a perceptual metric M , such as SSIM or VMAF, the goal is to optimize the full set of parameters θ , ϕ , such that the learned image codec can generate a reconstructed image $\hat{\mathbf{x}}$ that has a high perceptual quality score $M(\mathbf{x}, \hat{\mathbf{x}})$. Furthermore, the rate should be as small as possible. Therefore, we train our model using the following losses.

1) *Rate Loss*: The rate loss representing the bit consumption of an encode $\hat{\mathbf{y}}$ is defined by

$$\mathcal{L}_r(\theta) = -\log_2 p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}}), \quad (6)$$

where $p_{\hat{\mathbf{y}}}(\hat{\mathbf{y}})$ is the entropy model. This entropy term is minimized when the actual marginal distribution and $\hat{\mathbf{y}}$ have the same distribution.

During training, the latent presentation \mathbf{y} is approximately quantized to $\hat{\mathbf{y}}$ as additive quantization noise $\Delta \mathbf{y}$ (i.i.d uniform). Then, $\hat{\mathbf{y}}$ is used to estimate the rate via (6). Unlike the estimated entropy used when training the network, a variant

of the context-adaptive binary arithmetic coder (CABAC) [59] is used to encode the discrete-valued data into the bitstream during testing.

2) *Pixel Loss*: The pixel loss is the residual between the source image and the reconstructed image mapped by a distance function d . Given \mathbf{x} and $\hat{\mathbf{x}}$, the pixel loss is defined by

$$\mathcal{L}_d(\theta) = d(\mathbf{x} - \hat{\mathbf{x}}), \quad (7)$$

where $d(\cdot)$ can be mean squared error (i.e., $d(x) = \|x\|_2^2$) or mean absolute error (i.e., $d(x) = \|x\|_1$).

The original work in [42] used $d(x) = \|x\|_2^2$ as the pixel loss to maximize the PSNR of the reconstructed images. When combined with the rate loss, the image compression network is optimized by minimizing the objective function defined by

$$\mathcal{L}(\theta) = \lambda \mathcal{L}_d + \mathcal{L}_r, \quad (8)$$

which is a similar form of rate-distortion optimization (RDO) function as is found in conventional codecs. This equation indicates that the entropy model is jointly optimized with the compression network. We make use of a pixel loss to encourage training stability.

3) *Proxy Loss*: Instead of just minimizing an ℓ_p norm between \mathbf{x} and $\hat{\mathbf{x}}$, we introduce a novel loss term. The proxy loss \mathcal{L}_p is calculated from the output of ProxIQ network, denoted by \hat{M} , with fixed parameter ϕ :

$$\mathcal{L}_p(\theta; \phi) = M_{\max} - \hat{M}(\mathbf{x}, \hat{\mathbf{x}}). \quad (9)$$

Here M_{\max} denotes the upper bound of the model M , which is a constant to the loss function.

Finally, the total loss function for optimizing the compression network is the weighted combination of the losses from Eqs. (7), (6), and (9):

$$\mathcal{L}_t(\theta; \phi) = \lambda [\alpha \mathcal{L}_p + (1 - \alpha) \mathcal{L}_d] + \mathcal{L}_r, \quad (10)$$

where λ balances bitrate against distortion of the encoded bitstream, and α weights the proxy loss against the pixel



Fig. 5. Comparison of distorted patches from existing databases and three deep compression networks. For (a)(b), The first column shows the pristine patches while the other three columns are randomly selected distorted patches. In (c), each similar pair includes a reference patch (left), and a distorted patch (right). The distortions in (a)(b)(c) were generated from (i) patches reconstructed during different training steps of the deep compression network; (ii) synthetic distortions with different severity levels added to the original patches; (iii) the outputs of various convolutional neural networks, respectively.

loss. We empirically set $\alpha = 1.5e - 3$ for $M = \text{VMAF}$, and $\alpha = 3e - 3$ for both $M = \text{SSIM}$ and MS-SSIM . This was accomplished by selecting four values spaced along the reals and choosing the best performing value. However, the performance did not vary much over the range of tested values. The pixel loss plays a different role as a regularization term. Since the ProxIQa network is updated at each step, the loss function \mathcal{L}_p is also updated. The pixel loss serves to stabilize the training process.

4) *Metric Loss*: The ProxIQa network aims to mimic an image quality model M . Given two images x and \hat{x} , define a metric loss \mathcal{L}_m to attain this objective while updating ProxIQa network:

$$\mathcal{L}_m(\phi; \hat{x}) = \left\| \hat{M}(x, \hat{x}) - M(x, \hat{x}) \right\|_2^2. \quad (11)$$

Note that \hat{x} is a constant, since it is obtained from the reconstructed patches generated during the most recent update of the compression network. $M(x, \hat{x})$ is the ground-truth data obtained on-the-fly during training. We used the MSE loss, because it is more sensitive to outliers. We also tried ℓ_1 loss to update the ProxIQa parameters ϕ , obtaining very similar results.

C. What's Wrong With Using a Pre-Trained Network?

Another way of attempting to accomplish the same goal is to use a pre-trained network as the loss layer. That is, a proxy network is first learned to predict a metric score given a pristine patch and a distorted patch from an existing dataset. Next, the trained proxy network is inserted into the loss layer of the deep compression network with the goal of maximizing

the proxy score. Unfortunately, severe complications can arise when applying this simple methodology.

1) *New Distortion Types*: The success of a CNN model depends highly on the size of the training set. This is often an obstacle to learning DNN-based IQA models, due to the insufficient size, as compared to image recognition databases, of publicly available IQA databases. Luckily, training a proxy network on an existing model does not require human-labeled subjective quality scores such as mean opinion scores (MOS) or differential mean opinion scores (DMOS). The ground truth metric score for training the proxy network is easily obtained, given a pristine patch and a distorted patch. Therefore, we can make use of large-scale databases that do not include MOS, such as the Waterloo Exploration database [60].

Nevertheless, the distortion types provided by public-domain databases are generally quite different from the distortions created by a deep compression model (see Fig. 5(a)). In fact, most existing databases only provide synthetic distortions. As shown in Fig. 5(b), we randomly selected images distorted by JPEG, JPEG2000, Gaussian blur, and white Gaussian noise, from the Waterloo Exploration database. As may be easily observed, these distortions are drastically different from those created by CNNs.

The Berkeley-Adobe PPS (BAPPS) database [34] contains many distorted patches collected from the outputs of CNN models. However, these CNN-based distortion types are still different from the patches reconstructed by deep compression networks. In the second to fourth columns of Fig. 5(a), we show several randomly collected reconstructed patches output by the deep compression model during different training steps. We have observed that dissimilar distortions can be generated by the same network architecture by using

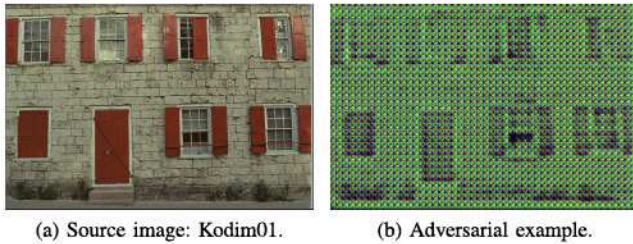


Fig. 6. An “adversarial” example produced by the compression network. The VMAF score calculated from (a) the source image and (b) the decoded image is 5.35 (which indicates a very poor-quality image), while the pre-trained ProxIQA network predicts a quality score of 97.74.

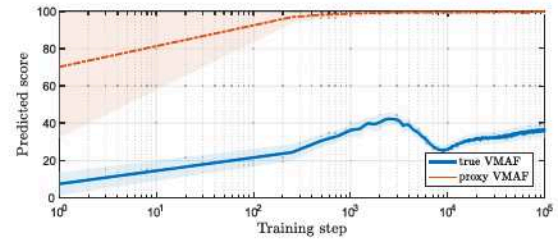
different training steps or parameters. By comparing these distortions, we noticed that learning a network from previously existing databases might not be the optimal solution to our problem. On the contrary, by applying the proposed alternating training, this problem is immediately resolved: the patches reconstructed by the compression network are directly used to learn the ProxIQA network.

2) *Adversarial Examples*: We also discovered that the deep compression network often generates “adversarial” examples when its loss layer is the output of a pre-trained network having fixed parameters. Fig. 6 shows such an “adversarial” example generated by the deep compression network using a proxy network as its loss function. In this example, the proxy network was well-trained to mimic the VMAF algorithm. However, comparing Fig. 6(a) with Fig. 6(b), it is apparent that the true VMAF score and the proxy VMAF score predicted by the ProxIQA network are very different. This can be understood by considering the training of the network to be an interpolation problem, whereby the neural networks maps a test image to an accurate quality score. However, when the input is too different from the training set, the ProxIQA network may produce a poor interpolation result.

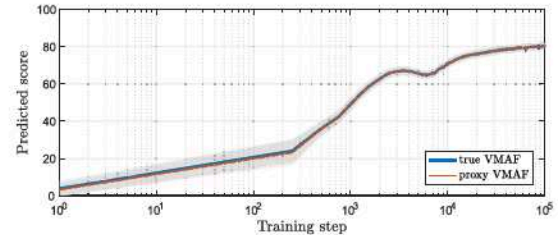
To further illustrate, Fig. 7 compares true VMAF scores with proxy VMAF scores. All of the scores were calculated on the reconstructed patches produced during training. Figure 7(a) shows that the proxy VMAF scores quickly approached 100, whereas much lower true VMAF scores were assigned to the patches produced by the compression model. This problem becomes significant when the previous discussed training strategy is applied. However, a straightforward way of improving the training stage is to simply feed the adversarial examples along with their objective quality scores into the ProxIQA network as additional training data. The ProxIQA network is then updated, which enables it to predict proxy quality much more accurately. As shown in Fig. 7(b), the true and proxy scores become highly coincident early in the training process. This also indicates that a CNN is able to closely approximate the responses delivered by VMAF.

D. Implementation and Training Details

The TensorFlow framework (version 1.12) was used to implement the proposed method. We use the Adam solver [61] to optimize both the ProxIQA network and the deep compression network, with $\beta_1 = 0.9$ and $\beta_2 = 0.999$. We set the initial



(a) Pre-trained model.



(b) Model learned from the proposed alternating training process.

Fig. 7. Comparison of true VMAF scores and proxy VMAF scores (quality scores predicted by ProxIQA) using two different optimization strategies during the training process. The two scores are plotted in mean values (lines) and standard deviations (shadows).

learning rates for both networks at fixed values of $1e-4$ for the first 2M steps and $1e-5$ for an additional 100K iterations. Thus, the network was trained on 2.1M iterations of back-propagation. All of the models were trained using NVIDIA 1080-TI GPU cards.

We used a subset of the 6507 processed images from the ImageNet database [62] as training data. As described in [42], small amounts of uniform noise were added to the images. The images were then down-sampled by random factors to reduce compression artifacts and high-frequency noise, and randomly cropped to a size of 256×256 . In each mini-batch, we randomly sampled 8 image patches from the subset. We then cropped the images to 128×128 patches. The source code is publicly available on <https://github.com/treammm/Compression>.

IV. EXPERIMENTS AND ANALYSIS

We compared the proposed perceptual optimization framework against the original MSE optimized image compression model and also against state-of-the-art image codecs. In order to experimentally evaluate the trained models, we conducted and reported the results of a quantitative evaluation, a subjective comparison, and by a runtime analysis. We first describe the experimental setups that were used, including the datasets on which the performance evaluation was conducted using standard evaluation criteria. We also performed a different series of experiments to probe the limitations of our design. In all the experiments conducted, we denote the deep image compression model [42] as the BLS model (BLS represents the authors’ first initials), which we use as a baseline for performance comparison. Also, we denote an optimized proxy model for a given IQA model M using (10) and (11) is denoted by M_p .

TABLE I

BD-RATE CHANGE (IN PERCENTAGE) OF THE OPTIMIZATION RESULTS OF DEEP IMAGE COMPRESSION MODELS ON THE KODAK DATASET, FOR FOUR DIFFERENT IQA MODELS. THE CORRESPONDING BASELINE IS THE MSE-OPTIMIZED BLS MODEL [42]. SMALLER OR NEGATIVE VALUES MEANS BETTER CODING EFFICIENCY

Optimization BD-rate Metric	SSIM _p				MS-SSIM _p				VMAF _p			
	PSNR	SSIM	MS-SSIM	VMAF	PSNR	SSIM	MS-SSIM	VMAF	PSNR	SSIM	MS-SSIM	VMAF
Kodim01	14.4	-17.4	-16.3	-2.1	12.6	-1.4	-3.9	-2.4	4.9	-2.9	-4.3	-25.3
Kodim02	11.0	-21.6	-17.9	15.4	6.5	-13.1	-21.5	3.0	8.6	-2.2	-2.6	-26.7
Kodim03	13.5	-19.5	-16.6	5.8	4.3	-14.6	-24.1	-10.6	7.8	-5.2	-7.2	-33.2
Kodim04	15.6	-23.5	-21.4	9.3	10.0	-15.6	-26.6	0.0	6.3	-6.4	-7.4	-26.9
Kodim05	13.8	-16.8	-14.8	1.4	14.0	-3.9	-11.2	-2.1	3.1	-6.1	-6.3	-18.1
Kodim06	17.0	-22.0	-20.0	0.5	12.2	-10.1	-19.2	-2.3	4.5	-6.0	-7.4	-23.6
Kodim07	11.6	-14.5	-10.9	8.3	7.1	-9.9	-21.1	-0.6	7.0	-10.0	-9.7	-24.7
Kodim08	13.0	-14.9	-15.1	-5.4	16.4	-0.6	-6.6	-7.3	2.5	-3.9	-5.2	-20.3
Kodim09	16.1	-17.6	-14.5	12.3	7.6	-12.9	-23.8	-1.2	6.4	-3.1	-4.4	-24.9
Kodim10	15.4	-25.3	-22.4	12.2	10.4	-17.3	-32.8	3.3	5.0	-9.6	-10.7	-23.2
Kodim11	16.6	-26.6	-25.3	12.8	14.1	-16.1	-25.9	6.9	5.1	-7.1	-8.5	-20.0
Kodim12	10.4	-30.8	-28.8	12.8	2.5	-19.7	-31.3	8.3	6.6	-3.4	-5.9	-22.4
Kodim13	19.9	-25.3	-24.5	-7.1	16.8	-9.1	-16.6	-7.3	1.2	-8.1	-9.2	-19.6
Kodim14	17.5	-22.6	-20.3	8.2	13.0	-10.5	-18.3	0.0	4.0	-8.5	-9.2	-20.1
Kodim15	17.4	-27.1	-26.0	6.0	9.8	-15.6	-30.7	0.1	6.7	-5.7	-7.6	-31.8
Kodim16	14.4	-20.4	-15.9	2.5	6.3	-13.9	-21.3	-3.2	6.5	-4.6	-6.1	-24.9
Kodim17	18.0	-20.7	-19.3	7.3	10.5	-15.0	-29.7	0.1	5.3	-9.6	-10.9	-23.6
Kodim18	19.7	-20.7	-18.3	12.1	17.9	-9.2	-17.9	2.4	2.7	-9.1	-9.8	-22.0
Kodim19	17.6	-18.2	-16.0	13.9	17.7	-8.7	-21.8	6.8	8.7	-4.7	-6.1	-17.8
Kodim20	19.7	-21.3	-22.4	15.3	12.8	-10.3	-25.3	0.3	5.1	-5.4	-8.2	-24.2
Kodim21	18.1	-16.6	-15.0	1.9	16.4	-7.5	-16.9	-4.1	4.1	-5.3	-6.8	-21.6
Kodim22	15.0	-23.0	-21.2	11.8	10.3	-15.5	-24.3	3.5	3.4	-10.9	-12.4	-25.1
Kodim23	14.7	-21.6	-17.5	15.5	10.5	-16.6	-29.6	4.1	7.8	-7.5	-8.7	-16.7
Kodim24	21.3	-23.4	-21.7	2.0	20.6	-11.3	-21.9	-1.7	2.2	-11.4	-12.2	-24.0

A. Experimental Setup

1) *Evaluation Datasets*: To evaluate the various image codecs, we utilized the well-known Kodak dataset [63] of 24 very high quality uncompressed 768×512 images. This publicly available image set is commonly used to evaluate image compression algorithms and IQA models. We also used a subset of the Tecnick dataset [64] containing 100 images of resolution 1200×1200 , and 223 billboard images collected from the Netflix library having acronym NFLX, yielding images having more diverse resolutions and contents. None of the test images were included in the training sets, to avoid bias and overfitting problems.

2) *Evaluation Criteria*: We measured the objective coding efficiency of each image codec using the Bjontegaard-Delta bitrate (BD-rate) [65], which quantifies differences in bitrate at a fixed distortion level relative to another reference encoder. To calculate BD-rate, we encoded the images at eight different fixed bitrates, ranging from 0.05 bpp (bit per pixel) to 2 bpp. The performances of all of the codecs were compared to the same baseline - the MSE-optimized BLS model. A negative number of BD-rate means the bitrate was reduced as compared with the baseline. To fairly compare deep compression models having different loss layers, we used 192 filters at every layer, and trained all of the models using the same number of steps. To ensure reproducibility, we report the version of each software codec used in Table II. The input

TABLE II

VERSION OF SOFTWARE IMAGE CODECS USED IN THE EXPERIMENTS

Codec	Software	Version
JPEG	JPEG XT	Release 1.53 ¹
JPEG2000	Kakadu	Version 7.10.2 ²
HEVC	HM	16.9 ³
Ballé [42]	Tensorflow Compression	Release 1.0 ⁴
Ballé [43]	Tensorflow Compression	Release 1.2 ⁵

image formats used were YUV444 for JPEG and JPEG2000, and both YUV420/444 for intra-coded HEVC, respectively. Lastly, the IQA algorithms that were used to evaluate the codecs were calculated using FFmpeg 4.0 with libavfilter (for PSNR) and libvmaf 0.6.1⁶ (for SSIM, MS-SSIM, VIF, and VMAF). Specifically, the PSNR calculation in libavfilter is defined by

$$\text{PSNR}_{\text{avg}} = (4 \times \text{PSNR}_Y + \text{PSNR}_U + \text{PSNR}_V) / 6, \quad (12)$$

which is commonly used for combining per-channel PSNRs.

¹<https://jpeg.org/downloads/jpegxt/reference1367abcd89.zip>

²<https://kakadusoftware.com/downloads/>

³https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.9/

⁴<https://github.com/tensorflow/compression/releases/tag/v1.0>

⁵<https://github.com/tensorflow/compression/releases/tag/v1.2>

⁶<https://github.com/Netflix/vmaf>

TABLE III

OVERALL COMPARISON OF DIFFERENT CODECS AND RESULTS OF OPTIMIZED DEEP IMAGE COMPRESSION: AVERAGE CHANGE OF BD-RATE AND STANDARD DEVIATION (INDICATED BY UNDERLINE) EXPRESSED AS PERCENTAGE, USING THREE DIFFERENT IQA MODELS TO TRAIN PROXIQ. THE BASELINE OF COMPARISON IS THE MSE-OPTIMIZED BLS MODEL [42]. SMALLER OR NEGATIVE VALUES INDICATE BETTER CODING EFFICIENCY

Image Dataset	Kodak					Tecnick					NFLX				
	PSNR	SSIM	MS-SSIM	VIF	VMAF	PSNR	SSIM	MS-SSIM	VIF	VMAF	PSNR	SSIM	MS-SSIM	VIF	VMAF
JPEG	113.99	129.49	149.86	95.62	78.36	119.33	218.04	171.59	94.59	89.73	102.28	143.99	168.20	79.34	89.95
	<u>15.86</u>	<u>21.98</u>	<u>25.04</u>	<u>15.20</u>	<u>20.33</u>	<u>31.65</u>	<u>34.11</u>	<u>38.88</u>	<u>32.73</u>	<u>24.01</u>	<u>28.59</u>	<u>34.94</u>	<u>37.19</u>	<u>25.88</u>	<u>23.53</u>
JPEG2000	-11.51	6.25	-1.02	-1.76	-33.39	-13.06	-1.55	-8.41	-8.91	-34.25	-27.81	1.43	-3.93	-14.55	-38.98
	<u>10.78</u>	<u>14.68</u>	<u>13.71</u>	<u>13.19</u>	<u>8.77</u>	<u>22.94</u>	<u>25.92</u>	<u>22.89</u>	<u>22.32</u>	<u>18.47</u>	<u>20.25</u>	<u>26.84</u>	<u>25.77</u>	<u>23.34</u>	<u>17.03</u>
HEVC	-26.35	-6.32	-6.12	-13.68	-28.23	-28.32	-8.97	-11.07	-14.87	-27.65	-49.43	-17.12	-16.06	-29.93	-35.03
	<u>11.06</u>	<u>18.65</u>	<u>19.05</u>	<u>14.14</u>	<u>12.15</u>	<u>22.48</u>	<u>29.84</u>	<u>27.56</u>	<u>25.07</u>	<u>24.05</u>	<u>16.07</u>	<u>25.63</u>	<u>26.59</u>	<u>21.22</u>	<u>21.52</u>
HEVC ₄₂₀	-27.33	-25.98	-24.97	-32.73	-42.18	-19.48	-28.97	-33.95	-40.15	-46.67	-37.63	-35.41	-33.88	-47.51	-50.91
	<u>8.56</u>	<u>15.05</u>	<u>15.53</u>	<u>11.77</u>	<u>10.32</u>	<u>19.25</u>	<u>22.74</u>	<u>19.93</u>	<u>18.05</u>	<u>18.37</u>	<u>18.13</u>	<u>19.01</u>	<u>20.08</u>	<u>15.66</u>	<u>16.34</u>
BLS MS-SSIM [42]	107.50	-21.89	-27.84	4.46	11.72	50.31	-15.09	-24.17	6.89	11.25	67.04	-10.58	-24.28	7.12	28.10
	<u>73.42</u>	<u>7.93</u>	<u>5.84</u>	<u>8.48</u>	<u>11.98</u>	<u>19.97</u>	<u>9.45</u>	<u>9.89</u>	<u>13.49</u>	<u>11.68</u>	<u>21.13</u>	<u>20.15</u>	<u>10.85</u>	<u>15.88</u>	<u>13.91</u>
BLS SSIM _p	15.89	-21.31	-19.25	-4.71	7.19	8.67	-10.79	-16.11	-5.72	8.68	16.79	-19.01	-17.73	-6.61	9.75
	<u>2.83</u>	<u>3.93</u>	<u>4.22</u>	<u>3.44</u>	<u>6.52</u>	<u>6.59</u>	<u>7.60</u>	<u>6.13</u>	<u>4.98</u>	<u>6.66</u>	<u>8.99</u>	<u>6.82</u>	<u>6.65</u>	<u>6.83</u>	<u>6.72</u>
BLS MS-SSIM _p	11.67	-11.58	-21.77	-3.69	-0.17	4.47	-17.40	-23.50	-5.00	0.19	12.28	-11.59	-23.53	-5.96	4.34
	<u>4.50</u>	<u>4.77</u>	<u>7.17</u>	<u>3.51</u>	<u>4.49</u>	<u>8.65</u>	<u>8.55</u>	<u>9.89</u>	<u>5.16</u>	<u>4.78</u>	<u>10.40</u>	<u>9.65</u>	<u>10.85</u>	<u>6.92</u>	<u>6.44</u>
BLS VMAF _p	5.23	-6.53	-7.78	-1.90	-23.35	6.23	-8.45	-5.97	-2.24	-23.78	7.00	-4.35	-5.43	-1.96	-21.97
	<u>1.97</u>	<u>2.57</u>	<u>2.40</u>	<u>1.82</u>	<u>3.92</u>	<u>2.91</u>	<u>4.34</u>	<u>4.16</u>	<u>2.90</u>	<u>7.16</u>	<u>2.37</u>	<u>5.96</u>	<u>5.57</u>	<u>3.66</u>	<u>5.01</u>
BMSHJ MSE [43]	-21.46	-10.94	-10.17	-16.52	-25.78	-26.03	-20.22	-16.71	-22.75	-33.75	-36.64	-21.21	-21.08	-28.92	-38.01
	<u>6.02</u>	<u>6.36</u>	<u>5.66</u>	<u>6.22</u>	<u>8.25</u>	<u>10.88</u>	<u>13.83</u>	<u>10.68</u>	<u>11.74</u>	<u>12.70</u>	<u>10.18</u>	<u>11.32</u>	<u>12.18</u>	<u>12.26</u>	<u>12.62</u>
BMSHJ MS-SSIM	93.94	-25.52	-34.04	50.77	-11.50	5.55	-32.53	-32.55	31.19	-20.90	-3.28	-37.82	-42.52	27.34	-25.74
	<u>77.19</u>	<u>21.98</u>	<u>25.04</u>	<u>15.20</u>	<u>20.33</u>	<u>12.77</u>	<u>34.11</u>	<u>38.83</u>	<u>32.73</u>	<u>24.01</u>	<u>18.47</u>	<u>14.43</u>	<u>13.93</u>	<u>34.49</u>	<u>16.75</u>
BMSHJ VMAF _p	-15.90	-13.57	-13.17	-17.02	-47.11	-19.64	-23.14	-16.73	-23.63	-53.18	-29.96	-18.87	-19.29	-29.00	-56.06
	<u>5.34</u>	<u>4.22</u>	<u>3.70</u>	<u>4.86</u>	<u>7.72</u>	<u>10.35</u>	<u>10.29</u>	<u>10.82</u>	<u>9.69</u>	<u>11.63</u>	<u>10.34</u>	<u>11.01</u>	<u>10.41</u>	<u>10.87</u>	<u>10.66</u>

B. Experimental Results on Kodak Dataset

The results on the images in the Kodak dataset are given Table I. The distortion levels that were used for BD-rate calculation were quantified using PSNR, SSIM, MS-SSIM, and VMAF. It may be observed that the RD performances that were measured using PSNR became worse on models optimized under other IQA models. This is not surprising since MSE, which is used by the baseline model, is the optimal loss function for PSNR. In fact, this phenomenon may also be related to the distortion-perception tradeoff described in [66]: given a fixed rate, the improvement of perceptual quality must be balanced against the loss in distortion. The second thing to notice is the performance attained by the SSIM_p and MS-SSIM_p optimizations. Both the SSIM and MS-SSIM BD-rates were improved significantly, as expected. Yet, the VMAF BD-rate performed slightly worse than the original MSE optimization, indicating there exists disagreements between VMAF and SSIM/MS-SSIM. Similar trends can also be observed from the MS-SSIM optimization given in Table III.

It may also be noted that, unlike using other IQA models as targets of the proposed optimization, VMAF_p optimization delivers coding gain with respect to all of the BD-rate measurements, except the PSNR BD-rate. Since PSNR is known not to correlate as well with human subjective quality judgments as do modern VQA models, the improvements on the PSNR

BD-rate may not always mean subjective improvements. This suggests that VMAF is a good optimization target.

C. Comparison With State-of-the-Art Codecs

Table III tabulates the percent change in BD-rate relative to the BLS baseline, with respect to different quality models. We comprehensively evaluated perceptual deep compression using different perceptual optimization protocols (highlighted in boldface), against three conventional image codecs: JPEG, JPEG2000, and intra-coded HEVC main-RExt (Format Range Extension) profile. Extensive experiments were carried out on the three aforementioned datasets, using three perceptual IQA models as optimization targets. In addition to the BLS model, we also deployed the proposed VMAF_p optimization framework on a more sophisticated deep compression model BMSHJ [43], representing the authors' first initials, to test its generality. We report the BD-rate changes obtained, averaged over all the images in each dataset. The reported standard deviation was also calculated from all the BD-rate changes within a test dataset. Similar results were obtained on the Tecnick and NFLX datasets, as shown in Tables I and III. These results show that our optimization approach is able to successfully optimize a deep image compression model relative to different IQA algorithms. Indeed, significant BD-rate reductions were obtained in many cases. We were able to demonstrate that the BLS model optimized by M_p achieved similar or better results than HEVC, when the result was

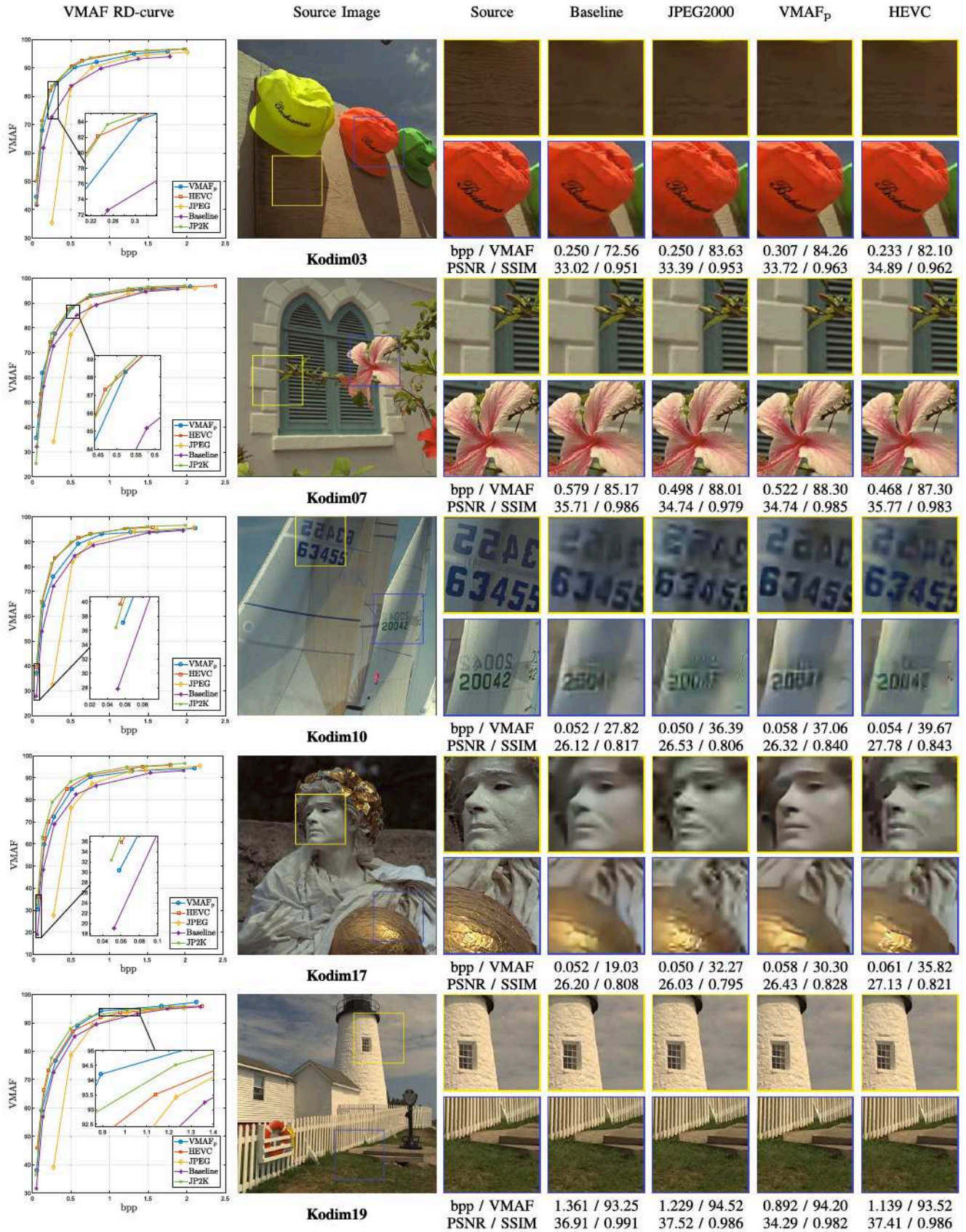


Fig. 8. Visual comparison of decoded images produced by different codecs as well as the corresponding VMAF RD-curve. Image crops from left to right: ground-truth, baseline model [42], JPEG2000, Ballé *et al.* optimized with VMAF_p (denoted by VMAF_p), and HEVC. Generally, the VMAF_p-optimized BLS model achieved visual quality comparable to intra HEVC and JPEG2000. The source images were cropped to resolution 512 × 512 in the second column for display purposes.

measured by the perceptual quality model M . It should be noted that the quality measurements made by SSIM, MS-SSIM, VIF, and VMAF only consider distortions of the luma channel, hence, the remaining scope for improvement by also accounting for the quality contributions of the chroma channels. Nevertheless, the VMAFp-optimized BMSHJ model still outperformed HEVC₄₂₀, in the sense of BD-rate as measured by VMAF.

In addition to the quantitative results, we also visually compared the decoded images. Fig. 8 plots VMAF Rate-distortion (RD) curves for several images. A subset of the images corresponding to the RD points obtained by the various codecs are also shown. As a basic test, we subjectively compare results yielding similar bitrates but different objective quality scores. The images Kodim10 and Kodim17 were subject to extreme compression at bitrates around 0.05 bpp. In these cases, the VMAFp-optimized model significantly outperformed the MSE-optimized baseline model, delivering performance comparable to HEVC and JPEG2000 with respect to VMAF score and subjective quality. At high bitrates, the distinctions between the codecs becomes subtle. Therefore, we isolated these RD points associated with similar VMAF scores and compared bitrate consumption. Generally, the VMAFp-optimized model yielded comparable subjective and objective (VMAF) quality as the baseline MSE-optimized model, while consuming significantly fewer bits. The encoding results on Kodim19 in Fig. 8 using the VMAFp-optimized model yielded similar VMAF scores as the other codecs, while consuming only 34.4%, 27.4% and 21.7% fewer bits than the Baseline, JPEG2000, and HEVC, respectively.

D. Subjective Quality Study

To better understand perceptual preferences on compressed images and the effects of perceptual optimization using ProxIQ, we conducted a human subject study where we compared deep compression engines using different optimization targets. Specifically, we randomly selected 10 image contents from the Kodak dataset. Each content was encoded at four different bitrates by VMAFp-optimized and MSE-optimized BLS models, resulting in eight different compression-distorted versions of that content. Since compressed images obtained using different algorithms often have subtle perceptual differences at similar bitrates, it is easier for humans to make comparisons between simultaneously displayed images. Therefore, we adopted the paired comparison method instead of the Absolute Category Rating (ACR) protocol in our experimental design. We designed a web-based user interface to carry out the human study, whereby participants could easily compare pairs of images and render relative quality decisions. On each image pair, the subject was asked to select the one they preferred. We recruited 41 volunteer participants, of whom about 70% were knowledgeable about image/video processing, while the others were naive. We asked each participant to compare 280 pairs of images, which required about 30 minutes. To obtain human opinion scores, we deployed the Bradley-Terry (BT) model [67] to estimate the Mean Opinion Score (MOS).

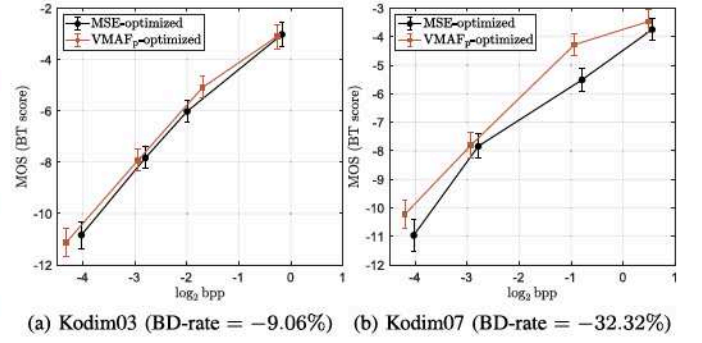


Fig. 9. Subjective performance comparison of rate-distortion curves (with 95% confidence intervals) for two different optimization protocols on two exemplar Kodak images. The y-axis denotes the MOS scores estimated by the BT model.

Given the collected subjective quality score and bitrate for each distorted picture, we compared the RD performances of the two optimization methods. On the 10 selected contents, the VMAFp-optimized model was able to achieve reduction of MOS BD-rate ranging from 9% to 32.3% relative to MSE optimization, with an average reduction of 17.7%. Fig. 9 shows the MOS RD curves of two contents, one of which was the most improved using VMAFp optimization, while the other was the least improved. It may be observed that the RD curves resulting from VMAFp optimization allow for reduced bitrates at similar levels of subjective quality, as compared to those produced by MSE optimization. For example, Fig. 9(b) shows that VMAFp delivers better predicted visual quality, with statistical significance, while consuming fewer bits at around 0.5 bpp. These results show that the proposed VMAFp optimization yields favorable compression results as compared to MSE optimization.

E. Limitations

When measuring RD performance using SSIM, optimizing a model using SSIM should approach the theoretical upper bound of SSIM-measured RD performance. Accordingly, we investigated the performance of our proposed framework by comparing SSIM and SSIM_p optimization of the BLS model. The results of comparison are presented in Table IV, indicating a 10% SSIM BD-rate performance gap between the two optimization approaches. A noticeable contributor to this performance drop is the pixel loss \mathcal{L}_d in (10). To validate this assumption, we conducted an ablation study to pinpoint the cause of this gap. We did this by fixing $\mathcal{L}_p = \mathcal{L}_{SSIM}$ for the SSIM optimization. The loss function is then

$$\mathcal{L} = \lambda [\alpha \mathcal{L}_{SSIM} + (1 - \alpha) \mathcal{L}_d] + \mathcal{L}_r, \quad (13)$$

where

$$\mathcal{L}_{SSIM} = 1 - SSIM(\mathbf{x}, \hat{\mathbf{x}}), \quad (14)$$

and $\alpha = 3e - 3$ is the same as in (10). The SSIM BD-rate that resulted from optimization of (13) is given in the fourth column of Table IV. It may be observed that the RD performance becomes very close to that of SSIM_p optimization,

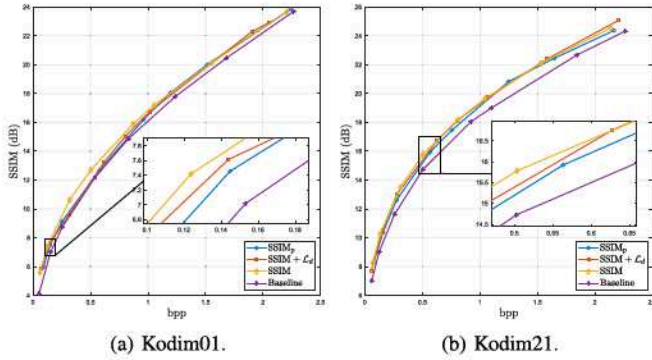


Fig. 10. Rate-distortion curves for different SSIM-oriented optimization protocols on two Kodak images. The baseline curve denotes the MSE-optimized BLS model [42].

TABLE IV

AVERAGE CHANGE IN BD-RATE (IN PERCENTAGE) OF DIFFERENT SSIM-DRIVEN OPTIMIZATION RESULTS ON KODAK DATASET

	SSIM (14)	SSIM _p (10)(11)	SSIM+ \mathcal{L}_d (13)
PSNR BD-Rate	133.79	15.56	15.43
SSIM BD-Rate	-29.41	-19.04	-22.39
MS-SSIM BD-Rate	-11.74	-17.29	-19.69
VMAF BD-Rate	40.09	7.63	12.03

which confirms that the pixel loss is the main contributor to the performance loss.

Moreover, Fig. 11 shows that similar visual results are obtained using SSIM_p optimization and the optimization described in (13), even at heavy compression levels. A close examination shows that true SSIM optimization nicely preserves high-frequency details but loses chromatic fidelity. The RD-curves in Fig. 10 further confirm the similar behavior of SSIM_p optimization and the optimization of (13). We present SSIM in decibels for readability, since small quantitative differences in SSIM may be associated with large visual differences.

All the models described in this subsection were trained using one million steps and a constant learning rate. Thus, the performance results of SSIM_p differ slightly from the results reported in Tables I and III.

F. Study of Training Steps

The instability introduced by the proxy loss can be further improved by training longer, and by reducing the learning rate. Fig. 12 plots the VMAF BD-rate as a function of the number of training steps. When measuring BD-rate against the same baseline (MSE optimization trained with 1M steps), VMAF_p achieves significant improvement relative to MSE optimization, by training longer or by lowering the learning rate. For fair comparison, we also evaluate VMAF_p using MSE optimization using the same training process as the baseline (dotted line). We have observed very relative results using other perceptual optimizers, like SSIM_p.

G. Execution Time

The encoding and decoding times of the various compared codecs are summarized in Table V. We compiled the source



Fig. 11. Visual comparison of model behavior among different SSIM-optimized models. First row: Kodim01 encoded at 0.12 bpp. Second row: Kodim21 encoded at 0.5 bpp.

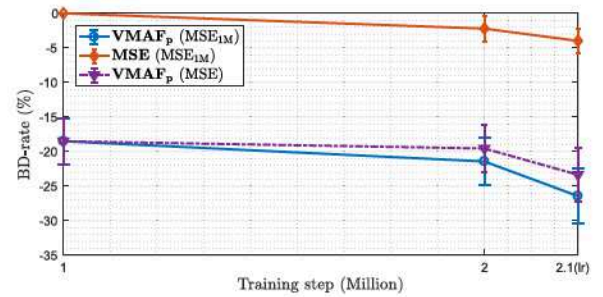


Fig. 12. VMAF BD-rate change (improvement) against the number of training steps and learning rate, using the Kodak images. The error bars represent the standard deviations of BD-rates. In the first 2M steps, a constant learning rate was used. After that, the learning rate was reduced by a factor of 10. We denote each result by “optimization (baseline)”. For BD-rate change calculations, the solid lines indicate MSE optimization over 1M steps, while the dotted line indicates that MSE optimization using the same training procedure as the optimization is used for baseline comparison.

TABLE V

RUN TIME COMPARISON OF CONVENTIONAL IMAGE CODECS AND DEEP COMPRESSION MODELS. MODEL LOADING TIME FOR DEEP COMPRESSION IS EXCLUDED. ALL TIMES ARE GIVEN IN MILLISECONDS

Codec		Encode	Decode
JPEG	CPU	43.02	62.88
JPEG2000	CPU	10.80	36.79
HEVC	CPU	4578.57	89.88
BLS MSE	CPU	251.01	117.93
	GPU	231.62	32.56
BLS VMAF _p (ours)	CPU	246.57	119.02
	GPU	229.26	29.22

code of state-of-the-art (standard) codecs, in order to be able to compare them on the same machine. The results were then calculated by averaging the runtime over all 24 Kodak images under different bitrate settings. From Table V, it may be observed that the time complexity of the MSE-optimized and VMAF_p-optimized BLS model are nearly identical, as they deploy the same network architecture in application. Furthermore, the encoding processes consume more time than

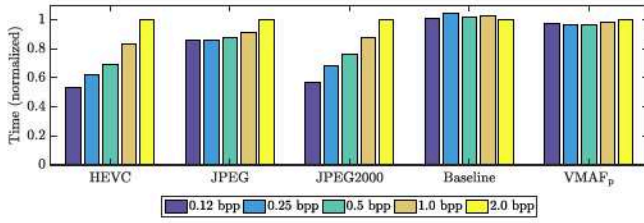


Fig. 13. Normalized encoding times for five different bitrate settings.

decoding, since the synthesis transform (g_s) is also included. Of course, the runtime of deep compression models can be reduced if implemented on a GPU. It should also be noted that the decoding time of HEVC was estimated from the reference software HM, which might be slow. This can be improved by using a third-party decoder such as FFmpeg.

We also compared the encoding times for different bitrate settings in Fig. 13. For each encoder, all of the runtimes were normalized by dividing by the value of 2.0 bpp. The conventional codecs required more time to encode at high bitrates, whereas the deep compression models have encoding times that do not vary much with bitrate.

V. CONCLUDING REMARKS

We have presented a learning framework for perceptually optimizing a learned image compression model. To optimize the ProxIQ network, we developed an alternating training method. We experimentally demonstrated that, for a fixed VMAF value, our proposed proxy approach achieved a 20% bitrate reduction, on average, relative to the MSE-based framework.

The idea behind the proposed optimization framework is general. We believe that, with proper modifications of the architecture of the ProxIQ network, the application scope should be applicable to a wide variety of image enhancement, restoration, and reconstruction problems, such as super-resolution, de-banding [68], [69], or de-noising.

Another future topic could be the study of new types of distortions caused by deep compression models. Like the examples shown in this paper, distorted images created by CNNs are very different from images afflicted by more traditional distortions, such as JPEG compression. Creating databases for assessing the subjective quality of these new types of distortions would be quite valuable.

ACKNOWLEDGMENT

The authors thank Johannes Ballé for providing the training images. The authors also thank the Texas Advanced Computing Center (TACC) at The University of Texas at Austin for providing HPC resources that have contributed to the research results reported within this article. URL: <http://www.tacc.utexas.edu>

REFERENCES

- [1] E. Shelhamer, J. Long, and T. Darrell, "Fully convolutional networks for semantic segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.
- [2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.
- [3] T.-M. Liu *et al.*, "A 0.76 mm2 0.22 nJ/pixel DL-assisted 4K video encoder LSI for quality-of-experience over smartphones," *IEEE Solid-State Circuits Lett.*, vol. 1, no. 12, pp. 221–224, Dec. 2018.
- [4] S. Paul, A. Norkin, and A. C. Bovik, "Speeding up VP9 intra encoder with hierarchical deep learning-based partition prediction," *IEEE Trans. Image Process.*, vol. 29, pp. 8134–8148, 2020.
- [5] A. Dosovitskiy *et al.*, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.
- [6] H. C. Burger, C. J. Schuler, and S. Harmeling, "Image denoising: Can plain neural networks compete with BM3D?" in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2392–2399.
- [7] V. Jain and S. Seung, "Natural image denoising with convolutional networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 769–776.
- [8] C. Dong, C. C. Loy, K. He, and X. Tang, "Learning a deep convolutional network for image super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 184–199.
- [9] W.-S. Lai, J.-B. Huang, N. Ahuja, and M.-H. Yang, "Deep Laplacian pyramid networks for fast and accurate super-resolution," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 624–632.
- [10] Y.-L. Liu, Y.-T. Liao, Y.-Y. Lin, and Y.-Y. Chuang, "Deep video frame interpolation using cyclic frame generation," in *Proc. AAAI*, 2019, pp. 8794–8802.
- [11] J. T. Barron, "A general and adaptive robust loss function," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4331–4339.
- [12] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
- [13] Z. Wang, E. P. Simoncelli, and A. C. Bovik, "Multi-scale structural similarity for image quality assessment," in *Proc. IEEE Asilomar Conf. Signals, Syst., Comput.*, Nov. 2003, pp. 1398–1402.
- [14] J. Snell, K. Ridgeway, R. Liao, B. D. Roads, M. C. Mozer, and R. S. Zemel, "Learning to generate images with perceptual similarity metrics," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 4277–4281.
- [15] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, "Loss functions for image restoration with neural networks," *IEEE Trans. Comput. Imag.*, vol. 3, no. 1, pp. 47–57, Mar. 2017.
- [16] Z. Wan, K. Gu, and D. Zhao, "Reduced reference stereoscopic image quality assessment using sparse representation and natural scene statistics," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 2024–2037, Aug. 2020.
- [17] Z. Tu, Y. Wang, N. Birkbeck, B. Adsumilli, and A. C. Bovik, "UGC-VQA: Benchmarking blind video quality assessment for user generated content," 2020, *arXiv:2005.14354*. [Online]. Available: <http://arxiv.org/abs/2005.14354>
- [18] D. M. Chandler and S. S. Hemami, "VSNR: A wavelet-based visual Signal-to-Noise ratio for natural images," *IEEE Trans. Image Process.*, vol. 16, no. 9, pp. 2284–2298, Sep. 2007.
- [19] H. R. Sheikh and A. C. Bovik, "Image information and visual quality," *IEEE Trans. Image Process.*, vol. 15, no. 2, pp. 430–444, Feb. 2006.
- [20] E. C. Larson and D. M. Chandler, "Most apparent distortion: Full-reference image quality assessment and the role of strategy," *J. Electron. Imag.*, vol. 19, no. 1, Jan. 2010, Art. no. 011006.
- [21] L. Zhang, L. Zhang, X. Mou, and D. Zhang, "FSIM: A feature similarity index for image quality assessment," *IEEE Trans. Image Process.*, vol. 20, no. 8, pp. 2378–2386, Aug. 2011.
- [22] L. Zhang, Y. Shen, and H. Li, "VSI: A visual saliency-induced index for perceptual image quality assessment," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4270–4281, Oct. 2014.
- [23] T.-J. Liu, W. Lin, and C.-C.-J. Kuo, "Image quality assessment using multi-method fusion," *IEEE Trans. Image Process.*, vol. 22, no. 5, pp. 1793–1807, May 2013.
- [24] S.-C. Pei and L.-H. Chen, "Image quality assessment using human visual DOG model fused with random forest," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 3282–3292, Nov. 2015.
- [25] V. V. Lukin, N. N. Ponomarenko, O. I. Jeremeiev, K. O. Egiazarian, and J. Astola, "Combining full-reference image visual quality metrics by neural network," *Proc. SPIE*, vol. 9394, Mar. 2015, Art. no. 93940K.
- [26] M. Oszust, "Decision fusion for image quality assessment using an optimization approach," *IEEE Signal Process. Lett.*, vol. 23, no. 1, pp. 65–69, Jan. 2016.

- [27] F. Gao, Y. Wang, P. Li, M. Tan, J. Yu, and Y. Zhu, "DeepSim: Deep similarity for image quality assessment," *Neurocomputing*, vol. 257, pp. 104–114, Sep. 2017.
- [28] S. Bosse, D. Maniry, K.-R. Müller, T. Wiegand, and W. Samek, "Deep neural networks for no-reference and full-reference image quality assessment," *IEEE Trans. Image Process.*, vol. 27, no. 1, pp. 206–219, Jan. 2018.
- [29] S. Bosse, S. Becker, K.-R. Müller, W. Samek, and T. Wiegand, "Estimation of distortion sensitivity for visual quality prediction using a convolutional neural network," *Digit. Signal Process.*, vol. 91, pp. 54–65, Aug. 2019.
- [30] Z. Li *et al.* VMAF: The Journey Continues. NETFLIX Tech. Blog. [Online]. Available: <https://netflixtechblog.com/vmaf-the-journey-continues-44b51ee9ed12>
- [31] Z. Sinno, A. K. Moorthy, J. De Cock, Z. Li, and A. C. Bovik, "Quality measurement of images on mobile streaming interfaces deployed at scale," *IEEE Trans. Image Process.*, vol. 29, pp. 2536–2551, 2020.
- [32] D. Brunet, E. R. Vrscay, and Z. Wang, "On the mathematical properties of the structural similarity index," *IEEE Trans. Image Process.*, vol. 21, no. 4, pp. 1488–1499, Apr. 2012.
- [33] K. Ding, K. Ma, S. Wang, and E. P. Simoncelli, "Comparison of image quality models for optimization of image processing systems," 2020, *arXiv:2005.01338*. [Online]. Available: <http://arxiv.org/abs/2005.01338>
- [34] R. Zhang, P. Isola, A. A. Efros, E. Shechtman, and O. Wang, "The unreasonable effectiveness of deep features as a perceptual metric," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 586–595.
- [35] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–14.
- [36] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9906, 2016, pp. 694–711.
- [37] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 3730–3738.
- [38] J. Bruna, P. Sprechmann, and Y. LeCun, "Super-resolution with deep convolutional sufficient statistics," in *Proc. Int. Conf. Learn. Represent.*, 2016, pp. 1–17.
- [39] C. Ledig *et al.*, "Photo-realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.
- [40] M. S. M. Sajjadi, B. Scholkopf, and M. Hirsch, "EnhanceNet: Single image super-resolution through automated texture synthesis," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 4491–4500.
- [41] C. Yang, X. Lu, Z. Lin, E. Shechtman, O. Wang, and H. Li, "High-resolution image inpainting using multi-scale neural patch synthesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4076–4084.
- [42] J. Ballé, V. Laparra, and E. P. Simoncelli, "End-to-end optimized image compression," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–27.
- [43] J. Ballé, D. Minnen, S. Singh, S. J. Hwang, and N. Johnston, "Variational image compression with a scale hyperprior," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–23.
- [44] D. Minnen, J. Ballé, and G. D. Toderici, "Joint autoregressive and hierarchical priors for learned image compression," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 10771–10780.
- [45] G. Toderici *et al.*, "Variable rate image compression with recurrent neural networks," 2015, *arXiv:1511.06085*. [Online]. Available: <https://arxiv.org/abs/1511.06085>
- [46] G. Toderici *et al.*, "Full resolution image compression with recurrent neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5306–5314.
- [47] N. Johnston *et al.*, "Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4385–4393.
- [48] E. Agustsson, M. Tschannen, F. Mentzer, R. Timofte, and L. Van Gool, "Generative adversarial networks for extreme learned image compression," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 221–231.
- [49] J. Löhdefink, A. Bär, N. M. Schmidt, F. Hüger, P. Schlicht, and T. Fingscheidt, "GAN- vs. JPEG2000 image compression for distributed automotive perception: Higher peak SNR does not mean better semantic segmentation," 2019, *arXiv:1902.04311*. [Online]. Available: <http://arxiv.org/abs/1902.04311>
- [50] C.-Y. Wu, N. Singhal, and P. Krähenbühl, "Video compression through image interpolation," in *Proc. Eur. Conf. Comput. Vis.*, Sep. 2018, pp. 416–431.
- [51] Z. Cheng, H. Sun, M. Takeuchi, and J. Katto, "Learning image and video compression through spatial-temporal energy compaction," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10071–10080.
- [52] S. Kim *et al.*, "Adversarial video compression guided by soft edge detection," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2193–2197.
- [53] S. S. Channappayya, A. C. Bovik, and R. W. Heath, "Rate bounds on SSIM index of quantized images," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1624–1639, Sep. 2008.
- [54] Y.-H. Huang, T.-S. Ou, P.-Y. Su, and H. H. Chen, "Perceptual rate-distortion optimization using structural similarity index as quality metric," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 11, pp. 1614–1624, Nov. 2010.
- [55] S. Wang, A. Rehman, Z. Wang, S. Ma, and W. Gao, "SSIM-motivated rate-distortion optimization for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 516–529, Apr. 2012.
- [56] K.-S. Lu, A. Ortega, D. Mukherjee, and Y. Chen, "Perceptually inspired weighted MSE optimization using irregularity-aware graph Fourier transform," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2020, pp. 3384–3388.
- [57] I. Goodfellow *et al.*, "Generative adversarial nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.
- [58] J. Balle, "Efficient nonlinear transforms for lossy image compression," in *Proc. Picture Coding Symp. (PCS)*, Jun. 2018, pp. 248–252.
- [59] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.
- [60] K. Ma *et al.*, "Waterloo exploration database: New challenges for image quality assessment models," *IEEE Trans. Image Process.*, vol. 26, no. 2, pp. 1004–1016, Feb. 2017.
- [61] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.
- [62] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
- [63] (2007). *Kodak Lossless True Color Image Suite*. [Online]. Available: <http://r0k.us/graphics/kodak/>
- [64] N. Asuni and A. Giachetti, "TESTIMAGES: A large-scale archive for testing visual devices and basic image processing algorithms," in *Proc. Eurographics Italian Chapter Conf.*, 2014, pp. 63–70.
- [65] G. Bjøntegaard, *Calculation of Average PSNR Differences Between RD-Curves*, document VCEG-M33, ITU-T Video Coding Experts Group (VCEG) 13th Meeting, Austin, TX, USA, Apr. 2001.
- [66] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6228–6237.
- [67] R. A. Bradley and M. E. Terry, "Rank analysis of incomplete block designs: The method of paired comparisons," *Biometrika*, vol. 39, nos. 3–4, pp. 324–345, Dec. 1952.
- [68] Z. Tu, J. Lin, Y. Wang, B. Adsumilli, and A. C. Bovik, "BBAND index: A no-reference banding artifact predictor," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2020, pp. 2712–2716.
- [69] Z. Tu, J. Lin, Y. Wang, B. Adsumilli, and A. C. Bovik, "Adaptive debanding filter," *IEEE Signal Process. Lett.*, vol. 27, pp. 1715–1719, 2020.



Li-Heng Chen (Graduate Student Member, IEEE) received the B.S. degree in electronics engineering from National Chiao Tung University, Hsinchu, Taiwan, and the M.S. degree in communication engineering from National Taiwan University, Taipei, Taiwan, in 2012 and 2014, respectively. He is currently pursuing the Ph.D. degree with the Laboratory for Image and Video Engineering (LIVE), The University of Texas at Austin. He was a Senior Engineer with the Video Encoder Team, MediaTek Inc., from 2014 to 2018. His research interests include perceptual image and video quality assessment, image and video compression, and machine learning.



Christos G. Bampis (Member, IEEE) received the M.Eng. diploma degree in electrical engineering from the National Technical University of Athens in 2014 and the Ph.D. degree in electrical engineering from The University of Texas at Austin in 2018. He is currently a Senior Software Engineer with the Video Algorithms Team, Netflix. His work interests include research and development of perceptual video quality prediction systems. His research interests include image and video quality assessment, computer vision, and machine learning.



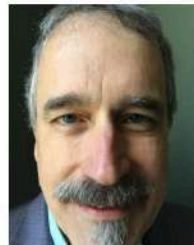
Andrey Norkin (Member, IEEE) received the M.Sc. degree in computer engineering from Ural State Technical University, Yekaterinburg, Russia, in 2001, and the D.Sc. degree in signal processing from the Tampere University of Technology, Tampere, Finland, in 2007.

From 2008 to 2015, he was with Ericsson, Sweden, conducting research on video compression and 3D video. In 2014, he worked on video encoding techniques for TV broadcasting products at Ericsson TV, Southampton, U.K. Since 2015, he has been with Netflix, Los Gatos, CA, USA, as a Senior Research Scientist working on encoding techniques for OTT video streaming, High Dynamic Range (HDR) video, and new video compression algorithms. He has participated in ITU-T and MPEG efforts on developing video compression standards, including HEVC, its extensions, and VVC, having multiple technical contributions and coordinating work in certain areas, such as deblocking filters. He has been actively contributing to the Alliance for Open Media (AOM) development of the AV1 video codec. He is currently a Co-Chair of the AOM Codec Working Group. His research interests include video compression, OTT streaming, HDR video, and machine learning techniques.



Zhi Li received the B.Eng. and M.Eng. degrees in electrical engineering from the National University of Singapore, Singapore, in 2005 and 2007, respectively, and the Ph.D. degree from Stanford University, Stanford, CA, USA, in 2012. He is currently with the Encoding Technologies, Netflix. His current interests include improving video streaming experience for consumers through understanding how humans perceive video and applying that knowledge to encoding/streaming design and optimization. He has broad interests in applying mathematics in

solving real-world engineering problems. His Ph.D. work focuses on source and channel coding methods in multimedia networking problems. He was a recipient of the Best Student Paper Award at IEEE ICME 2007 for his work on cryptographic watermarking, and a co-recipient of Multimedia Communications Best Paper Award from IEEE Communications Society in 2008 for a work on multimedia authentication.



Alan C. Bovik (Fellow, IEEE) is currently the Cockrell Family Regents Endowed Chair Professor with The University of Texas at Austin. His research interests include image processing, digital photography, digital television, digital streaming video, and visual perception. For his work in these areas he has been the recipient of the 2019 Progress Medal from The Royal Photographic Society, the 2019 IEEE Fourier Award, the 2017 Edwin H. Land Medal from The Optical Society, the 2015 Primetime Emmy Award for Outstanding Achievement in Engineering

Development from the Television Academy, and the Norbert Wiener Society Award and the Karl Friedrich Gauss Education Award from the IEEE Signal Processing Society. He has also received about ten Best Journal Paper awards, including the 2016 IEEE Signal Processing Society Sustained Impact Award. His books include *The Essential Guides to Image and Video Processing*. He co-founded and was longest-serving Editor-in-Chief of the IEEE TRANSACTIONS ON IMAGE PROCESSING, and also created/Chaired the IEEE International Conference on Image Processing which was first held in Austin, Texas, 1994.