Observer Dependent Lossy Image Compression

Maurice Weber$^{1(\S3)}$, Cedric Renggli$^1$, Helmut Grabner$^2$, and Ce Zhang$^1$

$^1$ Department of Computer Science, ETH Zürich, Switzerland
maurice.weber@inf.ethz.ch
$^2$ ZHAW School of Engineering, Winterthur, Switzerland

Abstract. Deep neural networks have recently advanced the state-of-the-art in image compression and surpassed many traditional compression algorithms. The training of such networks involves carefully trading off entropy of the latent representation against reconstruction quality. The term quality crucially depends on the observer of the images which, in the vast majority of literature, is assumed to be human. In this paper, we aim to go beyond this notion of compression quality and look at human visual perception and image classification simultaneously. To that end, we use a family of loss functions that allows to optimize deep image compression depending on the observer and to interpolate between human perceived visual quality and classification accuracy, enabling a more unified view on image compression. Our extensive experiments show that using perceptual loss functions to train a compression system preserves classification accuracy much better than traditional codecs such as BPG without requiring retraining of classifiers on compressed images. For example, compressing ImageNet to 0.25 bpp reduces Inception-ResNet classification accuracy by only 2%. At the same time, when using a human friendly loss function, the same compression system achieves competitive performance in terms of MS-SSIM. By combining these two objective functions, we show that there is a pronounced trade-off in compression quality between the human visual system and classification accuracy.

1 Introduction

Image compression algorithms aim at finding representations of images that use as little storage—measured in bits—as possible. Opposed to lossless image compression, where the goal is to achieve a high compression rate while requiring perfect reconstruction, lossy image compression enables even higher compression rates by allowing for a loss in reconstruction quality. Recently, image compression based on deep neural networks (DNNs) has achieved remarkable results in both lossless [33] and lossy image compression [2,4,32,35,41,43], outperforming many traditional codecs. One distinct advantage of such methods is their flexibility with regards to the term reconstruction quality which crucially depends on the observer.

Electronic supplementary material The online version of this chapter (https://doi.org/10.1007/978-3-030-71278-5_10) contains supplementary material, which is available to authorized users.
on the observer of the compressed images. Previous research in lossy image compression expressed quality largely in terms of human visual perception and optimized for the human visual system (HVS), using distortion measures such as multiscale structural similarity [47] (MS-SSIM) or mean squared error (MSE) as training objectives. However, due to recent advances in computer vision systems, increasingly more images are observed solely by machines and bypass humans. Consequently, a natural question that arises is whether or not there exists a relation between quality perceived by humans and quality perceived by computer vision systems, and if so, how can we trade off quality between different types of observers? In other words, is a compression system optimized for the human observer also optimal for machines? We investigate these questions by specifically looking at classification of natural images as one of the most well studied tasks in computer vision. The training of modern classifiers is typically a costly and time-consuming undertaking and parameters of the best performing classifiers are often publicly available. With that in mind, we are interested in a compression system that generalizes well in the following sense. Firstly, we want to compress images such that no retraining of classifiers on compressed images is required. Secondly, the compression system should be agnostic to classifier architectures. Thirdly, it should also generalize well to other visual tasks such as fine-grained visual categorization of natural images. Together, these generalization requirements encourage using publicly available, pretrained classifiers on compressed images from the same domain or on related tasks where classifiers were obtained with transfer learning. Our method for classification oriented compression relies on a feature reconstruction loss using deep features extracted from the hidden layers of a convolutional neural network trained for image classification. This type of loss function has been used in the context of super-resolution [7, 21, 27], style-transfer [15, 21] and variational autoencoders [12] with remarkable success.
In order to optimize for human visual perception, we make use of MS-SSIM as a measure of quality perceived by humans, since it has been reported to correlate better with the HVS than MSE. Finally, the convex combination of the two objectives allows to investigate the trade-off between human visual perception and classification in the context of image compression. In summary, the contributions of our work are threefold:

- We show that training deep image compression with a perceptual loss function preserves classification accuracy much better than human optimized compression systems. In addition, our experiments show that (1) we do not have to retrain classifiers on compressed images in order to preserve accuracy on highly compressed images, and (2) using VGG-based feature reconstruction loss generalizes to other models, indicating that deep CNN features are shared between CNN architectures.
- By looking at the convex combination between human and classification-friendly loss, we present a simple way to trade off compression quality in terms of human perception against image classification. Since we only rely on the training objective, our method can be integrated to any learned lossy image compression system.
- Our extensive experimental study indicates that there exists a pronounced trade-off between compression quality perceived by the human observer and classification accuracy. We show how improved compression quality for the human observer comes at the cost of degraded classification accuracy, and vice versa.

We emphasize that the contribution of this work is not presenting a new type of loss function nor in a new deep compression architecture. Rather, we make use of existing techniques in order to present a method to trading off compression quality depending on the observer and to show that it is possible to explicitly optimize compression for subsequent classification.

2 Related Work

Deep Image Compression. Image compression using DNNs has recently become an active area of research. The most popular types of architectures used for image compression are based on autoencoders [2,4,32,35,41] and recurrent neural networks [22,42,43] (RNNs). Typically, the networks are trained in an end-to-end manner to minimize a pixel-wise notion of distortion such as MSE, MS-SSIM or $L_1$-distance between original and decoded image.

Compression for Computer Vision. Image compression in combination with other computer vision tasks has been studied in a number of recent works. Liu et al. [29] propose an image compression framework based on JPEG that is favorable to DNN classifiers. Also starting from an engineered codec, Liu et al. [30] propose a 3D image compression framework based on JPEG2000 which is tailored to segmentation of 3-D medical images. Both works differ from ours in that we look at learned image compression, rather than modifying an engineered one.
A few examples exist in the literature, where a classifier is learned from features extracted from the encoded representations. Gueguen et al. [17] train a modified ResNet-50 directly on the blockwise discrete cosine transform coefficients from the middle of the JPEG codec. Torfason et al. [44] make use of the compressive autoencoder proposed in [41] and train neural networks for classification and segmentation on the latent (quantized) representations and on the decoded images. These works stand orthogonal to ours in that we do not allow training on compressed versions of images. Rather, we train the compression algorithm such that it maintains information relevant for subsequent classification, keeping the classifiers fixed. We furthermore focus on agnosticism to architectures of inference algorithms. Finally, since compression artifacts typically compromise the performance of classifiers, Dodge and Karam [11] study the effect of JPEG compression on image classification with neural networks.

**Feature Reconstruction Loss.** This class of similarity functions makes use of deep features extracted from convolutional neural networks. Recent advances in generative modelling have shown that using this type of loss functions, high quality images can be generated and have been applied to a variety of tasks. Gatys et al. [14,15] apply the idea to style transfer and texture synthesis, while Johnson et al. [21] and Bruna et al. [7] achieve remarkable results in super resolution [7,21] and style transfer [21]. Ledig et al. [27] further develop the idea and enhance the CNN feature loss with adversarial training to achieve state-of-the-art results in single image super resolution. In the image compression domain, steps in this direction have also been made. Agustsson et al. [3], Santurkar et al. [37] and Liu et al. [28] enhance pixel-wise distortion and adversarial training with a feature reconstruction loss. Furthermore, Chinen et al. [8] and Zhang et al. [48] both propose a similarity metric based on deep features extracted from VGG-16 trained for image classification. These works have in common that their focus is on the human observer, while we exploit properties of feature reconstruction loss in the context of compression geared towards subsequent image classification. Feature reconstructions loss has also been used in the context of compression artifact removal. Galteri et al. [13] train a generative adversarial network in combination with a VGG-based perceptual loss function to remove compression artifacts in images. It is shown that this can significantly increase the quality of compressed images in terms of MS-SSIM and in terms of object detection accuracy. However, contrary to our work, no clear trade-off between the human observer and image classification is investigated.

## 3 Method

In this section, we outline our approach to compressing images for human visual perception, classification accuracy and the interpolation between the two. Throughout this paper we adopt the compression architecture proposed by Toderici et al. [43], based on recurrent neural networks. We emphasize that we only focus on the objective functions to account for different types of observers.

**Compression Framework.** Let $\mathcal{X} \subseteq \mathbb{R}^d$ denote a set of training images, $\mathcal{Z} \subseteq \mathbb{Z}$ the quantization levels and $d: \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}$ a notion of distortion between images.
Our goal is to find a compression system consisting of an encoder $E: \mathbb{R}^d \rightarrow \mathbb{R}^m$ that maps input images $x$ to their latent representation $z = E(x)$, a quantizer $q: \mathbb{R}^m \rightarrow \mathbb{Z}^m$ that discretizes $z$ to $\hat{z} = q(z)$, and a decoder $D: \mathbb{Z}^m \rightarrow \mathbb{R}^d$ that maps the quantized representation back to image space, $\hat{x} = D(\hat{z})$. The goal is then to minimize the rate-distortion trade-off over the training set $\mathcal{X}$, i.e. for $\beta \geq 0$, we want to minimize $\sum_{x \in \mathcal{X}} d(x, \hat{x}) + \beta H(\hat{z})$, where $H$ denotes the entropy. As a compression architecture, we adopt the RNN-based model proposed in [43] with gated recurrent units (GRUs), allowing for variable bitrates. An input image $x$ is passed through the encoder and quantizer, mapping the latent codes stochastically to $\mathbb{Z}^m = \{-1, +1\}^m$. The quantized representation is subsequently decoded, yielding an estimate of the original image. This is repeated with the residual error fed to the encoder to obtain an estimate at the next bitrate, using information from the hidden states of the previous iterations. Formally, a single iteration at unrolling step $t \geq 1$, can be represented as $\hat{x}_t = \hat{x}_{t-1} + D_t(Q(E_t(r_t)))$ with $\hat{x}_0 = 0$ and $r_1 = x$ and where $E_t$ and $D_t$ are encoder and decoder carrying information from the previous unrolling steps. Finally, we remark that, since $\mathbb{Z}$ contains a finite number of quantization levels, we set $\beta = 0$ in the training objective.

**Optimizing for Human Visual Perception.** In order to optimize the compression system for the human observer, we choose a measure of distortion that approximately models human visual perception. The multiscale structural similarity index (MS-SSIM) [47] is based on the assumption that the human eye is adapted for extracting structural information from images and incorporates image details at multiple resolutions. It is furthermore reported to correlate better with human visual perception than MSE. Since MS-SSIM is differentiable, we follow [22,32,35] and minimize directly $d_H(x, \hat{x}) = 1 - \text{MS-SSIM}(x, \hat{x})$. We refer to compression optimized with $d_H$ as **RNN-H**. An alternative approach would be to use other, human-centric distortion metrics such as LPIPS [48] or the approach proposed in [8]. However, as these approaches are based on CNNs they bear the additional challenge of dealing with checkerboard-like artifacts [34].

**Optimizing for Classification.** Suppose we are given a CNN classifier $f$ trained on a set of images and labels $(\mathcal{X}', \mathcal{Y}')$ and corresponding training and validation splits $(\mathcal{X}'_{\text{train}}, \mathcal{Y}'_{\text{train}})$ and $(\mathcal{X}'_{\text{val}}, \mathcal{Y}'_{\text{val}})$. When we optimize compression for classification accuracy, we are interested in finding an encoder, quantizer and decoder such that the accuracy evaluated on the decoded validation set $D(q(E(\mathcal{X}'_{\text{val}})))$ is maintained as well as possible, without further retraining the classifier on decoded images. Formally, we wish to maximize $\sum_{x \in \mathcal{X}'_{\text{val}}} 1\{f(x) = f(\hat{x})\}$. We are thus not interested in matching decoded and original images on a pixel-wise basis, but rather on preserving features which are relevant for subsequent classification. Image classification is a task which is typically invariant to translations and local deformations (see e.g. [6,31]), which motivates the use of an objective function with similar properties. For example, using a pixel-wise distortion, such as MSE, which is not invariant to such deformations would be a suboptimal choice. Furthermore, minimizing MSE encourages the generator to produce
images that are pixel-wise averages of plausible solutions [12], resulting in overly smooth images. In other words, high frequency information such as textures will tend to get lost in the compression process. While this is less problematic for the HVS, which is more susceptible to low frequency changes, CNNs are sensitive to any change in frequency [29].

Features learned by convolutional neural networks [26] for image classification provide a promising alternative. The intuition is that, if such features are maintained in the compression process, then the compressed representations are encouraged to encode information relevant to classification rather than to the human observer. Moreover, it is known that CNNs provide stability to small geometric deformations and translations, thanks to rectification and pooling units [6]. This is beneficial for our purpose, since we do not want to put too much emphasis on such deformations as they do not affect classification. Finally, feature reconstruction loss typically leads to high frequency artifacts ([12] and references therein) and checkerboard patterns [34]. While this harms human perceived visual quality, our experiments indicate that this is not the case for classification. These considerations make distortion measures based on CNN features promising candidates for classification oriented image compression.

In order to define a distortion measure that incorporates these properties, we fix a CNN classifier \( f_L \) trained on a dataset \((X'', Y'')\). Denote by \( \phi_i \) the responses of the i-th convolutional layer after activation and let \( I \) be a set of such layers. Note that \( I \) is not required to include all layers. We then define the distortion measure associated with the loss network \( f_L \) and layers \( I \) to be MSE in feature space

\[
d_{C, I}(x, \hat{x}) = \sum_{i \in I} \gamma_i \| \phi_i(x) - \phi_i(\hat{x}) \|_2^2,
\]

where \( \gamma_i := (H_i \times W_i \times C_i)^{-1} \) and \( H_i, W_i, C_i \) represent the spatial dimensions of the corresponding layer. Note that we do not restrict the loss network to be trained on the same dataset as the compression system or the classifier \( f \), however we do require that \( X'' \cap X'_\text{val} = \emptyset \). Furthermore, the classifier \( f \) might have a different underlying architecture than the loss network \( f_L \). This formulation allows to investigate the generalizability of the compression system to new datasets and CNN architectures. We refer to compression optimized with \( d_{C, I} \) as RNN-C.

From Human Visual Perception to Classification. In a scenario where images are consumed by both humans and classifiers, we would like to be able to trade off reconstruction quality between the two observers. In other words, we want to have a compressed representation of an image that contains features relevant for classification and looks visually pleasing for the human observer. At the same time, this enables us to investigate the relation between human visual perception and classification accuracy. For that purpose, we consider the convex combination between distortions \( d_H \) and \( d_{C, I} \)

\[
d_{\alpha, I}(x, \hat{x}) = (1 - \alpha) \cdot \lambda_H \cdot d_H(x, \hat{x}) + \alpha \cdot d_{C, I}(x, \hat{x})
\]
Fig. 2. Sample image from the Stanford Dogs dataset. RNN-H results in smoother and blurrier images, RNN-C on the other hand produces sharp images but suffers from checkerboard-like artifacts stemming from the CNN based loss function.

and control the trade-off with the parameter $\alpha \in [0, 1]$. The parameter $\lambda_H$ is a scaling parameter which keeps the two losses on the same magnitude and is set to 5,000. We refer to compression optimized with $d_{\alpha, I}$ as RNN-$\alpha$.

4 Experiments

In this section we experimentally validate our approach to trading off compression quality between human visual perception and classification accuracy, making use of the proposed family of loss functions. All models are implemented in Python using the Tensorflow [1] library.$^1$

Image Compression. We use the RNN compression architecture proposed by Toderici et al. [43] with GRUs and the additive reconstruction framework. Our implementation differs from the original version in two aspects. Firstly, during training, we feed as input the full resolution images, rather than 32×32 image patches. And secondly, instead of optimizing the $L_1$-distance in image space, we use the family of loss functions (2) as training objective. Furthermore, we do not use the lossless entropy coding scheme proposed in their original work. While this would likely result in reduced bitrates, and thereby further improve our results, we omit this in order to reduce complexity and focus exclusively on the distortion during training. If not stated otherwise, we train the networks for 8 unrolling steps, yielding rates between 0.125 and 1.0 bpp. As training data $\mathcal{X}$, we use the training split of the ILSVRC-2012 [36] dataset, commonly known as ImageNet-1K. We preprocess the images by resizing such that the smallest side equals 256 pixels and aspects are preserved using bilinear interpolation. During training, we take random crops of size 224×224 and randomly flip them horizontally. During validation, we use the central crop of size 224×224. We follow [32] and normalize with a mean and variance obtained from a subset of the training set. We train all our networks using the Adam optimizer [24] for three epochs with the learning rate set to 4e-4 and minibatches of size four. All models are trained on eight Nvidia Titan X GPUs with 12GB RAM.

$^1$ The source code is available at https://github.com/DS3Lab/odlc.
Measures of Distortion. We train the compression networks using the loss function defined in Eq. (2) and use VGG-16 trained on the ImageNet-1K training split as our loss network $f_L$. Preliminary experiments have shown that choosing $I = \{\phi_{1,1}, \phi_{5,1}\}$ in (1), where $\phi_{i,j}$ denotes the $j$-th convolutional layer after activation in the $i$-th block of VGG-16, performed best. Including the entire set of convolutions or only the first or last layer did not yield any improvements. We provide detailed plots to compare the different loss compositions with respect to accuracy in the supplementary materials. The weights of the loss network are frozen and left unchanged during training. We experiment with different values for the parameter $\alpha$, starting the training each time from scratch. Namely, in order to optimize for human visual perception, we set $\alpha = 0$, while for classification oriented compression, we set $\alpha = 1$. To investigate the trade-off between human vision and classification, we train with $\alpha \in \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}\}$, also starting training from scratch each time.

Comparison with Other Methods. We compare our approach to the traditional compression algorithms JPEG [46], WebP [16] and BPG [5] which achieves state-of-the-art performance in HVS oriented compression. Following [32,35], BPG is used in the non-default 4:4:4 chroma format. Finally, we also compare against the state-of-the-art learned compression method presented in Mentzer et al. [32], using their publicly available weights and code. Since the available weights do not compress images below 0.3 bpp, we train two models with the same hyperparameters but different number of bottleneck channels to achieve lower bitrates.

Datasets. We evaluate our approach on four publicly available datasets. For classification we use the ImageNet-1K dataset [36], as well as two datasets used for fine-grained categorization, CUB-200–2011 [45] and Stanford Dogs [23]. In order to evaluate the performance in terms of human visual perception we use the Kodak Photo CD dataset [25] and the ImageNet-1K validation split.

CNN Architectures. On ImageNet-1K, we use DenseNet-121 [20], Inception-ResNet-V2 [39], Inception-V3 [40], MobileNet-V1 [19], ResNet-50 [18], Xception [9] and VGG-16 [38] for inference and use the weights provided by the Keras Library [10]. For fine-grained categorization on CUB-200–2011 and Stanford Dogs, we use Inception-V3, Inception-ResNet-V2, MobileNet-V1, ResNet-50 and VGG-16. To obtain the classifiers, we use ImageNet pre-trained networks and fine-tune all layers on the original uncompressed training split.

Evaluating Classification Accuracy. In order to compare the different compression algorithms with regard to classification accuracy, we evaluate a collection of CNN architectures on datasets compressed with different algorithms and at different bitrates. Note that all classifiers are trained on the uncompressed respective training datasets, without retraining on decoded data. The evaluation procedure is as follows. Since generally, the images do not have the same resolution, we resize them such that the smaller side equals $S_{comp}$ and aspects are preserved. We then take the central crop of size $S_{comp} \times S_{comp}$ yielding square images. After
this step, given a compression algorithm, we encode the images for a predefined grid of quality parameters and compute the bpp values for each image and quality parameter. For each quality level, we subsequently take the average over the entire validation set, yielding the final bpp values. Finally, we decode and take the central crop of size $S_{inf} \times S_{inf}$ of the decoded image, which is then fed to the classifier. This results in a set of (bpp, accuracy) points for each classifier and compression method. For CNNs that expect inputs of size $S_{inf} = 299$ we set $S_{comp} = 336$ and for those with $S_{inf} = 224$, we set $S_{comp} = 256$.

**Evaluating Human Visual Perception.** The procedure to compare the compression methods for human perceived visual quality is as follows. To account for the variable resolution on ImageNet-1K, we resize each image with bilinear interpolation such that the smallest side equals 256 pixels and aspects are preserved. We then take the central crop of size 256$\times$256. Since the Kodak images are all of equal resolution, we skip this first resizing step and keep the original resolution. We then compress the images using a predefined grid of quality parameters and compute their bpp values which are averaged over the validation set. Finally, we compute the MS-SSIM scores between decoded and original (resized) image. This yields a set of (bpp, MS-SSIM) points for each compression method.

### 4.1 Results

We start by investigating the trade-off between human perception and classification accuracy using compression trained with an increasingly more classification friendly loss. We then look at compression in terms of classification accuracy, followed by our results on human perception.

**From Human Visual Perception to Classification.** In order to investigate the relation between compression quality perceived by humans in terms of MS-SSIM, and by CNN classifiers, we train the compression networks with loss functions that interpolate between human friendly and classification friendly loss, i.e. for values of $\alpha$ in $\{0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1\}$. This trade-off can be seen qualitatively in Fig. 2. Optimizing for MS-SSIM, results in images that appear smoother and more blurry. Classification optimized compression on the other hand results in sharper images but suffers from checkerboard-like artifacts. This type of degradation is a known issue for feature visualization and super resolution (see e.g. [34]) and – in our case – stems from the convolution based loss function which incurs artifacts in gradients. In order to quantitatively investigate the trade-off, we visualize the relation in Fig. 3. We plot MS-SSIM on Kodak (left axis) and ImageNet-1K validation accuracy (right axis) against the tradeoff parameter $\alpha$ corresponding to RNN compression trained with different loss functions. The Figures indicate that we can indeed trade off accuracy against MS-SSIM by optimizing compression with our family of loss functions. Interestingly, we observe that by increasing the trade-off parameter $\alpha$ from 0 to 0.25, we substantially increase accuracy while the reduction in MS-SSIM is relatively small. The same holds for the other direction. Finally, we observe that the trade-off is much more pronounced in the low bitrate regime.
Observer Dependent Lossy Image Compression

(a) Inception-ResNet-V2, ~0.25 bpp
(b) Inception-ResNet-V2, ~1.0 bpp
(c) DenseNet-121, ~0.25 bpp
(d) DenseNet-121, ~1.0 bpp

Fig. 3. MS-SSIM evaluated on Kodak (left axis, grey), validation accuracy evaluated on ImageNet-1k (right axis, blue). As $\alpha$ increases, MS-SSIM decreases, while validation accuracy increases. The trade-off is especially pronounced for the low bitrates and DenseNet-121 is in general more sensitive to compression than Inception-ResNet-V2 (Color figure online).

**ImageNet Classification.** Table 1 shows the classification accuracies for a wider collection of CNN architectures. We see that our RNN-C outperforms both the traditional codecs BPG, WebP and JPEG as well as our RNN-H and the deep image compression method proposed in [32], across all architectures and bitrates considered. In the case of the loss network VGG-16 this is to be expected, since we explicitly train the compression network to produce images whose VGG-features – which are fed to the fully connected layers for classification – match the ones from their uncompressed version. Interestingly however, we see that RNN-C generalizes well to architectures different from the loss network and maintains the accuracy remarkably well, indicating that hidden representations are shared among CNN architectures. It is again noticeable that the advantage of RNN-C is much more pronounced for low bitrates.

**Fine-Grained Visual Categorization.** In order to explore the generalization properties of our compression system to new tasks, we evaluate our method on two well known datasets for fine-grained visual categorization, namely Stanford Dogs and CUB-200–2011. We emphasize that the compression system is trained on the ImageNet-1K training split. Figures 4(a) and 4(b) indicate that RNN-C outperforms both the traditional codecs, RNN-H compression and the approach
Table 1. Validation accuracy on ImageNet-1K. Our RNN-C compression consistently outperforms all other methods across bitrates and architectures.

| ImageNet-1K Validation Accuracy | 224 × 224 input | 299 × 299 input |
|---------------------------------|------------------|------------------|
|                                 | bpp              | Inception-V3     |
| Low Bitrates (≥0.13 bpp)        |                  |                  |
| RNN-C                           | 0.125            | 0.125            |
| RNN-H                           | 0.125            | 0.125            |
| Mentzer et al. [32]             | 0.142            | 0.142            |
| BPG                             | 0.157            | 0.157            |
| JPEG                            | 0.136            | 0.136            |
|                                 | **0.5914**       | **0.6998**       |
|                                 | **0.4903**       | **0.6998**       |
|                                 | **0.6092**       | **0.7037**       |
|                                 | **0.6009**       | **0.7037**       |
|                                 | **0.6691**       | **0.7037**       |
|                                 | **0.6815**       | **0.7037**       |
| Medium Bitrates (≥0.65 bpp)     |                  |                  |
| RNN-C                           | 0.625            | 1.000            |
| RNN-H                           | 0.625            | 1.000            |
| Mentzer et al. [32]             | 0.652            | 1.000            |
| BPG                             | 0.725            | 1.000            |
| WebP                            | 0.589            | 1.000            |
| JPEG                            | 0.582            | 1.000            |
|                                 | **0.7252**       | **0.7037**       |
|                                 | **0.6256**       | **0.7037**       |
|                                 | **0.6256**       | **0.7037**       |
|                                 | **0.7246**       | **0.7037**       |
|                                 | **0.6998**       | **0.7037**       |
|                                 | **0.625**        | **0.7037**       |
|                                 | **0.7678**       | **0.7037**       |
|                                 | **0.7787**       | **0.7037**       |
| High Bitrates (≥1.0 bpp)        |                  |                  |
| RNN-C                           | 1.000            | 1.000            |
| RNN-H                           | 1.000            | 1.000            |
| Mentzer et al. [32]             | 1.037            | 1.000            |
| BPG                             | 1.048            | 1.000            |
| WebP                            | 0.997            | 1.000            |
| JPEG                            | 1.087            | 1.000            |
|                                 | **0.7303**       | **0.7037**       |
|                                 | **0.6347**       | **0.7037**       |
|                                 | **0.7316**       | **0.7037**       |
|                                 | **0.7037**       | **0.7037**       |
|                                 | **0.7753**       | **0.7037**       |
|                                 | **0.7784**       | **0.7037**       |

* Loss network used to train RNN-C compression.

Fig. 4. Validation accuracy is displayed in (a) and (b) where our RNN-C consistently outperforms BPG, JPEG, WebP, RNN-H and deep compression [32] across bitrates. MS-SSIM on Kodak is shown in (c), indicating that our RNN-H is competitive to the state-of-the-art while RNN-C is comparable to JPEG. In each figure, RNN compression is trained from scratch on ImageNet-1K.
presented in [32] in terms of preserved classification accuracy with Inception-ResNet-V2 on both datasets. Similarly to ImageNet-1K classification, we see that the difference is especially pronounced below 0.5 bpp.

**Human Visual Perception.** Figure 4(c) shows that RNN-H outperforms RNN-C on Kodak across bitrates. In comparison to neural compression from [32], RNN-H performs competitive, although worse for the lowest bitrate. Comparing our method against the traditional codecs, RNN-H slightly outperforms BPG for bitrates above 0.2 bpp. Additionally, RNN-H clearly outperforms WebP and JPEG, while RNN-C performs competitive to JPEG.

5 Discussion

In this paper we investigate the trade-off in learned image compression with RNNs [43] between human visual perception and image classification. To that end, we use a family of loss functions that enables us to either optimize compression for the human observer, or towards subsequent image classification. Our experiments show that when using the human friendly loss, RNN compression performs competitive to a state-of-the-art learned compression method [32] and to the traditional codec BPG [5] in terms of MS-SSIM. JPEG and WebP perform consistently worse than our approach. We use MS-SSIM as a model for image similarity perceived by humans which, although being a widely adopted measure of distortion, is only an approximation to a true model of the human visual system. Our classification friendly loss, based on features extracted from VGG-16, induces a compression system which by a large margin outperforms both the other approaches in terms of preserved classification accuracy. Our experiments furthermore indicate that our approach is agnostic to the CNN architecture used for classification and does not require the classifiers to be retrained on compressed images. This suggests that we can indeed explicitly optimize image compression for subsequent classification. We observe a clear trade-off between quality perceived by the human visual system and classification accuracy, meaning that, for a fixed bitrate, an increase in accuracy always comes at the cost of degraded quality for the human observer, and vice versa. Across classifiers, this trade-off is much more pronounced for bitrates below 0.5 bpp. Finally, we find that by moving the loss function only marginally towards classification, we can substantially increase the preserved accuracy while incurring only a minor reduction in MS-SSIM. This improves compression in a scenario where images are consumed by humans and classifiers simultaneously and allows a user to trade off reconstruction quality accordingly.

An interesting line of future work could include investigating other types of distortion measures used for the human oriented training loss, for example metrics that are based on CNNs which have been reported to correlate better with human perceptual similarity. Additionally, while classification is one of the most basic computer vision tasks, it would be interesting to explore whether the approach presented here also generalizes to other tasks such as image segmentation and object detection.
Acknowledgments. CZ and the DS3Lab gratefully acknowledge the support from the Swiss National Science Foundation (Project Number 200021_184628), Innosuisse/SNF BRIDGE Discovery (Project Number 40B2-0_187132), European Union Horizon 2020 Research and Innovation Programme (DAPHNE, 957407), Botnar Research Centre for Child Health, Swiss Data Science Center, Alibaba, Cisco, eBay, Google Focused Research Awards, Oracle Labs, Swisscom, Zurich Insurance, Chinese Scholarship Council, and the Department of Computer Science at ETH Zurich.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). https://www.tensorflow.org/, software available from tensorflow.org
2. Agustsson, E., et al.: Soft-to-hard vector quantization for end-to-end learning compressible representations. Adv. Neural Inf. Process. Syst. 30, 1141–1151 (2017)
3. Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., Van Gool, L.: Generative adversarial networks for extreme learned image compression. arXiv preprint arXiv:1804.02958 (2018)
4. Ballé, J., Laparra, V., Simoncelli, E.P.: End-to-end optimized image compression. In: International Conference on Learning Representations (ICLR) (2017)
5. Bellard, F.: BPG image format (2014). https://bellard.org/bpg/
6. Bruna, J., Mallat, S.: Invariant scattering convolution networks. IEEE Trans. Pattern Anal. Mach. Intell. 35(8), 1872–1886 (2013)
7. Bruna, J., Sprechmann, P., LeCun, Y.: Super-resolution with deep convolutional sufficient statistics. In: International Conference on Learning Representations (ICLR), May 2016
8. Chinen, T., et al.: Towards a semantic perceptual image metric. In: 2018 25th IEEE International Conference on Image Processing (ICIP), October 2018
9. Chollet, F.: Xception: deep learning with depthwise separable convolutions. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
10. Chollet, F., et al.: Keras (2015). https://keras.io
11. Dodge, S., Karam, L.: Understanding how image quality affects deep neural networks. In: 2016 Eighth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2016)
12. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. Adv. Neural Inf. Process. Syst. 29, 658–666 (2016)
13. Galteri, L., Seidenari, L., Bertini, M., Del Bimbo, A.: Deep generative adversarial compression artifact removal. In: The IEEE International Conference on Computer Vision (ICCV), October 2017
14. Gatys, L., Ecker, A.S., Bethge, M.: Texture synthesis using convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 262–270 (2015)
15. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
16. Google: Webp image format (2015). https://developers.google.com/speed/webp/. Accessed 17 Mar 2019
17. Gueguen, L., Sergeev, A., Kadlec, B., Liu, R., Yosinski, J.: Faster neural networks straight from jpeg. Adv. Neural Inf. Process. Syst. 31, 3933–3944 (2018)
18. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
19. Howard, A.G., et al.: Mobilenets: efficient convolutional neural networks for mobile vision applications (2017). arXiv:1704.04861
20. Huang, G., Liu, Z., van der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
21. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9906, pp. 694–711. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46475-6_43
22. Johnston, N., et al.: Improved lossy image compression with priming and spatially adaptive bit rates for recurrent networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
23. Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L.: Novel dataset for fine-grained image categorization. In: First Workshop on Fine-Grained Visual Categorization, IEEE Conference on Computer Vision and Pattern Recognition, June 2011
24. Kingma, D., Ba, J.: Adam: a method for stochastic optimization. In: International Conference on Learning Representations (ICLR), May 2015
25. Kodak, E.: Kodak lossless true color image suite (PhotoCD PCD0992)
26. LeCun, Y., Kavukcuoglu, K., Farabet, C.: Convolutional networks and applications in vision. In: Proceedings of 2010 IEEE International Symposium on Circuits and Systems, pp. 253–256 (2010)
27. Ledig, C., et al.: Photo-realistic single image super-resolution using a generative adversarial network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017
28. Liu, H., Chen, T., Shen, Q., Yue, T., Ma, Z.: Deep image compression via end-to-end learning. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2018
29. Liu, Z., et al.: DeepN-JPEG: a deep neural network favorable JPEG-based image compression framework. In: 2018 55th ACM/ESDA/IEEE Design Automation Conference (DAC), pp. 1–6 (2018)
30. Liu, Z., et al.: Machine vision guided 3D medical image compression for efficient transmission and accurate segmentation in the clouds. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
31. Mallat, S.: Understanding deep convolutional networks. Philos. Trans. Roy. Soc. A 374(2065) (2016)
32. Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Van Gool, L.: Conditional probability models for deep image compression. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018
33. Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Van Gool, L.: Practical full resolution learned lossless image compression. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2019
34. Odena, A., Dumoulin, V., Olah, C.: Deconvolution and checkerboard artifacts. Distill (2016). http://distill.pub/2016/deconv-checkerboard/
35. Rippel, O., Bourdev, L.: Real-time adaptive image compression. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 2922–2930, August 2017
36. Russakovsky, O., et al.: Imagenet large scale visual recognition challenge. Int. J. Comput. Vis. 115(3), 211–252 (2015)
37. Santurkar, S., Budden, D., Shavit, N.: Generative compression. In: 2018 Picture Coding Symposium (PCS), pp. 258–262 (2018)
38. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (ICLR), May 2015
39. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, AAAI 2017, pp. 4278–4284. AAAI Press (2017)
40. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2016
41. Theis, L., Shi, W., Cunningham, A., Huszár, F.: Lossy image compression with compressive autoencoders. In: International Conference on Learning Representations (ICLR) (2017)
42. Toderici, G., et al.: Variable rate image compression with recurrent neural networks. In: International Conference on Learning Representations (ICLR) (2016)
43. Toderici, G., et al.: Full resolution image compression with recurrent neural networks. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5435–5443, July 2017
44. Torfason, R., Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Gool, L.V.: Towards image understanding from deep compression without decoding. In: International Conference on Learning Representations (ICLR), April 2018
45. Wah, C., Branson, S., Welinder, P., Perona, P., Belongie, S.: The Caltech-UCSD Birds-200-2011 Dataset. Technical report. CNS-TR-2011-001, California Institute of Technology (2011)
46. Wallace, G.K.: The jpeg still picture compression standard. IEEE Trans. Cons. Electron. 38(1), xviii-xxxiv (1992)
47. Wang, Z., Simoncelli, E.P., Bovik, A.C.: Multiscale structural similarity for image quality assessment. In: The Thirty-Seventh Asilomar Conference on Signals, Systems and Computers, vol. 2, pp. 1398–1402 (2003)
48. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018