Rate-Distortion-Classification tradeoff

Le Huy, Pierre Duhamel, Armelle Wautier

RATE - DISTORTION - CLASSIFICATION TRADEOFF

Huy LE
le.huy200073@gmail.com
Master student Paris-Saclay University and VNU-University of Engineering and Technology
Vietnam

Armelle WAUTIER
armelle.wautier@centralesupelec.fr
Professor, Head of departement Signal, Information, Communication
L2S, CentraleSupelec,
France

Pierre DUHAMEL
pierre.duhamel@centralesupelec.fr
"Emeritus research director" at CNRS,
Signals and Systems Laboratory,
CentraleSupelec
France

ABSTRACT

The purpose of the paper is to improve the classification performance upon reception of an image, for a given rate and distortion. It builds on the work of Blau and Michaeli [4] [3] who integrated perceptual quality in coding by introducing the divergence between input and output signal distributions as a criterion, thereby extending the rate/distortion tradeoff to include perception. This paper modifies this approach by incorporating image gradient statistics for enhanced segmentation in compressed images, therefore resulting in a slight modification of the Rate/Distortion/Perception model for improved classification performance. This is obtained by including the divergence between the high frequency components of the original and reconstructed images in the criterion to be optimized. Central to this approach is the use of Machine Learning, especially Wasserstein Generative Adversarial Networks (WGANs), marking a significant integration of traditional coding techniques with contemporary AI innovations.

KEYWORDS

Advanced compression methods, performance analysis, semantic communication, classification performance

1 INTRODUCTION

This work explores the classification performance in source coding, emphasizing a tradeoff among rate, distortion, and classification, extending the concepts introduced by Blau and Michaeli [4][3]. Unlike traditional image coding methods, which incorporate perceptual quality in a rather ad hoc manner, the work in [3] aims at using Machine Learning to obtain the best tradeoff between perception and distortion for a given rate. Our work shifts the focus from perception to classification performance, using deep feature-based distortion to enhance semantic similarity between original and reconstructed images. This study further investigates the use of image gradient statistics to preserve segmentation capabilities in compressed images, which could improve classification while balancing rate and distortion. More precisely, we make use of Wasserstein Generative Adversarial Networks (WGANs), to refine the Rate/Distortion/Perception tradeoff for improved classification. The efficiency of this approach is evaluated through classification performance comparisons between the original and transformed approaches on the MNIST dataset .

The proposed solution is based on a Generative Adversarial Network (GAN). The original image is processed by a first block (the generator) intending to provide a "good" reconstruction, while an adversarial part is modifying the parameters to minimize the distance between the original and reconstructed image based on some criterion. In other words, the generator is not trained to minimize the distance to a specific image, but rather to fool the discriminator.

2 BACKGROUND

2.1 Rate

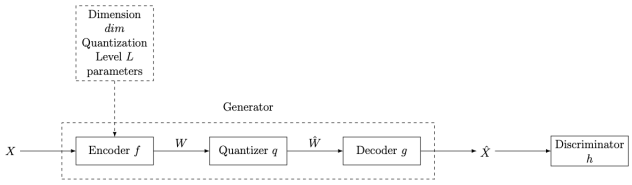

Figure 1: Architecture of encoder/decoder for deep image compression

In the deep learning framework for image compression [1], encoding occurs within a latent space. Here, the encoder aims to compress the input image into a latent space vector or matrix, which is a feature-rich, multidimensional domain. This compressed representation is quantized into discrete levels, represented as $C = \{c_1, \ldots, c_L\}$, forming the quantized output $\hat{W} = q(f(X))$. This output is then encoded into a bitstream for transmission or storage. As depicted in Figure 1, the process involves an encoder f, a decoder g, and a quantizer q that transforms the input image X into a quantized feature map $\hat{W}$, which is then reconstructed back to $\hat{X} = g(\hat{W})$.

The resulting rate is evaluated by the entropy $H(\hat{W})$:

$$H(\hat{W}) \leq dim \times \log_2(L), \tag{1}$$

where dim is the dimensionality of the encoder output, and L denotes the levels of quantization. The bottleneck layer, crucial for reducing data dimensionality, influences the network's capacity to encode and compress information.

Deep learning models are based on the use of back propagation, which would not be compatible with hard quantization, and the so-called "soft quantization" is preferred to allow back-propagation through the quantization layer. This approach is described by the soft quantization formula:

$$\tilde{w}_i = \sum_{j=1}^{L} \frac{\exp\left(-\sigma \left\| w_i - c_j \right\|_1\right)}{\sum_{l=1}^{L} \exp\left(-\sigma \left\| w_i - c_l \right\|_1\right)} c_j, \tag{2}$$

with $\sigma = 2/L$ and added noise $\mathcal{U}\left(-\frac{a}{2}, \frac{a}{2}\right)$, where $a = 2/(L-1)$.

2.2 Distortion

Classically, Distortion is evaluated via the mean-square-error (MSE) metric:

$$||x - \hat{x}||^2 = \frac{1}{MP} \sum_{k=1}^{M} \sum_{l=1}^{P} [x(k,l) - \hat{x}(k,l)]^2, \tag{3}$$

for pixel values $x(k,l)$ and $\hat{x}(k,l)$ in an image of dimensions $M \times P$. When evaluated over a set of N original and reconstructed image pairs $x_i, \hat{x}_i$, the average MSE is:

$$D = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{MP} \sum_{k=1}^{M} \sum_{l=1}^{P} [x_i(k,l) - \hat{x}_i(k,l)]^2. \tag{4}$$

2.3 Perception

In [4], naturalness of a reconstructed sample $\hat{X}$ is evaluated via the difference in distributions $p_{\hat{X}}$ versus p_X which is a different criterion from the distortion. The perceptual quality index measures this through the Wasserstein distance $d_{was}(p_X, p_{\hat{X}})$ [2], defined as:

$$P = d_{was}(p_X, p_{\hat{X}}) = \inf_{\gamma \in \Pi(p_X, p_{\hat{X}})} \mathbb{E}_{(X,\hat{X})\sim\gamma}[\|X - \hat{X}\|], \tag{5}$$

where $\Pi(p_X, p_{\hat{X}})$ includes all distributions γ that couple p_X and $p_{\hat{X}}$, aiming for minimal "mass" transfer for distribution alignment. GAN-based image restoration has enhanced perceptual quality by using adversarial loss to narrow the distribution gap $d(p_X, p_{\hat{X}})$, with the generator producing plausible data and the discriminator differentiating between real and generated images. Optimizing perceptual quality in GANs [6] involves maximizing the difference between expected values over a class of 1-Lipschitz functions $\mathcal{F}$, modified by a gradient penalty:

$$d_{was}(p_X, p_{\hat{X}}) = \max_{h \in \mathcal{F}} (\mathbb{E}[h(X)] - \mathbb{E}[h(\hat{X})]). \tag{6}$$

3 METHODOLOGY

3.1 WGAN model

Wasserstein Generative Adversarial Networks (WGAN) were proposed in 2017 in [2]. This is an extension of the first proposed Generative Adversarial Networks [6] in 2014. It improves the stability of learning, prevents the mode collapse and gives meaningful learning for debugging and hyperparamter searches. Table 1 provides the structure of the encoder-decoder pair (Generator block) and tha Discriminator block used in this paper : [3]: Once the architecture of each block is defined, we compute the loss of Generator k which consists in two components: encoder f, decoder g and Discriminator h. The WGAN architecture, as depicted in Figure 2, features the original dataset X, a Generator Block (k), and a Discriminator (h), utilizing subsets of X for training, validation, and testing.

The generator, an encoder-decoder, aims to minimize the difference between its outputs and real data, leveraging mean squared error and adversarial loss, controlled by λ_{WGAN}, for realism:

$$l_{gen} = \lambda_{MSE} \times l_{k-MSE} + \lambda_{adv} \times l_{k-WGAN}, \tag{7}$$

,

$$l_{k-WGAN} = \mathbb{E}[h(X)] - \mathbb{E}[h(\hat{X})] \tag{8}$$

Table 1: WGAN architecture of Blau and Michaeli [3]

Component	Size	Layer
Encoder	$28 \times 28 \times 1$	Input
	784	Flatten
	512	FC, BN, l-ReLU
	256	FC, BN, l-ReLU
	128	FC, BN, l-ReLU
	128	FC, BN, l-ReLU
	dim	FC, BN, Tanh
	dim	Quantize
Decoder	dim	Input
	128	FC, BN, l-ReLU
	512	FC, BN, l-ReLU
	$4 \times 4 \times 32$	Unflatten
	$11 \times 11 \times 64$	ConvT (st=2), BN, l-ReLU
	$25 \times 25 \times 128$	ConvT (st=2), BN, l-ReLU
	$28 \times 28 \times 1$	ConvT (st=1), Sigmoid
Discriminator	$28 \times 28 \times 1$	Input
	$14 \times 14 \times 64$	Conv (st=2), l-ReLU
	$7 \times 7 \times 128$	Conv (st=2), l-ReLU
	$4 \times 4 \times 256$	Conv (st=2), l-ReLU
	4096	Flatten
	1	FC

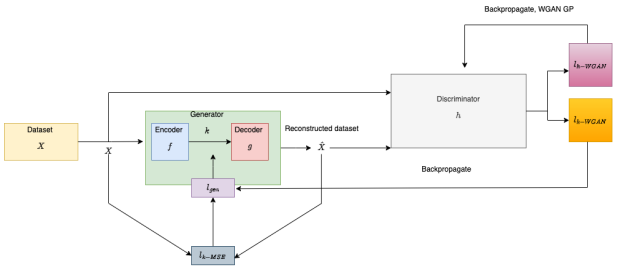

Figure 2: Structure of Generator and Discriminator in WGAN model

with λ_{MSE} and λ_{adv} adjusting the importance of MSE and adversarial loss, respectively.

The discriminator, distinguishing real from generated data, adjusts its training intensity for balance. Its loss, incorporating a gradient penalty, ensures 1-Lipschitz continuity for stable training:

$$l_{h-WGAN} = -(\mathbb{E}[h(X)] - \mathbb{E}[h(\hat{X})]) + \lambda_{penalty} \mathbb{E}\left[(\|\nabla_{\tilde{x}} h(\tilde{x})\|_2 - 1)^2\right], \tag{9}$$

with $\lambda_{penalty}$ fine-tuning the gradient penalty influence for training stability.

3.2 Training the WGAN model

The tuning of the coefficients is obtained via Algorithm 1, aiming to find the values of the generator parameters θ and of discriminator parameters w.

Algorithm 1 WGAN with gradient penalty [7]

Require: gradient penalty coefficient λ, number of discriminator iterations $n_{\text{discriminator}}$, batch size m, Adam hyperparameters α, β_1, β_2

Require: initial discriminator parameters v_0, initial generator parameters θ_0

while θ has not converged **do**
  for $t = 1, \ldots, n_{\text{discriminator}}$ **do**
    for $i = 1, \ldots, m$ **do**
      Sample real data $\boldsymbol{x} \sim p_X$, latent variable $\boldsymbol{w} \sim p(\boldsymbol{w})$, random number $\epsilon \sim U[0, 1]$
      $\hat{\boldsymbol{x}} \leftarrow k_\theta(\boldsymbol{w})$
      $\tilde{\boldsymbol{x}} \leftarrow \epsilon \boldsymbol{x} + (1 - \epsilon)\hat{\boldsymbol{x}}$
      $l^{(i)} \leftarrow h_v(\hat{\boldsymbol{x}}) - h_v(\boldsymbol{x}) + \lambda(\|\nabla_{\tilde{\boldsymbol{x}}} h_v(\tilde{\boldsymbol{x}})\|_2 - 1)^2$
    end for
    $v \leftarrow \text{Adam}(\nabla_v \frac{1}{m} \sum_{i=1}^{m} L^{(i)}, , \alpha, \beta_1, \beta_2)$
  end for
  Sample a batch of latent variables $\{\boldsymbol{w}^{(i)}\}_{i=1}^{m} \sim p(\boldsymbol{w})$
  $\theta \leftarrow \text{Adam}(\nabla_\theta \frac{1}{m} \sum_{i=1}^{m} -D_v(G_\theta(\boldsymbol{w}^{(i)})), \theta, \alpha, \beta_1, \beta_2)$
end while

4 CONTRIBUTION

4.1 Motivation

Increased bit-rates enhance image clarity and reduce blurriness for distortion-only images, as is the case when the system is tuned with both distortion and perception as a criterion.However, this these two criteria do not fully align with good classification performance. Therefore, we need to define the classification metric, and the tool that will be used in conjunction with Wasserstein GAN to reach our goal of improving classification performance.

4.2 Classification performance metrics

Given a confusion matrix:

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

- **True Positive (TP)**: The number of positive instances that were correctly predicted as positive by the classification model.
- **True Negative (TN)**: The number of negative instances that were correctly predicted as negative by the model.
- **False Positive (FP)**: The number of negative instances that were incorrectly predicted as positive by the model
- **False Negative (FN)**: The number of positive instances that were incorrectly predicted as negative by the model.

4.3 Haar transform

The two-dimensional Discrete Wavelet Transform (DWT) decomposes signals into four main frequency components. It starts with a matrix cA_j representing the initial signal level. Through sequential low-pass (LoD) and high-pass (HiD) filtering, followed by downsampling, the DWT isolates low and high-frequency details across rows and columns. This results in four outputs: with L means Lowpass, H means Highpass filter, cA_{j+1} (LL) for low-frequency approximation, $cD_{j+1}^{(h)}$ (LH) for horizontal details, $cD_{j+1}^{(v)}$ (HL) for vertical details, and $cD_{j+1}^{(d)}$ (HH) for diagonal high-frequency details.

(a) Original image **(b) Haar transform** **(c) Inverse Haar transform**

Figure 3: Illustration of Haar transform and Inverse Haar transform on the MNIST digit image

4.4 Proposed encoder and decoder architecture

The results indicate that deep learning-generated images are notably impacted by distortion and compression rate, especially in terms of perceptual quality, which can alter the image's contours and semantics. Since our aim is to enhance classification performance, we assumed that contour preservation in the image would be a reasonable criterion. Therefore we modified the perceptual component of the WGAN model for enhanced classification minimizing the contour differences between original and reconstructed images. This is obtained by the use of a 2D discrete Haar transform to split the image into four sub-bands: LL, LH, HL, and HH which are then processed individually, as shown in Figure. 4.

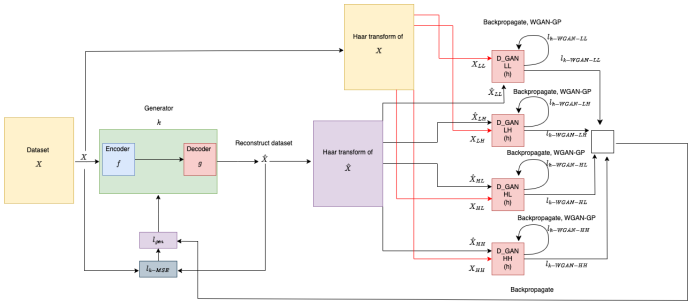

Figure 4: Proposed structure of Generator and Discriminator in WGAN model

This results in a WGAN architecture with one global Generator and four local Discriminators for the Haar transform coefficients. The losses $l_{h-WGAN-LL}, l_{h-WGAN-LH}, l_{h-WGAN-HL}, l_{h-WGAN-HH}$ are computed for each Discriminator. The total Generator loss is formulated as:

$$\begin{aligned} l_{gen} = \lambda_{MSE} \times l_{k-MSE} + \lambda_{LL} \times l_{k-WGAN-LL} \\ + \lambda_{LH} \times l_{k-WGAN-LH} + \lambda_{HL} \times l_{k-WGAN-HL} \\ + \lambda_{HH} \times l_{k-WGAN-HH}, \end{aligned} \tag{10}$$

prioritizing high frequencies by appropriately choosing λ_{LH}, λ_{HL}, and λ_{HH} higher than λ_{LL}.

4.5 Classification algorithm

For MNIST classification, a convolution neural network (CNN) was deployed with the following configuration: The cross-entropy loss

Table 2: CNN Architecture

Layer	Output Shape	Param #
Conv2d-1	[32, 28, 28]	320
MaxPool2d-2	[32, 14, 14]	0
Dropout2d-3	[32, 14, 14]	0
Conv2d-4	[64, 14, 14]	18,496
MaxPool2d-5	[64, 7, 7]	0
Dropout2d-6	[64, 7, 7]	0
Linear-7	[128]	401,536
Linear-8	[10]	1,290

formula, crucial for multi-class classification like MNIST with 10 classes, is:

$$\text{Cross-Entropy Loss} = -\sum_{c=1}^{M} y_{o,c} \log(p_{o,c}), \tag{11}$$

where $M = 10$, $y_{o,c}$ indicates if class c is the correct label for observation o, and $p_{o,c}$ is the predicted probability of class c.

5 EXPERIMENTAL RESULTS

5.1 Configuration

For MNIST digit classification, we followed the setup described in [5], utilizing a dataset with 60,000 training images, 20% reserved for validation, and 10,000 for testing. The WGAN model settings include: Generator learning rate of 0.0025, Discriminator learning rate of 0.0002, 25 epochs, batch size of 64, with a LR scheduler initiating at epoch 10, LR decay every 10 epochs by a factor of 0.2, and a WGAN GP-lambda of 10. Implementations were on a NVIDIA V100 GPU with 32GB VRAM on the Ruche server (University of Paris-Saclay). Classification was conducted under four scenarios:

- Original MNIST dataset
- Reconstructed images with $\lambda_{MSE} = 0.001$
- Deep feature pre-trained MNIST model with $\lambda_{MSE} = 0.001$
- A new method with four Discriminators as in 4

Tests were run at 6.000 and 8.000 bit-per-digit. For reconstructed images, we used a MSE coefficient of 0.001. Our novel approach involves four separate Discriminators:

- *NA1*: As per [3] with $\lambda_{MSE} = 0.001$ and $\lambda_{LL} = \lambda_{LH} = \lambda_{HL} = \lambda_{HH} = 0.00005$
- *NA2*: Focusing on high frequencies with $\lambda_{MSE} = 0.001$, $\lambda_{LL} = 0.00250$, and $\lambda_{LH} = \lambda_{HL} = \lambda_{HH} = 0.00350$
- *NA3*: Higher emphasis on high frequencies with $\lambda_{MSE} = 0.001$, $\lambda_{LL} = 0.00250$, and $\lambda_{LH} = \lambda_{HL} = \lambda_{HH} = 0.05$

5.2 Performance comparison

This section provides mostly preliminary results, aiming at illustrating the impact of a good representation of the high frequencies on classification performance. We did not have time to find the best tuning of the parameters. However, it is seen in the corresponding table that the set of parameters denoted as NA2 improves the classification performance over the compression only case, at the cost of a slight increase in terms of distortion, which was expected. What remains to be done (in the near future) is an evaluation of this tradeoff in comparison with the best possible one.

Table 3: Global Classification Metrics and Distortion Comparison

Case	Rate: 8.000 bit-per-digit		Rate: 12.000 bit-per-digit	
	Acc	Distortion	Acc	Distortion
Original Img	99.1%	-	99.1%	-
Only Comp	85.66%	0.032	90.48%	0.025
Deep Feat	86.39%	0.039	92.20%	0.029
NA1	86.04%	0.039	94.00%	0.031
NA2	**88.01%**	**0.045**	**95.26%**	**0.040**
NA3	87.02%	0.040	94.67%	0.034

6 CONCLUSION AND FUTURE WORKS

This paper is based on the study of the distortion-perception tradeoff in compression, notably at lower bit rates, and modifies the WGAN network, especially using Haar transform with four Discriminator blocks for enhanced classification performance. Future directions include (1) applying these findings to more complex datasets beyond MNIST, to assess model viability (2) compute the optimal tradeoff between distortion and classification for a given rate. These elements should be useful in the context of semantic communication.

REFERENCES

[1] Eirikur Agustsson et al. "Generative adversarial networks for extreme learned image compression". In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 2019, pp. 221–231.

[2] Martin Arjovsky, Soumith Chintala, and Léon Bottou. "Wasserstein generative adversarial networks". In: *International conference on machine learning*. PMLR. 2017, pp. 214–223.

[3] Yochai Blau and Tomer Michaeli. "Rethinking lossy compression: The rate-distortion-perception tradeoff". In: *International Conference on Machine Learning*. PMLR. 2019, pp. 675–685.

[4] Yochai Blau and Tomer Michaeli. "The perception-distortion tradeoff". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018, pp. 6228–6237.

[5] discusstech and sumeshk333. *Train a Deep Learning Model with PyTorch*. Sept. 2023. URL: https://www.geeksforgeeks.org/train-a-deep-learning-model-with-pytorch/.

[6] Ian Goodfellow et al. "Generative adversarial nets". In: *Advances in neural information processing systems* 27 (2014).

[7] Ishaan Gulrajani et al. "Improved training of wasserstein gans". In: *Advances in neural information processing systems* 30 (2017).