RDONet: Rate-Distortion Optimized Learned Image Compression with Variable Depth

Fabian Brand, Kristian Fischer, Alexander Kopte, Marc Windsheimer, André Kaup
Multimedia Communications and Signal Processing
Friedrich-Alexander-Universität Erlangen-Nürnberg
Cauerstr. 7, 91058 Erlangen, Germany
{fabian.brand, kristian.fischer, alex.kopte, marc.windsheimer, andre.kaup}@fau.de

Abstract

Rate-distortion optimization (RDO) is responsible for large gains in image and video compression. While RDO is a standard tool in traditional image and video coding, it is not yet widely used in novel end-to-end trained neural methods. The major reason is that the decoding function is trained once and does not have free parameters. In this paper, we present RDONet, a network containing state-of-the-art components, which is perceptually optimized and capable of rate-distortion optimization. With this network, we are able to outperform VVC Intra on MS-SSIM and two different perceptual LPIPS metrics. This paper is part of the CLIC challenge, where we participate under the team name RDONet_FAU.

1. Introduction

In recent years, neural-network-based image compression [3, 16, 9] has received wide attention in the research community. Similar to traditional coding methods, the image is transformed into a latent representation which is then compressed using certain probability models. In traditional compression, both the transform (e.g. discrete cosine transform) and the probability model (e.g. lower frequencies occur more likely) are hand-crafted and empirically determined. In deep image compression, both transform and probability model are learned from a large training set and are typically non-linear. The large success of these methods shows that traditional concepts can be improved by transferring them to an end-to-end trainable environment.

Another important concept of traditional image and video compression is rate-distortion optimization (RDO) [20]. In many cases the coder has additional degrees of freedom which influence the coding behavior locally. That way, an optimal configuration can be found and transmitted as structured side information. The epitome of rate-distortion optimization is certainly adaptive block partitioning as used for example in HEVC [19], VVC [8], VP9 [17], or AV1 [11]. Even though the possibility of locally adapting the coder behavior is a very powerful tool, it is not yet widely used in learning-based compression. A reason is that the network structures are usually fixed after training and do not allow for any local adaptation in inference.

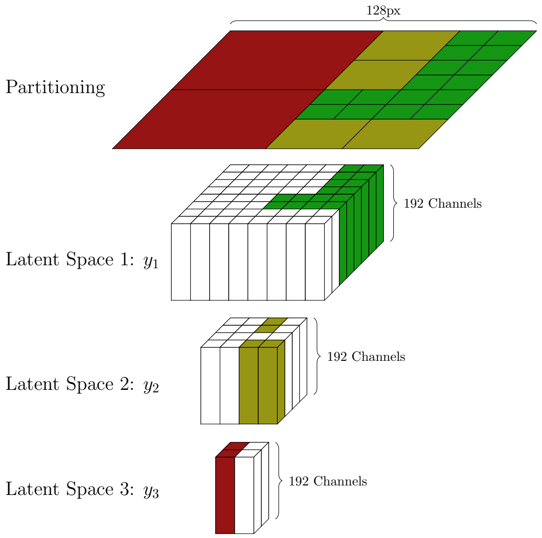


Figure 1. Visualization of the latent space coding in different layers for a 128×128 block. Each color represents one level. For u_{1-3}, a colored component indicates that this hyper-pixel is transmitted. For better visualization, x and y_{1-3} are not to scale.

In this paper, we present RDONet, a state-of-the-art neural compression network which is capable of rate-distortion optimization by employing multiple latent spaces. As shown in Fig. 1, we can thereby transfer block partitioning to neural compression. This work is based on two previous publications [6, 7], where we initially proposed the RDONet and extended the training. These publications studied the structure on a relatively simple autoencoder. In this publication we elevate the structure by combining it with state-of-the-art components to obtain a competitive performance. By using a suitable loss function, we also optimize our model on perceptual quality.

2. Related Work

The dominating technology for end-to-end trained image compression, the compressive autoencoder, goes back to a publication by Ballé *et al.* in 2017 [3] and extends the standard autoencoder [14] with an entropy bottleneck. This initial work was later extended by employing a more refined probability modeling using a hyperprior network [5] and a context model [16]. This way remaining spatial correlation in the latent space could be reduced. In subsequent research, the structure was extended by employing residual blocks, attention layers, and parametrized Gaussian mixture models [9].

Apart from the network structure, research has been performed to find optimal loss functions for the human visual system, since it is widely known that a mean squared error is not optimal in this respect. Multiple papers use discriminator networks in combination with perceptually motivated metrics like LPIPS [24] as loss functions and are able to greatly increase the visual quality [21, 15, 12].

Wang *et al.* also proposed a method to introduce RDO in neural compression [22]. There, multiple specialized networks were trained and the best one is chosen at encoding time. Another possibility was introduced by Schäfer *et al.*, where the transmitted latent space coefficients are optimized at encoding time to boost the performance [18] of the network. Furthermore, they also proposed a fast search algorithm.

3. RDONet

The basic principle of the rate-distortion optimization is inspired by adaptive block partitioning in HEVC. There, the image is split into blocks of different sizes for compression. As a rule of thumb, areas with stationary content are compressed with low rate using large blocks, while areas with fine details are compressed using small block, thus requiring a larger rate. We transfer this concept to neural-network-based compression by allowing our network to compress the image at different latent space levels. A compression at a deeper stage of the network is somewhat analogous to a compression with large blocks in HEVC. In both cases, a large area is represented jointly, usually with only few coefficients and only small rate. However, since the filters used in the decoder have a larger field of view, each position in the latent space not only corresponds to the respective block of the image but also influences neighboring areas. This implies that there is considerable redundancy between the different levels, which is considered by coding with a conditional hyperprior.

3.1. Network Structure

We split our network in two parts, the backbone feature analysis and synthesis on the one hand and the variable depth compression on the other hand. The former is responsible for generating a sparse feature representation of the image and reconstructing the image. The latter is tasked with compressing the feature on variable depth, enabling the eponymous RDO.

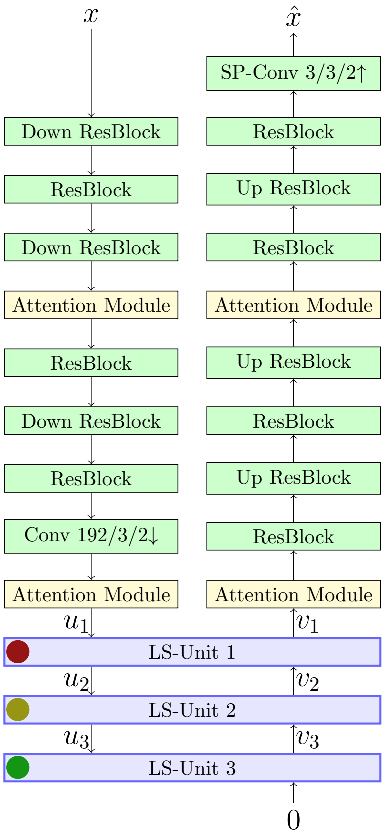


Figure 2. Architecture of the proposed RDONet. The left side of the figure shows the encoder components, while the right side shows the decoder components. *Conv c/k/s* ↓ denotes a convolution layer with c output channels, a kernel size of $k \times k$ and a downsampling factor of s. *SP-Conv* denotes an analogously parametrized subpixel convolution. *Down ResBlock* and *Up ResBlock* denote a residual block as used in [9] with a downsampling and upsampling, respectively. *ResBlock* is a residual block which does not change the resolution. The base number of channels for all components as used in [9] is 192.

As backbone for the encoder and decoder, we use the structure proposed by Cheng *et al.* [9]. This includes the use of residual blocks and attention layers. After the feature generation performed by the encoder network of the backbone, we proceed by using three latent space units (LS-Units) as we proposed in [6].

As shown in Fig. 2, we generate the feature representation u_1 from the image x using several residual blocks and attention modules in a configuration as in [9]. Using downsampling convolution layers, we generate the representations u_2 and u_3, which are each a factor of 2 smaller

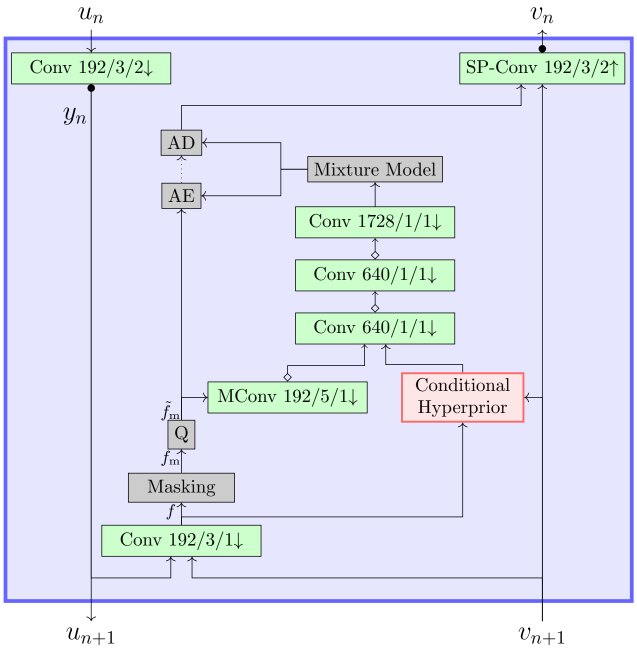


Figure 3. Structure of the LS-Unit responsible for latent space transmission. Each LS-Unit contains a Gaussian mixture model which is parametrized with a context model and a conditional hyperprior. *Conv* and *SP-Conv* are analoguos to Fig. 2, while *MConv* denotes a masked convolution with the same parameters. Filled circles after a convolution denote GDNs/IGDNs [4] and blank rhombuses denote leaky ReLUs. Whenever two signals enter the same convolutional layer, a concatenation along the channels is performed implicitly. The conditional hyperprior block is the same as in previous publications [6, 7]. For LS-Unit 1, the convolution and deconvolution, at the top have a stride of 1 to maintain the correct downsampling factors.

in each spatial dimension. From these features, we generate the latent spaces y_i inside the LS-Unit as shown in Fig. 3. We see that $u_{i+1} = y_i$ holds here. The three latent spaces y_1,y_2, and y_3 are now selectively transmitted. As we visualize in Fig. 1, for each spatial position in the features, we can choose whether this hyper-pixel is transmitted. Since we transmit the spatial positions, we can freely control the behavior externally. The only constraint is that each position (taking downsampling into account) is transmitted exactly once. This is done analogously to HEVC, where each pixel is transmitted with exactly one block size.

In Fig. 3, we see the exact diagram of a latent space unit. After generating the latent space, we use a convolutional layer to combine it with the previously decoded latent space from the lower layer. As described above, the latent spaces have redundancy between them. Using the convolution layer in the bottom, we can reduce this redundancy by only passing on information that can not be predicted from the previous latent space. This behavior develops during training time and is not enforced. The same deliberation leads us to the conditional hyperprior. It makes sense to include the information from the previous level in the context model of the current level. Finally, we also adapt the structure from [9] combining context model and hyperprior with a Gaussian mixture model yielding the probabilities for the arithmetic coder. The structure of the conditional hyperprior remains the same as in the previous version [7], except the for dimensions of encoder and decoder which were taken from [9].

3.2. Training

Since the network has externally controllable parameters, the block partitioning, these parameters have to be taken into account. For this work, we chose the combination of two possibilities. In the first training phase, we start by generating random block partitionings which do not depend on the content. From one point of view, this is suboptimal, since inference is performed with optimized masks, leading to a mismatch between training and inference. However, this way all layers see a large variety of content, leading to a more robust training. We then continue the training using specialized masks, which are estimated with a variance-based criterion we proposed in [7]. Preliminary experiments have shown a large advantage of starting the training with random masks and continuing with estimated masks.

3.3. Rate Distortion Optimization

After training a rate-distortion optimization has to be performed. However, an RDO typically consists of testing multiple possibilities and choosing the optimal one. Especially with large compression networks, this is rather cumbersome. Instead, we use a zero-pass RDO, which entirely relies on estimated masks. In [7], we showed that the performance of this fast version comes very close to the performance of a full RDO and is a suitable compromise between encoding time and RD performance.

4. Experiments

4.1. Setup

We train our network using a combination of the CLIC2021 training set, the full DIV2K [1] set, and the TECNICK [2] dataset. We train on image crops of size 512×512 and train for a total of 4750 epochs. In total, we use four distortion loss functions: MSE, MS-SSIM, LPIPS using a VGG backbone [24], and a Patch GAN discriminator, as in [10]. The overall distortion D is a linear combination of them. The final loss functions is given as

$$L = \lambda D + R \qquad (1)$$

To save time during training, we first trained a full model using $\lambda = 0.02$ which we then fine-tuned to three different rate points. The rate points where chosen to match the

Epoch	lr	λ	MSE	MSSSIM	LPIPS	Patch GAN	Mask
400	1e-4	0.02	1	0.1	0	0	Rand
1200	1e-5	0.02	1	0.1	0	0	Rand
2000	1e-6	0.02	1	0.1	0	0	Rand
3850	3e-5	0.02	1	0.05	0.015	0.0001	Var
4000	3e-6	0.02	1	0.05	0.015	0.0001	Var
4600	3e-5	0.0045 0.0275 0.08	1	0.05	0.015	0.0001	Var
4750	3e-6	0.0045 0.0275 0.08	1	0.05	0.015	0.0001	Var

Table 1. Schedule for training parameters. We give the learning rate (lr), λ, the weights of all individual distortion loss functions and the used mask. The left column states the epoch until which we use the given set of parameters.

target rates of 0.075, 0.15 and 0.3 bit per pixel. Tab. 1 summarizes the training procedure. Note that we decrease the learning rate over time and increase it every time we change other parameters. We use the Adam optimizer with standard parameters [13].

We perform our tests on the CLIC22 validation dataset. The set consists of 30 high-resolution images from Unsplash.

4.2. Results

We first compare RDONet with VVC Intra on objective metrics. We use the VVEnC implementation in version 1.3.1 [23] on the *slower* preset, which yields the best performance and is on par with the reference implementation. We compare the results using the PSNR, MS-SSIM, and LPIPS with two different backbones (AlexNet and VGG). Since our model was trained on a perceptual loss, we expect a low performance on the pixel-wise PSNR metric but a good performance on the other more perceptually motivated metrics. Indeed, as we see in Fig. 4, RDONet performs worse than VVEnC on PSNR. On MS-SSIM, we perform similarly to VVEnC, beating VVEnC for small rates but having a slightly worse performance for higher rates. On the LPIPS metrics however, we clearly outperform VVEnC for the entire range of rates. Note that we perform well both on the VGG backbone (which was used for training) and the AlexNet backbone (previously unseen). This shows that the results are not just a result of overfitting on one metric but that we actually achieve better objective perceptual quality.

We also want to demonstrate the performance in visual comparisons. In Fig. 5, we show a 128 × 128 pixel excerpt of one test image. We clearly see that VVC produces blurred results for fine details like the plant or the rooftiles. Our method keeps more details and produces visually more pleasant results. However, we also see a drawback of perceptually motivated losses, when we look at the color artifact in the center. RDONet overemphasizes the color compared to the original and even spreads it slightly into the plant. On the other hand, the original orange color is completely lost in VVC. Without comparison to the original however, the RDONet artifact is not strongly noticed and only slightly disturbs the visual quality.

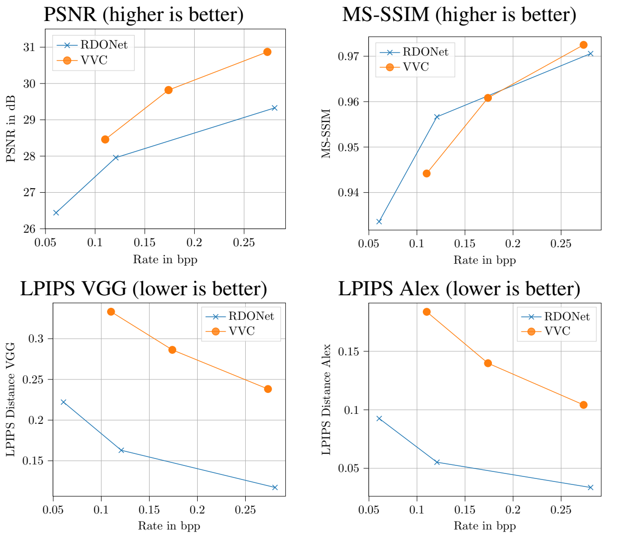
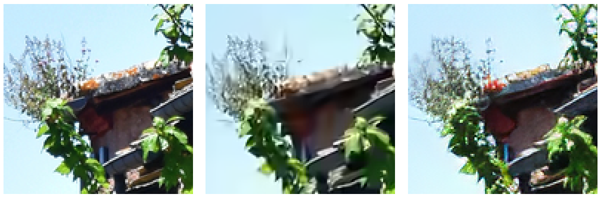


Figure 4. Objective quality evaluation comparing RDONet with VVEnC.



Original

VVEnC 0.215bpp

RDONet 0.196bpp

Figure 5. Visual examples of compressed images

5. Conclusion

In this paper, we have proposed a new RDONet, which contains several improvements over the previous versions. We have incorporated residual blocks and attention layers and also improved the performance using a Gaussian mixture model as probability model. We furthermore have shown that the RDONet structure is suitable for training on a perceptually motivated loss by combining LPIPS and a PatchGAN discriminator.

The proposed network outperforms VVC for perceptually motivated metrics like LPIPS and MS-SSIM. The coder was entered in the CLIC challenge 2022 with the team name *RDONet_FAU*.

References

[1] Eirikur Agustsson and Radu Timofte. NTIRE 2017 challenge on single image super-resolution: Dataset and study. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1122–1131, July 2017. 3

[2] Nicola Asuni and Andrea Giachetti. TESTIMAGES: A large data archive for display and algorithm testing. *Journal of Graphics Tools*, 17(4):113–125, 2013. 3

[3] Johannes Ballé, Valero Laparra, and Eero P Simoncelli. End-to-end optimized image compression. In *Proc. International Conference on Learning Representations (ICLR)*, pages 1 – 27, Apr 2017. 1, 2

[4] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. Density modeling of images using a generalized normalization transformation. In *Proc. International Conference on Learning Representations (ICLR)*, Jan, 2016. 3

[5] Johannes Ballé, David Minnen, Saurabh Singh, Sung Jin Hwang, and Nick Johnston. Variational image compression with a scale hyperprior. In *Proc. International Conference on Learning Representations (ICLR)*, pages 1–47, 2018. 2

[6] Fabian Brand, Kristian Fischer, and Andre Kaup. Rate-distortion optimized learning-based image compression using an adaptive hierachical autoencoder with conditional hyperprior. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, June 2021. 1, 2, 3

[7] Fabian Brand, Kristian Fischer, Alexander Kopte, and André Kaup. Learning true rate-distortion-optimization for end-to-end image compression. *ArXiv Preprint*, Jan. 2022. 1, 3

[8] Benjamin Bross, Ye-Kui Wang, Yan Ye, Shan Liu, Jianle Chen, Gary J. Sullivan, and Jens-Rainer Ohm. Overview of the versatile video coding (VVC) standard and its applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(10):3736–3764, oct 2021. 1

[9] Zhengxue Cheng, Heming Sun, Masaru Takeuchi, and Jiro Katto. Learned image compression with discretized gaussian mixture likelihoods and attention modules. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 1, 2, 3

[10] Shangyin Gao, Yibo Shi, Tiansheng Guo, Zhongying Qiu, Yunying Ge, Ze Cui, Yihui Feng, Jing Wang, and Bo Bai. Perceptual learned image compression with continuous rate adaptation. In *Proc. Workshop and Challenge on Learned Image Compression (CLIC)*, volume 101, 2021. 3

[11] Jingning Han, Bohan Li, Debargha Mukherjee, Ching-Han Chiang, Adrian Grange, Cheng Chen, Hui Su, Sarah Parker, Sai Deng, Urvang Joshi, Yue Chen, Yunqing Wang, Paul Wilkins, Yaowu Xu, and James Bankoski. A technical overview of AV1. *Proceedings of the IEEE*, pages 1–28, 2021. 1

[12] Chao Huang, Haojie Liu, Tong Chen, Qiu Shen, and Zhan Ma. Extreme image coding via multiscale autoencoders with generative adversarial optimization. In *Proc. IEEE Visual Communications and Image Processing (VCIP)*, Dec. 2019. 2

[13] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *Proc. International Conference on Learning Representations (ICLR)*, pages 1–15, May 2015. 4

[14] Alex Krizhevsky and Geoffrey E Hinton. Using very deep autoencoders for content-based image retrieval. In *Proc. European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, pages 489–494, 2011. 2

[15] Fabian Mentzer, George D Toderici, Michael Tschannen, and Eirikur Agustsson. High-fidelity generative image compression. *Advances in Neural Information Processing Systems*, 33, 2020. 2

[16] David Minnen, Johannes Ballé, and George D Toderici. Joint autoregressive and hierarchical priors for learned image compression. In *Advances in Neural Information Processing Systems*, volume 31, pages 1–10, Dec. 2018. 1, 2

[17] Debargha Mukherjee, Jim Bankoski, Adrian Grange, Jingning Han, John Koleszar, Paul Wilkins, Yaowu Xu, and Ronald Bultje. The latest open-source video codec VP9 - an overview and preliminary results. In *Proc. Picture Coding Symposium (PCS)*. IEEE, Dec. 2013. 1

[18] Michael Schäfer, Sophie Pientka, Jonathan Pfaff, Heiko Schwarz, Detlev Marpe, and Thomas Wiegand. Rate-distortion-optimization for deep image compression. In *Proc. IEEE International Conference on Image Processing (ICIP)*. IEEE, Sept. 2021. 2

[19] Gary J. Sullivan, Jens-Rainer Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (HEVC) standard. *IEEE Transactions on Circuits and Systems for Video Technology*, 22(12):1649–1668, Dec 2012. 1

[20] G. J. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *IEEE Signal Processing Magazine*, 15(6):74–90, Nov 1998. 1

[21] Vijay Veerabadran, Reza Pourreza, Amirhossein Habibian, and Taco S. Cohen. Adversarial distortion for learned video compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2020. 2

[22] Yefei Wang, Dong Liu, Siwei Ma, Feng Wu, and Wen Gao. Ensemble learning-based rate-distortion optimization for end-to-end image compression. *IEEE Transactions on Circuits and Systems for Video Technology*, 31(3):1193–1207, Mar. 2021. 2

[23] Adam Wieckowski, Jens Brandenburg, Tobias Hinz, Christian Bartnik, Valeri George, Gabriel Hege, Christian Helmrich, Anastasia Henkel, Christian Lehmann, Christian Stoffers, Ivan Zupancic, Benjamin Bross, and Detlev Marpe. VVEnC: An open and optimized VVC encoder implementation. In *Proc. IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, pages 1–2. 4

[24] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3