

Content-Oriented Learned Image Compression

Meng Li, Shangyin Gao, Yihui Feng, Yibo Shi, and Jing Wang 

Huawei Technologies, Beijing, China
wangjing215@huawei.com

Abstract. In recent years, with the development of deep neural networks, end-to-end optimized image compression has made significant progress and exceeded the classic methods in terms of rate-distortion performance. However, most learning-based image compression methods are unlabeled and do not consider image semantics or content when optimizing the model. In fact, human eyes have different sensitivities to different content, so the image content also needs to be considered. In this paper, we propose a content-oriented image compression method, which handles different kinds of image contents with different strategies. Extensive experiments show that the proposed method achieves competitive subjective results compared with state-of-the-art end-to-end learned image compression methods or classic methods.

Keywords: Image compression · Content-oriented · Loss metric.

1 Introduction

An uncompressed image usually contains tremendous data that is expensive to store or transmit. Therefore, image compression, or image codec, which aims to reduce redundant information in image data for efficient storage and transmission, is an essential part of real-world applications. The frameworks of traditional image compression methods such as JPEG [25], JPEG2000 [24] and BPG [6] are sophisticatedly designed, which usually include modules of prediction, transformation, quantization, and entropy coding. However, each module in the frameworks is optimized separately, making it hard to achieve global optimality.

In recent years, with the development of deep learning, end-to-end (E2E) learned image compression methods are proposed. Compared with traditional codecs, the biggest advantage of E2E methods is that the whole framework can be jointly optimized, making it has a greater potential in compression efficiency. Impressively, it takes only five years for E2E image compression to outpace the traditional methods that have developed for 30 years in terms of rate-distortion performance. Mainstream E2E image compression methods rely on the variational autoencoder(VAE) [3] [4] [15] [21] or generative adversarial network (GAN) [1] [2] [20] [22] to compress images, and various loss metrics such as Mean Squared Error (MSE), Multi-Scale Structural Similarity Index (MS-SSIM) [27], Learned Perceptual Image Patch Similarity (LPIPS) [28], etc. are used to optimize the model. Most existing methods optimized all regions of



Fig. 1. Reconstructed image with different methods at a similar bitrate.

the image in the same way, and have different distortions at low bit rates. For example, Minnen [21] used MSE metric to train a VAE image compression network. In this optimization, structural information such as text and lines are correctly preserved, but the reconstructed image will become too blurry. Mentzer [20] tried to improve the image quality by introducing GAN and LPIPS. In this optimization, the informative details are well preserved, but the structural information is distorted, such as distorted text, warped face (demonstrate in the left part of Fig.1).

Image content plays an important role in human perception, and human eyes have different sensitivities to different content. However, the influence of image content has been largely ignored in learning-based image compression. Different regions of the images have different properties, so the training strategy should also be different. For example, pixels in flat or texture regions are strongly correlated, and it is better to use loss metrics with a large receptive field. In contrast, pixels in edge or structure regions have little correlation with their neighborhoods, and it is better to use loss metrics with a small receptive field. The usage of paintbrushes is a good analogy to describe it, large paintbrushes are usually used to make large textures and to lay large color blocks, while small paintbrushes are usually used to draw fine lines and color dots. Besides, for some special regions, like small face, it is needed to use more strict constraints to avoid deformation. As a result, to address the problems above, we propose a content-oriented image compression scheme, which is suitable for most of the existing frameworks. Specifically, in the training stage, we divide the image into different regions, and use different loss metrics according to their characteristics.

To the best of our knowledge, this is the first content-oriented image compression method that improves the visual perceptual quality without changing the architecture. An instance is shown in Fig. 1, which demonstrates the superiority of our method. The contributions of this paper are summarized as follows:

1. We propose a content-oriented E2E image compression scheme, in which we use different loss metrics in different regions. The region masks are used only in the training stage, and no extra information is needed in the encoding and decoding stage.
2. To evaluate the scheme, we design a GAN-based architecture as an instance to show the effectiveness. Several classic and E2E methods are used for comparison, and our method shows the best performance.

2 Related Work

2.1 Loss metrics

MAE/MSE Loss MSE and the mean absolute error (MAE) are two of the most commonly used loss metrics in image compression. MAE and MSE metrics assume pixel-wise independence, and constrains the accuracy of the corresponding pixels. However, the main drawback of MSE loss metric mostly yields an oversmoothed reconstruction, which results in a lack of high-frequency details in the edges and textures. Given an image x and its reconstructed version \tilde{x} , the MAE and MSE loss are computed by:

$$\begin{aligned}\mathcal{L}_{mae} &= \mathbb{E}\|x - \tilde{x}\|_1 \\ \mathcal{L}_{mse} &= \mathbb{E}\|x - \tilde{x}\|_2^2\end{aligned}\tag{1}$$

Laplacian Loss Laplacian is a differential operator given by the divergence of the gradient, which is always used to detect the high-frequency component of images. The Laplacian loss is defined as the mean-squared distance between their Laplacians:

$$\mathcal{L}_{lap} = \mathbb{E}\|L(x) - L(\tilde{x})\|_2^2\tag{2}$$

Where \mathcal{L}_{lap} is the Laplacian loss value. As shown in Eq.2, it is also a point-to-point loss metric, which constrains the Laplacians of the corresponding location. By doing this, the high-frequency component of the area is preserved.

LPIPS Loss LPIPS is a state-of-the-art perceptual metric proposed by Zhang [28]. He uses architectures trained for classification as feature extractors, and leverages the power of the features to judge the similarity between the reconstructed image and the original one, which is computed by:

$$\mathcal{L}_{lips} = \sum_k \tau^k (f^k(x) - f^k(\tilde{x}))\tag{3}$$

where f denotes the feature extractor, and τ^k computes the score of features from the k -th layer of the architecture. The LPIPS value is the averaged score of all layers. It is a loss metric with a large receptive field, which constrains the

distribution of corresponding location. The LPIPS metrics makes images more semantically similar, which is more consistent with human perception. However, the absence of point to point constraints may result in geometric distortion.

GAN Loss GAN is widely used for perceptual image compression. By taking advantage of adversarial training, GAN-based architectures can produce more photo-realistic images. It contains two rivaling networks: the generator G is used to generate an image $\tilde{x} = G(\hat{y})$ that is consistent with the input image distribution p_x , and the discriminator D is used to predict if the input image is an image generated by G . The goal of GAN for G is to produce images that are real enough to fool D . In the procedure of training, the image produced by G becomes more and more authentic, and D becomes more and more discriminating, finally reaching a balance.

2.2 learned image compression methods

Many E2E image compression methods take advantage of VAE as its backbone, which usually consists of four components: the encoder, the quantizer, the entropy module and the decoder. The encoder is used to encode the input image x into a latent representation y , which is then quantized to \hat{y} by the quantizer. The entropy module, which is used to estimate the distribution of \hat{y} , plays an important role in minimizing the rate-distortion cost. Finally, the quantified feature \hat{y} is transformed to reconstruct the image \tilde{x} . The framework is directly optimized for the rate-distortion trade-off, that is: $\mathcal{L}_{RD} = R + \lambda d$. Where R represents the bitrate which is lower-bounded by the entropy of the discrete probability distribution of \hat{y} , d is the reconstruction distortion, λ controls the trade-off. The framework is first proposed by Ballé in reference [3], where they introduced the widely used generalized divisive normalization (GDN) transform. In their following works, they proposed the hyper prior [4] to better exploit the probability distribution of the latent feature. And in [21] and [15], spacial context structure is proposed to improve the performance of the entropy module.

On the basis of above works, some researchers proposed to use GAN and some perceptual metrics (e.g. LPIPS) to improve the visual perceptual quality. Rippel [22] first introduced the effectiveness of GAN in generating perceptual friendly images at an extremely low bitrate. In the following works, Agustsson [1], Akutsu [2], Dash [10] also make use of GAN loss to improve the perceptual quality of reconstruction images. According to rate-distortion-perception theory presented in [5], Mentzer combines MSE (Mean squared error), LPIPS and GAN loss to generate images of competitive perceptual quality [20]. In this paper, we also use a GAN-based architecture, and what distinguishes us from previous works is that we take the image contents into consideration, and adopt different strategies on different target regions.

Another related topic of the work is content-related image compression. Li introduced a content-weighted importance map to guide the allocation of local bit rate [17]. Similarly, Cai proposed a CNN based ROI image compression

method to allocate more bits to the ROI areas [7]. Besides, Zhang proposed an attention-guided dual-layer image compression, which employs a CNN module to predict those pixels on and near a saliency sketch within ROI that are critical to perceptual quality [29]. However, most of the existing content-related works need to change the architecture and allocate more bits in important fields, thus increasing the network complexity and reducing the compression efficiency. Different from above, we proposed a content-oriented image compression method by adopting different loss strategies on different areas. The advantage is that our scheme can be applied in most of the previous works without changing the architecture. Besides, the image content masks are learned by the network during the training, so no extra information is needed when encoding or decoding an image.

3 Method

3.1 Framework of our method

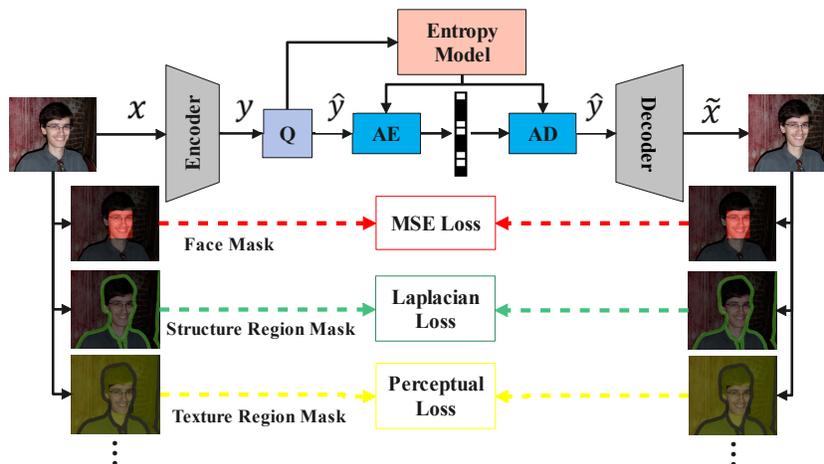


Fig. 2. Framework of our method.

From Section 2.1, it is easy to deduce that different loss functions are suitable for different image contents. However, existing learned image compression always optimizes the whole image with the same loss function. In existing content-related image compression methods, important maps [17], ROI masks [7], or other extra information need to be encoded as part of the codestream. On the one hand, such methods require changing the encoder-decoder structure to make the architecture more complex by adding a special module to extract the important maps or ROI masks, on the other hand, they also increase the extra

bits that inevitably degrade the coding efficiency. This problem can be overcome if the architecture is ‘clever’ enough to distinguish different image contents and employ different compression strategies. The whole framework of our method is shown in Fig.2. The mask information is only used to select different loss functions during the training phase and not needed in the inference phase.

As shown in Fig.2, E2E image compression is transformed from unlabeled to labeled learning. The choice of the loss functions depends on the classification of image regions. According to the previous analysis, we divide the image into three image regions, namely texture region, structure region, and small face region.

Structure Region Structure regions usually have strong gradients and have little statistical correlation with very close neighborhoods due to the abrupt changes in pixels. And loss function with large receptive field will introduce additional noise, which is not acceptable for a precise edge reconstruction. The human eye is sensitive to the sharpness and the pixel-wise correctness of the structure, so a point-wise loss function is more suitable. However, MSE will lead to a blurred reconstruction, affecting the subjective visual perception. As shown in Fig.3 [19], considering a simple 1-dimensional case. If the model is only optimized in image space by the L1 or MSE loss, we usually get a reconstructed sequence as Fig.3(b) given an input testing sequence whose ground-truth is a sharp edge as Fig.3(a). The model fails to recover sharp edges for the reason that the model tends to give a statistical average. If a second-order gradient constraint is added to the optimization objective as Fig.3(f), the probability of recovering Fig.3(c) is increased significantly. As a result, we choose the Laplacian loss in structure region. The structure loss function is as follows:

$$\mathcal{L}_{stru} = M_{stru} \circ \mathcal{L}_{lap} \quad (4)$$

where M_{stru} denotes the mask of the structure region, \mathcal{L}_{lap} is computed with Eq.2. We use Laplacian edge detector to generate the M_{stru} , which will be introduced in Section 4.1.

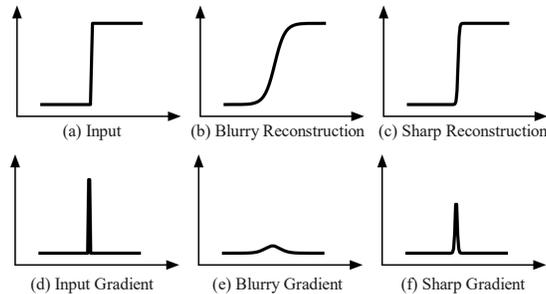


Fig. 3. An illumination of a simple 1-D case. The first row shows the pixel sequences and the second row shows their corresponding gradients.

Texture Region Unlike structure regions, texture regions always have more details and domain pixels are highly correlated. It is difficult for the human eyes to perceive pixel-by-pixel correctness, and people are more concerned about the distribution of the texture. Existing perceptual optimization based methods [20] have achieved very compelling texture reconstruction results. Therefore, we use the perceptual loss in this region. In our work, the perceptual loss are constituted of three part: MAE loss, LPIPS loss, and GAN loss. The formulation can be denoted by:

$$\mathcal{L}_{per} = M_{tex} \circ (\alpha \cdot \mathcal{L}_{mae} + \beta \cdot \mathcal{L}_{LPIPS} + \delta \cdot \mathcal{L}_{GAN}) \quad (5)$$

where M_{tex} denotes the mask of the texture region, \mathcal{L}_{mae} , \mathcal{L}_{LPIPS} are computed with Eqs.1, 3, \mathcal{L}_{GAN} will be introduced in Section 3.2. In our work, the VGG [23] network is chosen as the backbone to compute the LPIPS loss.

Small Face Region In our framework, the facial regions will be classified as texture regions and optimised with perceptual loss if not intervened. In this case, larger faces can be well reconstructed, but small faces may be warped, as shown on the left side of Fig.1. Therefore, we adopt a different loss function for regions of the small faces. Generally, people are very sensitive to the correctness of the face structure, for which an accurate reconstruction is necessary. Therefore, we use a stricter constraint loss, the MSE loss, for the facial image reconstruction.

$$\mathcal{L}_{sface} = M_{sface} \circ \mathcal{L}_{mse} \quad (6)$$

where the M_{sface} denotes the mask of the small face regions, and \mathcal{L}_{mse} is computed with Eq.1. We use the well-known YOLO-face to detect the faces in the image, and \mathcal{L}_{sface} is only adopted to small faces. The bitrate of the quantized latent representation \hat{y} is estimated by the entropy module denoted by P, $R(\hat{y}) = -\log(P(\hat{y}))$. Finally, the loss function of the whole image is summarized as:

$$\mathcal{L}_{total} = \eta R(\hat{y}) + \epsilon \mathcal{L}_{stru} + \mathcal{L}_{per} + \gamma \mathcal{L}_{sface} \quad (7)$$

where η , ϵ and γ are weights of corresponding loss metrics. Because people usually pay more attention to faces of the images, we intentionally allocate more bits on small faces by using a larger γ . Note that the masks are binary-valued and mutually exclusive. And the priority of the masks is different, specifically, from high to low: the facial mask, the structure mask and the texture mask. In other words, the structure mask doesn't cover the facial areas, and the texture mask doesn't cover the facial and structure areas.

3.2 Architecture

To prove the effectiveness of our method, we design an E2E image compression architecture, as shown in Fig.4. It consists of four parts: the encoder E, the decoder/generator G, the entropy module P, and the discriminator D.

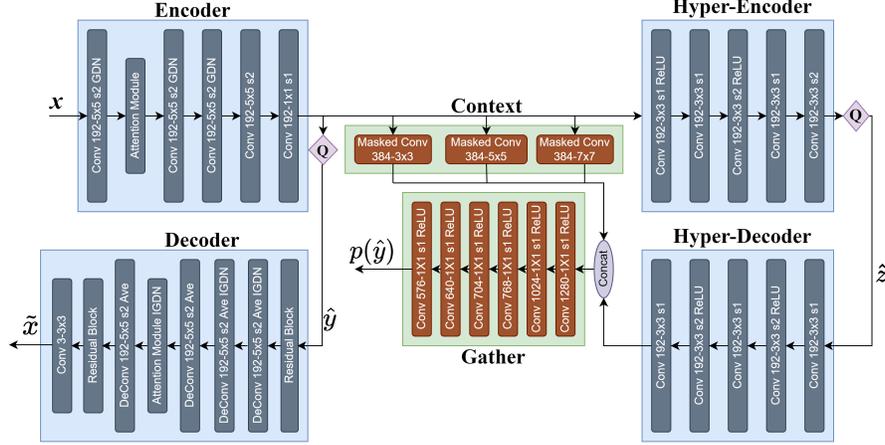


Fig. 4. The architecture of our method. *Conv* $192 - 5 \times 5$ is a convolution with 192 output channels, with 5×5 filters. *DeConv* is a deconvolution operation. *s2* means the stride of this convolution or deconvolution is 2. *Ave* means average pooling. *GDN* or *ReLU* is used to increase the non-linearity.

Autoencoder For the encoder and generator/decoder, GDN [3] is used to normalize the intermediate feature and also play a role of non-linearity element. Besides, to capture the global dependencies between features, we introduced the attention module (the residual non-local attention block, RNAB) [30] integrated into the architecture. It is well-known that deconvolution operation, or transposed convolution always generates checkerboard artifacts on the reconstructed image. The main reason is, the up-sampled feature map generated by deconvolution can be considered as the result of periodical shuffling of multiple intermediate feature maps computed from the input feature map by independent convolutions [12]. As a result, adjacent pixels on the output feature map are not directly related, resulting in checkerboard artifacts. To alleviate this issue, we add an extra average pooling layer after every deconvolution layer to strengthen the association between adjacent pixels.

Entropy model We adopt the context-based model [21] to extract side information z , which is used to model the distribution of latent representation \hat{y} . And uniform noise $U(-1/2, 1/2)$ is used to simulate quantization in the hyper-encoder and when estimating $p(\hat{y}|z)$. The distribution of \hat{y} is modeled with an asymmetric Gaussian entropy model [9], which can be denoted by:

$$p_{\hat{y}|z}(\hat{y} | \hat{z}) \sim N(\mu, \sigma_l^2, \sigma_r^2) \quad (8)$$

where μ represent the mean of the latent representation, σ_l^2 and σ_r^2 represent the estimated left-scale and right-scale parameter, respectively. The asymmetric

Gaussian model has stronger representation ability when the entropy estimation do not obey the Gaussian distribution.

Discriminator In our framework, adversarial training is adopted to improve the perceptual quality of the reconstructed images. Instead of a standard discriminator, we borrow the relativistic average discriminator [13] used in [26], which tries to predict the probability that a groundtruth image x is relatively more realistic than a generated one \tilde{x} , on average. The loss is divided into two parts, the generator loss \mathcal{L}_D^{Ra} , and the discriminator loss \mathcal{L}_G^{Ra} :

$$\begin{aligned}\mathcal{L}_D^{Ra} &= -\mathbb{E}_x [\log (D_{Ra} (x, \tilde{x}))] - \mathbb{E}_{\tilde{x}} [\log (1 - D_{Ra} (\tilde{x}, x))] \\ \mathcal{L}_G^{Ra} &= -\mathbb{E}_x [\log (1 - D_{Ra} (x, \tilde{x}))] - \mathbb{E}_{\tilde{x}} [\log (D_{Ra} (\tilde{x}, x))]\end{aligned}\quad (9)$$

where $D_{Ra} (x, \tilde{x}) = \sigma (C (x) - \mathbb{E}_{\tilde{x}} [C (\tilde{x})])$. $C (x)$ and $C (\tilde{x})$ are the the non-transformed discriminator output. σ is the sigmoid function, and $\mathbb{E}_x [\cdot]$ computes the average output. Moreover, the PatchGAN discriminator [16], which has been proven to improve the quality of the generated images, is also utilized in our architecture. The PatchGAN has fewer parameters. And it not only preserves more texture but can also be applied to images of arbitrary sizes. The detailed architecture of our discriminator is shown in Fig.5.

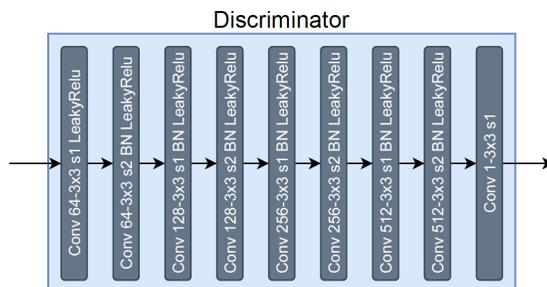


Fig. 5. The architecture of discriminator. Same notation is used as in Fig.4.

3.3 Implementation of masked perceptual loss

Usually, Adding a pixel-level mask to MSE or MAE is easy with simple point-wise multiplication. But it is a little harder to used on LPIPS or GAN losses, because these two loss functions compute the feature losses and cannot correspond to mask pixel-to-pixel. In this section, we propose a method called ‘real value replacement’ to solve the problem.

Consider the mask M_{tex} , the original image x , and the reconstructed image

\tilde{x} , we replace the value of the mask part of \tilde{x} with the corresponding value of x to get the replaced reconstructed image \tilde{x}' :

$$\tilde{x}' = (1 - M_{tex}) \circ x + M_{tex} \circ \tilde{x} \quad (10)$$

Then calculate the loss function directly with x and \tilde{x}' to estimate the masked loss:

$$\begin{aligned} M_{tex} \circ \mathcal{L}_{LPIPS}(x, \tilde{x}) &\approx \mathcal{L}_{LPIPS}(x, \tilde{x}') \\ M_{tex} \circ \mathcal{L}_{GAN}(x, \tilde{x}) &\approx \mathcal{L}_{GAN}(x, \tilde{x}') \end{aligned} \quad (11)$$

4 Experiment

4.1 Training Details

Due to the adoption of face loss, a large number of face images are required for training. For this purpose, we use the well-known MSCOCO dataset [18] as our training set, and Kodak dataset [14] as our testing set. It is notoriously hard to train GANs for image generation, so the training procedure is divided into two stages. In the first stage, we only use the MSE as the distortion loss to guide the optimization at the pixel level reconstruction. The optimization target in the first stage is to minimize $\mathcal{L}_{RD} = \eta R + \mathcal{L}_{mse}$. Using the first stage result as the pre-trained model, we can train the perceptual optimized model with the loss function Eq.7 mentioned in section 3.2 in the second stage. During the training phase, we randomly crop the images into patches of size 256×256 and set the batch size to 8. The initial learning rate is set to 0.0001 and halved at 160k and 500k iteration. We use kaiming initialization [11] to initialize all our models. The weights of the loss are set: $\alpha = 0.01$, $\beta = 1$, $\gamma = 0.2$, $\delta = 0.0005$, and $\epsilon = 0.3$. We only modify η for different target bit-rates.

In order to use different loss metrics on different regions of the images, the masks of different content regions are required. In our experiment, the faces in the images are detected using the well-known YOLO-face [8]. Then the coordinate information of the faces is stored in an XML file, which is used to generate the face masks in the training phase. By doing this, we can save the face detection time during training the model. For structure region masks, due to the low complexity of edge detection, we do not generate masks in advance, but directly use Laplacian edge detectors to detect structure regions during training.

4.2 Ablation Study

Ablation on the loss metric In order to show the effectiveness of the proposed, we conduct several comparative experiments as follows.

- Case 1, the model is optimized only with MSE loss. (MSE)
- Case 2, the model is optimized with MAE, LPIPS and GAN loss. (Perceptual)
- Case 3, the model is optimized with MAE, LPIPS, GAN and Laplacian loss. (Perceptual + Lap)

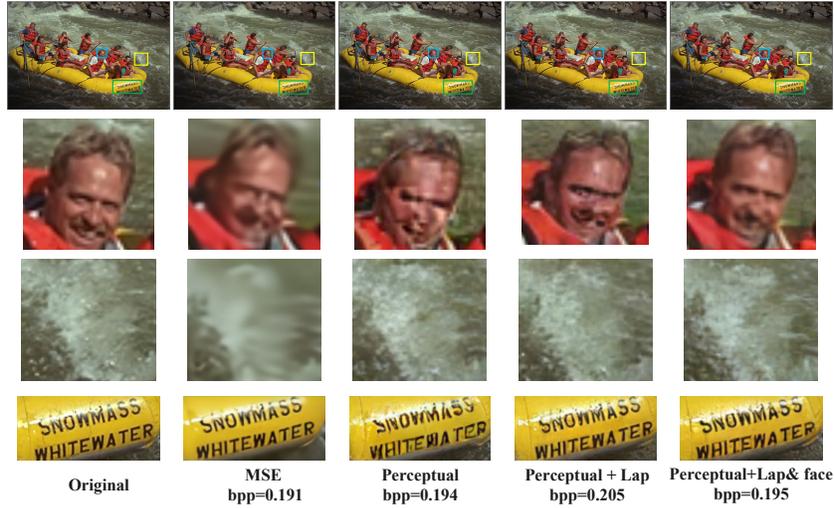


Fig. 6. Comparison of different loss strategies.



Fig. 7. Comparison of the discriminators. Fig. 8. Ablation on the average pooling.

- Case 4, the model is optimized with MAE, LPIPS, GAN, Laplacian and face loss. (Perceptual + Lap & face)

Take the image kodim14 as an example, the visual comparison of different cases is shown in Fig.6. All of these cases are compressed at a similar bitrate, around 0.2 bpp. Overall, it can be clearly observed that the reconstructed image in case 5 has high-fidelity facial features, clear and correct text, informatively detailed backgrounds and an overall harmonious visual quality. Specifically, in case 1, the faces and water waves are too blurred, which attributes to the property of the MSE loss function. In case 2, with the help of additional loss functions GAN and LPIPS, the informative details of water waves are restored, but the facial features and text are severely warped. In case 3, with the help of additional edge loss, not only the informative details of water waves are restored, but the text distortion phenomenon also disappears. However, the facial distortion remains. Until all the losses are added, in case 5, the best visual effect is achieved in all the areas. Through the ablation study, we can draw two conclusions: (1) the architecture proposed in 3.3 is able to distinguish different image contents; (2) adopting appropriate loss function for different image content helps improve the visual quality of reconstructed images.

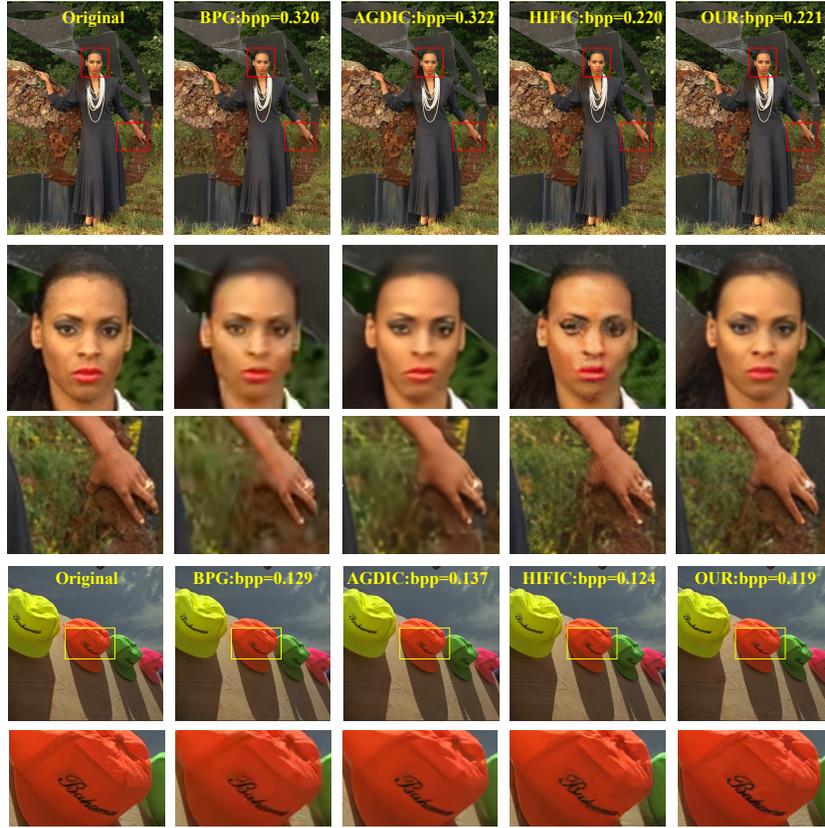


Fig. 9. Compressed results of kodim18 using different methods.

Ablation on the architecture In order to get better result, we choose the better components in our architecture, such as the asymmetric Gaussian entropy model, the relatively average discriminator, and the average pooling. We have done relevant experiments to demonstrate the effectiveness of our architecture. The results show that with the help of the asymmetric Gaussian model, we achieve about 0.68% BD-rate reduction compared with the general Gaussian entropy model. We also tested the effectiveness of our discriminator, and the result shown in Fig.7 proves that our discriminator can preserve more details than that of HIFIC. Besides, Fig.8 shows that the average pooling can alleviate the checkerboard artifacts.

4.3 Visual Results

To show the advantage of our method, we compare our method (named COLIC) with BPG, Asymmetric Gained Deep Image Compression (AGDIC) optimized with MSE [9] and High-Fidelity Generative Image Compression (HIFIC, SOTA

perceptual optimized method) [20]. Since BPG and AGDIC are not perceptual-optimized, we select relatively higher bitrates for them. The visual results are shown in Fig.9, in which the cropped blocks highlight the reconstruction of certain regions, such as textures, texts, and small faces. As can be seen from Fig.9, although BPG and AGDIC can maintain the correct facial structure and achieve relatively better results in text area, the texture results are over-smoothed. As shown in the third row, the background generated by BPG and AGDIC is too blurry. On the contrary, although HIFIC can reconstruct the informative details, it will lead to the distortion of face and text. As shown in the second and fifth rows, the face and text generated by HIFIC are overwarpped. Compared to these results, it is clear that our proposed COLIC achieves the best visual effects. It handles all these situations better, recovering more texture detail, correct small face and text structure.

More examples of comparison with HIFIC are shown in Fig.10. Compared with HIFIC, COLIC can not only recover similar informative texture regions, such as the spray, but also better structured regions such as lines, textures and small faces. In a word, COLIC can achieve better visual effects.

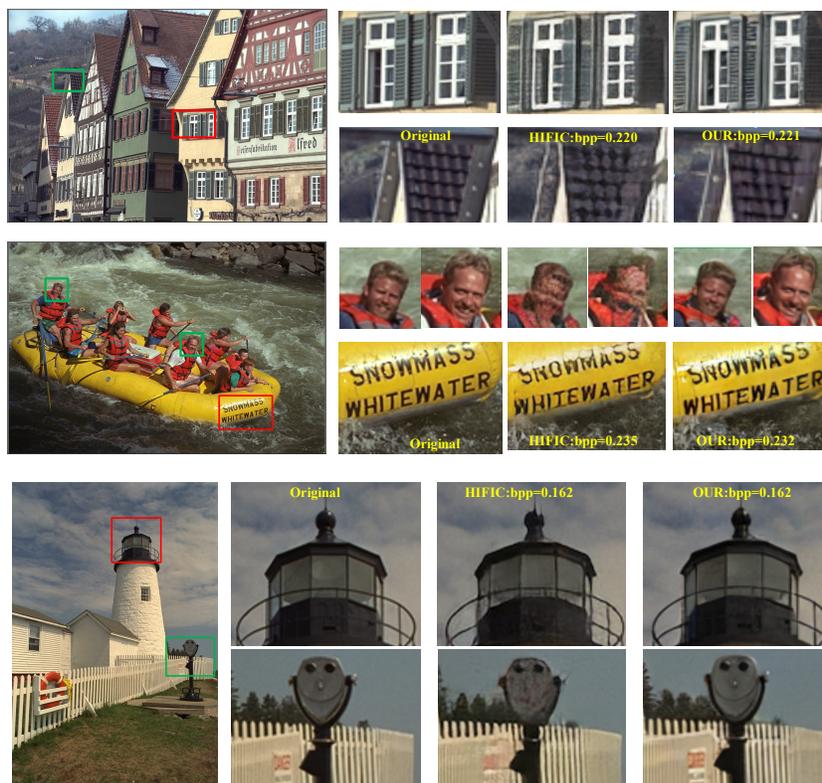


Fig. 10. Compressed results of kodim18 using different methods.

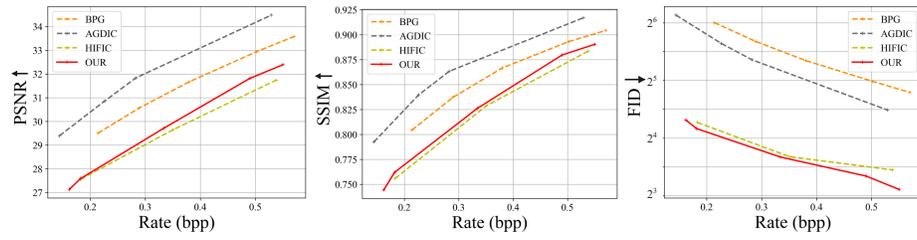


Fig. 11. Rate-distortion and -perception curves on Kodak. Arrows indicate whether lower is better (\downarrow), or higher is better (\uparrow).

4.4 Statistics result

In this section, we use metrics such as PSNR, SSIM, and FID to quantify our results. PSNR and SSIM are widely used in various digital image evaluations, which compute the average pixel-wise distance and structural similarity between the two images. Unlike PSNR, SSIM, which measure the similarity between individual images pairs, FID assesses the similarity between the distribution of the reference images and the generated/distorted images. FID is a good metric for evaluating the subjective results and is widely used for super-resolution or generative tasks.

The R-D curves of COLIC, BPG, AGDIC and HIFIC are shown in Fig.11. Compared with BPG and AGDIC, as expected, COLIC and HIFIC dominate in perceptual metric FID, but relatively poor on objective metrics PSNR and SSIM, which can be explained by the rate-distortion-perception theory [5] that they sacrifice some objective performance to improve the perceptual quality. In order to improve the visual quality, we sacrifice some of the objective metrics. Compared with HIFIC, COLIC achieves better results on both objective and perceptual metrics. It attributes to the better optimization loss that imposes the correct constraints on the structure region, the better network structure that with better reconstruction ability, and better training strategy.

5 Conclusion

In this work, we propose a content-oriented image compression scheme, which can be used in most existing methods. We suggest that different loss metrics should be used on different image contents according to their characteristics. And a GAN-based architecture is designed to prove the effectiveness of our scheme. Experiments clearly show the superiority of our method on visual quality as well as different metrics. In fact, the effectiveness of this method demonstrates that existing encoders and decoders are 'smart' enough to distinguish different image regions and employ different reconstruction strategies with the guidance of different training loss. Therefore, human perceptual priors can be better utilized through supervised training methods to obtain better subjective results.

References

1. Agustsson, E., Tschannen, M., Mentzer, F., Timofte, R., Van Gool, L.: Generative adversarial networks for extreme learned image compression. In: 2019 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 221–231 (2019). <https://doi.org/10.1109/ICCV.2019.00031>
2. Akutsu, H., Suzuki, A., Zhong, Z., Aizawa, K.: Ultra low bitrate learned image compression by selective detail decoding. In: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). pp. 524–528 (2020). <https://doi.org/10.1109/CVPRW50498.2020.00067>
3. Ballé, J., Laparra, V., Simoncelli, E.: End-to-end optimized image compression. In: Proceedings of the International Conference on Learning Representations (ICLR) (April 2017)
4. Ballé, J., Minnen, D., Singh, S., Hwang, S., Johnston, N.: Variational image compression with a scale hyperprior. In: Proceedings of the International Conference on Learning Representations (ICLR) (April 2018)
5. Blau, Y., Michaeli, T.: Rethinking lossy compression: The rate-distortion-perception tradeoff. In: Chaudhuri, K., Salakhutdinov, R. (eds.) Proceedings of the 36th International Conference on Machine Learning. vol. 97, pp. 675–685 (Jun 2019)
6. BPG: Bpg image format (2014), accessed 7 March 2022. <https://bellard.org/bpg/>
7. Cai, C., Chen, L., Zhang, X., Gao, Z.: End-to-end optimized roi image compression. *IEEE Transactions on Image Processing* **29**, 3442–3457 (2020). <https://doi.org/10.1109/TIP.2019.2960869>
8. Chen, W., Huang, H., Peng, S., Zhou, C., Zhang, C.: Yolo-face: a real-time face detector. *The Visual Computer* **37** (04 2021). <https://doi.org/10.1007/s00371-020-01831-7>
9. Cui, Z., Wang, J., Gao, S., Guo, T., Feng, Y., Bai, B.: Asymmetric gained deep image compression with continuous rate adaptation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10532–10541 (June 2021)
10. Dash, S., Kumaravelu, G., Naganoor, V., Raman, S.K., Ramesh, A., Lee, H.: Compressnet: Generative compression at extremely low bitrates. In: 2020 IEEE Winter Conference on Applications of Computer Vision (WACV). pp. 2314–2322 (2020). <https://doi.org/10.1109/WACV45572.2020.9093415>
11. He, K., Zhang, X., Ren, S., Sun, J.: Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In: 2015 IEEE International Conference on Computer Vision (ICCV). pp. 1026–1034 (2015). <https://doi.org/10.1109/ICCV.2015.123>
12. Hongyang, G., Hao, Y., Zhengyang, W., Shuiwang, J.: Pixel deconvolutional networks. In: Proceedings of the International Conference on Learning Representations (ICLR) (April 2018)
13. Jolicoeur-Martineau, A.: The relativistic discriminator: a key element missing from standard gan. In: Proceedings of the International Conference on Learning Representations (ICLR) (May 2019)
14. Kodak, E.: Kodak lossless true color image suite: Photocd pcd0992 (1993), accessed 7 March 2022. <http://r0k.us/graphics/kodak/>
15. Lee, J., Cho, S., Beack, S.K.: Context-adaptive entropy model for end-to-end optimized image compression. In: Proceedings of the International Conference on Learning Representations (ICLR) (August 2018)

16. Li, C., Wand, M.: Precomputed real-time texture synthesis with markovian generative adversarial networks. In: European Conference on Computer Vision (ECCV). pp. 702–716 (September 2016)
17. Li, M., Zuo, W., Gu, S., Zhao, D., Zhang, D.: Learning convolutional networks for content-weighted image compression. In: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3214–3223 (2018). <https://doi.org/10.1109/CVPR.2018.00339>
18. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: European Conference on Computer Vision (ECCV). pp. 740–755. Springer International Publishing (2014)
19. Ma, C., Rao, Y., Cheng, Y., Chen, C., Lu, J., Zhou, J.: Structure-preserving super resolution with gradient guidance. pp. 7766–7775 (06 2020). <https://doi.org/10.1109/CVPR42600.2020.00779>
20. Mentzer, F., Toderici, G., Tschannen, M., Agustsson, E.: High-fidelity generative image compression. In: Advances in Neural Information Processing Systems (NeurIPS) (06 2020)
21. Minnen, D., Ballé, J., Toderici, G.: Joint autoregressive and hierarchical priors for learned image compression. In: Proceedings of the 32nd International Conference on Neural Information Processing Systems. p. 10794–10803 (2018)
22. Rippel, O., Bourdev, L.: Real-time adaptive image compression. In: Proceedings of the 34th International Conference on Machine Learning. vol. 70, pp. 2922–2930. PMLR (August 2017)
23. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: International Conference on Learning Representations (2015)
24. Taubman, David, S.: Jpeg2000: Image compression fundamentals, standards and practice. *Journal of Electronic Imaging* **11**(2), 286 (2002)
25. Wallace, G.: The jpeg still picture compression standard. *IEEE Transactions on Consumer Electronics* **38**(1), 18–34 (1992). <https://doi.org/10.1109/30.125072>
26. Wang, X., Yu, K., Wu, S., Gu, J., Liu, Y., Dong, C., Qiao, Y., Change Loy, C.: Esrgan: Enhanced super-resolution generative adversarial networks. In: Proceedings of the European Conference on Computer Vision (ECCV) Workshops (September 2018)
27. Wang, Z., Simoncelli, E., Bovik, A.: Multiscale structural similarity for image quality assessment. In: The Thirty-Seventh Asilomar Conference on Signals, Systems Computers, 2003. vol. 2, pp. 1398–1402 Vol.2 (2003). <https://doi.org/10.1109/ACSSC.2003.1292216>
28. Zhang, R., Isola, P., Efros, A.A., Shechtman, E., Wang, O.: The unreasonable effectiveness of deep features as a perceptual metric. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2018)
29. Zhang, X., Wu, X.: Attention-guided image compression by deep reconstruction of compressive sensed saliency skeleton. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13354–13364 (June 2021)
30. Zhang, Y., Li, K., Li, K., Zhong, B., Fu, Y.R.: Residual non-local attention networks for image restoration. *ArXiv abs/1903.10082* (2019)