# CoCliCo: Extremely low bitrate image compression based on CLIP semantic and tiny color map

Tom Bachard, Tom Bordin, Thomas Maugey

# CoCliCo: Extremely low bitrate image compression based on CLIP semantic and tiny color map

Tom Bachard[†]
Univ Rennes, INRIA
Rennes, France

Tom Bordin[†]
INRIA
Rennes, France

Thomas Maugey
INRIA
Rennes France

***Abstract*—Coding algorithms are usually designed to pixel-wisely reconstruct images, which limits the expected gains in terms of compression. In this work, we introduce a semantic compressed representation for images: CoCliCo. We encode the inputs into a CLIP latent vector and a tiny color map, and we use a conditional diffusion model for reconstruction. When compared to the most recent traditional and generative coders, our approach reaches drastic compression gains while keeping most of the high-level information and a good level of realism.**

## I. Introduction

Compressing images at extremely low bitrates represents a challenge, notably because of the increasing amount of data produced. It is particularly the case, when it comes to cold data – rarely or never accessed data – where highly compressed method of storage should be optimized.

When targeting extremely low rates, every bit of information becomes crucial to the description and reconstruction of the signal. The way bits are spent to describe the signal must be optimized to be as faithful to the source as possible. To measure the fidelity to the source, from low to high bitrates, the mean squared error (MSE) came as a natural criterion to evaluate compression. However, the MSE has its drawbacks. When optimizing compression, Blau *et al.* [1] observe a tradeoff between the distortion (the *pixel fidelity* defined by the MSE metric) and the perception (the *realism* of the decoded image or the perceived quality). This tradeoff is even more important at extremely low bitrates [2] where providing little information on the signal is not enough to produce images that are both realistic and faithful. Optimizing the MSE will provoke a decrease in perception because of compression artifacts. On the other hand, decoding a *realistic* image with few bits leads to high-level differences with the original image. A recent trend is to target realism at the expense of the MSE. In this context, fidelity is expressed at the semantic level.

Several methods already integrate perceptual objectives on top of the distortion, notably [3] with a framework allowing navigation in the rate-perception-distortion tradeoff up to relatively low bitrates. Going further and completely discarding distortion for extremely low bitrates, several methods rely on semantic descriptions for image representation. The semantic description can be done using text as first showcased in [4] using a fully human compression scheme for description and decoding. In [5], the authors discuss the interest of using a textual coding scheme for image compression, favoring semantic fidelity and realism. A generative approach is proposed in [6] using textual inversion to generate captions from images on top of a light sketch of the image to add positional information. In [7], we proposed a framework for semantic compression relying on a representation using segmentation and color maps to condition a generative model. This representation of the semantic is limited by a finite number of labels, and with a fixed representation, the bitrate cannot variate. Motivated by the compact semantic representation that the foundation model CLIP [8] offers, as shown in [9], we propose a coding scheme relying on CLIP semantic representation in a context of semantic based generative compression using a latent diffusion model (LDM) [10] as our decoder. Our proposed codec CoCliCo (COmpression, CLIp, COlor map) encodes an image as a quantified CLIP latent vector together with a quantized down-sized color of the image. This method allows us to achieve extremely low bitrates while still being able to reconstruct faithfully the input images at a high level. As we can see in Fig. 1, the semantic and realism of the image is maintained at the cost of the pixel fidelity, contrary to what is done is classical codecs.


Fig. 1: Comparison of the decoded image at very low bitrate of our model and VVC. Image taken from the Wikimedia Common files.

## II. Generative Compression Paradigm

Fig. 2 presents the semantic based generative compression framework. The input image $\mathbf{x}$ is encoded into a latent semantic representation $\sigma$ via the semantic encoder $\mathcal{E}$. The image generator $\mathcal{D}$, acting as the decoder, reconstructs the decoded input $\tilde{\mathbf{x}}$ using the semantic present in the latent representation. Unlike classical compression, the error is not evaluated with a classical pixel-based loss (MSE), but rather

[†]These authors contributed equally to this work. The names are arranged in alphabetical order.

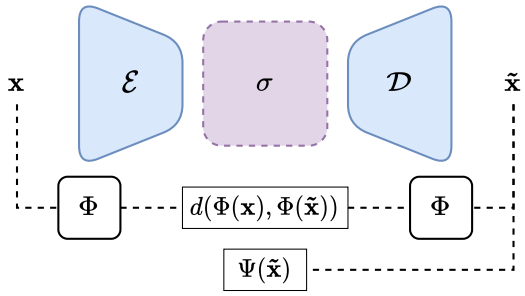


Fig. 2: Semantic based generative compression framework. The codec is made of $(\mathcal{E}-\mathcal{D})$ and we propose to use the semantic representation $\sigma$. The semantic projection function is $\Phi$.

with a realism metric $\Psi$ evaluating to which extent the image is likely to be a natural image. To also ensure that inputs and outputs are correlated, in terms of semantic, we propose, to project the images, $\mathbf{x}$ and $\tilde{\mathbf{x}}$, to a semantic space via $\Phi$, a non-linear projection function. We then express the (semantic) distortion error as $d_\Phi(\mathbf{x}, \tilde{\mathbf{x}}) = d(\Phi(\mathbf{x}), \Phi(\tilde{\mathbf{x}}))$, where $d$ is a similarity function in the semantic space. Two images can be semantically close while being pixel-wise different.

Given $\Psi$ and $\Phi$, we define the problem as maximizing the realism of the reconstruction under extremely low bitrates constraints $R < R_t$ and semantic fidelity to the input $d(\Phi(\mathbf{x}), \Phi(\tilde{\mathbf{x}})) < d_t$. See as follows:

$$\min_{\mathcal{E}} \; R(\mathcal{E}(x)) \text{ s.t.} \tag{1}$$
$$\Psi(x) > \Psi_t \text{ and } d_\Phi(\mathbf{x}, \tilde{\mathbf{x}}) < d_t$$

## III. Proposed approach

### A. Semantic of an image

The representation of the image in our framework non longer relies on the classical pixel fidelity but on a semantic fidelity instead. In this work, we define semantic fidelity as the combination of two complementary semantic aspects of the image:

- the semantic content of the image;
- the semantic organization of the image.

The *semantic content* of an image represents every material concept present in an image. It can be either in the foreground (a person, a cat, a tree, ...) or in the background (a mountain, a forest, some buildings, ...). This part of the semantic, however, does not consider the different relationships between the different concepts; their relative places or even their colors. In this work, we use a CLIP-based model to extract the semantic content $\sigma_{clip}$ of an image.

The *semantic organization* of an image represents every other, non-material, concepts present in an image. This can range from the position of the objects, their colors (or the main dominant color in a specific area), or even the ambiance of the picture. In this work, we represent the semantic organization $\sigma_{color}$ of an image with a down-sized color map of this image, as it encapsulates the position of the objects. This method is inspired by [7].

All in all, the encoded semantic of our image is a pair:

$$\mathcal{E}(\mathbf{x}) = (\sigma_{clip}, \sigma_{color}) \tag{2}$$

We illustrate, in Fig. 3, our implementation of the encoder of the semantic and the generative decoder of the proposed CoCliCo framework.

### B. Extracting and encoding the semantic content

We propose to encode the semantic content of an image with a CLIP latent vector $\sigma_{clip}$. Since foundation models, like CLIP [8], are used as for solving multiple different tasks, the latent spaces of such models are expected to project the data into high-level but low-dimensional representation spaces. Specifically, this foundation model was trained to align images and their captions in the same projection space, so our hypothesis is that such a latent space can encapsulate the semantic of the data compactly.

We implement the following quantization process $Q$: we clamp the vector in $[-1, 1]$ before uniformly quantizing among each of its dimension into $\hat{\sigma}_{clip}$, as illustrated in Fig. 3. Setting a fixed CLIP dimension to 768 and a quantization step $q$, typically a negative power of 2 representing the number of bits, we have the following compression scheme and bitrate:

$$\hat{\sigma}_{clip} = \lfloor \frac{\sigma_{clip}}{q} \rfloor q + \frac{q}{2} \tag{3}$$
$$R(\hat{\sigma}_{clip}) = -\log_2(q) * 768 \tag{4}$$

### C. Extracting and encoding the semantic organization

To extract the semantic organization of an image, we resize this image to a $n \times n$ color map using bilinear interpolation. This representation of the input encapsulates a global idea of the positions, color, and ambiance of the objects present in the inputs. As shown in Fig. 3, the color map $\sigma_{color}$ is computed via the positional encoder $\mathbf{E}_{color}$.

We also implement a quantization process $Q$ to reduce the bitrate dedicated to the color map. On the one hand, we can vary the resolution of the color map $n$, but we also choose to code the color pallet on different numbers of bits $b_{color}$. Our color maps are saved in the YUV $4:2:0$ format rather than an RGB format to further lower the rate. All in all, the bitrate for $\hat{\sigma}_{color}$ is computed as:

$$R(\hat{\sigma}_{color}) = 1.5 * b_{color} * n^2$$

### D. Generating the decoded image

The decoder is based on a generative approach. In this work, we opt for a conditional latent diffusion model (LDM) that was trained on conditional CLIP latent vectors. Diffusion models iteratively remove the noise of a random Gaussian distribution, converging towards realistic images. In terms of probability density, this optimizes perception of generated outputs. Latent diffusion models do the same process but in the latent space of a VAE instead of in the pixel space.

In other words, for $\epsilon_\theta$ the conditional diffusion model trained on $T$ time steps, we iterate $z_{t-1} = \epsilon_\theta(z_t, t, \sigma_{clip})$

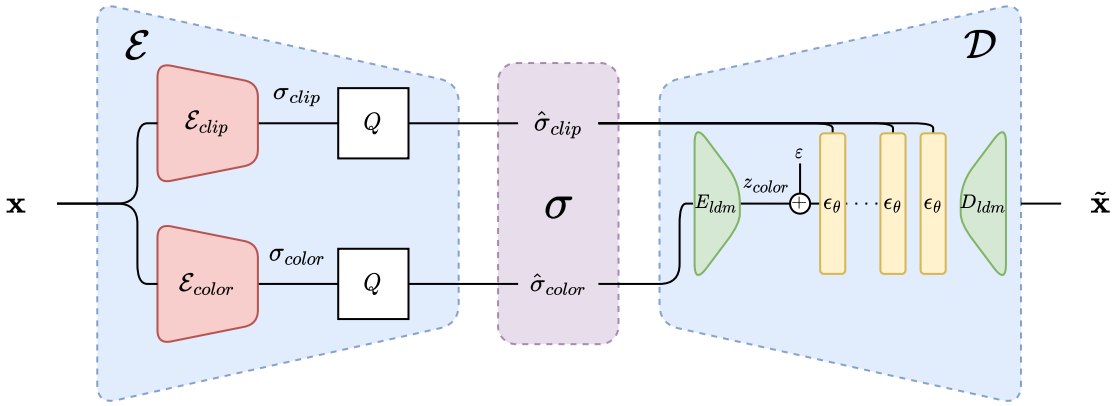


Fig. 3: CoCliCo encoding and decoding schemes. The encoder $\mathcal{E}$ is separated between extracting and quantizing the CLIP of the image and the color map. The decoder $\mathcal{D}$ relies on a latent diffusion model, slightly modified to integrate the color map.

starting from random noise $z_T$. The semantic description $\sigma_{clip}$ conditions the generation. The VAE decoder $D_{ldm}$ is used at the end of the process to decode the generated latent vector $z_0$.

Our generative decoder $\mathcal{D}$ is illustrated in the right-hand part of Fig. 3. It slightly differs from the standard LDM to integrate the information from the color map. In the vein of [7], the quantized color map $\hat{\sigma}_{color}$ is first upscaled to the dimension of the original image, then given to the LDM encoder. We then obtain the latent vector of the color map $z_{color}$. We start the diffusion at a later time step, skipping the $t_{start}$ first denoising steps, and use the latent color with the corresponding level of noise as $z_{t_{start}}$. The choice of the $t_{start}$ changes the amount of noise that is added to the latent, and was discussed in [11]. The later we start, the less amount of noise is added, but the less denoising steps is left. We apply the conditional diffusion model on the noised latent, using the quantified CLIP $\hat{\sigma}_{clip}$ as side information at each step. The generated $z_0$ is then fed to the VAE to obtain $\tilde{\mathbf{x}}$. Integrating the color map this way has the advantage of not requiring any retraining or fine-tuning of the diffusion model.

## IV. Experimental Results

### A. Dataset, architectures, and models

As mentioned in the previous section, the CoCliCo encoder is separated into two parts. For the CLIP encoder, we use the image encoder of the version ViT-L/14 of the CLIP model. In this version, images are encoded in a $768-$dimensional vector coded on 16 bits. For the color maps downsizing, we use the PyTorch bilinear interpolation method. For the LDM decoder, we used the Stable Diffusion model [10] model fine-tuned for CLIP latent vectors conditioning, also called Stable unCLIP, the weights we used can be found in [12].

The images, that are used for comparison and metric evaluation, come from the Landscape dataset [13]. Images are generated using 20 time steps using the DPM-solver scheduler from [14], with a guidance scale of 12. All the images presented are cropped in the center to obtain an $768 \times 768$ image, smaller images are discarded. We compare ourselves with the intra coder of VVC(v1.6) [15], SGC [7] and PICS [6].

### B. Compression parameters

For the CLIP vector, prior experiments strongly suggested that clip quantization did not impact the semantic faithfulness. The impact of quantization is measured using cosine similarity between clip vectors of the generated image and the input image. We noted that the images generated with quantized clip vectors, even up to 1 bit per dimension, give similar results in terms of CLIP alignment. For our encoder, we thus use 1 bit per dimension for the CLIP vector, *i.e.*, 768 bits to code $\sigma_{clip}$.

For the color map, we measured the effect of quantization over two parameters: the resolution of the color map, and the number of bits to encode each channel. We observe the pallet being more expressive as the number of bits increases, as illustrated in Fig. 4. The effects of the color map resolution are shown in Fig. 5: an image generated from a high-resolution color map is more faithful, in terms of semantic organization, to the input.

To set the best parameters, we measure the MSE between the color map of the input with resolution $32 \times 32$ with 16 bits per channel, to the color map of the output with the respective parameters upscaled in $32 \times 32$ to match the resolution as an average over 100 images. We retain only the parameters forming the convex hull of the curve in Fig. 6. Notably, we choose the $8 \times 8$ with 2 bits per channel for extremely low rates. Empirically, we find that for small color maps, choosing $t_{start} = 0.88T$ works best. However, for higher resolution color maps, this value should be reduced to add less noise at initialization.

### C. Evaluation

The images are coded using the parameters presented previously, $8 \times 8$ resolution and 2 bits per channel for the color map and 1 bit per dimension for the CLIP latent. We compare our images with different methods at the same bitrates when possible, and otherwise we try to reduce it as much as possible.

Fig. 4: (First image) Input image. (Second to fifth image) Decoded images with increasing number of bits for the color palette: 1, 2, 3 and 4 bits. Clip quantization is set to 1 bit and color map size to a $8 \times 8$ resolution.

Fig. 5: (First image) Input image. (Second to fifth image) Decoded images with different color map sizes: $4 \times 4$, $8 \times 8$, $16 \times 16$ and $32 \times 32$ pixels. Color palette quantization set to 2 bits and clip quantization to 1 bits.

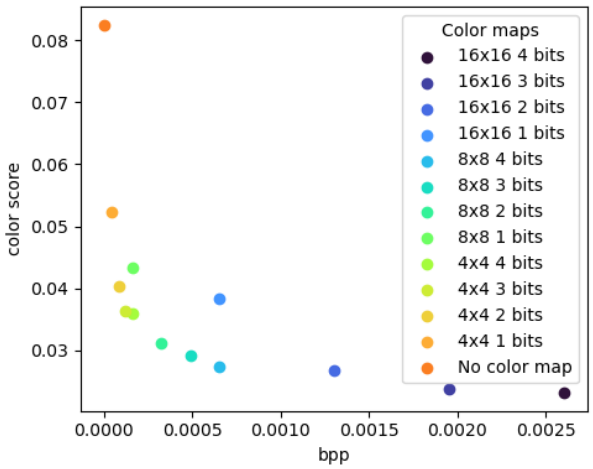


Fig. 6: Comparison between different parameters of color maps. Reducing the size is sometimes more advantageous than harsher quantization at the same bitrates.

In Table I, we compare CoCliCo with other generative compression methods based on semantics. We evaluate realism $\Psi$ using image quality assessment metric (IQA). We evaluate the semantic faithfulness $d(\Phi)$ with CLIP alignment, *i.e.*, with the cosine similarity between the clip latent vectors of the input and the output. We can see that VVC at similar bitrates yields images with low realism due to the high number of artifacts. Moreover, a lot of the semantic information is not only degraded but also lost. The semantic representation proposed in [7] lacks in precision. Indeed, the segmentation maps are limited by the number of labels, the information on colors is not always enough to compensate, as it can be seen in the second row of images. In Fig. 7, visual results are displayed for the different methods. PICS images, even though more realistic according to the metrics, are less faithful to the semantic. While images produced by our method yield a similar level of realism than the other methods, our choice of semantic representation using CLIP brings more fidelity to the input.

TABLE I: Evaluation of SGC

| | **IQA metrics** $\uparrow \Psi$ | | $d_\Phi \downarrow$ | | $R \downarrow$ |
|---|---|---|---|---|---|
| | **DBCNN** [16] | **MUSIQ** [17] | **Color score** | **CLIP** [8] | **bpp** |
| Input | 62.0 | 69.0 | - | - | - |
| VVC | 15.0 | 17.8 | 0.01 | 0.39 | 0.0072 |
| SGC [7] | 49.8 | 60.6 | 0.10 | 0.20 | 0.0209 |
| PICS [6] | 72.0 | 73.9 | 0.16 | 0.14 | 0.0244 |
| Ours | 46.3 | 61.0 | 0.03 | 0.18 | 0.0016 |

## V. Conclusion

In this paper, we introduce CoCliCo, a semantic-based generative codec that encodes images at extremely low bitrate and decodes them with a closely related semantic from their originals while keeping a high level of quality for the images. We define the semantic using the CLIP foundation model that encapsulates a high-level description of the image and complements it with a tiny color map. We use this representation in a generative compression framework. An interesting continuation would be to integrate user in the coding loop, coding using semantic only a part of the image.


Fig. 7: Visual comparison of several methods at minimal rate.

## VI. Acknowledgements

This work was funded by the French National Research Agency (*MADARE*, Project-ANR-21-CE48-0002 and *Contrats doctoraux en intelligence artificielle*, ANR-20-THIA-0018).

## References

[1] Y. Blau and T. Michaeli, "The perception-distortion tradeoff," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6228–6237.

[2] ——, "Rethinking lossy compression: The rate-distortion-perception tradeoff," in *International Conference on Machine Learning*. PMLR, 2019, pp. 675–685.

[3] E. Agustsson, D. Minnen, G. Toderici, and F. Mentzer, "Multi-realism image compression with a conditional generator," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 22 324–22 333.

[4] A. Bhown, S. Mukherjee, S. Yang, S. Chandak, I. Fischer-Hwang, K. Tatwawadi, J. Fan, and T. Weissman, "Towards improved lossy image compression: Human image reconstruction with public-domain images," *arXiv preprint arXiv:1810.11137*, 2018.

[5] T. Weissman, "Toward textual transform coding," *arXiv preprint arXiv:2305.01857*, 2023.

[6] E. Lei, Y. B. Uslu, H. Hassani, and S. S. Bidokhti, "Text+ sketch: Image compression at ultra low rates," *arXiv preprint arXiv:2307.01944*, 2023.

[7] T. Bordin and T. Maugey, "Semantic based generative compression of images at extremely low bitrates," *IEEE Multimedia Signal Processing 2023*, 2023.

[8] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark *et al.*, "Learning transferable visual models from natural language supervision," in *International conference on machine learning*. PMLR, 2021, pp. 8748–8763.

[9] T. Bachard and T. Maugey, "Is clip latent space resistant to quantization for generative compression?" *Preprint*, 2023, available at https://project.inria.fr/dare/publications/quant-clip/.

[10] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10 684–10 695.

[11] C. Meng, Y. He, Y. Song, J. Song, J. Wu, J.-Y. Zhu, and S. Ermon, "Sdedit: Guided image synthesis and editing with stochastic differential equations," *arXiv preprint arXiv:2108.01073*, 2021.

[12] P. von Platen, S. Patil, A. Lozhkov, P. Cuenca, N. Lambert, K. Rasul, M. Davaadorj, and T. Wolf, "Diffusers: State-of-the-art diffusion models," https://github.com/huggingface/diffusers, 2022.

[13] P. Esser, R. Rombach, and B. Ommer, "Taming transformers for high-resolution image synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 12 873–12 883.

[14] C. Lu, Y. Zhou, F. Bao, J. Chen, C. Li, and J. Zhu, "Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models," *arXiv preprint arXiv:2211.01095*, 2022.

[15] A. Wieckowski, J. Brandenburg, T. Hinz, C. Bartnik, V. George, G. Hege, C. Helmrich, A. Henkel, C. Lehmann, C. Stoffers, I. Zupancic, B. Bross, and D. Marpe, "Vvenc: An open and optimized vvc encoder implementation," in *Proc. IEEE International Conference on Multimedia Expo Workshops (ICMEW)*, 2021, pp. 1–2.

[16] W. Zhang, K. Ma, J. Yan, D. Deng, and Z. Wang, "Blind image quality assessment using a deep bilinear convolutional neural network," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 30, no. 1, pp. 36–47, 2018.

[17] J. Ke, Q. Wang, Y. Wang, P. Milanfar, and F. Yang, "Musiq: Multi-scale image quality transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 5148–5157.