Towards learning-based image compression for storage on DNA support

Sophie Strebel, Noémie Monnier, Davi Lazzarotto, Michela Testolina, and Touradj Ebrahimi

Multimedia Signal Processing Group (MMSPG)
Ecole Polytechnique Fédérale de Lausanne (EPFL)
CH-1015 Lausanne, Switzerland
sophie.strebel@epfl.ch, noemie.monnier@epfl.ch, davi.nachtigalllazzarotto@epfl.ch,
michela.testolina@epfl.ch, touradj.ebrahimi@epfl.ch

ABSTRACT

The demand for data storage has grown exponentially over the past decades. Current archival solutions have significant shortcomings, such as high resource requirements and a lack of sufficient longevity. In contrast, research on DNA-based storage has been advancing notably due to its low environmental impact, larger capacity, and longer lifespan. This has led to the development of compression methods that adapted the binary representation of legacy JPEG images into a quaternary base of nucleotides while taking into account the biochemical constraints of current synthesis and sequencing mechanisms. In this work, we show that DNA can also be leveraged to efficiently store images compressed with neural networks even without a need for retraining, by combining a convolutional autoencoder with a Goldman encoder. The proposed method is compared to the state of the art, resulting in higher compression efficiency on two different datasets when evaluated by a number of objective quality metrics.

Keywords: DNA-based archival, lossy image compression, end-to-end compression, deep learning

1. INTRODUCTION

The constant demand for increased storage capabilities poses challenges to conventional storage technologies for archival, which are approaching their physical limitation. DNA-based storage currently represents a promising alternative to such methods, offering the use of DNA molecules as a medium for encoding and storing different types of digital information such as images. For this purpose, data is converted to quaternary codes corresponding to the four nucleotide bases found in DNA, namely adenine (A), cytosine (C), guanine (G), and thymine (T). DNA storage offers a large number of advantages when compared to conventional storage:

1. **High information density**: DNA molecules can store a large amount of data in a limited amount of space. For example, recent studies have demonstrated the potential of storing data with a density up to 1 exabyte (or 10^{18} bytes) per mm^3, approximately six orders of magnitude denser than the modern physical storage devices.[1,2]

2. **Long-term stability**: DNA has been demonstrated to be more stable than conventional storage devices over time, making it a promising candidate for long-term archival. DNA can, in fact, be preserved for millennia, or even longer if stored in ideal conditions.[3]

3. **Energy efficiency**: DNA-based storage allows for reducing the energy cost, both at rest and per access. For example, the energy consumption at rest can be approximately eight orders of magnitude less than conventional storage devices.[4] This makes DNA-based storage a more sustainable alternative to conventional storage technologies.[2]

4. **Redundancy and error correction**: DNA-based data storage methods can incorporate redundancy for error correction. In fact, DNA molecules can be replicated, for example using PCR, with a little added cost, allowing for a simple and effective solution for correctly retrieving data.[1] Moreover, distributing data across multiple DNA molecules lowers the risk of data loss due to damage or decay, ensuring better data integrity over time.

Despite the numerous advantages offered by DNA-based storage, there are still a number of challenges that need to be taken into account, namely high costs of synthesis and sequencing, and slow read and write. In this context, data compression is fundamental to reduce the storage requirements and therefore limiting the cost and accelerate the synthesis/sequencing process. Moreover, in the context of image storage, technologies should consider that a large percentage of stored images are already compressed with well-known compression standards, such as JPEG or JPEG 2000. Therefore, an effective DNA-based storage solution should allow the transcoding of such image formats to DNA.

As DNA is highly susceptible to errors occurring both during sequencing and synthesis of the DNA strand, as well as during storage, a number of constraints need to be considered while designing a robust and effective DNA-based coding:[5,6]

- Strand length limitation: as longer DNA strands are more difficult to produce, they are typically divided into smaller strands, also known as *oligos*. Acceptable oligo lengths range from 100 to 300 nucleotides.
- Homopolymer run: homopolymers, or repetitions of the same nucleotide consecutively in a strand, might impact the stability of the DNA molecules. For this reason, homopolymers of length 3 or more should be avoided, and in particular, homopolymers of length 7 and above strictly forbidden.
- GC content balance: to limit errors during the sequencing operations, the percentage of G and C nucleotides in each strand should be between 40% and 60%.
- Pattern repetition: or repetitions of the same sequence of nucleotides more than 3 times (pattern), increases the probability of errors during encoding and the stability of the produced DNA strands, and should therefore be avoided.

Regardless, researchers and professionals in this field are constantly working towards making DNA storage technology feasible not only by designing new synthesis/sequencing techniques, but also presenting effective ways of encoding data in DNA. Therefore the above constraints are still evolving and could change as a result of features offered by newer technologies.

The JPEG Committee is currently working on an activity, known as JPEG DNA, with the goal of standardizing efficient image coding solutions for storage on DNA support. The standardized coding is expected to be able to respect the biochemical constraints and offer robustness to errors. During the 99^{th} JPEG Meeting, in April 2023, a Final Call for Proposals on Digital Media Storage on DNA Support[7] was issued, covering both coding and transcoding solutions. The proposals are expected to be received in early October 2023, and the collaborative process will be initiated during the 101^{st} JPEG meeting, in October 2023.

In this paper, a novel deep learning-based method for compressing images into DNA code is presented. The proposed method is able to outperform previous works and is compliant with a number of biochemical constraints. Following the Final Call for Proposals on Digital Media Storage on DNA Support,[7] the performance of the submitted solutions is compared to two anchors and on two different datasets, both through rate-distortion plots and by presenting a number of visual examples.

2. RELATED WORK

The first attempts to store digital information using DNA recorded in the literature were conducted by Church et al.,[8] which translated each zero bit of a file either to A or C and each one bit either to T or G. Since the selected mechanism for translation from binary to nucleotides did not take into account the inherent biochemical constraints, the data could not be fully recovered. This work illustrated the need for the use of robust encoding mechanisms as well as the challenges related to DNA-based storage. In a later work, Goldman et al.[9] proposed an algorithm for constrained DNA coding, where ternary symbols were translated to nucleotides with a rotating dictionary. The dictionary assigned one nucleotide to each symbol, ensuring the absence of homopolymers by not allowing the repetition of the last produced nucleotide. The produced sequence was finally partitioned into oligos, each representing a section of the source sequence. Three-quarters of the section represented by each oligo overlapped with the section represented by the previous one, ensuring that all fragments of the source sequence were represented by four different oligos. Yet, after synthesizing these oligos into DNA and sequencing them back to digital information, there were fragments of the original data that could not be fully retrieved.

In order to improve the information retrieval process despite such errors inserted in the DNA channel, Grass et al.[3] proposed to represent the data as symbols from a Gallois Field (GF(47)) and use Reed-Solomon codes for error correction. Each GF(47) symbol was translated into three nucleotides using a mechanism that avoided homopolymers by ensuring that the last two nucleotides in the triplet were different. This experiment was able to completely retrieve a total of 83 kB of data. Later, Blawat et al.[10] proposed a method where each byte was mapped into five nucleotides. The first three pairs of bits were translated into one nucleotide each, and the last two bits were translated into two nucleotides using a dictionary allowing for multiple translation options. Erlich et al.[11] presented a method leveraging Fountain codes to encode data into DNA. The source binary data was partitioned into packets which were randomly combined into data chunks using the XOR operation following the Luby transform.[12] The binary chunks were then translated into DNA oligos using a simple scheme where each two bits were mapped into one nucleotide. The obtained oligo was then only added to the oligo pool if it met the imposed set of biochemical constraints, and discarded otherwise. Since a virtually infinite number of oligos can be generated from the source data, this method also has the advantage of producing redundant oligos if desired, allowing the retrieval of the source data even if full oligos are lost prior to sequencing. Lately, Schwarz et al.[13] have implemented a library with not only a similar approach as Erlich et al.,[11] but also more recent fountain codes such as Online codes[14] and Raptor codes.[15]

In addition to the above efforts, which focused on the encoding of any type of data into DNA, other works studied the implementation of methods directly tailored to specific data types. Dimopoulou et al.[16] proposed an image compression algorithm based on the Discrete Wavelet Transform (DWT) that used dictionaries to associate quantized coefficients to nucleotides while avoiding biochemical constraints. The encoded oligos were used to synthesize DNA strands, which were amplified both with PCR and, during sequencing, with BA (bridge amplification), allowing for a full recovery of the information using only a consensus mechanism without any error correction scheme. The same authors later developed an algorithm[17] based on the JPEG coding that translated quantized DCT coefficients into nucleotides using Goldman encoding. The JPEG Committee selected this algorithm as the JPEG DNA Benchmark Codec (BC). Inspired by this solution, Secilmis et al. developed an algorithm[18] able to retrieve the coefficients from already compressed JPEG files and encoded them with a similar method. Ramos et al.[19] explored the impact of the errors added during the DNA synthesis, storage, and sequencing on the decoding of such files, proposing a method to protect these nucleotide sequences. With a similar approach to the JPEG DNA BC, Pic et al.[20] replaced the MQ-coder of the JPEG 2000 coding with an arithmetic encoder designed to directly produce nucleotides respecting the homopolymer constraint.

The majority of the works are inspired by conventional image compression methods relying on handcrafted transforms to generate coefficients that are later quantized and entropy coded. However, learning-based approaches have been receiving increased attention as the applied transforms are obtained from data in order to maximize rate-distortion performance. Early works proposed[21] an autoencoder architecture based on convolutional neural networks with downsampling operations in the encoder and upsampling in the decoder. The latent features generated in the bottleneck were then compressed using a range encoder assuming learned independent probability distributions for each variable. Although this approach already outperformed, on average, conventional image coding such as JPEG and JPEG 2000, it was not able to optimally explore spatial redundancies within the image. The previous method was therefore later extended with a hyperprior[22] that estimated the probability distribution of the latent features and used predicted variance values to reduce the bitrate of compressed representation. Autoregressive methods[23] taking previously decoded symbols into consideration for probability estimation using masked convolutions were proposed as an alternative way of increasing performance. Solutions based on Generative Adversarial Networks[24] have also been investigated as a technique to increase the perceptual quality of images at low bitrates. Inspired by learning-based solutions for image compression, Pic et al. proposed a method[25] using two distinct DNA entropy coders[9, 26] to translate quantized latent features into nucleotides by extracting the statistics from the training data, without however being able to outperform previous JPEG-based methods.[17, 18]

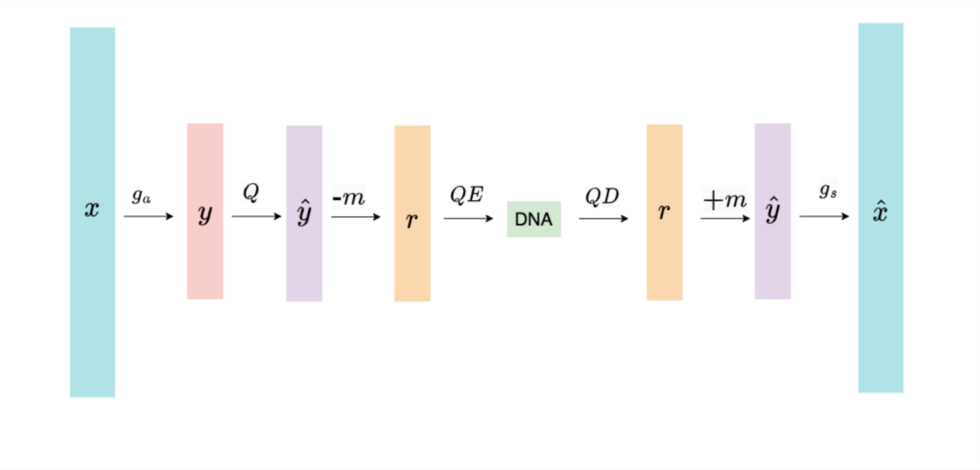


Figure 1: Workflow of the codec proposed in this paper

3. LEARNING-BASED IMAGE COMPRESSION FOR DNA-BASED STORAGE

The learning-based algorithm for image compression presented by Ballé et al.,[21] using factorized probability distributions for the entropy coding of latent features, was used in this work. The input image is represented as a tensor x containing one channel for each color component, which serves as an input to the analysis transform g_a. This transform is composed of stacked downsampling convolutions interleaved with generalized divisive normalization layers acting as non-linear operations. The output tensor y is then quantized into $Q(y) = \hat{y}$ and entropy coded following a probability distribution learned during training. The quantization operation does not round the values of y directly, but instead first measures the offset between the latent features and their median values m. The rounding operation is then applied and the integer values contained in $r = \lfloor y - m \rceil$ are fed to the entropy coder, with the value of m being derived from the probability function estimated during training. During decoding, the median values for each channel are added again as represented by Equation 1. Note that since y is composed of a total of N channels, this probability function is estimated separately for each channel, and m is represented as a tensor of size N.

$$Q(y) = \hat{y} = \lfloor y - m \rceil + m \tag{1}$$

On the decoder side, $\hat{y}$ goes through a synthesis transform g_s that mirrors the analysis g_a. The decoded output $\hat{x}$ can be related to the input through Equation 2.

$$\hat{x} = g_s(Q(g_a(x))) \tag{2}$$

The entire network was trained end-to-end to minimize both the distortion of the decoded image and the rate of the compressed representation. The distortion is obtained through the mean squared error (MSE) between the input x and output $\hat{x}$, and while the rate cannot be directly computed, it was estimated using the binary entropy of the tensor $\hat{y}$. The relative importance given to each of these values is controlled by a multiplier term λ, which is set as a hyperparameter. The final loss term is represented in 3.

$$L = R + \lambda D = -\sum \log_2 p(\hat{y}) + \lambda ||\hat{x} - x|| \tag{3}$$

Since the rounding operation within Q cannot be differentiated, it is replaced by additive uniform noise as a proxy function during backpropagation. In this paper, the implementation provided by CompressAI[27] was used. Contrary to previous works,[25] the pre-trained models used to compress images into binary representation were also used in this paper, since the binary entropy was considered to be a suitable estimator for the rate, independently from binary or quaternary representations used for entropy coding. The employed implementation contains eight different pre-trained models each using a different value of λ. The three models trained using the highest values contained $N = 320$ latent channels, while for the remaining ones, the value of $N = 192$ was set.

The main difference between the method proposed in this paper and the compression model described above is that the latter stores the information of the image as binary data, while here, DNA is used as a storage medium. In the original work, the latent features from y are quantized and the values of $r = \lfloor y - m \rceil$ are entropy coded into binary. This step is modified in the proposed work to generate quaternary nucleotides respecting biochemical constraints. In particular, the same ternary Huffman coding module coupled with Goldman used by the JPEG DNA BC[17] is here used. The statistics from the r tensor itself were used to build the frequency tables of the Huffman encoder. A workflow of the proposed codec is depicted in Figure 1.

Interestingly, during the training process of the model, the analysis transform converged to a point where only a subset of the N channels of the latent features contained actual information about the input image. The remaining channels contained only zero values no matter which image was encoded and therefore did not provide useful information for the decoder to reconstruct the image. This behavior is encouraged during training since setting all values of a channel to a constant effectively neutralizes its entropy, reducing overall the first term of the loss function. The channels with zero entropy are here referred to as trivial, and the remaining ones as non-trivial. This is illustrated in the plots in Figure 2, which shows a histogram with logarithmic scale of the values contained in each of the non-trivial channels of r after encoding the first image of the Kodak dataset with quality levels 1, 3, 5, and 8. The histogram for each channel is represented with a different color. It is clear that the amount of non-trivial channels increases with the quality level even when the total number of channels is the same, as for the first three plots. Likewise, the range of the values r increases as well.

Since the original model uses an arithmetic encoder, the presence of these channels does not increase the size of the bitstream, since all the zero symbols have a probability equal to 1. However, the ternary Huffman encoder can be regarded as a dictionary where each input value is translated into a sequence of $\{0, 1, 2\}$ symbols, with shorter sequences being assigned to values with higher probability. Repeated input values, therefore, increase the bitrate if they are included in this dictionary. For this reason, and since the trivial channels are always the same for any given input image, their values are excluded from the encoding process in this paper, and automatically set to 0 at the decoder side.

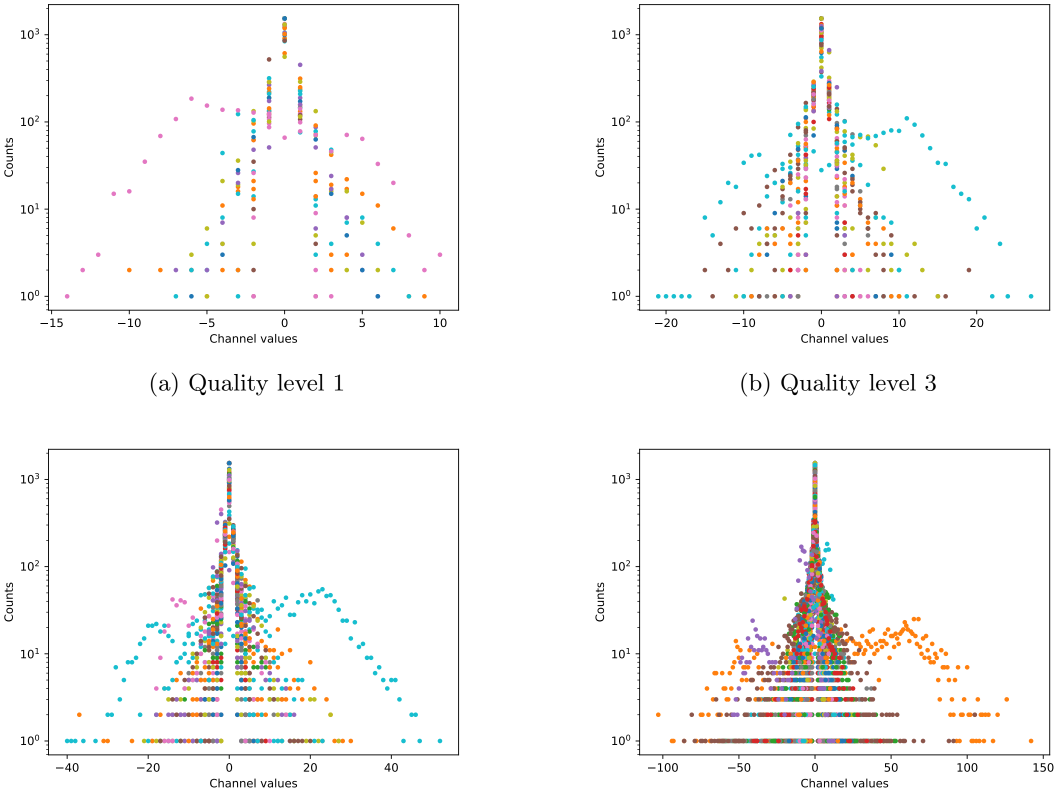
4. EXPERIMENTAL RESULTS AND DISCUSSION

The performance of the proposed learning-based image compression for DNA storage is assessed on Kodak and JPEG DNA datasets:

- **Kodak dataset**:[28] consists of 24 images with small size (768 × 512) in both portrait and landscape orientation, including a variety of contents. In this dataset, only the original uncompressed images are provided.
- **JPEG DNA dataset**: derived from the JPEG AIC-3 dataset,[29] includes 10 high-quality reference images with different sizes and resolutions (from 560 × 888 up to 2592 × 1946). In addition to the reference images, the dataset includes their encoded and decoded versions with three different compression methods, namely JPEG, JPEG 2000, and JPEG XL. Ten different compressed images per codec are provided, having visual quality levels, in Just Noticeable Difference (JND) units, between 0 to -2.5 JND. Information on the procedure adopted for the selection of images in the dataset is provided in the paper on the JPEG AIC-3 dataset.[29]

The proposed learning-based image compression for storage on DNA support is compared to two anchor methods, namely the JPEG DNA Benchmark Codec (JPEG DNA BC)[17] and the JPEG DNA Benchmark Transcoder (JPEG DNA BT).[18]

For the JPEG DNA BC, the rate control is managed by the *alpha* value. By varying this parameter, each image in each dataset was compressed with 8 different quality levels. A similar procedure was adopted for the images in the Kodak dataset for the JPEG DNA BT, which needed to be encoded to JPEG prior to transcoding. Notably, the original images were compressed with JPEG at 8 different quality levels and subsequently transcoded



(a) Quality level 1

(b) Quality level 3

(c) Quality level 5

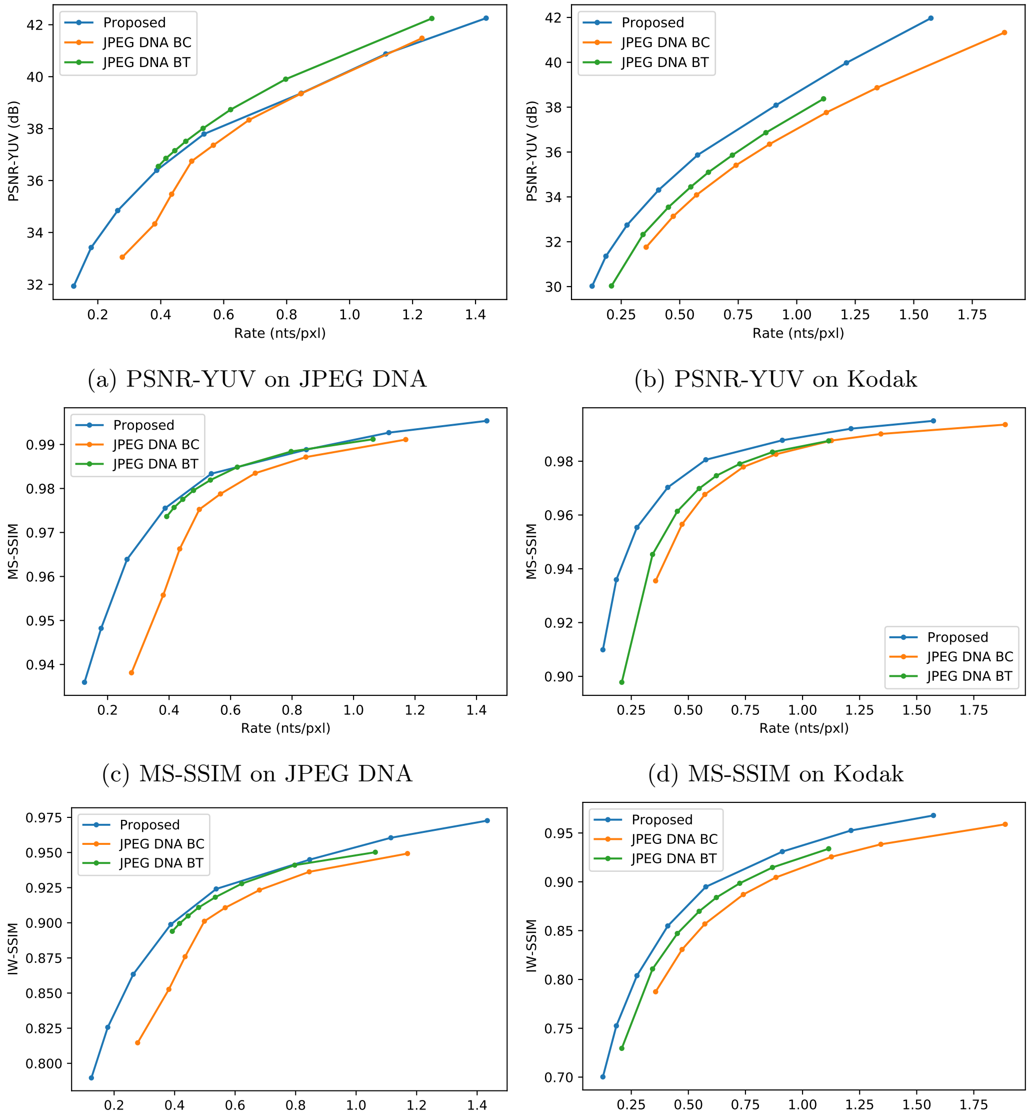
(d) Quality level 8

Figure 2: Values in each channel of r for kodim01 at various quality levels

using the JPEG DNA BT. For the JPEG DNA dataset, this process was not necessary as the encoded images with JPEG are provided as part of the dataset, and therefore they were directly transcoded using the JPEG DNA BT.

Figures 3 and 4 present the rate-distortion performance of the proposed learning-based image compression for storage on DNA support in comparison to the two anchors. Notably, the rate is reported as nucleotides per pixel, and the distortion is computed using several objective image quality metrics: PSNR YUV, MS-SSIM,[30] IW-SSIM,[31] FSIM,[32] NLPD,[33] VIF,[34] and VMAF.[35] For the majority of the analyzed objective metrics, except for NLPD for which the opposite applies, a high metric score expresses better image quality.

A trend can be observed for the PSNR YUV, MS-SSIM, IW-SSIM, VIF, and NLPD objective quality metrics. While the proposed learning-based image compression for storage on DNA support always performs better than the anchors on the Kodak dataset, the performance gap between the proposed method and JPEG DNA BT narrows on the JPEG DNA dataset, which even outperforms the proposed method in a few cases. A closer analysis shows that the performance of the proposed learning-based image compression for storage on DNA support is comparable on both datasets, presenting similar metric values at similar rates. On the other hand, the performance of the JPEG DNA BT increases on the JPEG DNA dataset and, in some cases, approaches the proposed codec. A similar trend can also be observed on the JPEG DNA BC, which also presents higher performance on the JPEG DNA dataset, showing objective values approaching those of the proposed codec when evaluated on this dataset. This could be explained by the fact that both the JPEG DNA BC and JPEG DNA BT are based on JPEG compression, which inherently removes the high-frequency AC coefficients. As the images in the JPEG DNA dataset are larger than those in the Kodak dataset, they most likely contain fewer



(a) PSNR-YUV on JPEG DNA

(b) PSNR-YUV on Kodak

(c) MS-SSIM on JPEG DNA

(d) MS-SSIM on Kodak

(e) IW-SSIM on JPEG DNA

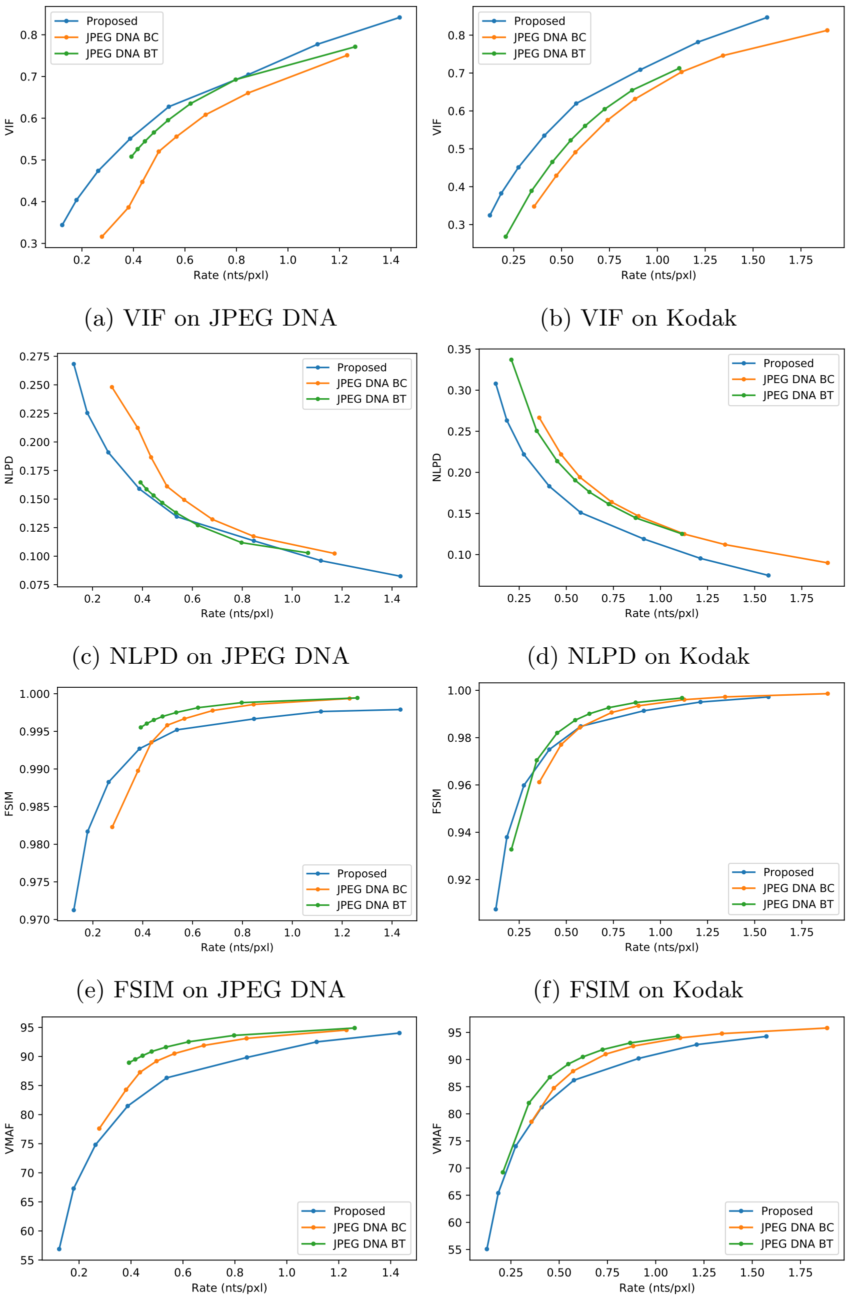
(f) IW-SSIM on Kodak

Figure 3: Average PSNR YUV, MS-SSIM, and IW-SSIM across all the images of both JPEG DNA and Kodak datasets, for different models and at different rates

high-frequency regions in each 8x8 DCT block, leading to a better reconstruction quality.

This behavior does not emerge when VMAF and FSIM are considered. For both metrics, in fact, the proposed codec is the model with the lowest performance on the JPEG DNA dataset and the lowest performance on the Kodak dataset at the highest rates. Nevertheless, VMAF was designed to assess the performance of video compression, and therefore its usage might not be adequate in the use case of interest to this paper. In this context, a visual inspection of the reconstructed images is necessary to validate the discussion above.

Figures 5 and 6 show crops of images from the JPEG DNA and Kodak datasets respectively. In Figure 5, it is possible to observe that while the images compressed with the proposed codec (central column) present smoothing artifacts and loss of texture, the JPEG DNA BC (left column) presents typical JPEG artifacts namely blocking and posterization. Similarly, Figure 6 shows stair-casing artifacts in the sky and water areas



(a) VIF on JPEG DNA

(b) VIF on Kodak

(c) NLPD on JPEG DNA

(d) NLPD on Kodak

(e) FSIM on JPEG DNA

(f) FSIM on Kodak

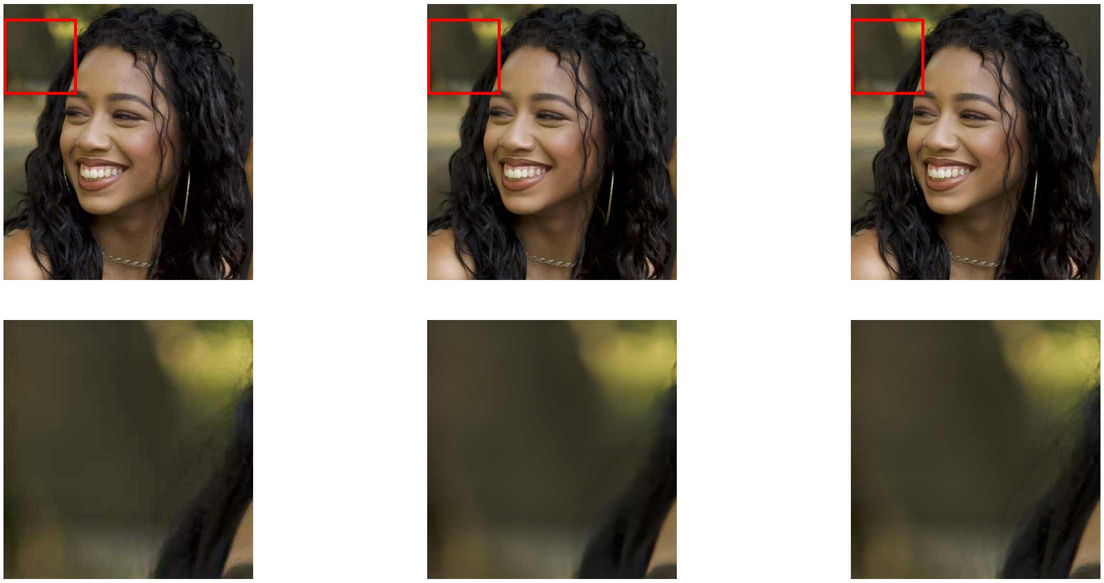
(g) VMAF on JPEG DNA

(h) VMAF on Kodak

Figure 4: Average VIF, NLPD, FSIM, and VMAF across all the images of both JPEG DNA and Kodak datasets, for different models and at different rates

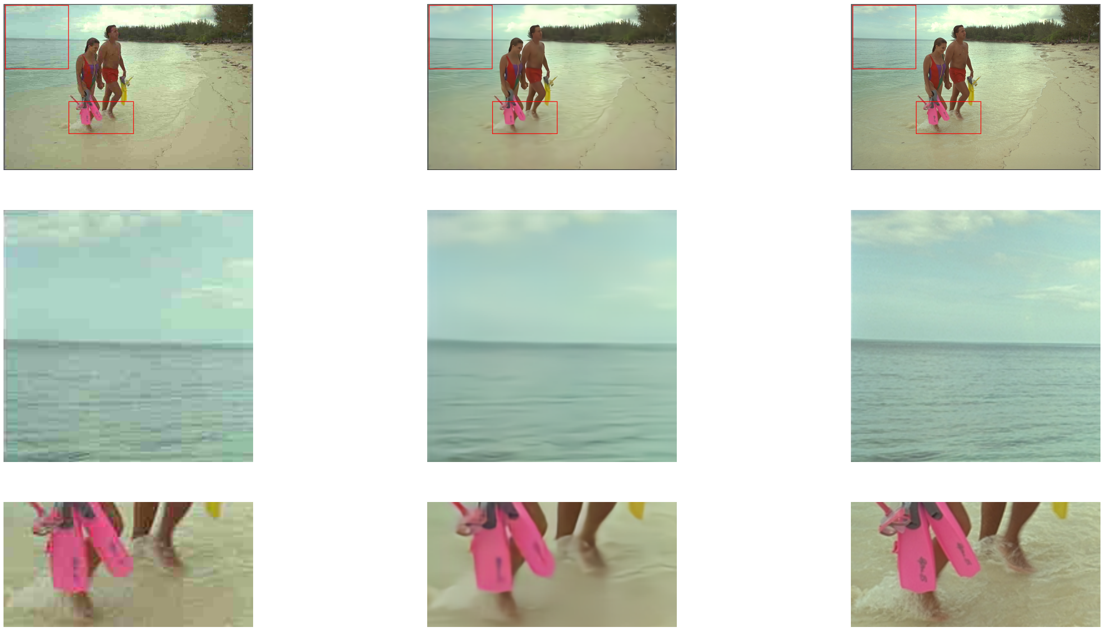
of the image compressed with the JPEG DNA BC, whereas the proposed codec produces a smoother texture. In general, the learning-based image compression for storage on DNA support presents a higher accuracy in reconstructing the colors when compared to the JPEG DNA BC, making the proposed method more visually appealing (right column).

More visual examples, both for the Kodak dataset as well as on the JPEG DNA dataset, are available in



(a) JPEG DNA BC (b) Proposed (c) Original

Figure 5: Image *00002* of the JPEG DNA dataset, at a rate of ≈ 0.37 nucleotides/pixel



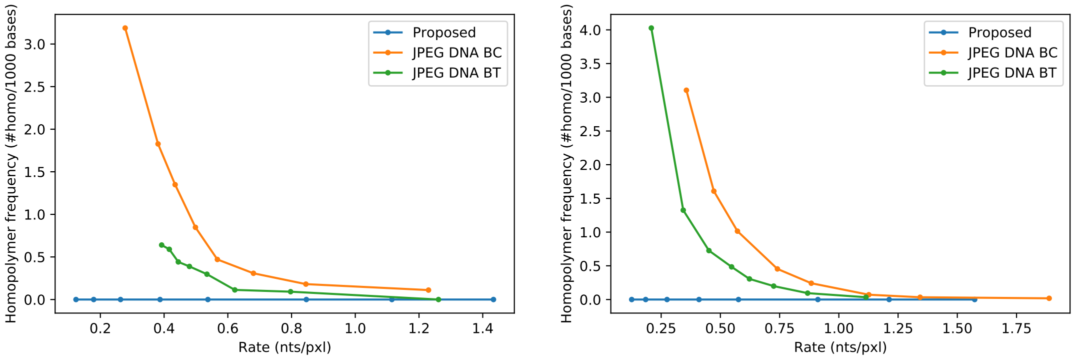
(a) JPEG DNA BC (b) Proposed (c) Original

Figure 6: Image *kodim12* of the Kodak dataset, at a rate of ≈ 0.26 nucleotides/pixel

Annexes A, B and C.

4.1 Biochemical constraints analysis

As reported in Section 1, a number of biochemical constraints should be taken into account during the design of an efficient DNA-based compression method. In this paper, only an analysis of the homopolymers is conducted. Moreover, to simplify the analysis, this paper considers only homopolymers of length 7, corresponding to the



(a) JPEG DNA

(b) Kodak

Figure 7: Average homopolymer frequency on the different datasets.

most severe type of error, disregarding the homopolymers of length between 3 and 6 from the analysis. Moreover, the homopolymer frequency is defined as the number of homopolymers per 1000 bases in the DNA. Figure 7 shows that the proposed method is able to respect the homopolymer constraint, regardless of the rate or test dataset. On the other hand, both the JPEG DNA BC and the JPEG DNA BT present a high homopolymer frequency at the lowest encoding rates, rapidly decreasing when the encoding rate is increased. It can also be observed that, when the JPEG DNA BC was used to compress the JPEG DNA dataset, the homopolymer frequency never reaches zero.

5. CONCLUSIONS

In this paper, a method that combines a learning-based compression algorithm with a Goldman encoder applied in the latent domain is proposed for the representation of images based on DNA codes. The proposed method is used to compress images from two datasets and is evaluated with objective quality metrics. Results confirm that the proposed method outperforms previous methods designed to compress images for storage on DNA support. Moreover, contrary to the other evaluated approaches, the homopolymers constraint is found to be respected. Future works will focus on comparing the proposed approach to methods directly producing DNA from binary data from compressed image files, in order to better evaluate whether producing DNA code directly from source images can actually provide benefits. Moreover, since Goldman encoding does not ensure that other biochemical constraints, such as restricting the percentage of CG content or avoiding pattern repetitions, are met, the development of other mechanisms for the translation of latent features into DNA code respecting those constraints is desirable.

ACKNOWLEDGMENTS

The authors would like to acknowledge support from the Swiss National Scientific Research project entitled ”Compression of Visual information for Humans and Machines (CoViHM)” under grant number 200020_207918.

REFERENCES

[1] Ceze, L., Nivala, J., and Strauss, K., “Molecular digital data storage using dna,” *Nature Reviews Genetics* **20**(8), 456–466 (2019).

[2] Nguyen, B., Sinistore, J., Smith, J. A., Praneet Singh, A., Johnson, L. M., Kidman, T., DiCaprio, T., Carmean, D., and Strauss, K., “Architecting datacenters for sustainability: greener data storage using synthetic dna,” *Proc. Electronics Goes Green 2020* (2020).

[3] Grass, R. N., Heckel, R., Puddu, M., Paunescu, D., and Stark, W. J., “Robust chemical preservation of digital information on dna in silica with error-correcting codes,” *Angewandte Chemie International Edition* **54**(8), 2552–2555 (2015).

[4] Meiser, L. C., Nguyen, B. H., Chen, Y.-J., Nivala, J., Strauss, K., Ceze, L., and Grass, R. N., "Synthetic dna applications in information technology," *Nature communications* **13**(1), 352 (2022).

[5] ISO/IEC JTC1/SC29/WG1 N100517, "JPEG DNA Common Test Conditions version 2.0." https://jpeg.org/jpegdna/documentation.html.

[6] Antonini, M., Cruz, L., da Silva, E., Dimopoulou, M., Ebrahimi, T., Foessel, S., San Antonio, E. G., Menegaz, G., Pereira, F., Pic, X., et al., "DNA-based Media Storage: State-of-the-Art, Challenges, Use Cases and Requirements version 8.0," (2022).

[7] ISO/IEC JTC1/SC29/WG1 N100476, "Final Call for Proposals on Digital Media Storage on DNA Support." https://jpeg.org/jpegdna/documentation.html.

[8] Church, G. M., Gao, Y., and Kosuri, S., "Next-generation digital information storage in dna," *Science* **337**(6102), 1628–1628 (2012).

[9] Goldman, N., Bertone, P., Chen, S., Dessimoz, C., LeProust, E. M., Sipos, B., and Birney, E., "Towards practical, high-capacity, low-maintenance information storage in synthesized DNA," *nature* **494**(7435), 77–80 (2013).

[10] Blawat, M., Gaedke, K., Huetter, I., Chen, X.-M., Turczyk, B., Inverso, S., Pruitt, B. W., and Church, G. M., "Forward error correction for dna data storage," *Procedia Computer Science* **80**, 1011–1022 (2016).

[11] Erlich, Y. and Zielinski, D., "Dna fountain enables a robust and efficient storage architecture," *science* **355**(6328), 950–954 (2017).

[12] Luby, M., "Lt codes," in [*The 43rd Annual IEEE Symposium on Foundations of Computer Science, 2002. Proceedings.*], 271–271, IEEE Computer Society (2002).

[13] Schwarz, P. M. and Freisleben, B., "Norec4dna: using near-optimal rateless erasure codes for dna storage," *BMC bioinformatics* **22**(1), 1–28 (2021).

[14] Maymounkov, P., "Online codes," tech. rep., Technical report, New York University (2002).

[15] Shokrollahi, A., "Raptor codes," *IEEE transactions on information theory* **52**(6), 2551–2567 (2006).

[16] Dimopoulou, M., Antonini, M., Barbry, P., and Appuswamy, R., "A biologically constrained encoding solution for long-term storage of images onto synthetic dna," in [*2019 27th European Signal Processing Conference (EUSIPCO)*], 1–5, IEEE (2019).

[17] Dimopoulou, M., San Antonio, E. G., and Antonini, M., "A JPEG-based image coding solution for data storage on DNA," in [*2021 29th European Signal Processing Conference (EUSIPCO)*], 786–790, IEEE (2021).

[18] Secilmis, L., Testolina, M., Lazzarotto, D., and Ebrahimi, T., "Towards effective visual information storage on dna support," in [*Applications of Digital Image Processing XLV*], **12226**, 29–35, SPIE (2022).

[19] Ramos, J. E., Lazzarotto, D., Testolina, M., and Ebrahimi, T., "Analysis of the influence of errors in dna-based image coding," in [*CORESA (22nd edition of the CORESA COmpression et REprésentation des Signaux Audiovisuels)*], (2023).

[20] Pic, X., Dimopoulou, M., Antonio, E. G. S., and Antonini, M., "Mq-coder inspired arithmetic coder for synthetic dna data storage," *arXiv preprint arXiv:2306.12708* (2023).

[21] Ballé, J., Laparra, V., and Simoncelli, E. P., "End-to-end optimized image compression," in [*International Conference on Learning Representations*], (2016).

[22] Ballé, J., Minnen, D., Singh, S., Hwang, S. J., and Johnston, N., "Variational image compression with a scale hyperprior," *arXiv preprint arXiv:1802.01436* (2018).

[23] Minnen, D., Ballé, J., and Toderici, G. D., "Joint autoregressive and hierarchical priors for learned image compression," *Advances in neural information processing systems* **31** (2018).

[24] Mentzer, F., Toderici, G. D., Tschannen, M., and Agustsson, E., "High-fidelity generative image compression," *Advances in Neural Information Processing Systems* **33**, 11913–11924 (2020).

[25] Pic, X. and Antonini, M., "Image Storage on Synthetic DNA Using Autoencoders," *arXiv preprint arXiv:2203.09981* (2022).

[26] Pic, X. and Antonini, M., "A constrained shannon-fano entropy coder for image storage in synthetic dna," in [*2022 30th European Signal Processing Conference (EUSIPCO)*], 1367–1371, IEEE (2022).

[27] Bégaint, J., Racapé, F., Feltman, S., and Pushparaja, A., "Compressai: a pytorch library and evaluation platform for end-to-end compression research," *arXiv preprint arXiv:2011.03029* (2020). https://github.com/InterDigitalInc/CompressAI.

[28] "Kodak Lossless True Color Image Suite (PhotoCD PCD0992)," (accessed: July 2023). "http://r0k.us/graphics/kodak/".

[29] Testolina, M., Hosu, V., Jenadeleh, M., Lazzarotto, D., Saupe, D., and Ebrahimi, T., "JPEG AIC-3 Dataset: Towards Defining the High Quality to Nearly Visually Lossless Quality Range," in [*2023 15th International Conference on Quality of Multimedia Experience (QoMEX)*], 55–60, IEEE (2023).

[30] Wang, Z., Simoncelli, E. P., and Bovik, A. C., "Multiscale structural similarity for image quality assessment," in [*The Thrity-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*], **2**, 1398–1402, Ieee (2003).

[31] Wang, Z. and Li, Q., "Information content weighting for perceptual image quality assessment," *IEEE Transactions on image processing* **20**(5), 1185–1198 (2010).

[32] Zhang, L., Zhang, L., Mou, X., and Zhang, D., "Fsim: A feature similarity index for image quality assessment," *IEEE transactions on Image Processing* **20**(8), 2378–2386 (2011).

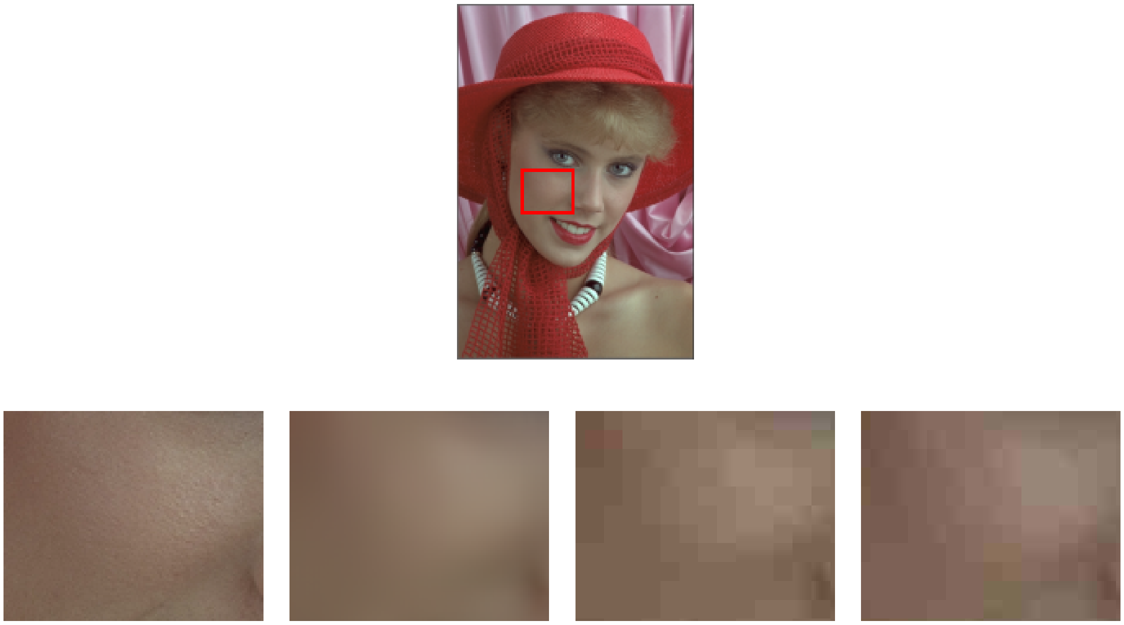
[33] Laparra, V., Ballé, J., Berardino, A., and Simoncelli, E. P., "Perceptual image quality assessment using a normalized laplacian pyramid," *Electronic Imaging* **2016**(16), 1–6 (2016).

[34] Sheik, H. and Bovik, A., "A visual information fidelity measure for image quality assessment," *IEEE T. Img. Proc* **15**(2), 430–444 (2006).

[35] Li, Z., Aaron, A., Katsavounidis, I., Moorthy, A., and Manohara, M., "Toward a practical perceptual video quality metric," *The Netflix Tech Blog* **6**(2) (2016).

APPENDIX A. VISUAL INSPECTION ON THE KODAK DATASET: KODIM04

Figure 8 presents a crop from image *kodim04* of the Kodak dataset compressed with the proposed compression method and the two anchor methods considered in this paper. All the reported crops were encoded targeting a similar rate of approximately 0.25 nucleotides/pixel. The crop decoded with the proposed codec presents smoothing artifacts and loss of details. On the other hand, the other anchor methods present the typical JPEG artifacts, namely blocking artifacts and color distortions. Among the presented crops, the proposed method is the method that presents the least amount of distortions, resulting in more visually-pleasing results.



(a) Original

(b) Proposed (0.26 nts/pxl)

(c) JPEG DNA BC (0.32 nts/pxl)

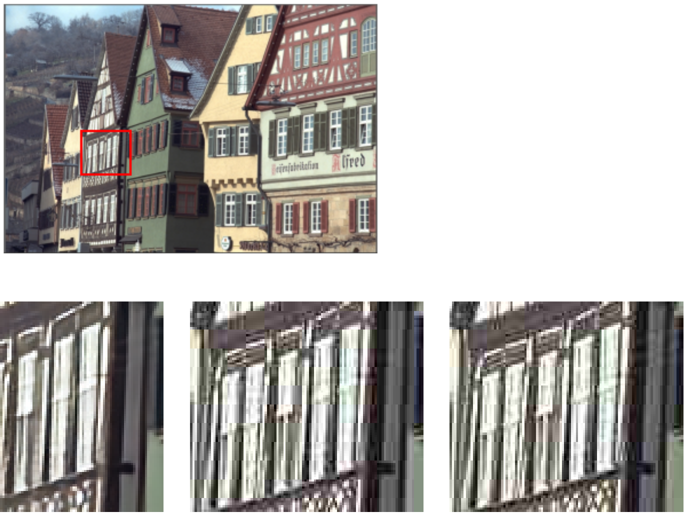
(d) JPEG DNA BT (0.27 nts/pxl)

Figure 8: Visual comparison of a crop from image *kodim04* from the Kodak dataset, compressed with the proposed codec and the two anchor methods considered in this paper.

APPENDIX B. VISUAL INSPECTION ON THE KODAK DATASET: KODIM08

Figure 9 presents a crop from image *kodim08* of the Kodak dataset compressed with the proposed learning-based DNA compression method and the two anchor methods considered in this paper. All the reported crops were encoded targeting a similar rate of approximately 0.55 nucleotides/pixel. Notably, the presence of small details and texture in the image allows the evaluation of all models on such image types.

In this example, the performance of the proposed codec, JPEG DNA BC, and JPEG DNA BT are equivalent to each other: on one side, the proposed codec generates blurriness in the image, therefore losing details in the white panels of the windows; on the other side, the JPEG DNA BC and JPEG DNA BT present blocking artifacts and color distortions.



(a) Original

(b) Proposed (0.48 nts/pxl)

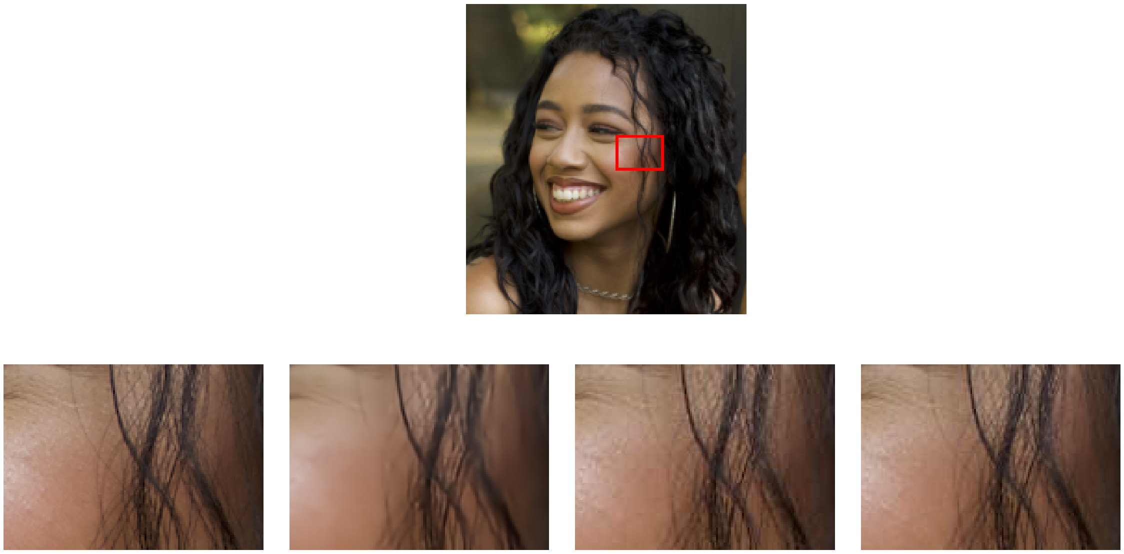
(c) JPEG DNA BC (0.64 nts/pxl)

(d) JEPG DNA BT (0.56 nts/pxl)

Figure 9: Visual comparison of a crop from image *kodim08* from the Kodak dataset, compressed with the proposed codec and the two anchor methods.

APPENDIX C. VISUAL INSPECTION ON THE JPEG DNA DATASET: IMAGE 00002

As mentioned in Section 4, the performances of JPEG DNA BC and the JPEG DNA BT are, on average, higher when measured on the JPEG DNA dataset compared to the Kodak dataset. Figure 10 presents a crop from image *00002* of the JPEG DNA dataset compressed with the proposed codec and the two anchor methods considered in this paper. All the reported crops were encoded targeting a similar rate of approximately 0.5 nucleotides/pixel. This region was selected to analyze the effect of compression on fine texture areas, e.g. skin and hair. In this case, the proposed codec presents severe smoothing artifacts, which entirely smooth the texture of the skin and degrade the lines of the hair. The other anchor methods, on the other hand, present light-blocking artifacts, better preserving the texture and information. For all methods, no color distortions can be observed.



(a) Original

(b) Proposed, (0.50 nts/pxl)

(c) Benchmark Codec (0.49 nts/pxl)

(d) BC transcoder (0.53 nts/pxl)

Figure 10: Visual comparison of a crop from image *00002* from the JPEG DNA dataset, compressed with the proposed codec and the two anchor methods.