Quantization-free Lossy Image Compression Using Integer Matrix Factorization

Pooya Ashtari[1][*][†]
pooya.ashtari@esat.kuleuven.be

Pourya Behmandpoor[1][†]
pourya.behmandpoor@esat.kuleuven.be

Fateme Nateghi Haredasht[2]
fnateghi@stanford.edu

Jonathan H. Chen[2]
jonc101@stanford.edu

Lieven De Lathauwer[1]
lieven.delathauwer@kuleuven.be

Sabine Van Huffel[1]
sabine.vanhuffel@esat.kuleuven.be

[1]Department of Electrical Engineering (ESAT), STADIUS Center, KU Leuven, Belgium
[2]Department of Medicine, Stanford University, Stanford, CA, USA

Abstract

one paragraph

1 Introduction

2 Related Work

3 Method

3.1 Overall Framework

Figure 1 illustrates an overview of the encoding pipeline for our proposed image compression method using integer matrix factorization (IMF). The encoder accepts an RGB image with dimensions $H \times W$ and a color depth of 8 bits, represented by the tensor $\boldsymbol{\mathcal{X}} \in \{0, \ldots, 255\}^{3 \times H \times W}$. Each step of encoding is described in the following.

Color Space Transformation. Analogous to the JPEG standard, the image is initially transformed into the $\mathrm{YC_BC_R}$ color space:

$$\begin{bmatrix} Y \\ C_B \\ C_R \end{bmatrix} \triangleq \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.168736 & -0.331264 & 0.5 \\ 0.5 & -0.418688 & -0.081312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}, \tag{1}$$

where Y represents the *luma* component, and C_B and C_R are the blue-difference and red-difference *chroma* components, respectively. Note that as a result of this transformation, the elements of the luma ($\boldsymbol{Y}$) and chroma ($\boldsymbol{C}_B$, $\boldsymbol{C}_R$) matrices are no longer integers and can take any value within the range $[0, 255]$.

[*] Corresponding author. Emails:pooya.ashtari@esat.kuleuven.be, pooya.ash@gmail.com
[†] Equal contribution

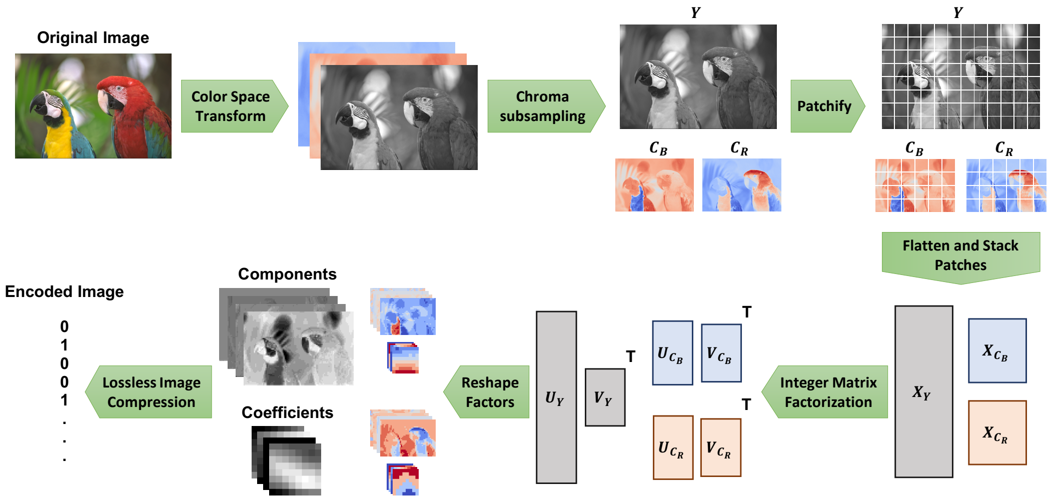


Figure 1 An illustration of the encoder for our image compression method, based on integer matrix factorization.

Chroma Downsampling. After conversion to the $\mathrm{YC_BC_R}$ color space, the chroma channels C_B and C_R are downsampled by a factor of 2, similar to the process used in JPEG. This results in three components: the luma matrix $\boldsymbol{Y} \in [0,255]^{H \times W}$ and the chroma matrices $\boldsymbol{C}_B, \boldsymbol{C}_R \in [0,255]^{\frac{H}{2} \times \frac{W}{2}}$. This downsampling leverages the fact that the human visual system perceives far more detail in brightness information (luma) than in color saturation (chroma).

Patchification. Each of the matrices $\boldsymbol{Y} \in [0,255]^{H \times W}$, $\boldsymbol{C}_B \in [0,255]^{\frac{H}{2} \times \frac{W}{2}}$, and $\boldsymbol{C}_R \in [0,255]^{\frac{H}{2} \times \frac{W}{2}}$ is split into non-overlapping 8×8 patches. If a dimension of a matrix is not divisible by 8, the matrix is first padded to the nearest size divisible by 8 using reflection of the boundary values. These patches are then flattened into row vectors and stacked vertically to form matrices $\boldsymbol{X}_Y \in [0,255]^{\frac{HW}{64} \times 64}$, $\boldsymbol{X}_{C_B} \in [0,255]^{\frac{HW}{256} \times 64}$, and $\boldsymbol{X}_{C_R} \in [0,255]^{\frac{HW}{256} \times 64}$. Later, these matrices will be low-rank approximated using integer matrix factorization (IMF). Note that this patchification technique differs from the block splitting in JPEG, where each block is subject to discrete cosine transform (DCT) and processed independently. The patchification technique not only captures the locality and spatial dependencies of neighboring pixels but also performs better with the matrix decomposition approach to image compression.

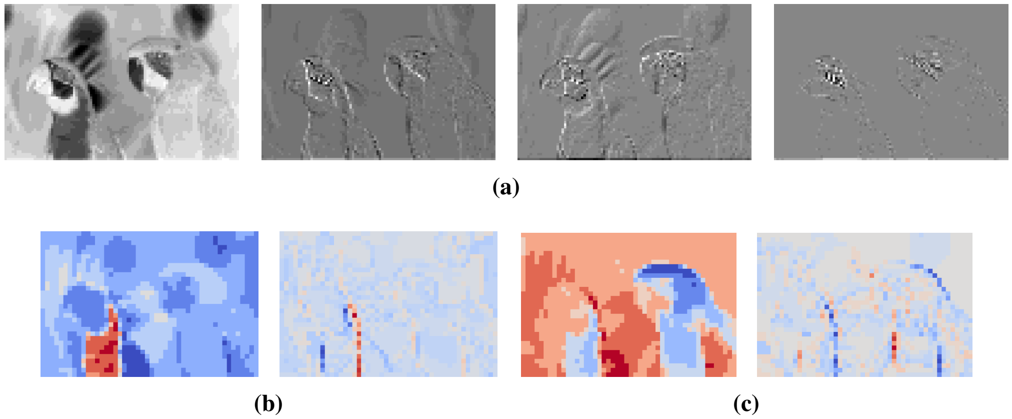
Low-rank approximation. We can now low-rank approximate the matrices $\boldsymbol{X}_Y$, $\boldsymbol{X}_{C_B}$, $\boldsymbol{X}_{C_R}$, which is the core step in our compression method that provides a lossy compressed represention of these matrices. The low-rank approximation [1] seeks to approximate some given matrix $\mathbf{X} \in \mathbb{R}^{M \times N}$ by

$$\boldsymbol{X} \approx \boldsymbol{U}\boldsymbol{V}^{\mathsf{T}} = \sum_{r=1}^{R} U_{:r} {V_{:r}}^{\mathsf{T}}, \tag{2}$$

where $\boldsymbol{U} \in \mathbb{R}^{M \times R}$ and $\boldsymbol{V} \in \mathbb{R}^{N \times R}$ are *factor matrices*, and $R \leq \min(M, N)$ is known as the *rank*. By setting R to sufficiently small value, the factor matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ with a combined number of $(M+N)R$ elements provide a compressed representation of the original matrix with MN elements, encapsulating the most significant patterns in the image. With

Reshape factors.

Lossless image compression.



(a)

(b)

(c)

Figure 2 IMF components of the `kodim23` image from the Kodak dataset. Panels (a), (b), and (c) show the IMF components corresponding to luma (Y), blue-difference (Cb), and red-difference (Cr) chroma, respectively.

3.2 Integer Matrix Factorization (IMF)

3.3 Block Coordinate Descent Scheme for IMF

Theorem 1. *The IMF cost function, $\|\boldsymbol{X} - \boldsymbol{U}\boldsymbol{V}^\mathsf{T}\|_F^2$, is monotonically nonincreasing under each of the multiplicative update rules.*

Proof. See Appendix A for the proof. □

3.4 Implementation Details

4 Experiments

In this section, the proposed IMF-based compression method is assessed against SVD-based and JPEG [] methods. The performance is reported qualitatively and also based on two criteria, namely rate-distortion performance and image classification performance. Moreover, ablation studies are presented to investigate the effect of different hyperparameters in IMF.

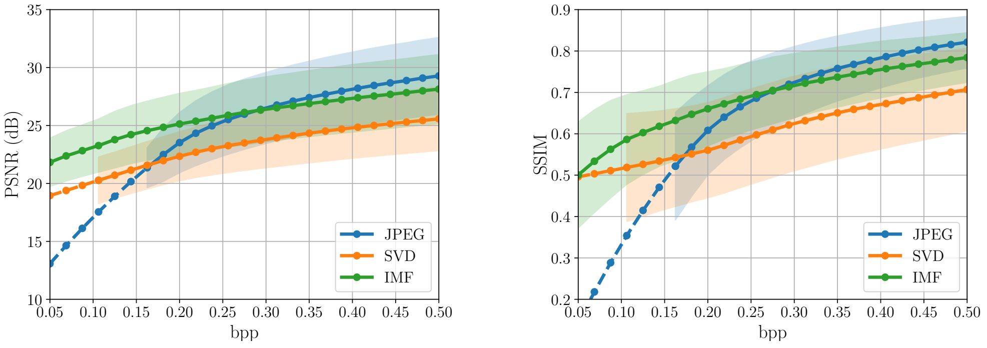
4.1 Qualitative Performance

Qualitative performance is shown on an image selected from the Kodak [] dataset. Fig.2 depicts the first IMF components U_Y, U_{Cb}, and U_{Cr} which are extracted following the procedure elaborated in Section 3.4. It is evident in the figure that the IMF components with higher energy maintain the overall texture of the image in each channel, while components with lower energy focus more on subtle changes.

The quantitative comparison is made in Fig.**??** on an image selected from the Kodak [] dataset. The images compressed by the considered compression methods are shown in different bits per pixel (bpp) values, a standard compression measure in the literature []. As can be seen, the IMF-based compression method is capable of maintaining quality compressed images in bpp values as low as ?, outperforming JPEG and SVD-based methods which suffer from maintaining balanced colors as soon as bpp drops below ? and ?, respectively. The artifacts in color are visible, e.g. by JPEG in bpp values starting ?.

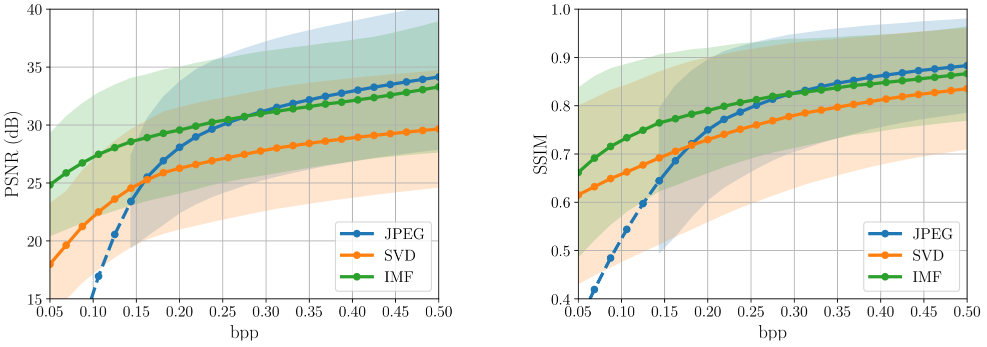
4.2 Rate-Distortion Performance

Peak signal-to-noise ratio (PSNR), as well as structural similarity index measure (SSIM) [], are reported versus bpp for the considered methods. The considered datasets are Clic [] and Kodak, consisting of respectively 32 and 24 colored images of various sizes. For each image compressed by any of the considered methods, bpp, PSNR, and SSIM are calculated. Then, for each compression method, PSNR and SSIM values are interpolated linearly in the fixed bpp values in the range



(a) **(b)**

Figure 3 Rate-distortion performance on the Kodak dataset. In panels (a) and (b), the average PSNR and SSIM are plotted against bpp, respectively.



(a) **(b)**

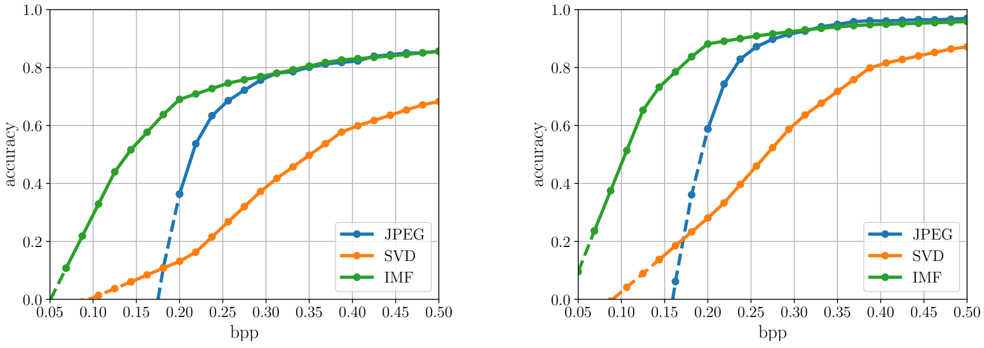
Figure 4 Rate-distortion performance on the CLIC dataset. In panels (a) and (b), the average PSNR and SSIM are plotted against bpp, respectively.

$(0.05, 0.5)$. In the following plots, the average performance over all images is reported, along with the standard deviation presented as shadows. For the missing bpp values, the average is extrapolated quadratically and is shown by dashed lines.

In Fig.3, the compression performance on the Kodak dataset is reported. In this figure, the proposed IMF-based method outperforms the SVD-based method, which can be attributed to the quantization errors that SVD is prone to during encoding and decoding, deteriorating its performance in both criteria. In this view, the quantization-free property of IMF effectively guarantees higher performance in different bpp values. It is also evident that the IMF-based method outperforms JPEG in low bpp values. The same performance regime can also be concluded for all the mentioned compression methods on the Clic dataset in Fig.4.

4.3 ImageNet Classification Performance

As another criterion, classification performance is investigated on the images compressed by the considered compression methods. This criterion focuses on the higher-level information required for object recognition and classification embedded in each image. Furthermore, it highlights the importance of image compression where various vision tasks such as classification are the main objective—rather than maintaining the perceived image quality—while keeping the requirement of resources such as memory, communication bandwidth, computation power, latency budget, etc. as limited as possible. ImageNet [] validation set, consisting of 50000 224×224 colored images in 1000 classes, is considered for this classification task done by a ResNet-50 classifier [], pre-trained on the original ImageNet dataset. The classification performance comparison is made in Fig.5, showing



(a) **(b)**

Figure 5 Impact of different compression methods on ImageNet classification accuracy. Panels (a) and (b) show the validation top-1 and top-5 accuracy plotted against bits per pixel (bpp), respectively. A ResNet-50 model pre-trained on the original ImageNet images is evaluated using validation images compressed by different methods.

the higher compression performance of IMF for classification, reaching top-1 accuracy of 70% in bpp values as low as 0.2.

4.4 Ablation Studies

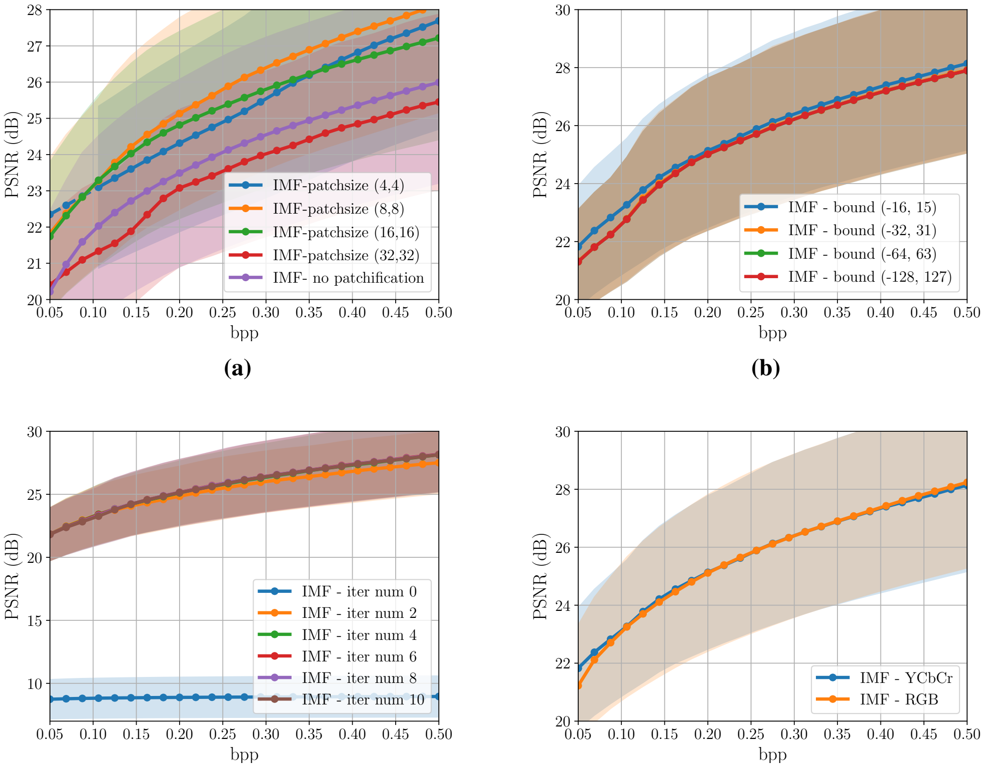
In this section, ablation studies are performed, focusing on various hyper-parameters in the IMF-based compression method and their effect on the compression performance.

Patchification. In Fig.6a, patchification effect with different patch sizes is investigated. First, it can be concluded that patchification has a positive effect on performance. This performance boost is mainly due to the fact that in each patch (local) pattern variation is limited and hence IMF has more representation power to reconstruct the original pattern in each patch with fewer components. The performance on various datasets has shown that patches of size 8×8 lead to the best performance. The same conclusion is evident for the Kodak dataset example presented in Fig.6a.

Factor bounds. As elaborated in Section 3.4, during the IMF optimization, IMF components are constrained into a bound $(-\alpha, \alpha - 1)$. Fig.6b studies the compression performance versus different bounds. According to the results, the bound $(-16, 15)$ leads to the best performance since the value distribution of IMF components lies mostly in this bound. Hence, dedicating fewer bits to represent this narrower bound compared to the other bounds results in higher compression rates without sacrificing performance.

BCD iteration. The next parameter to study is the inner iteration number required in IMF BCD updates. According to the numerical results on various datasets, the objective value in the IMF optimization drops drastically after 2 iterations, while more iteration numbers have marginal improvement. This observation is shown for the Kodak dataset in Fig.6c. This feature makes the IMF computationally efficient since with a limited number of iterations a high compression performance can be achieved.

Color space. The compression performance of IMF is studied with two color spaces, namely RGB and YCbCr, in Fig.6d. Although the compression performance remains unchanged in terms of PSNR, qualitative results reported in Fig.? indicate that YCbCr color space can maintain natural colors and brightness more effectively. Consistently, the JPEG method employs this color space as well.



(a)

(b)

(c)

(d)

Figure 6 Ablation experiments for the IMF compression method. In all cases, we plot PSNR as a function of bits per pixel (bpp) on the Kodak dataset. (a) Compares IMF compression performance without patchification and different patch sizes. (b) Compares IMF compression performance for different bound values of factor matrices. (c) Compares IMF compression performance for different numbers of BCD iterations. (d) Compares IMF compression performance between RGB and YCbCr color space transform.

5 Conclusion and Future Work

Acknowledgments and Disclosure of Funding

References

[1] Carl Eckart and Gale Young. The approximation of one matrix by another of lower rank. *Psychometrika*, 1 (3):211–218, 1936.

A Proof of Theorem 1