Quantization-free Lossy Image Compression Using Integer Matrix Factorization

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

Pourya Behmandpoor[1†]
pooya.ashtari@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

Preprint. Under review.

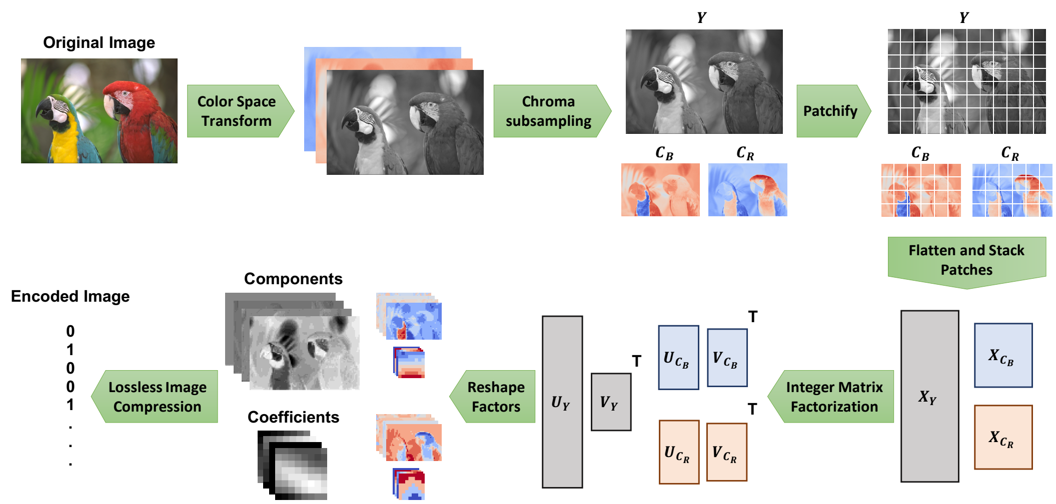


Figure 1 An illustration of the encoder for our image compression method, based on integer matrix factorization.

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)

Figure 2 Rate-distortion performance on the Kodak dataset. In panels (a) and (b), the average PSNR and SSIM are plotted against bits per pixel (bpp), respectively.

(a) (b)

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

3.2 Integer Matrix Factorization (IMF)

3.3 Block Coordinate Descent Scheme for IMF

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

Proof. See Appendix A for the proof. □

3.4 Implementation Details

4 Experiments

4.1 Rate-Distortion Performance

4.2 ImageNet Classification Performance

4.3 Ablation Studies

Patchification. without patchification, patch size 4, 8, 16, 32

Factor bounds.

BCD iteration.

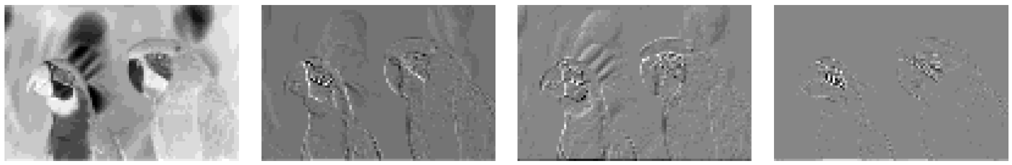
Color space.

5 Conclusion and Future Work

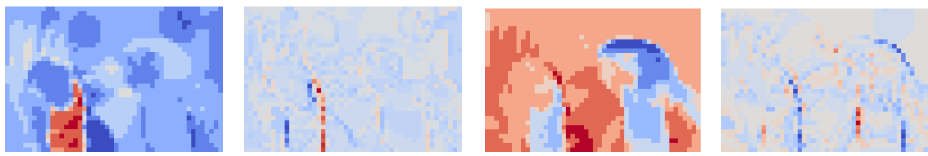
Acknowledgments and Disclosure of Funding

References

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



(a)



(b) **(c)**

Figure 4 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.

(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 pretrained on the original ImageNet images was evaluated using validation images compressed by different methods.

(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.

A Proof of Theorem 1