Homework 3 - SVD

Computational linear algebra for large scale problems

Gaudino Andrea s346119
Grivet Talocia Lorenzo s346559

A.A. 2024/25

1.1 Introduction

Abstract

This study explores the application of *Singular Value Decomposition* (SVD) as an effective technique for data compression for image and audio sources. By decomposing the data matrix into singular vectors and singular values, we identify the most significant components while discarding less important ones. Performing the compression, the method effectively reduces the data size while maintaining acceptable quality.

1.1.1 Source description

We perfomed our analysis on two different categories of media: image and audio. To better highlight the definition of the outlines using different compression factors, we selected an high-quality image, 2000×1377 pixels, with few well-defined colors: the famous painting *La Danse* by Matisse, presented in figure 1.1.

Figure 1.1: Matisse, *La Danse*

Regarding the audio file, we wanted to conduct the *SVD* on a voice recording which could show clearly, by observing the spectrogram, the differences of the original track and the compressed one. Therefore, we decided to record a short audio file of us counting from one to ten. In the figure 1.2 we display the spectrogram of the original audio file in order to observe the intensity of frequencies as a function of time. We can clearly notice the frequencies and the seconds corresponding to the words, associated with a lighter color.

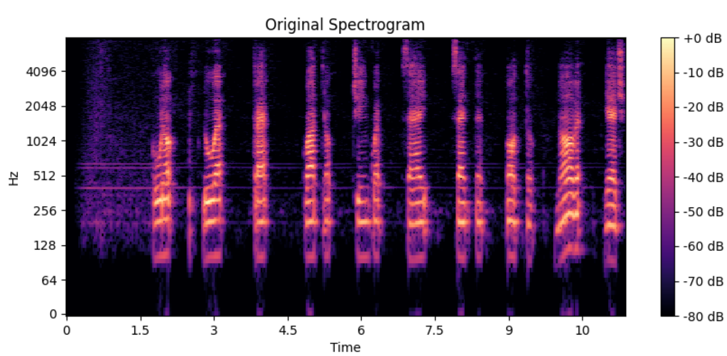


Figure 1.2: Spectrogram of the original audio file

1.2 Singular Value Decomposition

Both of our sources are represented as a $m \times n$ matrices, that lead to the creation of three matrices:

- U: a $m \times m$ column-orthonormal matrix
- Σ: a $m \times n$ diagonal matrix with non negative entries
- V: a $n \times n$ column-orthonormal matrix

such that:

$$A = U \Sigma V^T \tag{1.1}$$

$$U \Sigma V^T = \begin{pmatrix} u_{11} & u_{12} & \cdots & u_{1m} \\ u_{21} & u_{22} & \cdots & u_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ u_{m1} & u_{m2} & \cdots & u_{mm} \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_{\min(m,n)} \end{pmatrix} \begin{pmatrix} v_{11} & v_{12} & \cdots & v_{1n} \\ v_{21} & v_{22} & \cdots & v_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ v_{n1} & v_{n2} & \cdots & v_{nn} \end{pmatrix} \tag{1.2}$$

The columns of the matrix U, called *left-singular vectors*, and the rows of the matrix V^T, called *right-singular vectors*, represent the orthogonal eigenvectors associated with the eigenvalues of AA^T and A^TA, respectively.

Therefore, we can affirm that the columns of U correspond to the non-zero singular values of the column space of the matrix A, and the rows of V^T correspond to the non-zero singular values associated with the row space of the matrix A.

Furthermore, the kernel and the image of the matrix A can be written as a function of the matrices U and V. Given the rank of A as r, the kernel of A is the subspace of R^n generated by the $n - r$ columns of V corresponding to the singular values $\sigma_{r+1} = \cdots = \sigma_n = 0$. The image of A, instead, is the subspace of R^m generated by the columns of U corresponding to the non vanishing singular values.

1.3 Image compression

The first application of SVD that we decided to study is the image compression, which is especially useful for high-dimensional data. The image indeed can also be represented by using less number of singular values, thus, presenting necessary features of an image while compressing it.

We can decompose the given image into the three color channels red, green and blue. Each channel can be represented as a $m \times n$ matrix with values ranging from 0 to 255. In the figure 1.3 we display the three separated channels of the image.

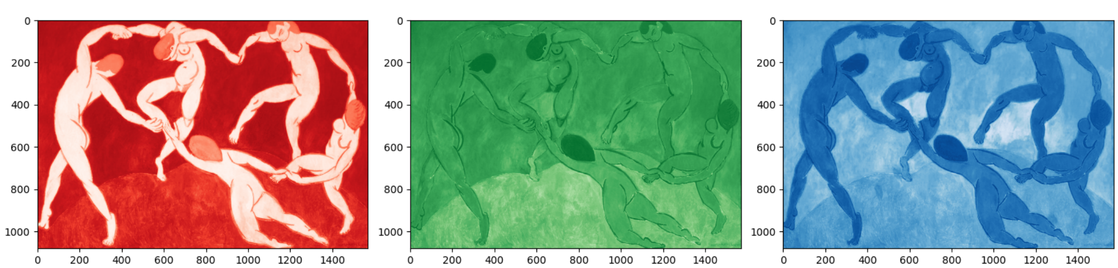


Figure 1.3: RGB channels

We performed the SVD on each of the three channels and we created the graph displayed in figure 1.4 in order to observe the magnitude of the singular values and understand which components have the most significant importance.

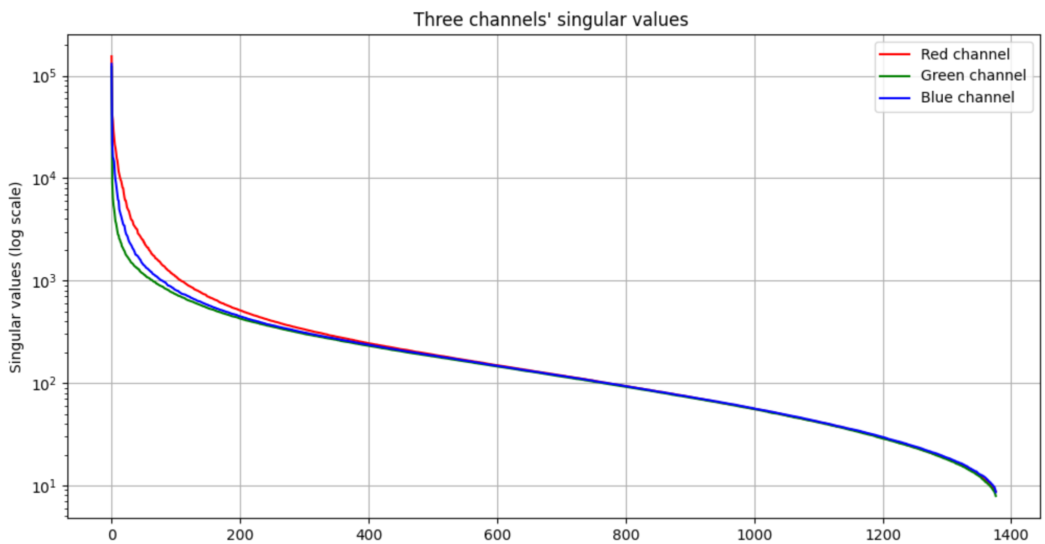


Figure 1.4: Singular values of each channel

By interpreting the graph 1.4, we observe that the first few dozens of singular values are significantly higher than the others, indicating that they have a substantial impact on the image. Furthermore, the graph demonstrate an property of the SVD: the data in the matrices U, Σ and V are sorted by how much they contribute to the matrix A in the product.

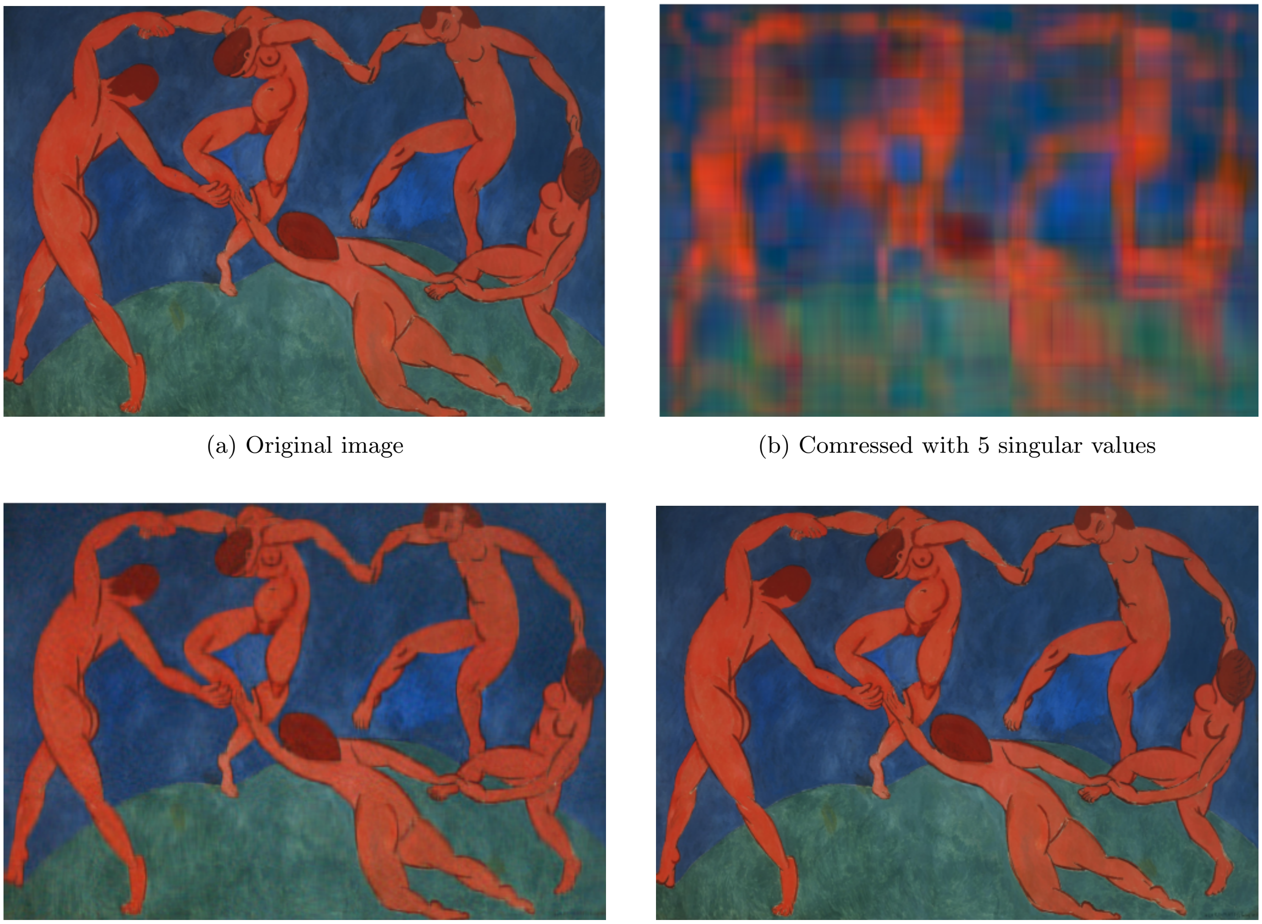
As a means to inspect how the reconstruction of the image would appear when selecting different number of singular values, we reassembled the matrix A by combining the RGB channels.

The compressed image requires less storage space as compared to the original image: in the table 1.1 we show the compression ratio of the trials that we conducted. We kept only the first k columns of U, the first k rows of V and the upper left $k \times k$ sub matrix of Σ, containing the k largest (and therefore most important) singular values. That enables us to get a quite good approximation by simply using only the most important parts of the matrices.

Number of singular values (k)	Compression ratio
5	99.39%
50	93.87%
500	38.67%

Table 1.1: Compression ratios

Choosing the number of singular values for compression and reconstruction of the image is an important decision for acceptable reconstruction. In the figure 1.5 we display the three trials that we conducted (with k equals to 5, 50 and 500) in order to draw attention to the actual difference in the image reconstruction.



(a) Original image

(b) Comressed with 5 singular values

(c) Comressed with 50 singular values

(d) Compressed with 500 singular values

Figure 1.5: Reconstruction of original image with different number of singular values (k)

Immediately, we observed that as the number of k decreases the image quality degrades. The higher this number, the better the quality of the approximation gets but also the more data are needed to encode it.

The reconstruction of the image displayed in Figure 1.5b demonstrates how the first and most significant components capture the essential shapes and structure of the original image, however we are still not able to recognize them and the quality is very poor. By increasing the number of singular values used in the reconstruction to 50, the shapes become distinct and understandable, although the definition of the color outlines and finer details are still lacking. The last image, obtained with 500 singular values is, to the bare eye, unrecognizable from the original one and the colors and shapes are completely defined.

In order to examine the differences between the images in greater detail, we decided to compute the *Peak signal-to-noise ratio* (PSNR), which could help us to determine the accuracy of the compressed images, calculated as follows:

$$PSNR = 10 \cdot \log_{10}\left(\frac{MAX^2}{MSE}\right) \tag{1.3}$$

In figure 1.6 we display the graph representing the PSNR as a function of the number of singular values chosen, from which we understood that choosing to keep 200 singular value components or more could restore a good image quality.

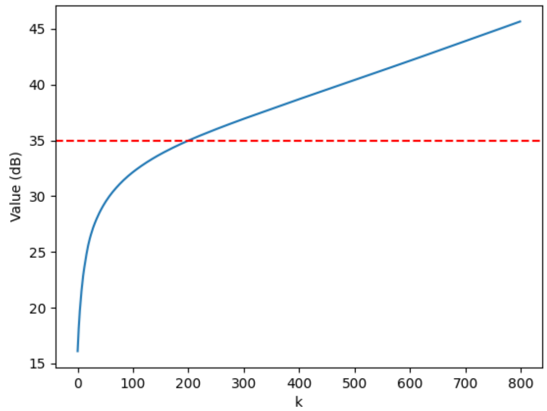


Figure 1.6: Image quality reconstruction evaluation based on PSNR

As a consequence, the choice of k represents a trade-off between image quality and the amount of data required for encoding, making it crucial to balance these factors based on the application.

1.4 Audio compression

In this chapter we analyze how the SVD behave when the source is an audio file; as we were describing in section 1.1.1 the track we chose features a voice counting from one to ten. The algorithm we applied is nearly the same of the one implemented for the image compression in section 1.3: the file compression process relies on restricting the selection of singular values.

Firstly, employing the *Python Librosa* package, we loaded the track and computed the *Fourier transform* to convert the signal from the time domain into the frequency domain, enabling the separation of magnitude and phase components.

We performed the SVD on the signal magnitude matrix to inspect how the corresponding components would impact the audio segment. In the figure 1.7 we display the graph representing the singular values size.

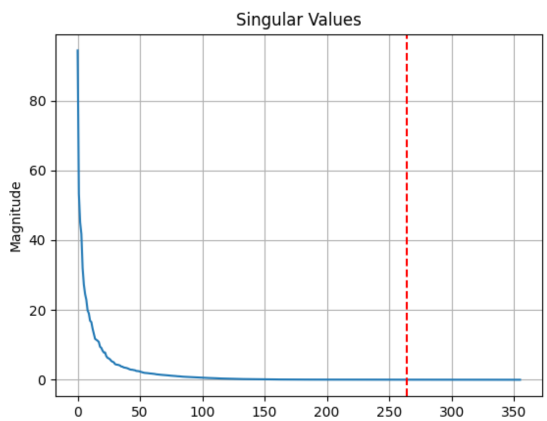


Figure 1.7: Singular values

We could instantly notice a difference in the graph shown in Figure 1.4: the singular values derived from the audio file decrease much more rapidly compared to those of the image. This means that the first components of the track have a crucial impact on it.

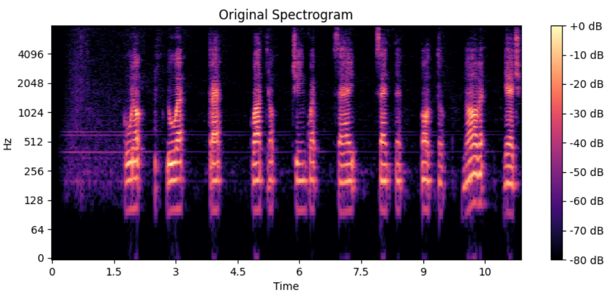
Examining the graph more carefully, we can extract additional insights. Starting from the hundredth singular value, the curve of the graph no longer shows significant decreases, hence we expect that truncating the original matrix to the first hundred components would have a very satisfactory audio quality.

In order to stress the differences between the file reconstructions obtained by keeping different number of singular values (denoted with k), we decided to apply the decomposition using k equals to 1, 100 and the maximum possible (in our case it is 264, where the red vertical lines is drawn, figure 1.7). In the figure 1.2 we show the corresponding compression ratios.

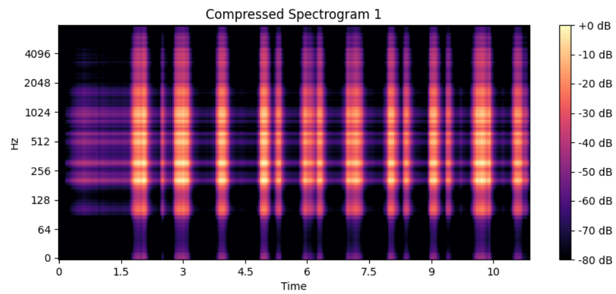
Number of singular values (k)	Compression ratio
1	99.62%
100	62.13%
264	0%

Table 1.2: Compression ratios

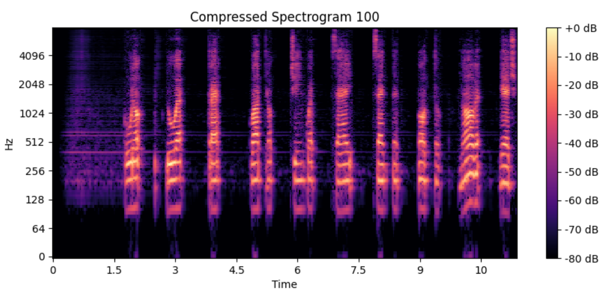
In the figure 1.8 we show the corresponding spectrograms.



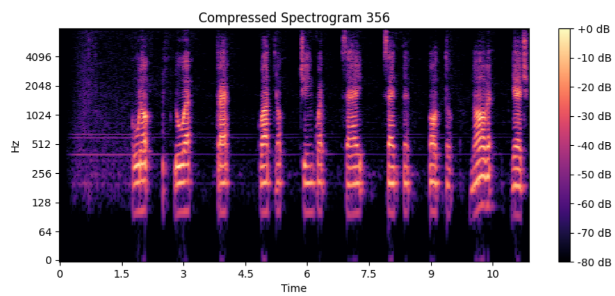
(a) Original image



(b) Comressed with 1 singular value



(c) Comressed with 100 singular values



(d) Comressed with 264 singular values

Figure 1.8: Reconstruction of original spectrogram with different number of singular values (k)

Figure 1.8b, representing the spectrogram created using only the first component, demonstrates that, by observing the time axis, we can still identify when the words were pronounced. However, it is not possible to determine the exact frequency at each time instant. Indeed, listening to the reconstructed audio track, it is impossible to guess which were the pronounced words.

Observing the graph in Figure 1.8c and listening to the reconstructed audio file, it is evident that there are no significant differences between the spectrograms of the original and reconstructed version. Furthermore, these differences are nearly imperceptible when listening to the audio tracks. As a result, selecting $k = 100$ proves to be an effective choice for compressing the file.

Last spectrogram in figure 1.8d is obtained keeping the maximum number of singular values in order not to drop the compression ratio under 0%, storing bigger data than the original track.

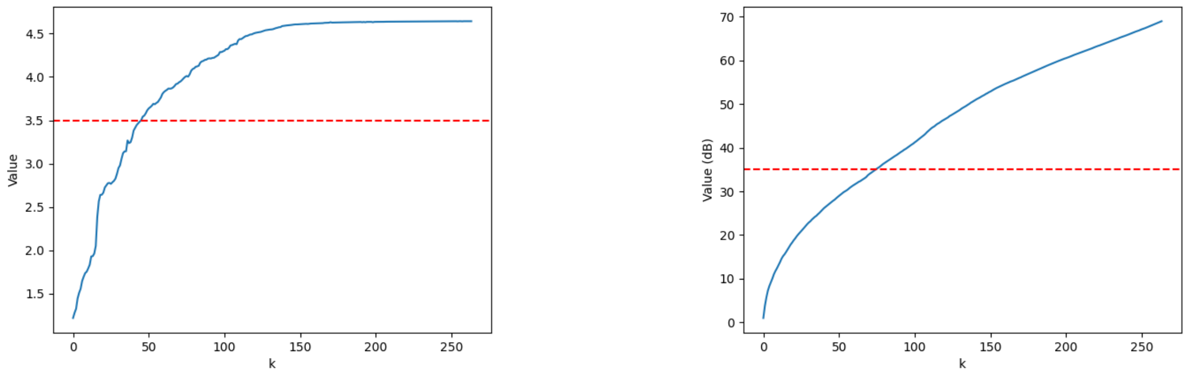
To advance the analysis further we conducted an evaluation of the outcomes we obtained based on two different metrics:

- *Perceptual Evaluation of Speech Quality* (PESQ), that analyzes the speech quality taking into account numerous characteristics such as: audio sharpness, volume or background noise.

  Minimum: -0.5, maximum: 4.5.

- *Signal-to-Distortion* (SDR), that represents how much louder the original signal is compared to the distortion or noise present in the degraded version.

  It is measured in Decibel.



(a) PESQ

(b) SDR

Figure 1.9: Audio evaluation as a function of the number of singular values (k)

The red horizontal lines represent the thresholds of an adequate quality of the compression, therefore we observed that choosing a number of singular values ranging from 50 to 60 is the best choice because it allows us to keep an understandable audio track removing components with minimal effect on the overall quality. This is in accordance with what was demonstrated in figure 1.7; indeed, from the 50th singular value onwards, the graph curve decreases at a much slower rate.

1.5 Conclusion

The study demonstrates how SVD is an effective tool for data compression either for image an audio files. The quality of the data reconstructions, indeed, can be maintained excellent regardless the increase of the compression factor. By choosing the right value of singular values and vectors to keep into account, thanks to the SVD we can find the best balance between data quality and storage space.