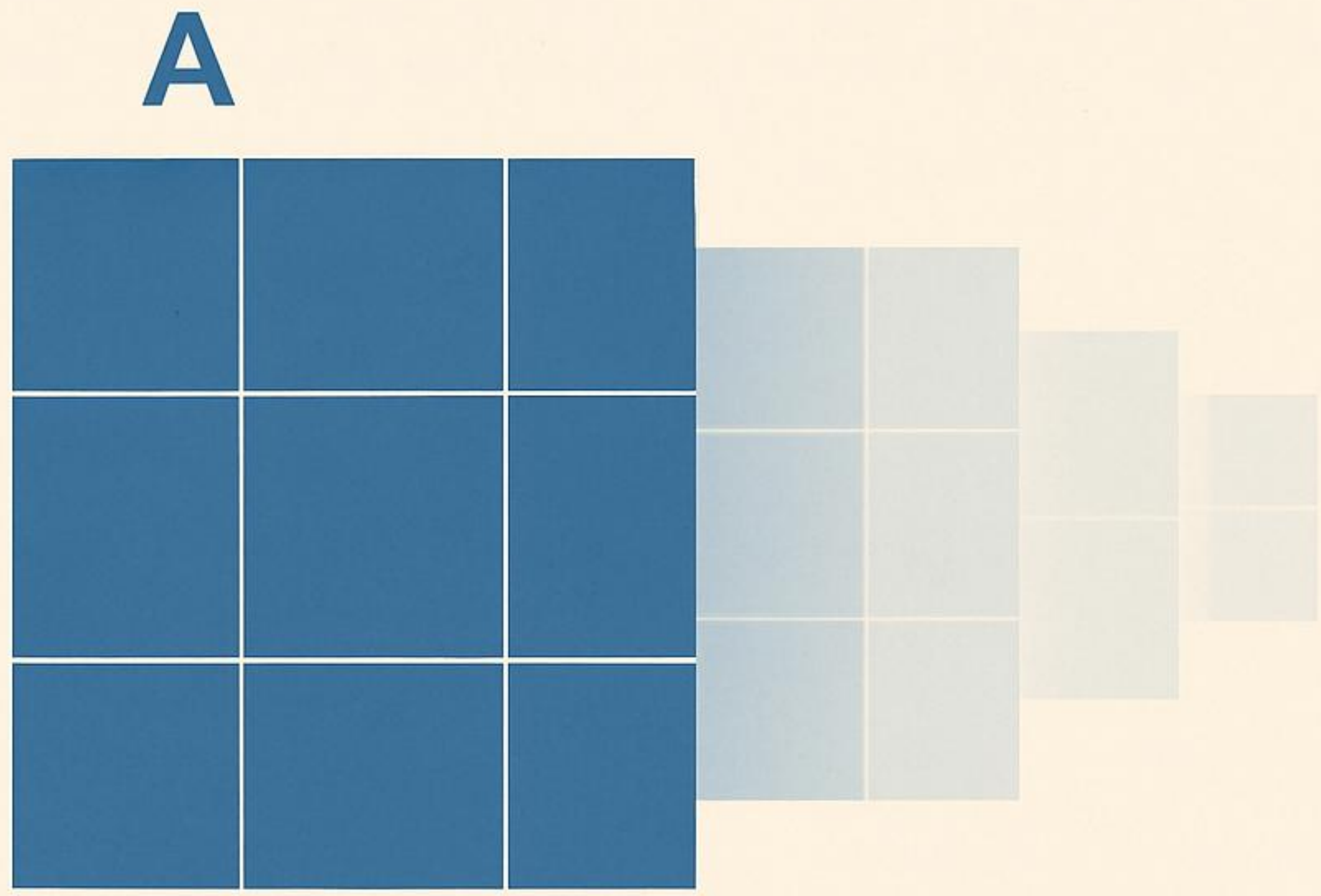
P346 DIY Presentation

Singular Value Decomposition:

Image Compression

&

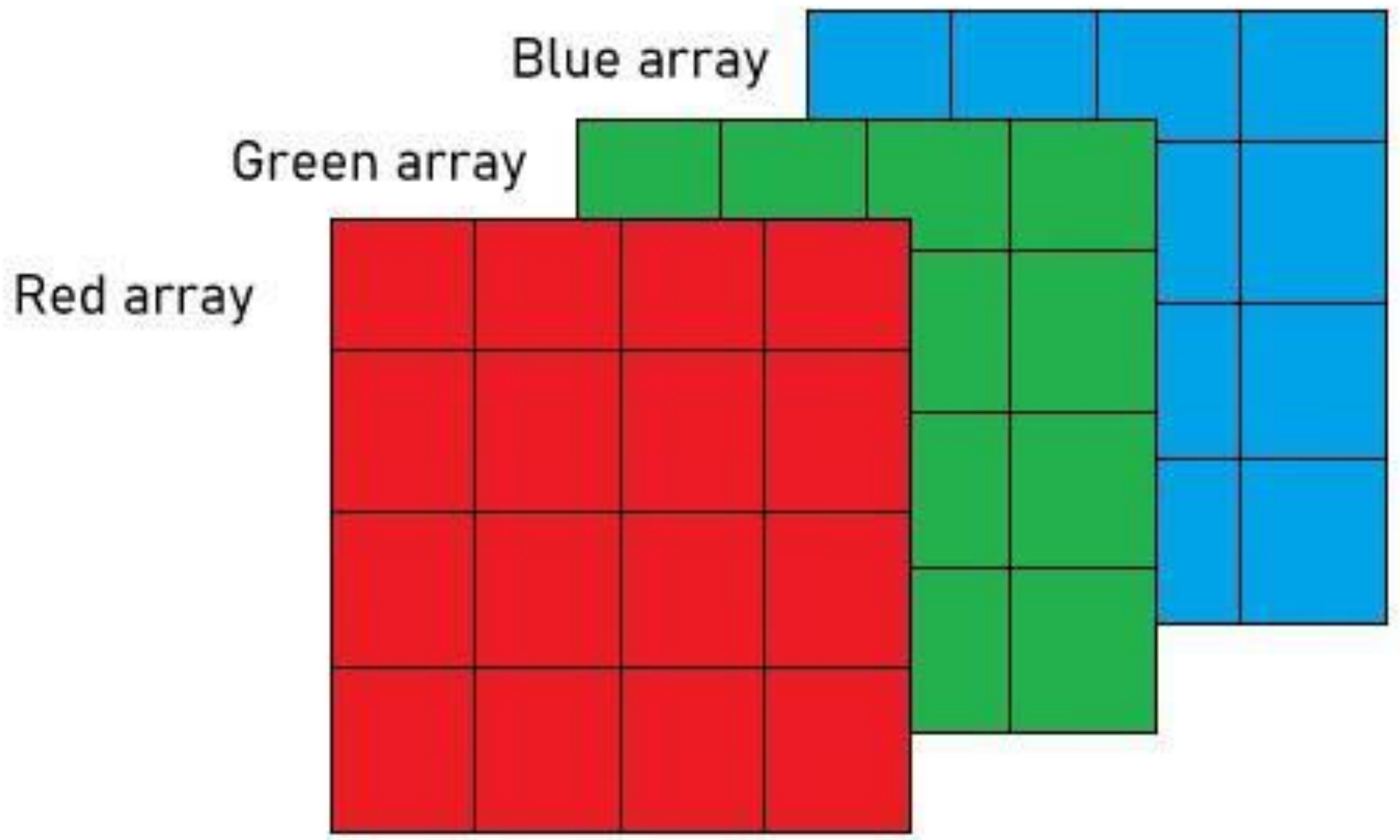
Noise Reduction



by
Aditya Raj
2311013

RGB Structure of Images

- An Image is made up of pixels, each with an Intensity value represented by numbers.
- It can be represented by an m x n matrix, where each element corresponds to a pixel.
- For example in an 8-bit image, pixel value ranges from 0 to 255.
- In a color image, these values are distributed across different channels of Red, Green and Blue as shown in the figure.



Arrrays stacked over each other to form a Digital Image.

Grayscale Image Formation

- A grayscale image contains only intensity information, with no separate color channel.
- Each pixel is represented by a single value indicating brightness.
- For 8-bit grayscale images, pixel values range from 0 (black) to 255 (white)
- Grayscale images are often created by combining RGB channels using a weighted formula to match human visual perception.
- The luminance formula used by python OpenCV BT.601 weighted grayscale conversion: looks like –

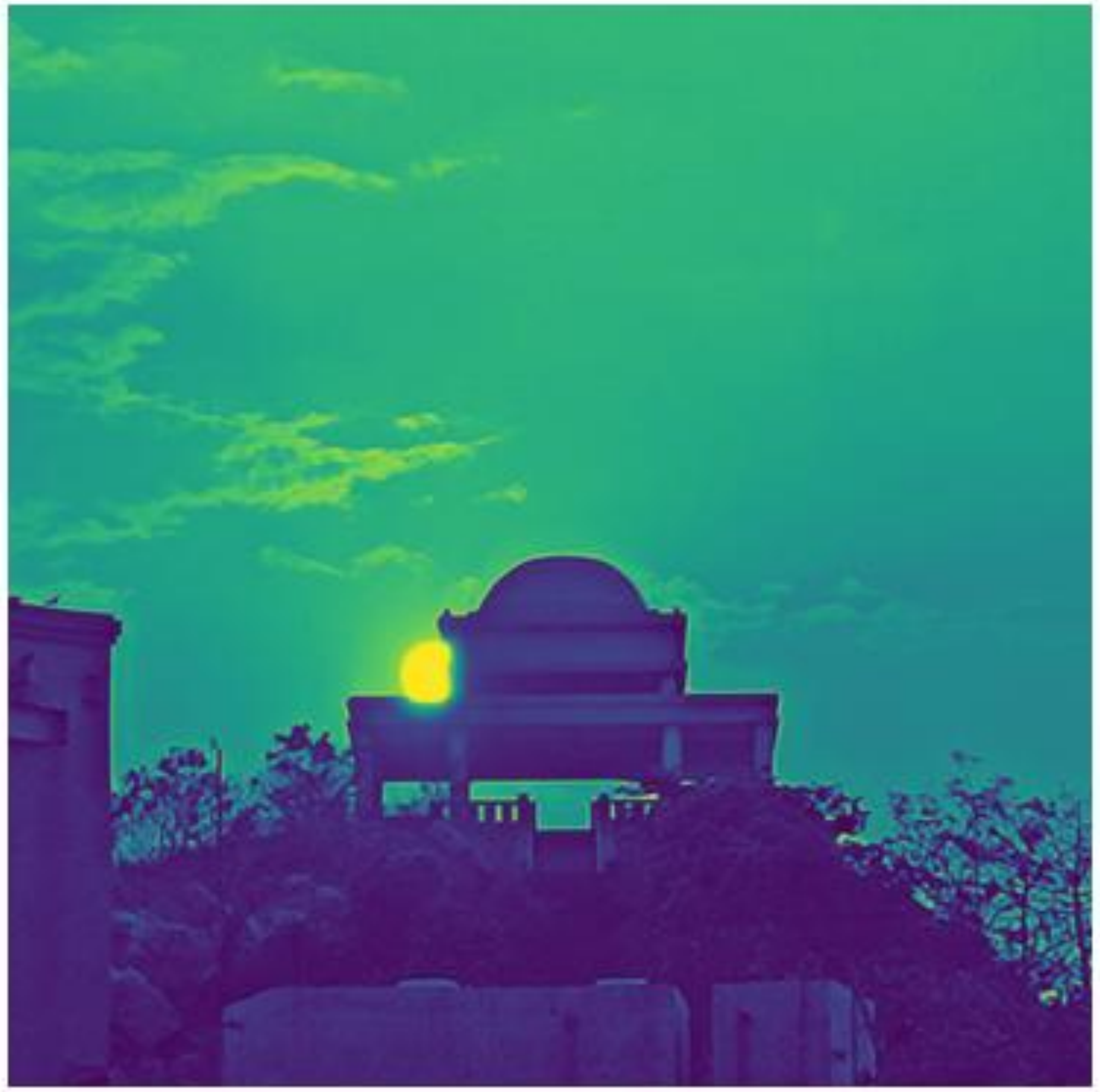
$$GRAY = 0.2996 * R + 0.587 * G + 0.114 * B$$

ah pixel is represented by a single value indicating brightness.

Example Image -

```
image_path = r"test_img1.jpg"
noisy_image =
cv2.imread(image_path,
cv2.IMREAD_GRAYSCALE)

plt.imshow(noisy_image)
plt.show()
```



Why SVD?

- SVD is the foundation of Recommender Systems that are at the heart of huge companies like Google, YouTube, Amazon, Facebook, Netflix...

Key Benefits in Decomposition

Universal Application

Singular Value Decomposition (SVD) operates effectively on all real matrices including non-square and non symmetric ones, making it a versatile tool for various applications in image processing and data analysis across multiple fields.

Optimal Low-Rank

SVD achieves optimal low-rank approximations by retaining the essential structure of the matrix, allowing for efficient data compression while minimizing the loss of important features and information.

Stability

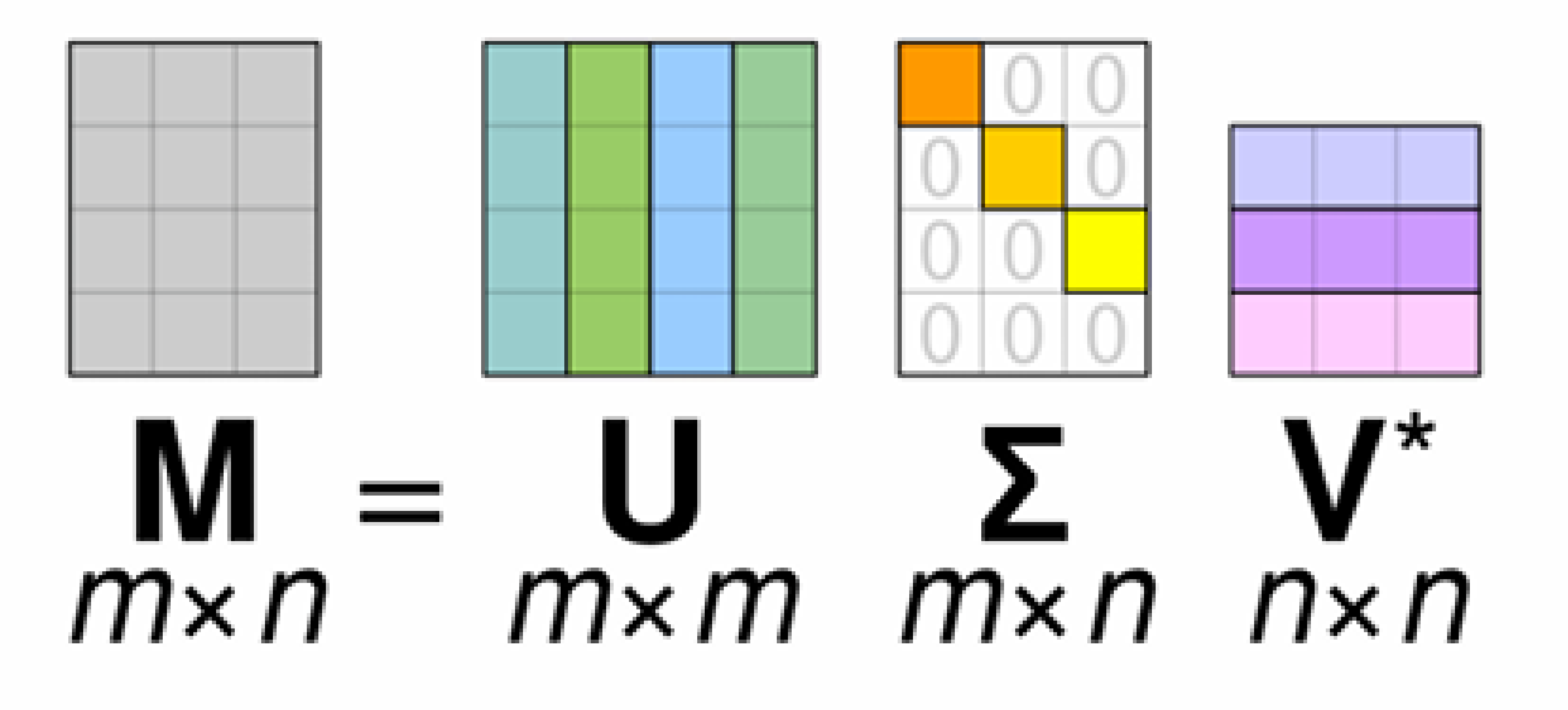
SVD is known for its numerical stability, providing consistent and reliable results under various conditions, which is crucial when handling real-world data that may exhibit noise and other discrepancies.

SVD Equation

he fundamental equation of Singular Value Decomposition is

$$M_{(m \times n)} = U_{(m \times m)} * \Sigma_{(m \times n)} * V^T_{(n \times n)}$$

where U and V are orthogonal matrices, and Σ is a diagonal matrix containing singular values that represent information strength.



Understanding U, Σ, and V^T

Components of SVD

Column Basis

The matrix U contains the **orthonormal basis** for the column space of A. Each column corresponds to a singular eigenvector of AA^T matrix.

Singular Values

$$Av_i = \sigma_i u_i$$

By computing the singular values, we can quantify the strength of the information contained in the matrix, allowing for effective low-rank approximations and efficient storage solutions.

Row Basis

The transpose matrix V^T provides the **orthonormal basis** for the row space of A, with each row is eigenvector of $A^T A$ matrix.

Numerical Example

$$A = \begin{bmatrix} 3 & 1 & 1 & 0 \\ 0 & 2 & 0 & 1 \\ 1 & 0 & 2 & 1 \end{bmatrix} = U \Sigma V^T$$

$$U \approx \begin{bmatrix} 0.80 & -0.30 & 0.52 \\ 0.25 & 0.90 & 0.35 \\ 0.54 & -0.31 & -0.78 \end{bmatrix}, \quad \Sigma \approx \begin{bmatrix} 4.00 & 0 & 0 & 0 \\ 0 & 2.45 & 0 & 0 \\ 0 & 0 & 1.59 & 0 \end{bmatrix},$$

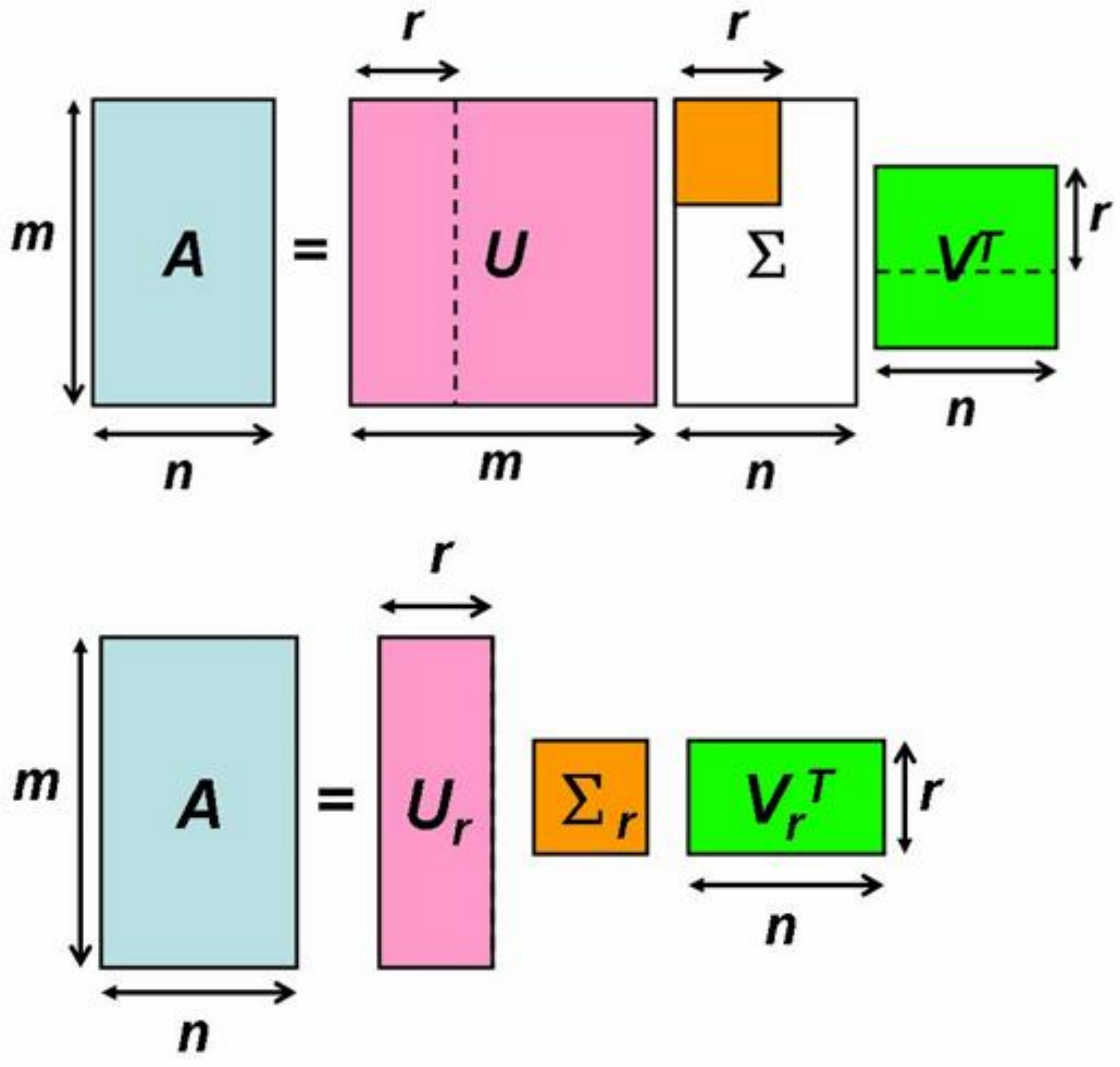
$$V^T \approx \begin{bmatrix} 0.67 & 0.21 & 0.67 & 0.25 \\ -0.48 & -0.39 & 0.08 & 0.78 \\ 0.42 & -0.82 & 0.39 & -0.15 \\ 0.37 & 0.36 & -0.62 & 0.57 \end{bmatrix}.$$

Low-Rank Approximation

- SVD breaks a matrix into components sorted by importance.
- These singular values are arranged diagonally in descending order.
- All entries in the diagonal of Σ atisfy-

$$\sigma_1 \geq \sigma_2 \geq \sigma_3 \geq \cdots \geq 0$$

- The first few singular values capture most of the structure of the image.

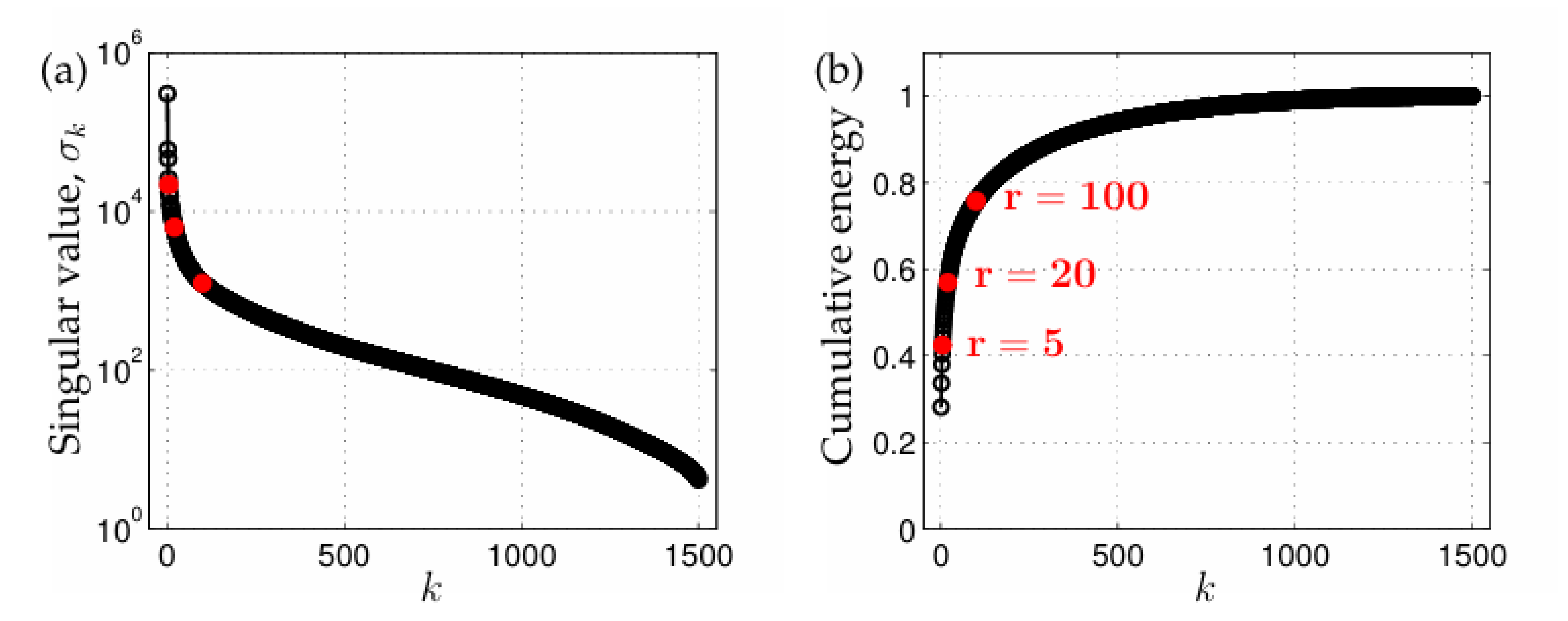


- This property of the Σ matrix allows us take only first k σ_{k} values and reconstruct the matrix.

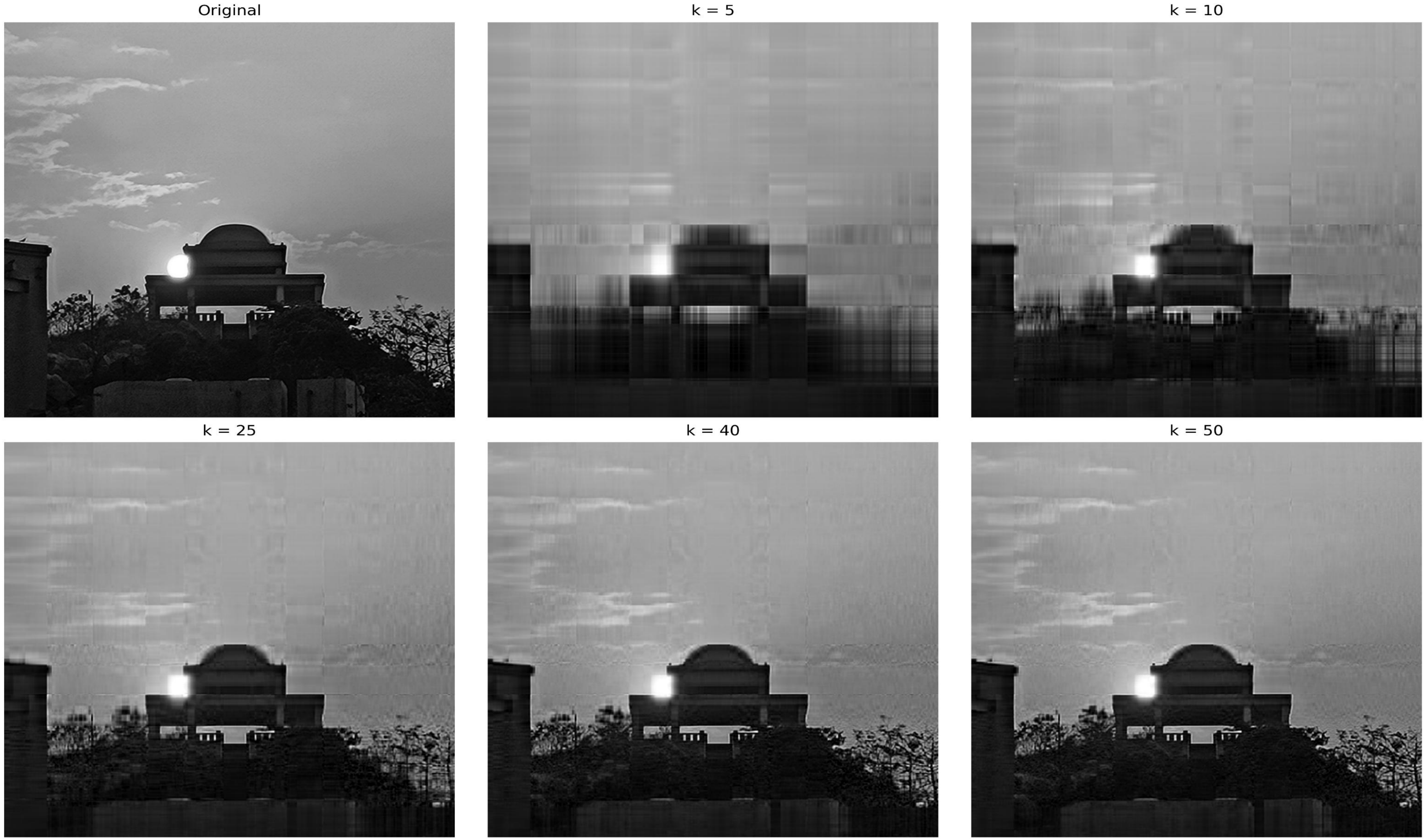
The formula with $\Sigma_{(\mathrm{k}\times\mathrm{k})}$ looks like :-

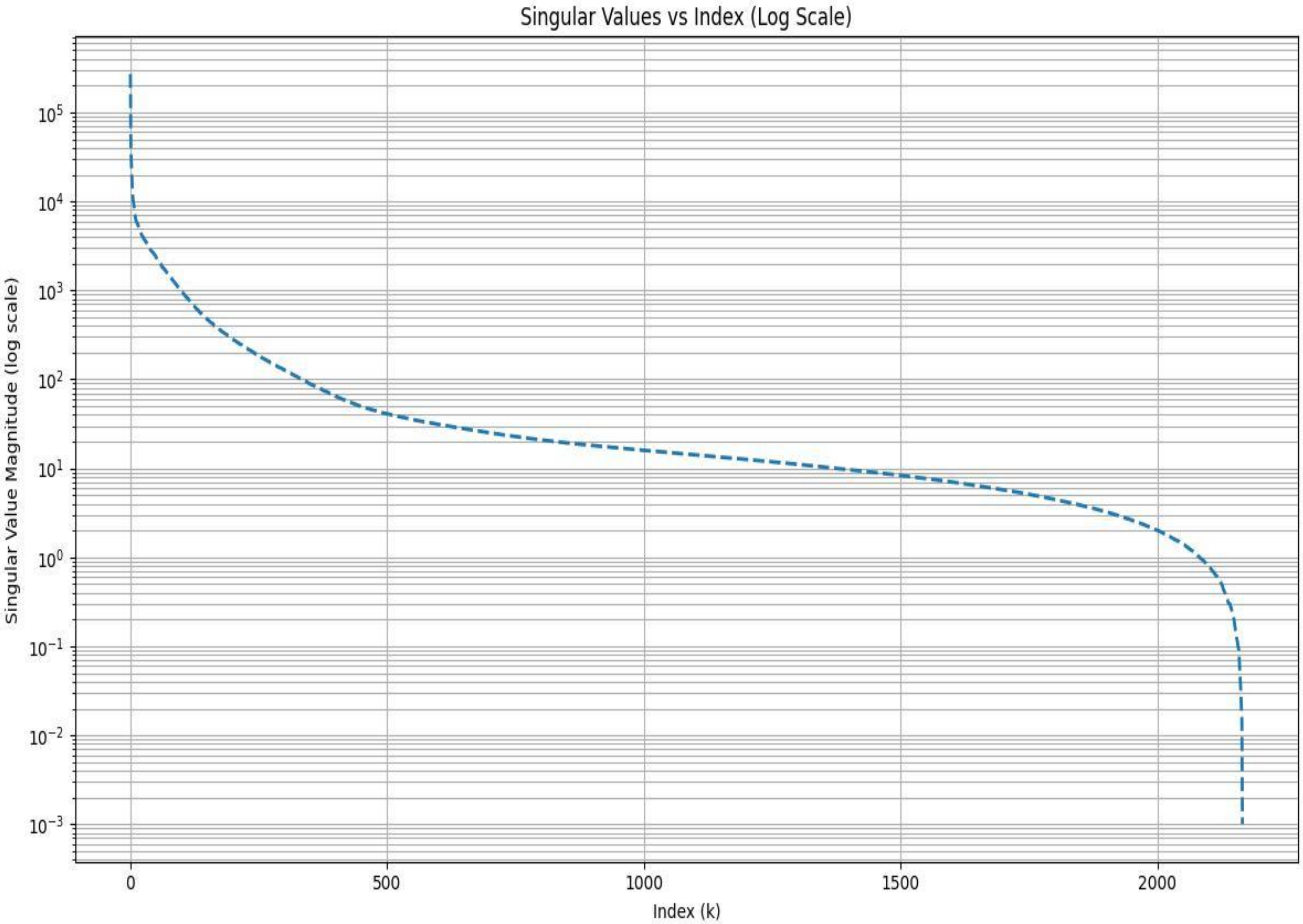
$$\mathrm{M}_{(m\times n)} = U_{(m\times \mathrm{k})} * \Sigma_{(\mathrm{k}\times\mathrm{k})} * V^{T}_{(\mathrm{k}\times n)}$$

We can plot singular values vs k to showcase the drop in value and cumulative energy vs k to show how much of the total information (variance) in the image is captured by the first k singular values.



Reconstructed image for different values of k -



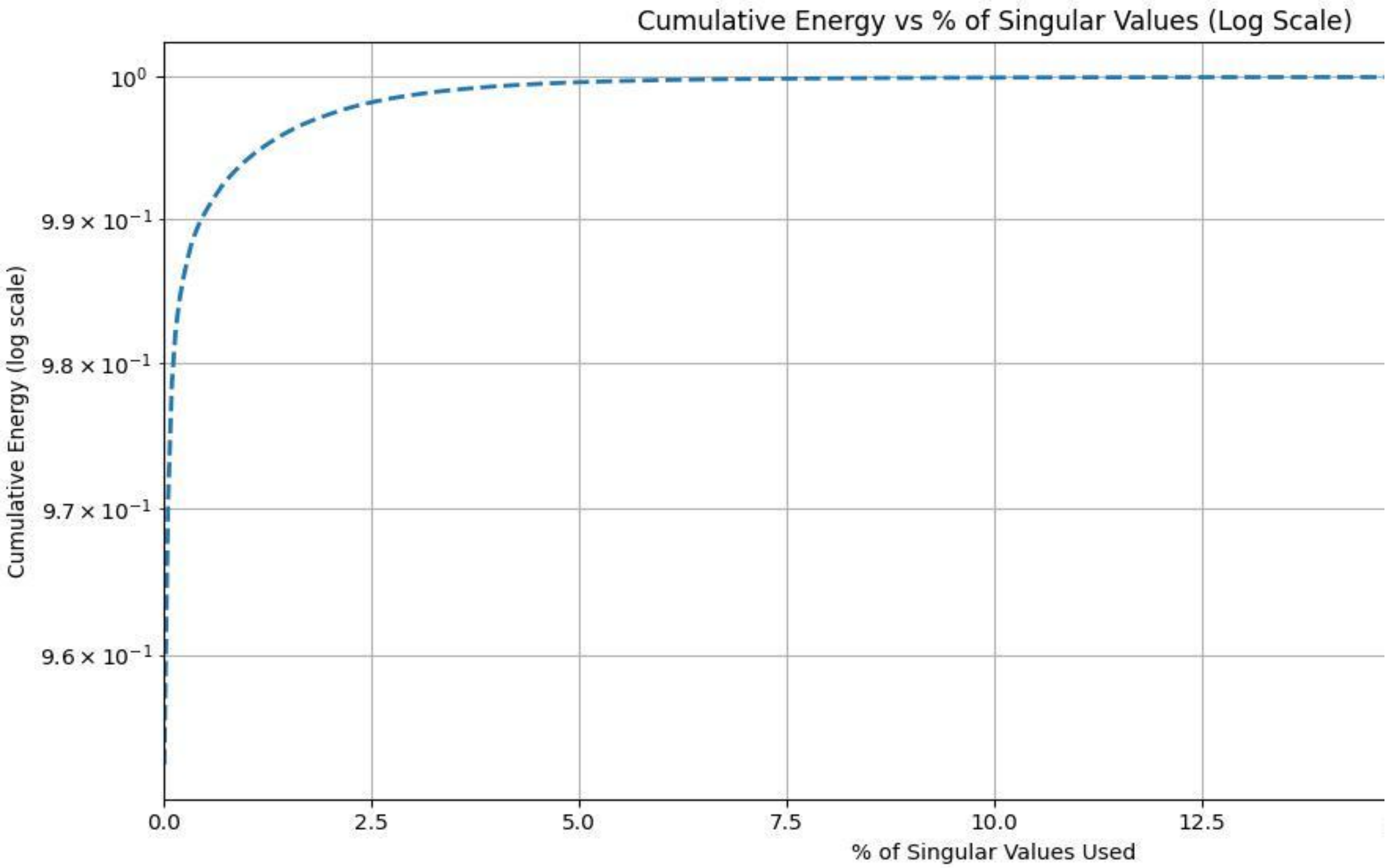


Singular Values vs Index (Log Scale)
Singular Value Magnitude (log scale)
10^5
10^4
10^3
10^2
10^1
10^0
10^{-1}
10^{-2}
10^{-3}
0
500
1000
1500
2000
Index (k)

Cumulative Energy can be calculated by the formula -

$$E_k = \frac{\sum_{i=1}^{k} \sigma_i^2}{\sum_{i=1}^{r} \sigma_i^2}$$

The figure shows that most of the energy (approx 80%) are captured within the first 2% of the total rank (k) of the image.



To measure the quality of the reconstructed images, we evaluate two widely-used image quality metrics:

- PSNR — Peak Signal-to-Noise Ratio

Indicates how close the reconstructed image is to the original in terms of pixel intensity.
Higher PSNR = better numerical accuracy.

$$\mathrm{PSNR} = 10\,\log_{10}\left(\frac{MAX^2}{\mathrm{MSE}}\right)$$

Where,
MAX = The maximum possible pixel intensity value of the image. For an 8-bit image: 255
MSE (Mean squared error) = The average squared difference between corresponding pixels of the original and reconstructed image.

- SSIM — Structural Similarity Index Measure

Evaluates how well the reconstruction preserves the structural and perceptual features of the image. Ranges from 0 to 1
Higher SSIM = better preservation of texture and structure

Example – let's take a 8x9 matrix

$$A = \begin{bmatrix} 5 & 7 & 6 & 8 & 32 & 7 & 5 & 6 & 4 \\ 6 & 8 & 7 & 5 & 6 & 7 & 34 & 6 & 7 \\ 7 & 5 & 4 & 33 & 5 & 6 & 4 & 5 & 7 \\ 6 & 7 & 8 & 6 & 5 & 35 & 7 & 6 & 5 \\ 4 & 5 & 6 & 5 & 4 & 6 & 5 & 31 & 4 \\ 5 & 6 & 5 & 7 & 8 & 6 & 5 & 4 & 33 \\ 7 & 8 & 7 & 6 & 7 & 6 & 7 & 6 & 7 \\ 6 & 4 & 5 & 7 & 6 & 5 & 4 & 5 & 6 \end{bmatrix}$$

For different rank k

k	PSNR	SSIM
1	12.5895	0.1558
2	13.7979	0.4996
3	15.2114	0.7034
5	20.1070	0.9128
7	44.5547	0.9998

$$A_{k=1} = \begin{bmatrix} 6.0904 & 6.7781 & 6.4506 & 10.4449 & 10.0335 & 11.2552 & 10.1609 & 8.9313 & 10.0192 \\ 6.5145 & 7.2501 & 6.8997 & 11.1722 & 10.7321 & 12.0388 & 10.8684 & 9.5532 & 10.7168 \\ 5.8912 & 6.5564 & 6.2396 & 10.1033 & 9.7054 & 10.8871 & 9.8286 & 8.6392 & 9.6915 \\ 6.6850 & 7.4398 & 7.0803 & 11.4646 & 11.0130 & 12.3540 & 11.1529 & 9.8033 & 10.9973 \\ 5.1096 & 5.6866 & 5.4118 & 8.7629 & 8.4177 & 9.4426 & 8.5246 & 7.4930 & 8.4057 \\ 6.0655 & 6.7504 & 6.4242 & 10.4023 & 9.9925 & 11.2092 & 10.1194 & 8.8948 & 9.9783 \\ 4.3954 & 4.8917 & 4.6553 & 7.5380 & 7.2412 & 8.1228 & 7.3331 & 6.4457 & 7.2308 \\ 3.5252 & 3.9232 & 3.7336 & 6.0456 & 5.8074 & 6.5145 & 5.8812 & 5.1695 & 5.7991 \end{bmatrix}$$

$$A_{k=3} = \begin{bmatrix} 6.3011 & 6.4190 & 6.0153 & 14.7473 & 11.6135 & 10.3029 & 4.9815 & 8.5928 & 10.9707 \\ 5.9288 & 8.4675 & 7.7742 & -1.6343 & 6.5945 & 9.6808 & 29.1056 & 7.5263 & 13.1407 \\ 6.4300 & 5.6701 & 5.0778 & 20.9818 & 13.7827 & 7.7246 & -2.8572 & 7.3421 & 12.8617 \\ 6.4128 & 7.4315 & 8.3633 & 7.7335 & 8.4263 & 24.3455 & 9.5795 & 16.6223 & -1.1315 \\ 4.9849 & 5.6245 & 6.0882 & 7.2794 & 7.1657 & 16.2617 & 6.7848 & 11.4030 & 1.5067 \\ 6.3339 & 6.7487 & 5.1738 & 14.1207 & 12.5325 & -0.3930 & 11.4980 & 2.3027 & 21.7130 \\ 4.3816 & 4.9438 & 4.6400 & 7.1458 & 7.1709 & 7.5320 & 8.1732 & 6.0805 & 7.8300 \\ 3.5949 & 3.8184 & 3.5679 & 7.4150 & 6.3467 & 5.8776 & 4.4138 & 4.8666 & 6.4400 \end{bmatrix}$$

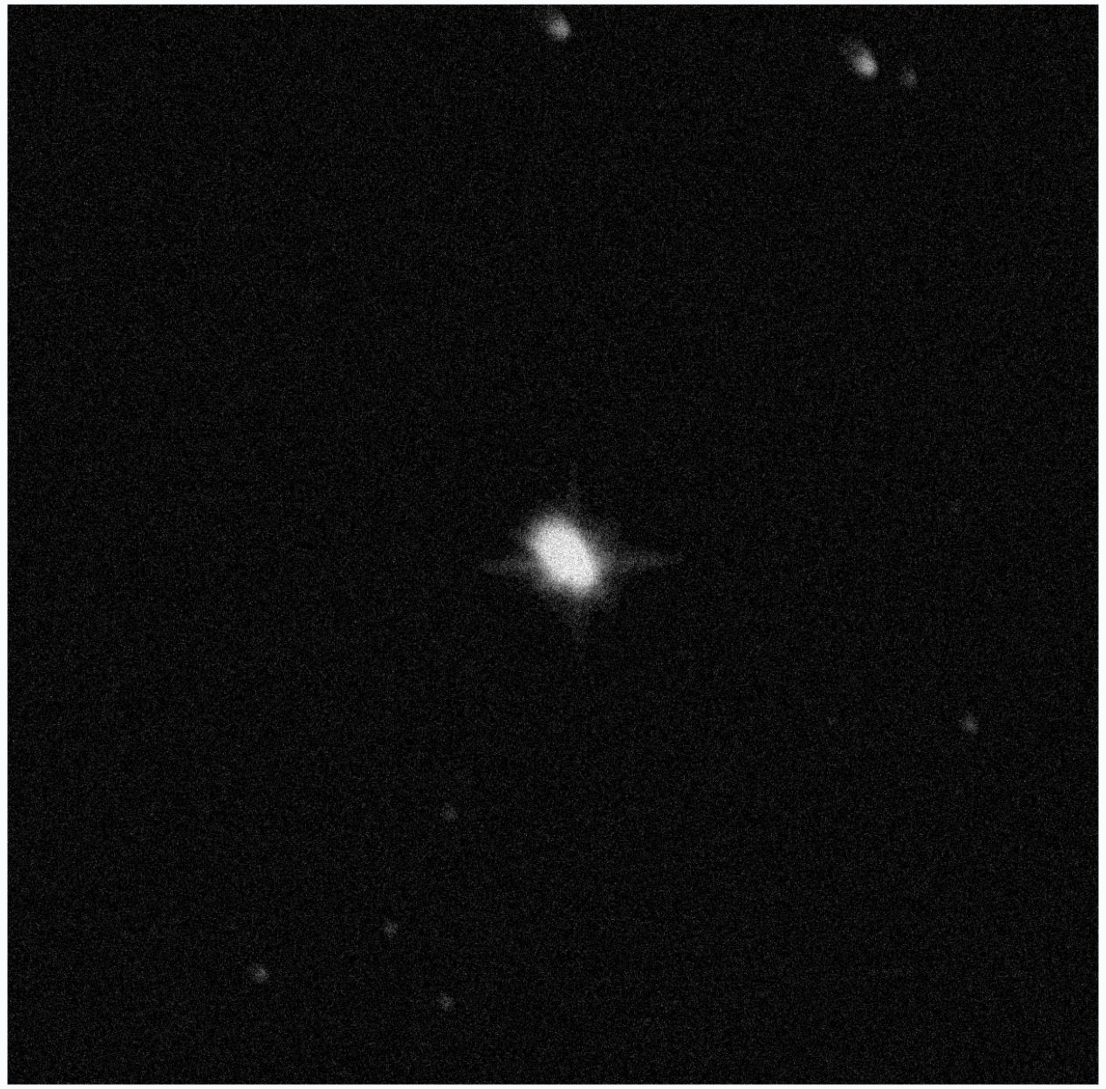
$$A_{k=7} = \begin{bmatrix} 5.0291 & 6.9589 & 6.0059 & 7.9978 & 32.0028 & 7.0009 & 5.0025 & 6.0008 & 4.0011 \\ 5.9770 & 8.0322 & 6.9953 & 5.0017 & 5.9977 & 6.9992 & 33.9979 & 5.9993 & 6.9990 \\ 7.0720 & 4.8987 & 4.0146 & 32.9946 & 5.0070 & 6.0023 & 4.0063 & 5.0021 & 7.0028 \\ 6.0135 & 6.9809 & 8.0027 & 5.9989 & 5.0013 & 35.0004 & 7.0011 & 6.0004 & 5.0005 \\ 4.0212 & 4.9700 & 6.0043 & 4.9984 & 4.0020 & 6.0006 & 5.0018 & 31.0006 & 4.0008 \\ 5.0296 & 5.9583 & 5.0060 & 6.9977 & 8.0028 & 6.0009 & 5.0026 & 4.0008 & 33.0011 \\ 7.4327 & 7.3919 & 7.0878 & 5.9676 & 7.0422 & 6.0140 & 7.0380 & 6.0130 & 7.0172 \\ 5.2243 & 5.0898 & 4.8426 & 7.0580 & 5.9243 & 4.9748 & 3.9317 & 4.9766 & 5.9691 \end{bmatrix}$$

FOR NOISE REDUCTION -

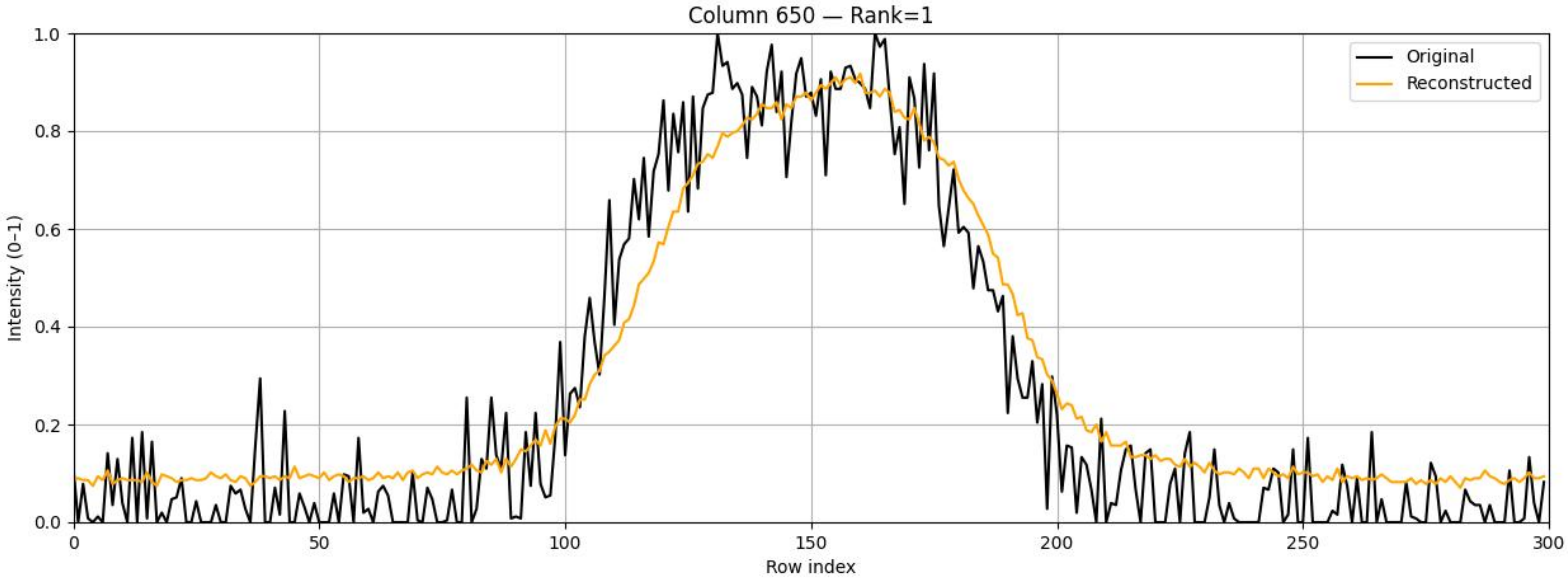
Image of a star in M46 open cluster

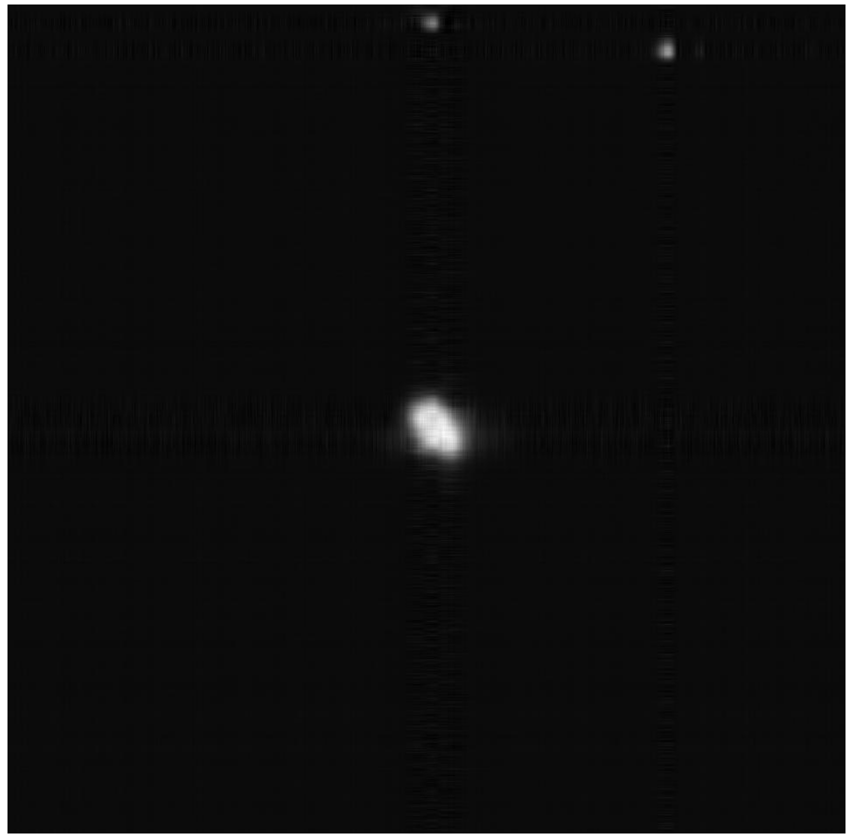
Added gaussian noise with sigma = 30

Comparing noisy image with the clean one
PSNR = 21.67 dB
SSIM = 0.0201



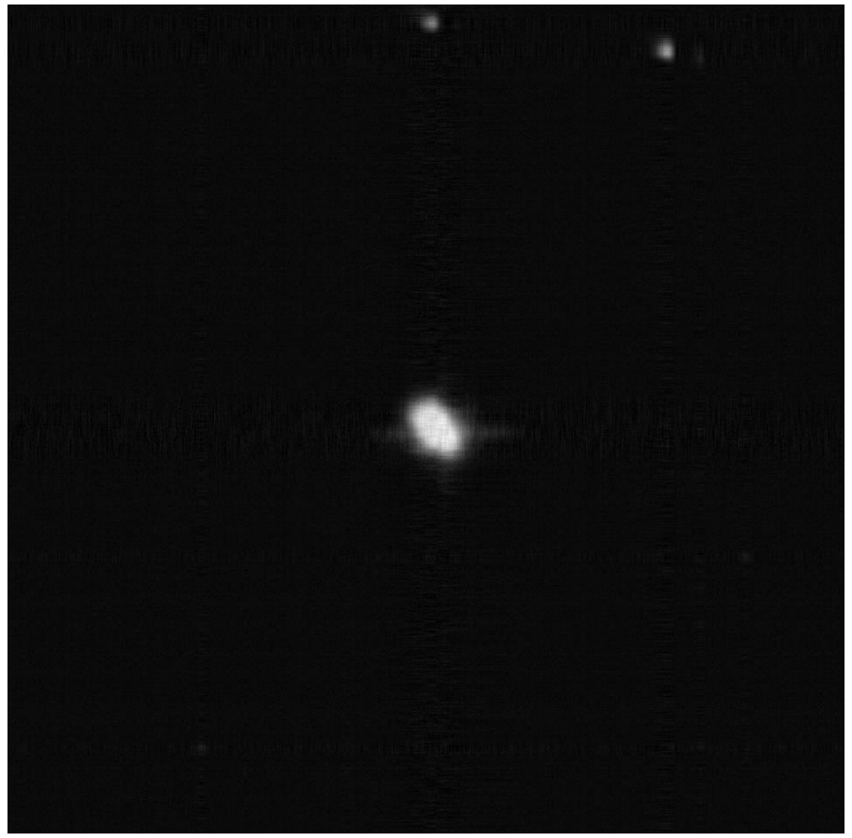
GIF plot of row index vs normalised intensity of a randomly chosen column





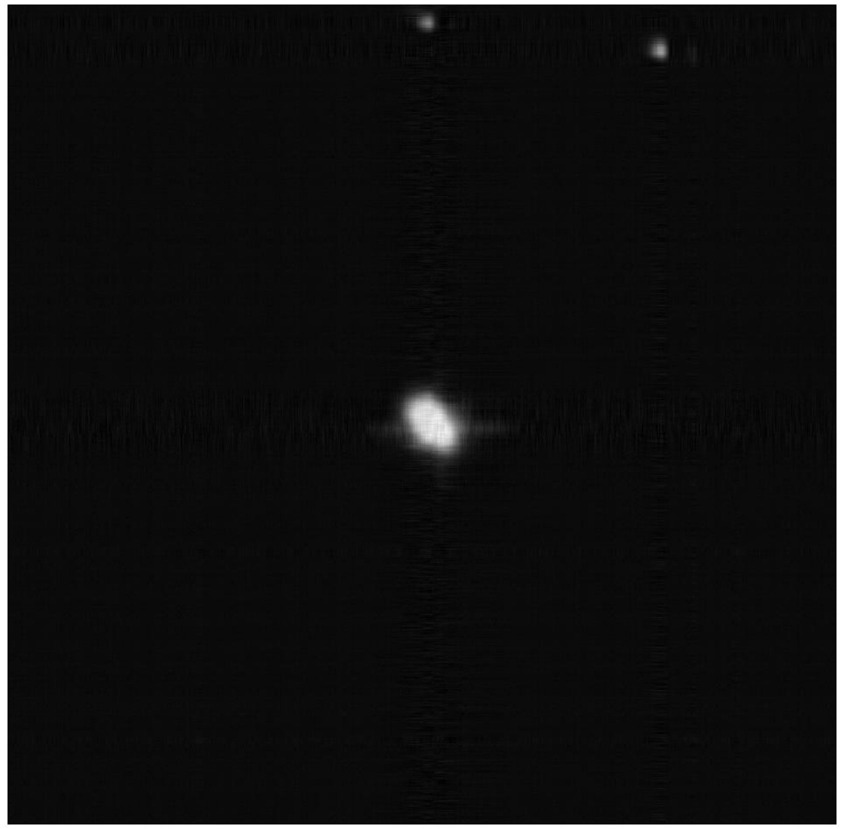
k = 5, PSNR = 27.05, SSIM = 0.0909

For k = 5, the image retains minimal detail.



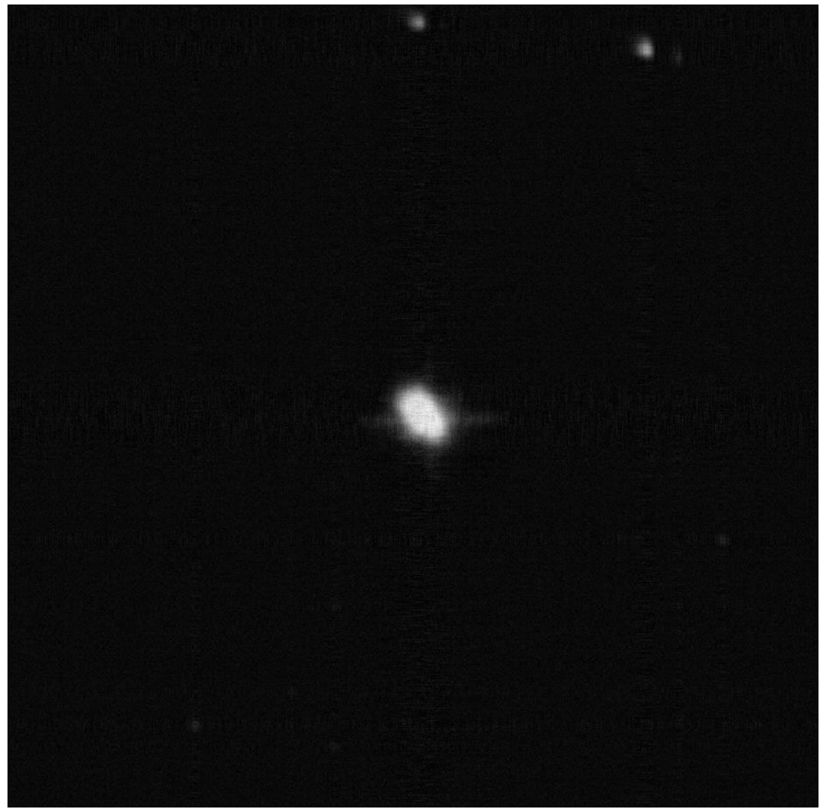
k = 20, PSNR = 26.64, SSIM = 0.0768

At k = 20, significant structure remains visible.



k = 10, PSNR = 26.92, SSIM = 0.0858

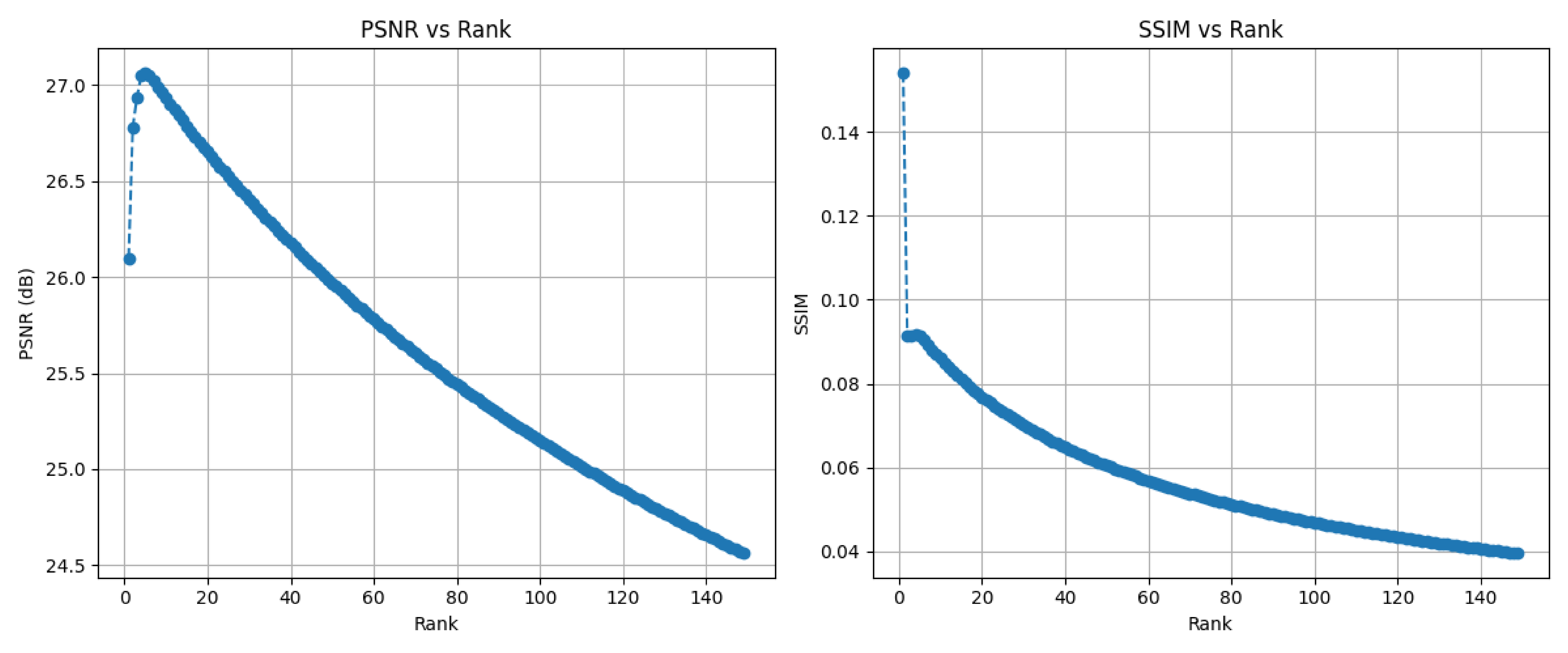
The k = 10 image shows improved clarity and structure.



k = 50, PSNR = 25.96, SSIM = 0.0604

At k = 50, nearly all original detail is preserved.

Reconstruction Graphs



THANK YOU.

References -

- Steven L. Brunton & J. Nathan Kutz, *Data-Driven Science and Engineering*.
- Jiaxing Shen, “Singular Value Decomposition (SVD) and Its Applications,” Tutorial Presentation, April 2022.
- Andrew Gibiansky, “Cool Linear Algebra: Singular Value Decomposition.”
- GeeksforGeeks, “Singular Value Decomposition (SVD) in Machine Learning.”
- GeeksforGeeks, “Python — Peak Signal-to-Noise Ratio (PSNR).”
- Jethro Jens Norbert Simatupang, “Noise Reduction in Satellite Imagery Using Singular Value Decomposition,” Institut Teknologi Bandung (ITB).