UC San Diego

Image Compression
with
Singular Value Decomposition

Group 1
Yashi Shukla, Shreya Reddy Pakala, Sagarika Sardesai, Sai Kaushik Soma
Dec 6, 2022

Sec 1: Introduction

Recap and Agenda

RECAP: In our midterm presentation, we had discussed -

- ❑ The different types of image compression (lossy vs lossless).
- ❑ The need for image compression.
- ❑ Our approach to solve Image compression using Numerical Linear Algebra
- ❑ The comparison between our approach and the state of the art.
- ❑ Information about the dataset, tools, logistics involved.

AGENDA: In this final presentation, we will be covering -

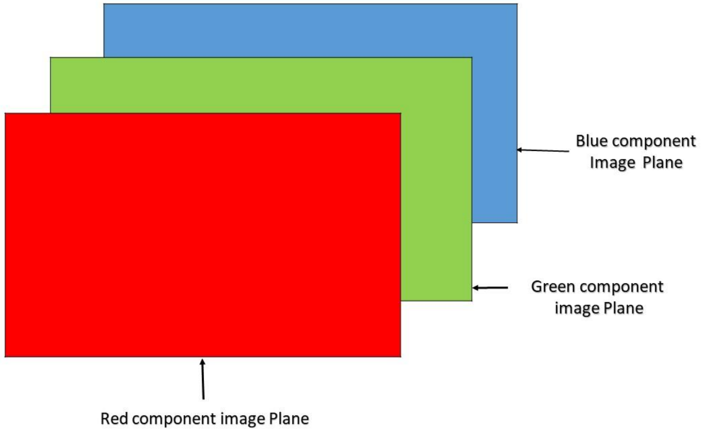
- ❑ The overview of image compression using SVD
- ❑ Our Numerical Linear Algebra approach for SVD and the results of the experiment.
- ❑ SOTA and comparison with SVD
- ❑ Conclusions drawn from experiment

Overview of Image Compression with SVD

Definition	Singular Value Decomposition or SVD refactors a digital image into three matrices. ‘K’ Singular values refactor the image and at the end of this process, image is represented with smaller set of values.
Goal	Reduce psycho visually redundant data while preserving the important features which describe the original image
Drawbacks	Image degradation for excessive compression

Why do people care?	Reduced transmission and storage costs for smaller size images.
Applications	Lowering image dimensions, watermarking an image, computing weighted least squares, and optimal prediction, principal component analysis

Why compress images?



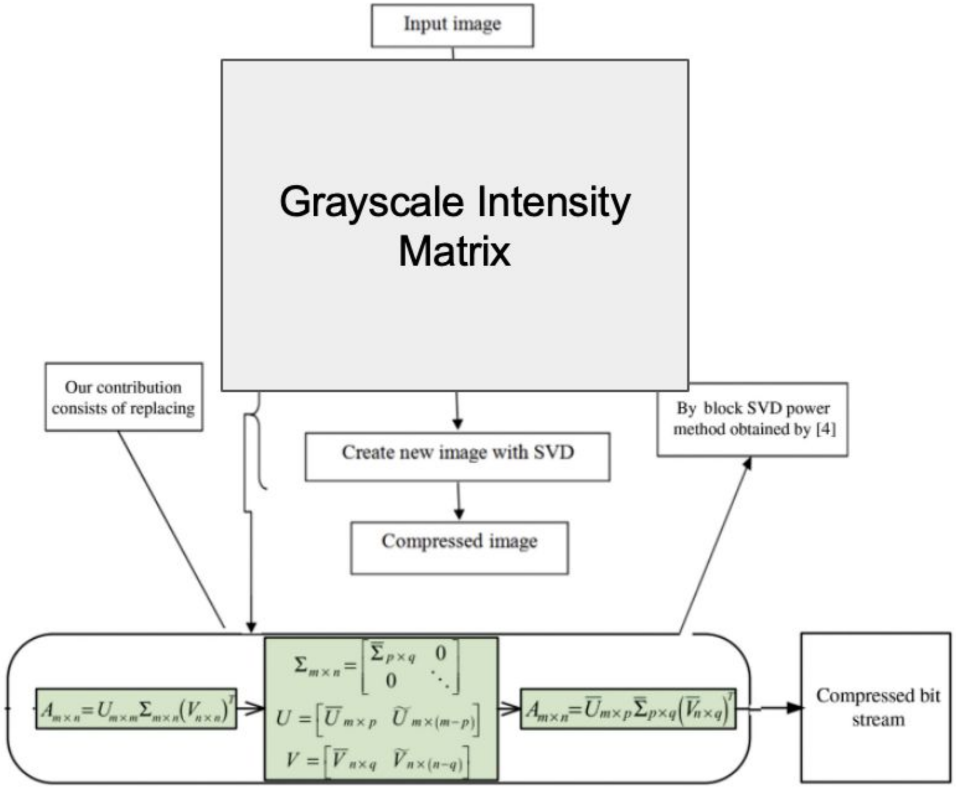
[RGB image is a three layered image]

A mxn grayscale image - 1 number per pixel [8 bits (1 byte)]

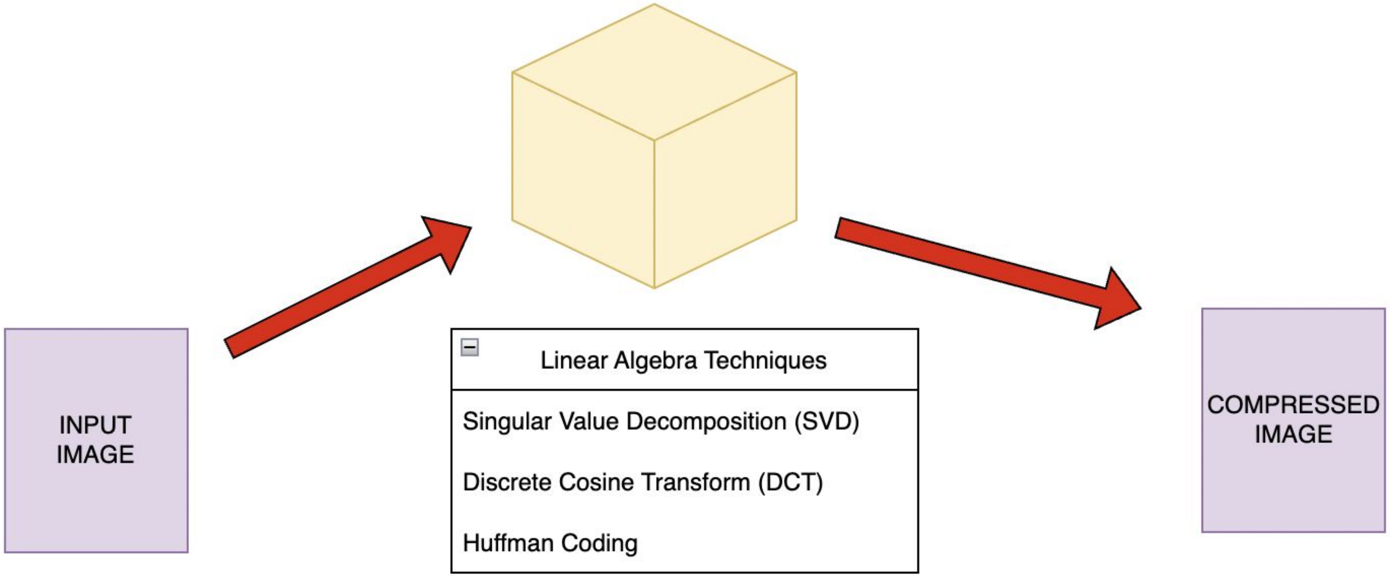
The same mxn image but in colour - 3 numbers (rgb) per pixel [24 bits (3 bytes)]

Sec 2: Problem Formulation

Problem Formulation and Output



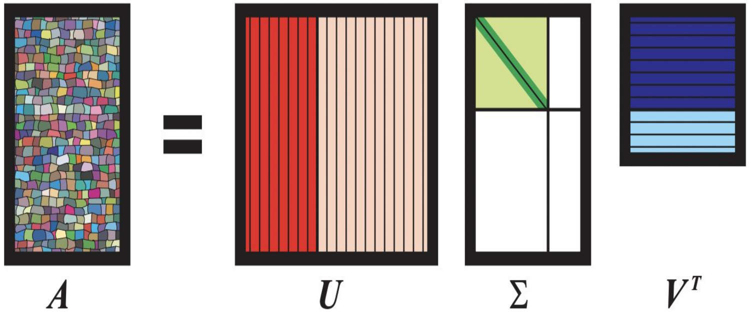
Relation to Numerical Linear Algebra



Approach of Numerical Linear Algebra (NLA)

Singular value decomposition is a factorization of a real or complex matrix. It generalizes the eigendecomposition of a square normal matrix with an orthonormal eigenbasis to any m x n matrix

$$\underset{m \times n}{A} = \underset{m \times m}{U} \; \underset{m \times n}{\Sigma} \; \underset{n \times n}{V^T}$$



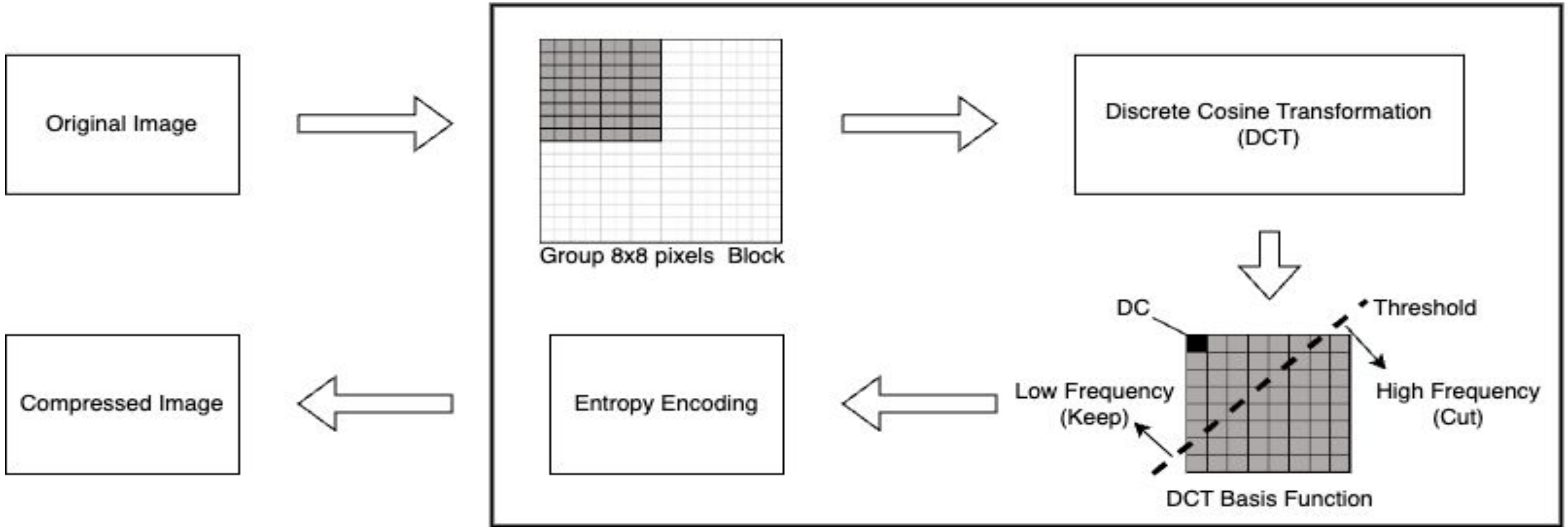
A : m × n matrix with rank r
U : orthogonal m x m matrix
Σ : m x n matrix with singular entries
V : orthogonal n x n matrix

- Calculating the SVD consists of finding the eigenvalues and eigenvectors of AA^T and A^TA.
- The eigenvectors of A^TA make up the columns of V
- The eigenvectors of AA^T make up the columns of U.
- Also, the singular values in **Σ** are square roots of eigenvalues from AA^T or A^TA.

Sec 3: State of the Art (SOTA)

Discrete Cosine Transform (DCT)

- The DCT technique works by segmenting the image into blocks, each block makes its way through each processing step and yields a compressed output.
- In principle, the DCT introduces no loss to the source image samples; it transforms them to a domain in which they can be more efficiently encoded.



Why DCT is a better approach

- SVD and DCT can easily compress an image to 30% of its original size with almost no visual difference.
- However, DCT appears to be more effective because it takes its losses throughout the entire photo evenly, while SVD removes entire vectors of data at a time.
- DCT’s block based approach saves a lot of computation memory compared to SVD which is performed over an entire image.
- DCT provides better quality with higher compression ratios than SVD.

Sec 4: Experiment

Experimental Setup

Dataset

The Images 4k dataset - performed the compression experiments on 100 jpg images taken from Images 4k

Libraries & Tools

- Python - Programming language
- Matplotlib, IPyWidgets - Visualization, Interactive environment
- Numpy, Numpy.linalg, Scipy, Skimage, Opencv, Math - For loading the images, performing SVD, DCT and calculating the performance metrics

Logistics

- Colab - Jupyter Notebook Environment
- Notion - Report Template
- GitHub - Version Control

Performance Metrics

Compression Ratio (CR)	❏ Compression Ratio is the ratio between the sizes of original image and compressed image. ❏ Higher the compression ratio, smaller the size but lower the quality of the compressed image.
Mean Squared Error (MSE)	❏ Mean Squared Error is the average of squared errors where error is the difference between each corresponding pixel in the actual and compressed image. ❏ Lower the MSE, lower the error, closer the compressed image is to the original image.
Peak Signal to Noise Ratio (PSNR)	❏ PSNR measures the ratio between maximum pixel intensity and the power of distorting noise (error produced by compression) affecting the quality of its representation. ❏ Higher the PSNR value, better the quality of compressed image.
Structural Similarity Index Measure (SSIM)	❏ In this method, the structural similarity index calculation is based on three aspects namely: image luminance, contrast and structure. The index measures similarity between two images. ❏ Closer the SSIM value is to 1, similar the compressed image is to original image.

Sec 5: Results

Image Compression with SVD - Results

Hyperparameter: K = rank of Σ matrix with singular values as diagonal entries

K value chosen here to perform SVD = 30

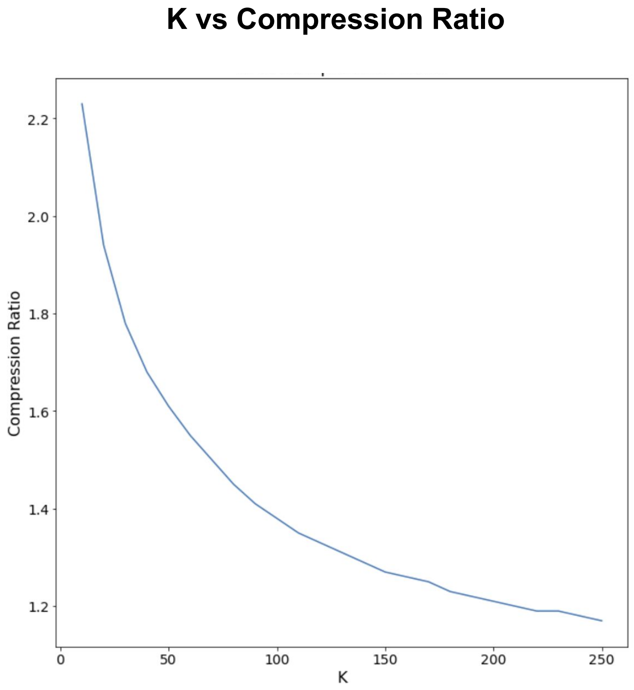
Original Image
Size = 2384 KB

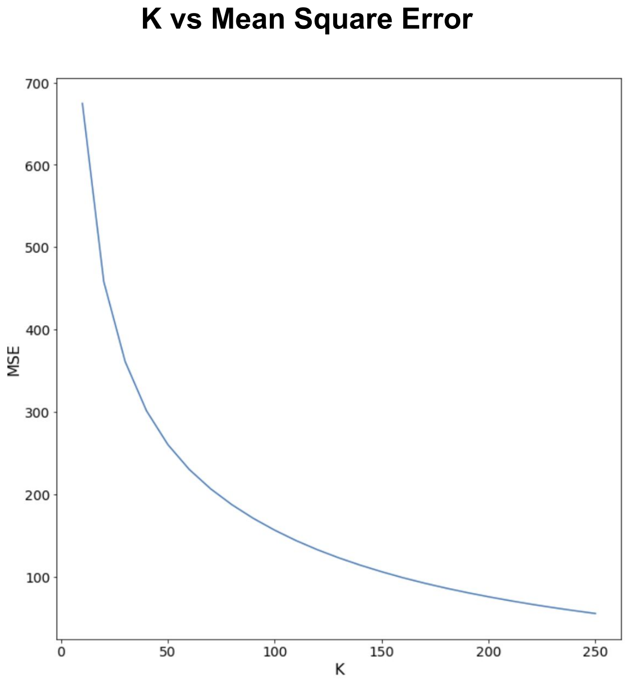
Compressed Image
Size = 1342 KB

Performance Metrics for this image

CR	1.78
MSE	361.31
PSNR	22.5
SSIM	0.68

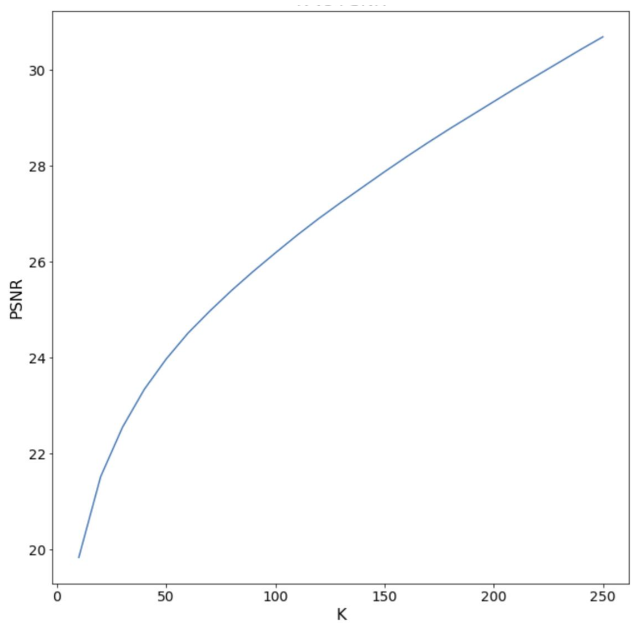
Performance Metrics for different K values





Performance Metrics for different K values (contd.)

K vs PSNR



K vs SSIM

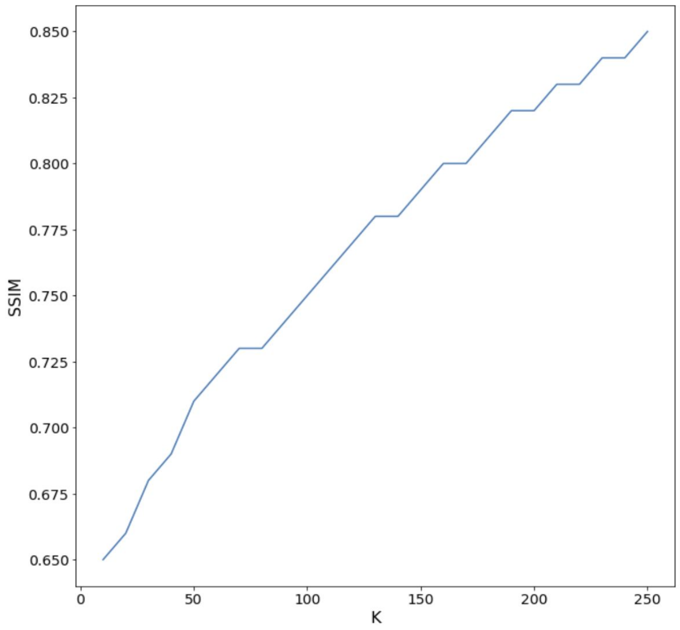


Image Compression with DCT - Results

Hyperparameter: Threshold = discarding those values of DCT matrix that fall below this threshold value

Threshold value chosen here to perform DCT = 0.0050

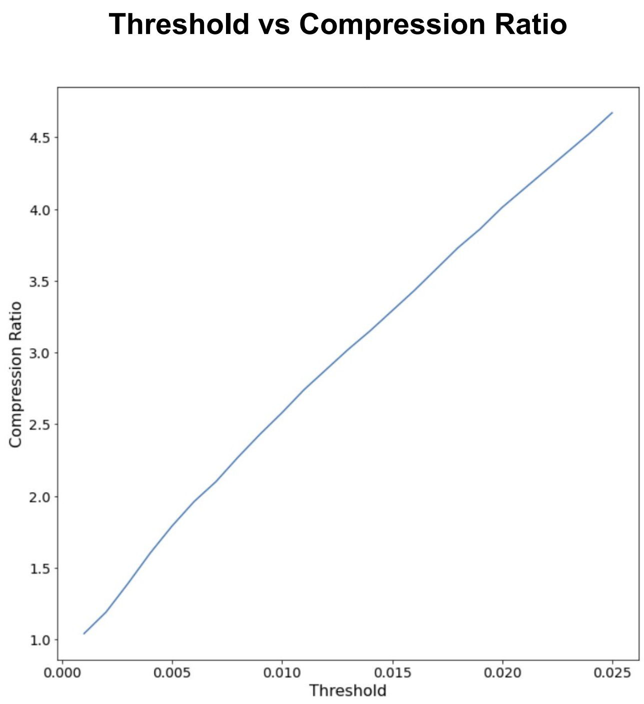
Original Image
Size = 2384 KB

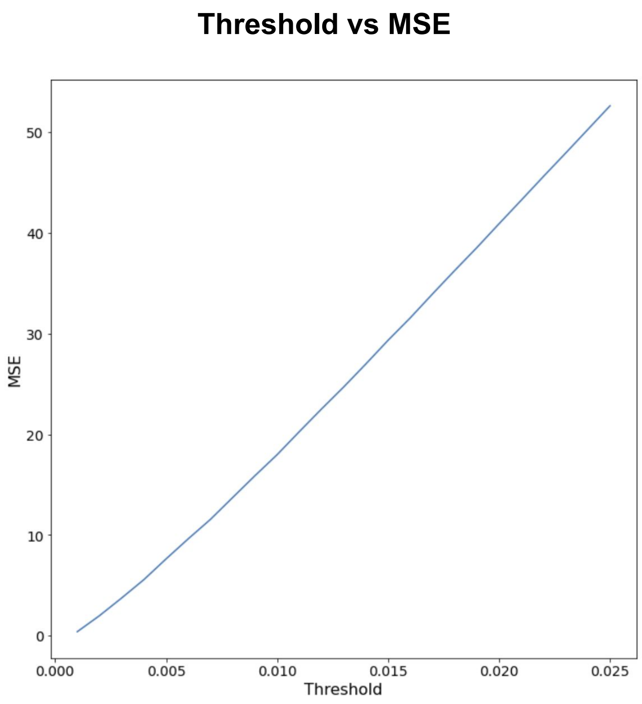
Compressed Image
Size = 1329 KB

Performance Metrics
for this image

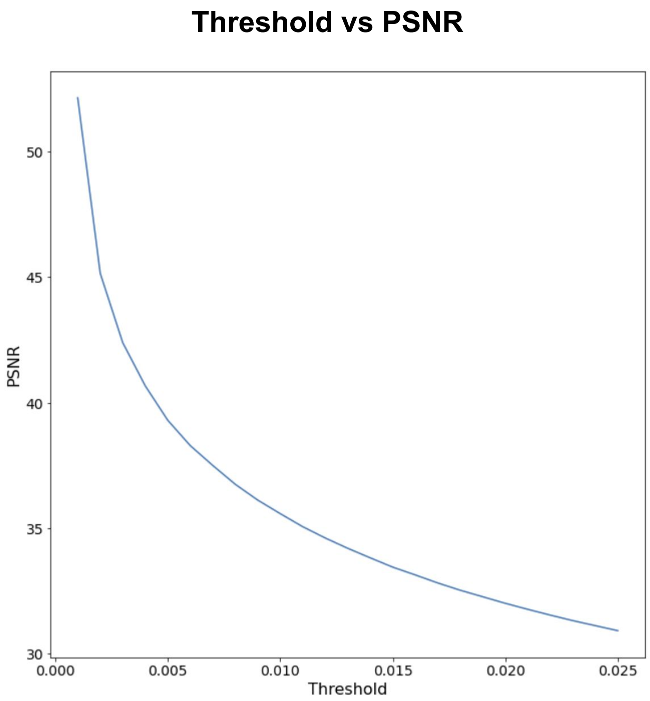
CR	1.79
MSE	7.63
PSNR	39.3
SSIM	0.94

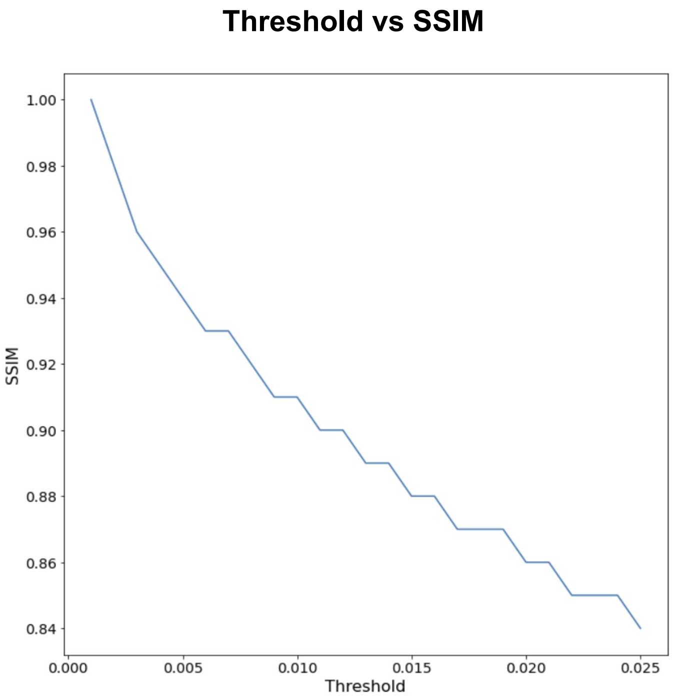
Performance Metrics for different threshold values





Performance Metrics for different threshold values (contd.)





SVD VS DCT - Comparison

Original Image
Size = 2384 KB

Compressed Image - **SVD**
Size = 1342 KB

Compressed Image - **DCT**
Size = 1329 KB

Evaluation metrics

	SVD	DCT
CR	1.78	1.79
MSE	361.31	7.63
PSNR	22.5	39.3
SSIM	0.68	0.94

SVD VS DCT - Comparison on entire data set

The SVD and DCT algorithms have been implemented on all the 100 images in our data set and performance metrics were calculated for every image. We have taken the average value of each metric and summarized the results as shown below:

Average image size: 2238 KB **Average compressed image size: 1224 KB**	**SVD** **K = 30**	**DCT** **Threshold = 0.0050**
CR	1.86	1.83
MSE	359.79	4.01
PSNR	24.75	42.65
SSIM	0.65	0.96

Limitations

- As we increase the image size, the compression time increases as the size of matrices increase.
- Performing compression on RGB images becomes heavy as we need to process each channel separately.
- The performed compression algorithms are lossy compression techniques, so we lose data after compression.
- To overcome these limitations, we need to adapt more efficient algorithms which are also lossless in nature.
- One of the more efficient algorithms being adapted these days is Huffman Coding which is a lossless data compression technique.
- Huffman coding is based on the frequency of occurrence of a data item i.e. pixel in images. The technique is to use a lower number of bits to encode the data in to binary codes that occurs more frequently. It is used in JPEG files.

Sec 6: Concluding Remarks

Conclusion

- In this project, we implemented image compression using SVD and DCT techniques.
- We observed reduced image sizes post compression for both these methods.
- For SVD, large values of hyperparameter give good results in terms of performance metrics and for DCT, low values of the hyperparameter give good results.
- For a given compression ratio, DCT resulted in a less distorted image as compared to SVD.
- Finally, we saw the challenges/limitations of the implementation and how to overcome them.

References

[1] Sara, U. et al. (2019) Image Quality Assessment through FSIM, SSIM, MSE and PSNR—A Comparative Study. *Journal of Computer and Communications*, **7**, 8-18

[2] Asnaoui, K. (2020) Image Compression Based on Block SVD Power Method. *Journal of Intelligent Systems*, 29(1), 1345-1359

[3] http://people.missouristate.edu/jrebaza/assets/10compression.pdf

[4] https://www2.karlin.mff.cuni.cz/~tuma/Aplikace15/prezentace_Hnetynkova.pdf

[5] https://mse.redwoods.edu/darnold/math45/laproj/fall2016/ChristopherBingham/presentation.pdf

[6] https://web.mit.edu/be.400/www/SVD/Singular_Value_Decomposition.htm

[7] Medium. 2020. *Compressing Images Using Linear Algebra*. [online] Available at: https://medium.com/analytics-vidhya/compressing-images-using-linear-algebra-bdac64c5e7ef

[8] http://ijarse.com/images/fullpdf/1467705501_8_Research_Paper.pdf