Lossy Image Compression using SVD Coding Algorithm

- K M Aishwarya

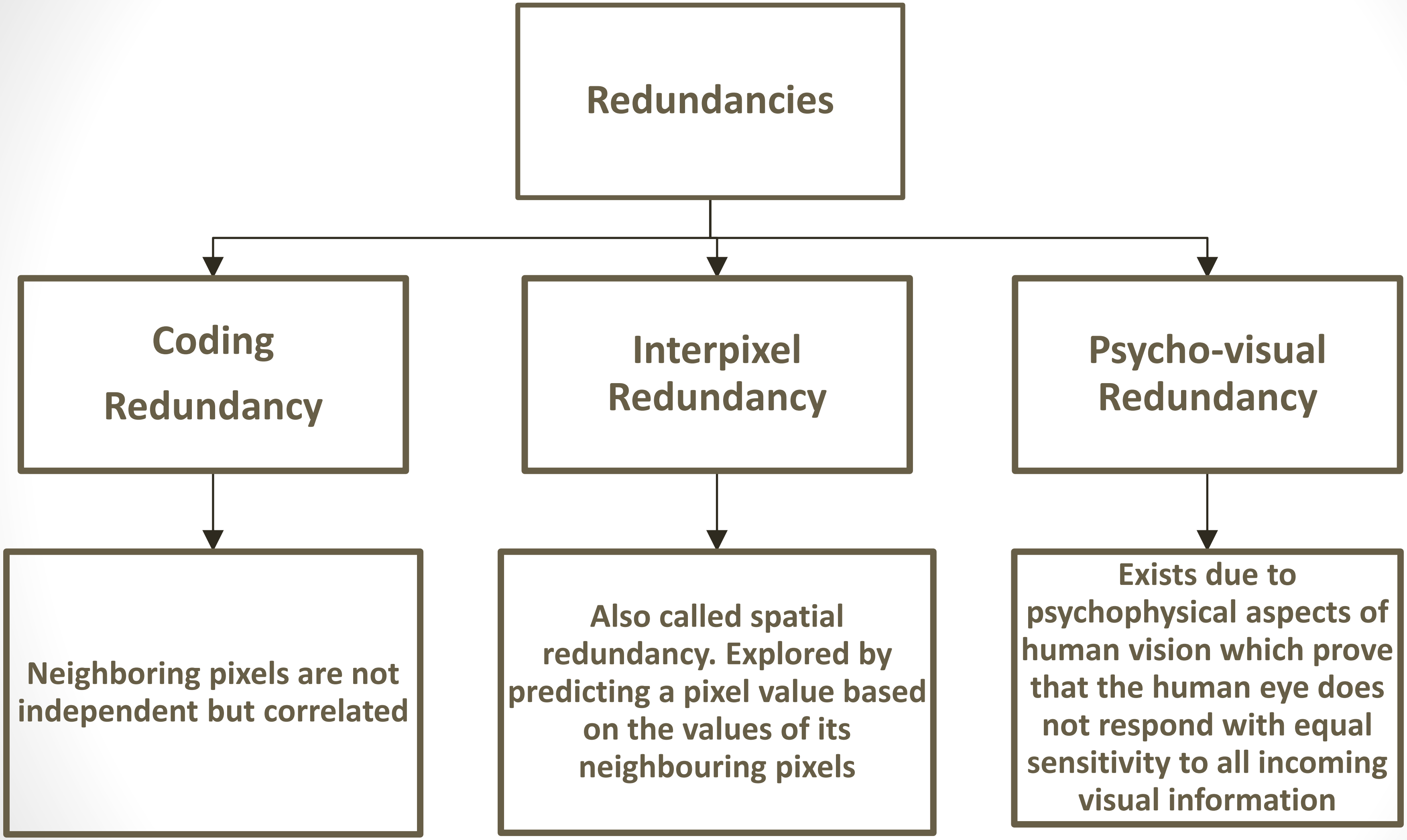
Why is compression needed?

- To store data efficiently
- To transmit data efficiently
- To save:
    - Memory
    - Bandwidth
    - Cost

Why SVD?

- Image compression alone can experience redundancies which can be overcome by Singular Value Decomposition (SVD) of the image matrix.
- Image compression and coding techniques explore 3 types of redundancies:
  - Coding redundancy
  - Interpixel redundancy
  - Psycho-visual redundancy

How to compress?



What is SVD?

- Basic idea:

  Taking a high dimensional, highly variable set of data points and reducing it to a lower dimensional space that exposes the substructure of the original data more clearly and orders it from most variation to the least.

- Definition:

  In linear algebra, the singular value decomposition is a factorization of a real or complex, square or non-square matrix. Consider a matrix A with *m* rows and *n* columns with rank *r*. Then A can be factorized into three matrices:

$$A = USV^T$$

SVD – Overview I

- Diagrammatically,

$$A = \begin{bmatrix} u_1 & \cdots & u_r & \cdots & u_m \end{bmatrix} \begin{bmatrix} \sigma_1 & & & & \\ & \ddots & & & \\ & & \sigma_r & & \\ & & & \ddots & \\ & & & & 0 \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_r^T \\ \vdots \\ v_n^T \end{bmatrix}$$

where,

- U is an $m \times m$ orthogonal matrix
- V^T is the conjugate transpose of the $n \times n$ orthogonal matrix.
- S is an $m \times n$ diagonal matrix with non-negative real numbers on the diagonal which are known as the singular values of A.
- The m columns of U and n columns of V are called the left-singular and right-singular vectors of A respectively.
- The numbers $\sigma_1^2 \geq \ldots \geq \sigma_r^2$ are the eigenvalues of AA^T and A^TA.

SVD – Overview II

- SVD takes a matrix, square or non-square, and divides it into two orthogonal matrices and a diagonal matrix
- Simply applying SVD on an image does not compress it
- To compress an image, after applying SVD:
  - Retain the first few singular values (containing maximum image info)
  - Discard the lower singular values (containing negligible info)
- The singular vectors form orthonormal bases, and the important relation

$$A\,v_i = s_i\,u_i$$

- shows that each right singular vectors is mapped onto the corresponding left singular vector, and the "magnification factor" is the corresponding singular value

Mathematical steps to calculate SVD of a matrix

1. Given an input image matrix A.
2. First, calculate AA^T and A^TA.
3. Use AA^T to find the eigenvalues and eigenvectors to form the columns of U: $(AA^T - \lambda I)\ddot{x} = 0$.
4. Use A^TA to find the eigenvalues and eigenvectors to form the columns of V: $(A^TA - \lambda I)\ddot{x} = 0$.
5. Divide each eigenvector by its magnitude to form the columns of U and V.
6. Take the square root of the eigenvalues to find the singular values, and arrange them in the diagonal matrix S in descending order: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_r \geq 0$

In MATLAB: `[U,W,V]=svd(A,0)`

SVD Compression Measures

- Parameters for quantitative and qualitative compression of image:
  - Compression Ratio (C_R):

$$C_R = \frac{\text{uncompressed image file size}}{\text{compressed image file size}}$$

  - Mean Square Error (MSE):

$$MSE = \frac{1}{mn} \sum_{y=1}^{m} \sum_{x=1}^{n} (f_A(x, y) - f_{A_k}(x, y))$$

  - Signal to Noise Ratio (SNR):

$$SNR = \frac{P_{signal}}{P_{noise}} = \left(\frac{A_{signal}}{A_{noise}}\right)^2$$

  - Peak Signal to Noise Ratio (PSNR):

$$PSNR = 10log_{10}\left(\frac{MAX_I^2}{MSE}\right) = 20log_{10}\left(\frac{MAX_I}{\sqrt{MSE}}\right)$$

Storage Space Calculations

$$\Longrightarrow \quad A = USV^T$$

$$\Longrightarrow \quad A = \sum_{i=1}^{r} \sigma_i u_i v_i^T$$

$$\Longrightarrow \quad A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_r u_r v_r^T$$

- When performing SVD compression, the sum is not performed to the very last Singular Values (SV's); the SV's with small enough values are dropped. The values which fall outside the required rank are equated to zero. The closet matrix of rank *k* is obtained by truncating those sums after the first *k* terms:

$$\Longrightarrow \quad A_k = \sigma_1 u_1 v_1^T + \sigma_2 u_2 v_2^T + \ldots + \sigma_k u_k v_k^T$$

- Total storage for A_k will be:

$$\Longrightarrow \quad A_k = k(m + n + 1)$$

- The digital image corresponding to A_k will still have very close resemblance to the original image

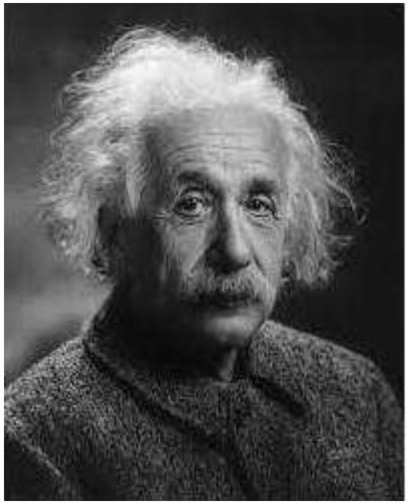
Lowest rank-*k* approximations

- Since $\sigma_1 > \cdots > \sigma_n > 0$, in the singular matrix, the first term of this series will have the largest impact on the total sum, followed by the second term, then the third term, etc.
- This means we can approximate the matrix A by adding only the **first few terms of the series**.
- As *k* increases, the image quality increases, but so too does the amount of memory needed to store the image. This means smaller ranked SVD approximations are preferable.
- Hence *k* is limit bound to ensure SVD compressed image occupies lesser space than original image.

Necessary condition: $\boldsymbol{k < \frac{mn}{m+n+1}}$

Examples of SVD compression in MATLAB

Original Image:

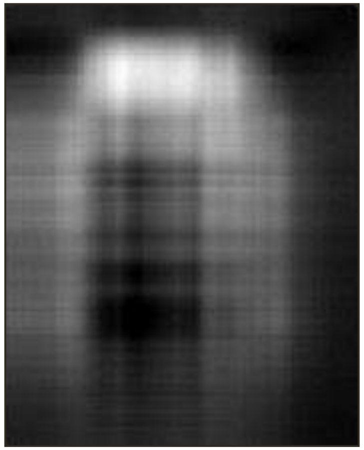


Rank of image = 202

A_k = 31840

C_R = 0.580

SVD compressed images for different values of *k*

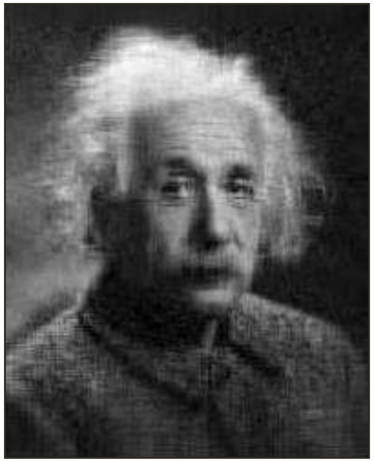


$A_k = 30796$
$C_R = 61.046$

k = 2

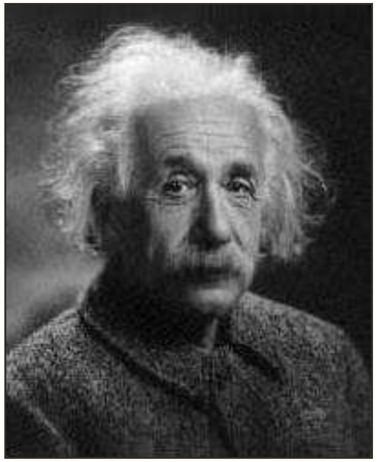
$A_k = 31679$
$C_R = 23.588$

k = 12



$A_k = 31699$
$C_R = 12.101$

k = 22



$A_k = 31781$
$C_R = 5.72$

k = 52

Observation & Inference

- *k* represents the number of Eigen values used in the reconstruction of the compressed image
- Smaller the value of *k*, more is the compression ratio but image quality deteriorates
- As the value of *k* increases, image quality improves (i.e. smaller MSE & larger PSNR) but more storage space is required to store the compressed image
- When *k* is equal to the rank of the image matrix (202 here), the reconstructed image is almost same as the original one

Conclusion

- SVD's applications in world of image and data compression are very useful and resource-saving.
- SVD allows us to arrange the portions of a matrix in order of importance. The most important singular values will produce the most important unit eigenvectors.
- We can eliminate large portions of our matrix without losing quality.
- Therefore, an optimum value for 'k' must be chosen, with an acceptable error, which conveys most of the information contained in the original image, and has an acceptable file size too.

Thank You