Using Linear Algebra in Image Compression: SVD and DCT

By: Andrew Fraser

Outline:

- Explain file compression and its linear algebra representations
    - How are files stored?
    - Colored image storage vs black and white image storage
    - How are images stored in matrix form?
    - How many bytes does this form of storage take up, and why is compression so necessary?
    - What is the difference between a lossy and lossless compression?
- Explain SVD and DCT as methods of compression
    - How is Singular Value Decompression (SVD) used, and how are eigenvalues and eigenvectors applied?
    - What is the compression ratio that results in a mostly lossless compression for the average SVD?
    - How is the Discrete Cosine Transform (DCT) used, and how is it considered a Fourier Transformation?
    - What is the compression ratio that results in a mostly lossless compression for the average DCT?
- Introduce the code
    - Show an example of using SVD on a picture, and how compression ratios can be altered for higher or lower quality (code snippets)
    - Show an example of using DCT on a picture, and how compression ratios can be altered for higher or lower quality (code snippets)
- Explain the results
    - Show multiple examples of using SVD and DCT on various pictures with different compression ratios
    - Explain why certain pictures require higher ratios, and what makes a picture easier or harder to compress
    - Explain any other results that I found in doing the project, including comparing SVD and DCT and their upsides and downsides