Final Project - Singular Value Decomposition in Image Compression
Math 214
Manas Taneja (CS), Ali Almasmari (CS), Joseph Lucchesi (IOE)

Topic: Singular Value Decomposition in Image Compression

In linear algebra, Singular Value Decomposition (SVD) is a factorization of real or complex matrices, breaking down any matrix into its final and best factorization of $U\Sigma V^T$ (U is orthogonal, Σ is diagonal, and V is orthogonal). It is particularly useful in image processing, specifically image compression. In our project, we'll discuss how SVD is used in image processing along with some code and live examples to demonstrate.

Purpose Outside of Math: Image compression is crucial given limited storage for consumers yet massive amounts of data generated/used/sent.

In Math: The purpose of SVD is to reduce a large dataset into a significantly smaller dataset while preserving most of the variability from the original dataset. Additionally, we'll be working with low rank approximations, which allow us to further reduce the amount of data we actually need to keep a record of. Another useful aspect of SVD is that the matrices resulting from the decomposition are much simpler than the original matrix, in the sense that they have certain properties that make them much easier to work with for general computations (both in and outside of compression).

Data: Since our topic is image compression, our dataset will consist of images (each of which can be thought of as 3 n by m matrices, one each for red, green, and blue color values). The exact images haven't been picked yet, and the dataset will likely change multiple times throughout the span of the project depending on what our code can handle in a timely manner (note to instructor: please do let me know if you'd like a finalized dataset immediately, I can put it together and provide it within a day or two).

Assumptions: Some basic assumptions will be the processing power (specifically RAM) of the computer being used. This limits our image size as it simply doesn't make any sense (for the sake of the project and demo) to have an image that takes forever to compress and/or decompress. Lastly, for the sake of simplicity, we'll be taking a guess-and-check approach with the low rank approximation, though we'll explain the concept in theory.

Resources found so far:
https://medium.com/analytics-vidhya/compressing-images-using-linear-algebra-bdac64c5e7ef
(won't be using code from here, but might use images)
http://iridl.ldeo.columbia.edu/dochelp/StatTutorial/SVD/index.html
https://inst.eecs.berkeley.edu/~ee127/sp21/livebook/l_svd_low_rank.html
https://inst.eecs.berkeley.edu/~ee127/sp21/livebook/exa_low_rank_4by5.html

Computations: Convert image to 3 m*n matrices (one for each color in RGB), perform SVD on all 3, perform low rank approximation on all 3, and then use the approximation matrices to restore the original image. This will all be done in python and jupyter notebook.

Final Project - Singular Value Decomposition in Image Compression
Math 214
Manas Taneja (CS), Ali Almasmari (CS), Joseph Lucchesi (IOE)

Background & Skills:

Ali & Manas - CS background, which is useful for writing code
Joseph - IOE background, useful for optimization problems
All 3 members - poster design and content

Deadline & Work Timeline:

November 30 - Start poster design, content, and code rough draft
November 30 - Poster design draft done
December 1 - content rough draft done
December 3 - Content solidified draft done
December 5 - Code done
December 4/5 - Presentation practice
December 6/7 - Tweaks to code, design, and content to make them fit together
December 8 - Final checks & submit

Delegation:

Live presentation - All 3 members
Code - Manas, Ali
Poster content - Mostly Ali and Joe with input from Manas
Poster design - All 3 members