Project 3: SVD Image Compression
Due date: 2023-11-15

Project

1. Write computer code to use SVD to compress an image. Follow the theory in Section 12.4.3 from the book and the sample code "SVD - Image Compression" uploaded on Canvas. The sample code is for gray scale images. For the project, you are required to develop a similar code for color images. Search the internet to learn how to process color images.

2. Write your own code to calculate the SVD (rather than using a linear algebra library).

3. Take a picture of one of the team members with the camera and use the SVD to compress it. One parameter in your code should the number p of singular values that are used in the reduction (refer to Section 12.4.2). In your report provide the following information:

   (a) The percentage of the storage memory used for p singular values versus the storage memory used for the original picture, for various values of p.

   (b) The error between the matrix A storing the original image and the reduced matrix A_p obtained when the first p singular values are used.

Project Report

- This is a typed-up report (does not need to be very long, around 10 pages including figures is okay) that carefully discusses and presents your solutions to the project questions. A solution with no explanation may not receive full credit.

- If the activity asks for a numerical solution, always include the numerical solution in the report. (The numerical solution may be a number, a vector, a list of numbers, a table, ...).

- Be sure to use enough significant digits! Remember we are striving for accuracy: why stop at tolerance 10^{-3} if you can easily compute the solution with tolerance 10^{-12}?

- If the activity asks for a plot, always include the plot in the document.

The report should be submitted to the instructor **via Canvas as a PDF** file by the due date.

Code Listing

Your code should be clearly written, consistent with best practices, and **reproducible**. In particular:

- **Write your own numerical code**. The goal of this course is to learn numerical methods. You must program the **numerical methods explained in class** yourselves. Don't use readily-available numeric libraries, such as SciPy's `fsolve` to find roots of an equation.

  Of course, you may use libraries for all other things (e.g. for data manipulation, plotting, etc.). You may also use libraries for standard numerical computations (e.g. standard functions like trigonometric functions, matrix algebra, ...)

- **Reproducibility**. Your code must be **ready to run** and **produce the same results** documented in the project report.
- **Structure your code**. Divide your code into files with meaningful names (e.g. activity1.m, activity2.m, ...). Use **functions** to encapsulate relevant pieces of code. Use meaningful names for constants/variables
- **Comment your code**. Be as verbose as necessary, adding comments profusely whenever code is complex. Here is a pointer to a short, sensible page on commenting code: `https://www.cs.utah.edu/~germain/PPS/Topics/commenting.html`. In particular, add a function header for every numerical method that you write (e.g. Newton's method, Least squares, or Power iteration). File headers and (short) inline comments are also useful. Please read the reference above for more details.
- **Test key functions**. If a function performs some complex task and is very important for your project, test it on a few known examples. The more complex the function, the more exhaustive the testing.

The code should be submitted to the instructor **via Canvas as a jupyter notebook** file by the due date.

CODE OF ETHICS

0.1 Academic Integrity

The submission by a student of any examination, course assignment, or degree requirement is assumed to guarantee that the thoughts and expressions therein not expressly credited to another are literally the student's own. Evidence to the contrary will result in appropriate penalties, described below.

0.2 Cheating on Assignments and/or Exams

Cheating is an affront on academic integrity and ethics. Any instance of dishonesty undermines your work and the work of your classmates and the University.

0.3 Plagiarism

In defining plagiarism, this policy distinguishes between Intentional Misrepresentation (which is deemed to constitute plagiarism) and Misuse of Sources. These are two clear extremes, but this policy also recognizes that there can be a continuum between them.

Intentional Misrepresentation occurs when a student deliberately uses someone else's language, ideas, or other original (not common knowledge) work without acknowledging the source. Examples include but are not limited to when a student submits an Assignment that: a) is downloaded from an Internet source and/or obtained from a paper mill; b) is obtained from someone else (including another student); c) contains part or all of the writings of another person (including another student), without acknowledgment of the source; or d) contains passages that were cut and pasted from an Internet source, without acknowledgement of the source.

Misuse of Sources is the unintentional misappropriation of the language, ideas, and work of others due to a lack of understanding of the conventions of citation and documentation, including paraphrasing, quoting, and the parameters of common knowledge.

Students are responsible for knowing how to quote from, paraphrase, summarize, and cite sources correctly. However, when a student has attempted to acknowledge a source but has not done so fully

or completely, the instructor, perhaps in consultation with other faculty, administrators, or an academic integrity panel, may determine that the issue is Misuse of Sources or unsuccessful writing, rather than Intentional Misrepresentation.

0.4 Penalties and Procedures for Violating Academic Integrity Standards

Accordingly, students who act in a dishonest manner by cheating on written exams or plagiarizing are subject to penalty under the procedures stated in the Katz Graduate Catalog. This may include **reduced credit or zero on the assignment, reduced final grade or failing grade**.