Final Project: Report

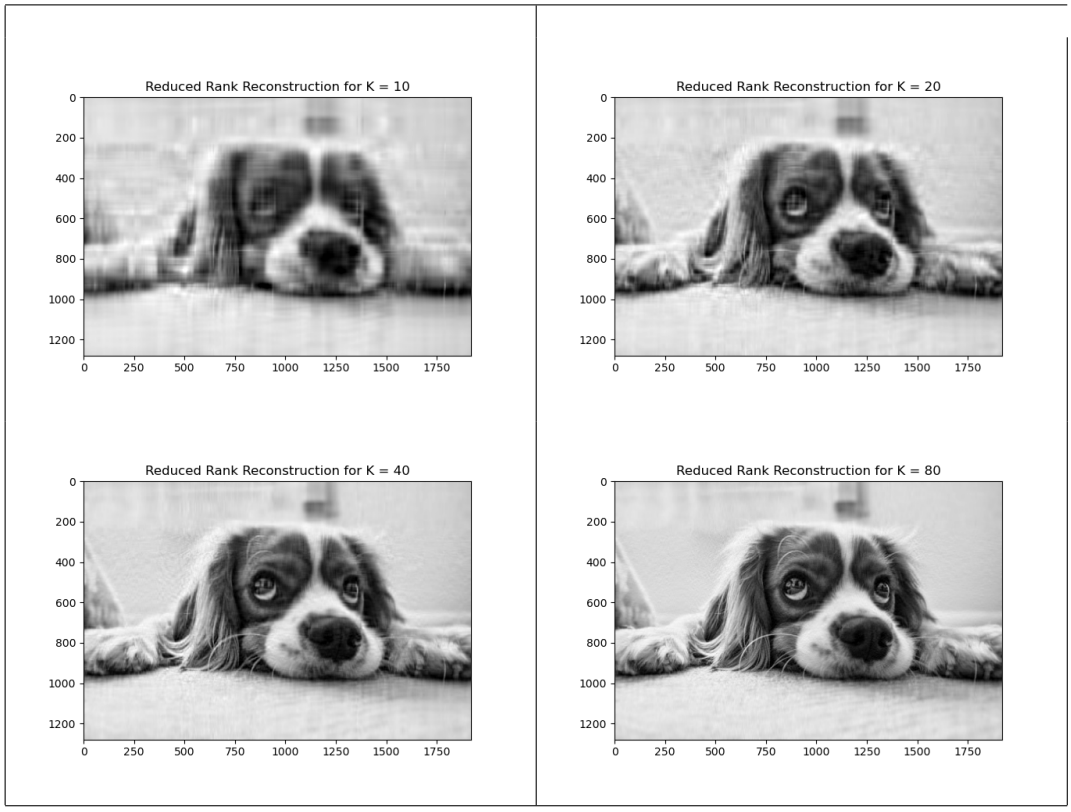
Dante Buhl

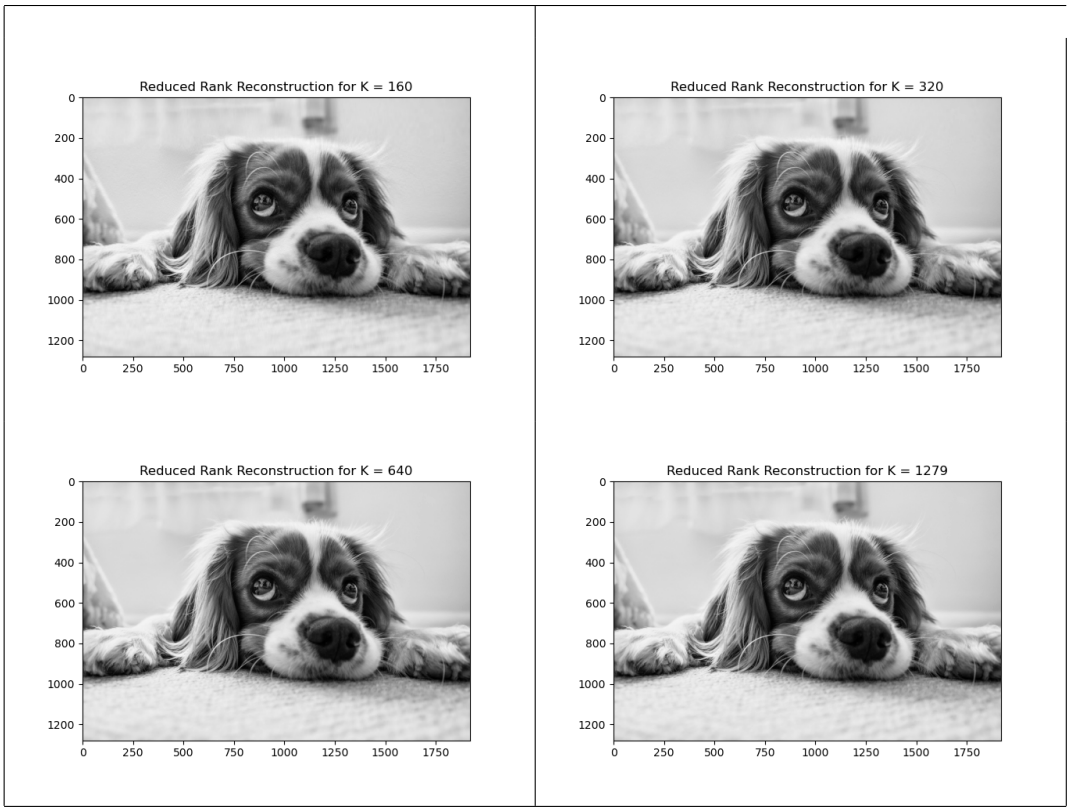
March 18, 2024

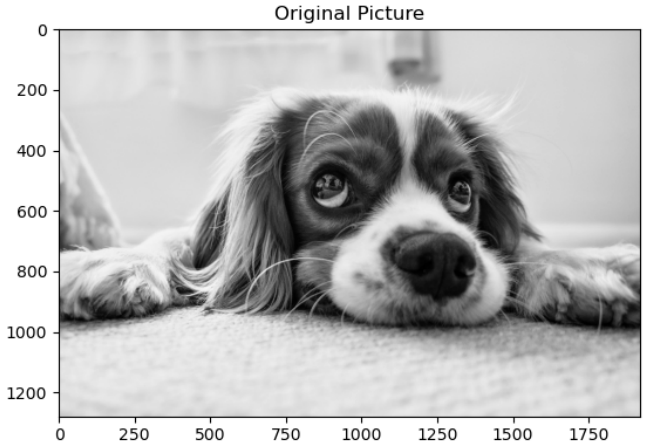
1 Part A: Reduced Rank Image Reconstruction

1. The first ten largest singular values are {...}. Furthermore the "rest singular values" are as follows $\{\sigma_{10} =, \sigma_{20} =, \sigma_{40} =, \sigma_{80} =, \sigma_{160} =, \sigma_{320} =, \sigma_{640} =, \sigma_{1279} =\}$.

2. Here are the following eight images generated by the reduced rank approximation for the image.







3. It is easily noticable that as we reduce the rank of our appoximations more and more, that the quality of the image decreases. Some of the differences between rank approximations aren't as noticable as others. Between ranks of the reconstruction 40 and 80 there is a very noticable sacrifice in image quality. There seem to be smudges in the picture and a sort of typical graininess which is present in low resolution images. This is what is noticable here. Looking at the error reports of these approximations we find that we only find a normalized error below 10^{-3} for our rank 10 approximation. All other normalized errors are of order 10^{-3} or lower. The lowest being the rank 1279 error at 10^{-16} exactly the order of machine double precision.

2 Iterative Methods

2.1 Gauss-Jacobi/Gauss-Seidel

1. Both the Gauss-Jacobi and Gauss-Seidel methods involve an iterative method in which we recompute a vector x and after (hopefully) a finite number of iterations we will reach the desired level of error. They differ in the compution of the next x vector from the proceeding one. In Gauss-Jacobi, it is simply one line of matrix multiplcation and subtraction, i.e.

$$x^{(n)} = D^{-1}(b - Rx^{(n-1)}), \quad D = \text{diag}(A), \quad R = A - D$$

This is much easier to compute since $x^{(n)}$ only depends on the givens A, D, R, b and the previous iteration. A slightly more complex algorithm, Gauss-Seidel, is derived from this where we break R into an upper and lower matrix, i.e.

$$A = D + R + L, \quad D = \text{diag}(A), \quad R = \text{upper}\,\Delta(A - D), \quad L = \text{lower}\,\Delta(A - D)$$

Here the iterative method is

$$x^{(n)} = 0 \to x_i^{(n)} = \frac{1}{a_{ii}}(b_i - L_{i,1:i-1}x_{1:i-1}^{(n)} - R_{i,i+1:m}x_{i+1:m}^{(n-1)})$$

Here we have that going down the elements of $x^{(n)}$ each element depends on the term before it. This means that rather than one line of matrix opertions, we have to express the iterative process for $x^{(n)}$ as a loop over the m elements of the vector.

2. Here are the plots obtained for all values of D.

3. Some of the cases didn't converge, specifically $D = 2$ and $D = 5$ for the Gauss-Jacobi method. For both of these cases the number of iterations reached into the order of 10^2 and the final result is NaN, usually indicating an underflow. For the cases where both methods converge, we see that Gauss-Seidel converges more quickly. This matches the theoretical predictions that we went over in lecture yielding better convergence and stability for Gauss-Seidel with the trade off of not being able to parallelize.

4. For the case of $A_{ii} = i$ we have that the Gauss-Jacobi method does not converge, but Gauss-Seidel does.

2.2 Conjugate Gradient

1. The Conjugate Gradient Method is a method which

2. DON'T FORGET TO WRITE THE DAMN PROOF

3. This method converges very quickly and with high accuracy for each value of D! In the table below we have the number of iterations needed to converge to order 10^{-5} accuracy (note that though the tolerance parameter is given as 10^{-5} we have that the actual value of the resultant error is below 10^{-14} in only 2 iterations).

$D =$	2	5	10	100	1000
$N =$	2	2	2	2	2

4. Specifically the Diagonal preconditioner is not strong for these matrices. The reason is that since each diagonal element is the same we are effectively dividing the whole matric by the same value, thereby not scaling the eigenvalues of $M^{-1}A$ in any significant manner whatsoever.

5.

6.