CS5691 Pattern Recognition and Machine Learning Assignment 1

Name - Vedant Saboo
Roll no. - CS19B074

Motivation

Working with grey scale images, I explored the data points in it. I converted the picture into a 2d matrix of 256 by 256 pixels, each pixel a single byte. The purpose, to use this matrix in several matrix operations and explore the effects on the image.
The square matrix could be decomposed in two ways

1. **Eigenvalue decomposition**, where we find two matrices Q, and Λ, where Q is an complex invertible matrix, of eigen vectors of original matrix A, and Λ is a diagonal matrix, its diagonal values comprising of eigenvalues corresponding to the eigen vectors of A. These values, of course, can be complex. $A = Q \times \Lambda \times Q^{-1}$

2. **Singular Value Decomposition**, where we find three matrices U, D, V for $A : m \times n$such that

   (a) $U : m \times m$ matrix of left singular eigen vectors of A

   (b) $V : n \times n$ matrix of right singular eigen vectors of A

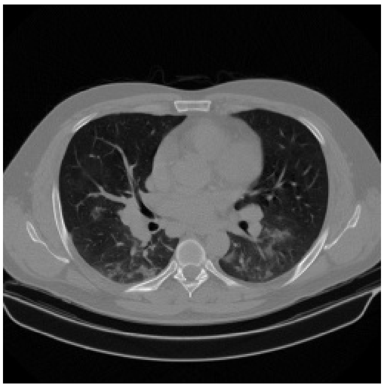
   (c) $D : m \times n$ diagonal matrix, diagonal values being eigen values of A

   (d) $A = U \times D \times V^T$

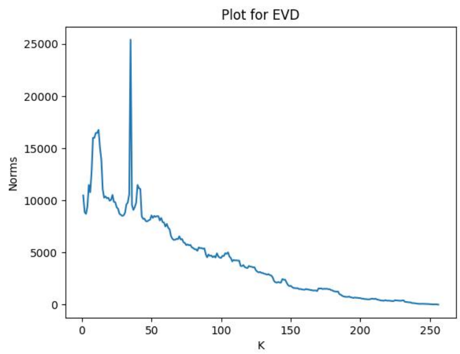
The motivation behind this paper is to understand, analyse and appreciate the effects these decompositions, followed by limited reconstruction (taking k largest magnitude eigenvalues) have on grey scale images.
As a bonus, we will attempt to use the same for RGB images.

Eigenvalue Decomposition (EVD) Analysis

I used the method `numpy.linalg.eig` provided by the `numpy` library in python for the evd calculation. The grey scale image that I was given to work with was **Figure 1 (a)**



(a) Original grey scale image



(b) Plot of norm versus k for EVD

Figure 1: EVD

$$A = Q \times \Lambda \times Q^{-1}$$

To reconstruct the matrix, I identified k highest eigen values (by magnitude) and all the other eigen values were made 0 This reduced the rank of the final matrix and thereby reducing the image quality, and compressing the image, preserving the essential features of the image, and ignoring the rest. We can see the resulting reconstructions in **Figure 2**

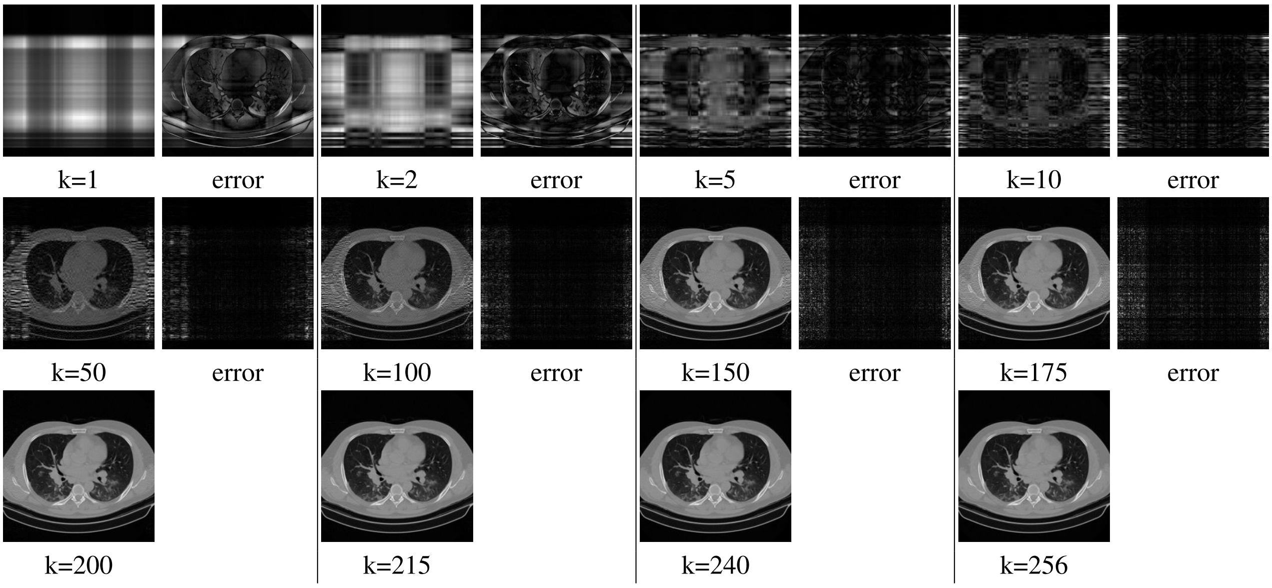


Figure 2: Reconstruction images using k eigen values in the EVD of original image

I used the **Frobenius norm** to measure the dissimilarity between the original image and the reconstructed image. Between two matrices A and B of same order, frobenius norm is defined as

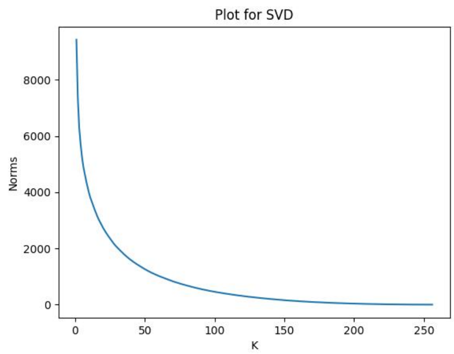
$$Fnorm(A, B) = \sqrt{\sum_{(i,j)} |a_{ij} - b_{ij}|^2}$$

I plotted the values of frobenius norm versus the eigen value count. The results are described in **Figure 1 (b)**. Conclusion, the graph is not smooth, but in general, dissimilarity decreases as the rank is increased.

Singular Value Decomposition (SVD) Analysis

The grey scale image that I was given to work with was **Figure 3 (a)**

(a) Original grey scale image



(b) Plot of norm versus k for SVD

Figure 3: SVD

$$A = U\Sigma V^T$$
$$A^T A = (U\Sigma V^T)^T (U\Sigma V^T)$$
$$A^T A V = V(\Sigma\Sigma^T)$$

The algorithm that I used to calculate the SVD:

1. Form $A^T A$

2. Use EVD on $A^T A$ to find V and the diagonal values of $\Sigma\Sigma^T$. Sort them corresponding to descending eigen values.

3. Square root the eigen values found in step two, to get diagonal values of Σ

4. Find columns of U using $u_i = \frac{1}{\sigma_{ii}} A v_i$ where u_i and v_i are ith column vectors of U and V.

To reconstruct the matrix, I identified k highest eigen values (by magnitude) and all the other eigen values were made 0 in the Σ matrix
This reduced the rank of the final matrix and thereby reducing the image quality, and compressing the image, preserving the essential features of the image, and ignoring the rest. We can see the resulting reconstructions in **Figure 4**
I used the **Frobenius norm** to measure the dissimilarity between the original image and the reconstructed image. Between two matrices A and B of same order, frobenius norm is defined as

$$Fnorm(A, B) = \sqrt{\sum_{(i,j)} |a_{ij} - b_{ij}|^2}$$

I plotted the values of frobenius norm versus the eigen value count. The results are described in **Figure 3 (b)**.

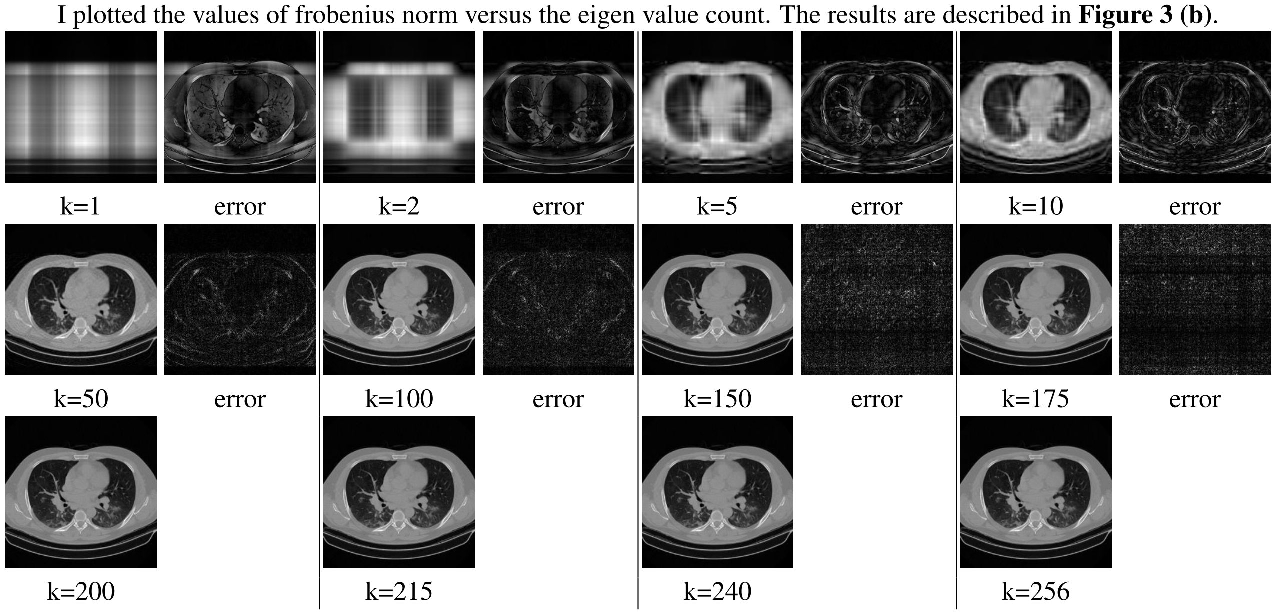


Figure 4: Reconstruction images using k eigen values in the svd of original image

Conclusion and Inference

In EVD, the graph is not so smooth, but the general pattern of dissimilarity decreasing, as we decrease the rank can be easily understood. As we go to a rank as high as **185**, the dissimilarity becomes almost unnoticeable to human eye.
In SVD, the graph is smooth and dissimilarity decreases as the rank is increased, very close to hyperbolic curve. Near the rank of **52**, the dissimilarity becomes unnoticeable to human eye. This suggests that we can isolate the eigen values that contribute the most to the features in the image based on magnitudes of the eigen values. We can then form images with lesser ranks and yet, very similar to original image.
In a general trend, it can be seen through the frobenius norms that **SVD is better than EVD** at all rank values. In fact, we can think of SVD at a rank k to be the **closest matrix to actual image (closeness measured using norm) of rank k**.

Use in image compression: The reconstruction leaves us with a lot of zeroes, which makes us wonder whether we can use this technique in image compression. Many compression techniques are based on the fact that some pattern in the pixel data, and if we have more 0s, we have more pattern. The theory coincides with the fact that indeed, reconstructed images, although reasonable ”retained” have significantly lower sizes. The image with k=100 occupied 9.7kB although, 256*256 = 64k would be consuming 64kB if it wasn’t compressed.