CS5691 - Assignment1

Eigen Value Decomposition and Singular Value Decomposition

Motivation:-

Processing Image data is very crucial in Machine Learning. Extracting useful features from an image is very essential. Before performing any task related to images, it is necessary to process the image to make more suitable as input data. So,for processing an image,we can consider image as Matrix of pixel values.

Eigen value decomposition and Singular value decomposition can be done on these image matrices. These are used for Image Compression, approximating images, removing noise and also for removing unnecessary features like facial expressions from an image.

Eigen Value Decomposition:-

Eigen Value decomposition is factorizing a square matrix A into three matrices which are comprised of eigen values and eigen vectors.

$$A = UDU^{-1}$$

where D is the diagonal matrix with diagonal entries as eigen values and the corresponding eigen vectors form the columns of matrix U.

Eigen decomposition is decomposing image in terms of Eigen Vectors. We can note that if we assume the eigen values have been sorted in decreasing order based magnitude (as eigen values can be complex), then we can consider top k terms in eigen decomposition to have a good approximation for the original matrix. The following images shows how the reconstructed images looks for different values of k.

(a) original image (b) k=100 (c) k=200

Figure 1: Original image vs Reconstructed images

Singular Value Decomposition:-

Singular Value decomposition is factorizing a square matrix A into three matrices which are comprised of singular values and singular vectors.

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

Σ is the diagonal matrix with diagonal entries as singular values and the corresponding left singular vectors form the columns of matrix U and the corresponding right singular vectors form the columns of matrix V.

- Eigen vectors of $A^T A$ form the columns of V.
- Eigen vectors of AA^T form the columns of U.

Singular decomposition is decomposing image in terms of Singular Vectors. we can consider top k terms in singular decomposition to have a good approximation for the original matrix. The following images shows how the reconstructed images looks for different values of k.

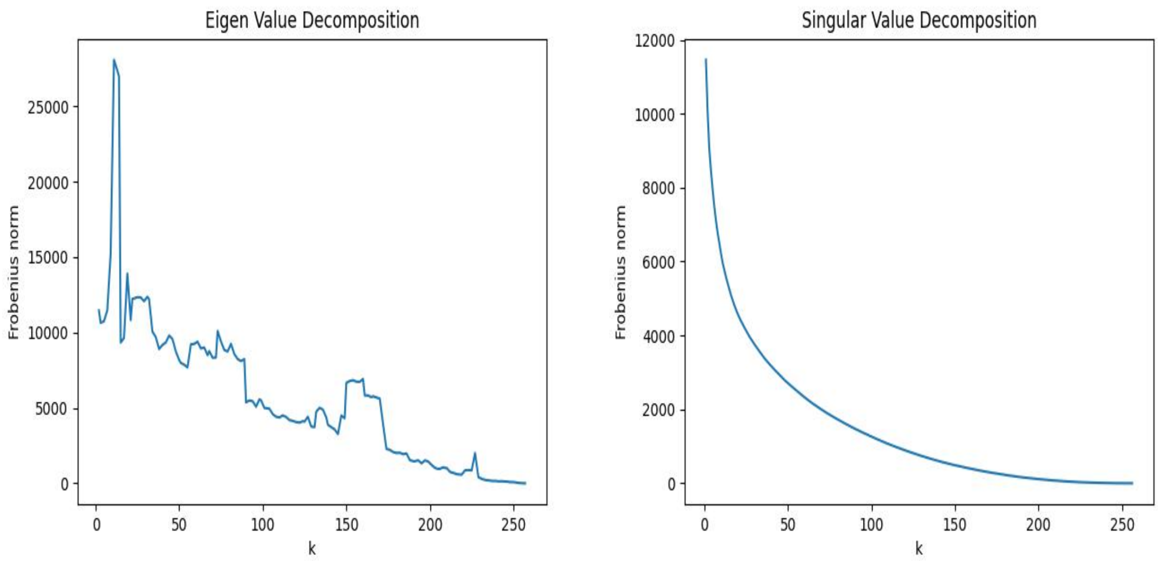
(a) original image (b) k=100 (c) k=200

Figure 2: Original image vs Reconstructed images

Experimental Results:- In this Experiment, EVD is computed by finding eigen values and eigen vectors of A and making eigen values as diagonal elements in D and making eigen vectors as coloumns in U. SVD is computed by finding eigen values and eigen vectors of $A^T A$ and these eigen vectors form the columns of V and U is computed using $AV = U\Sigma$ and Σ is computed by $U^T AV$.

The main theme of the experiment is to know how the error in the image varies with k.If we compute the Frobenius norm between the reconstructed image and the original image for different values of k, we can observe that frobenius norm decreases with increase in k.

The graphs for eigen decomposition and singular value decomposition are as follows.



(a) EVD

(b) SVD

Figure 3: Frobenius Norm vs k

Inferences:-

- With increase in the value of k,the frobenius norm continuously decreases in case of SVD but in EVD, frobenius norm decreases but with some spikes. SVD graph is nearly similar to $y = 1/x$.
- In Eigen value decomposition, eigen values can be complex but in Singular value decomposition, singular values are real. So,when considering eigen values, we have to include both eigen value and its complex conjugate to get the real image matrix.
- If $A = UDU^{-1}$ (EVD) then U need not be orthogonal but if $A = U\Sigma V^T$ (SVD) then U and V are orthonormal matrices.
- The decrement in frobenius norm in case of SVD is at higher rate than EVD. For $k = 100$ the image in case of SVD is more clear than that in EVD.In SVD, if k is nearly 50 then image is clear but in EVD for image to be clear k should be nearly 150.
- Neglecting smaller eigen values or singular values provides good approximation of images and also helpful in reducing noise. This also decreases condition number of the matrix. That is the matrix will become less sensitive to errors in input vector.