CS571 Project Report

K Bharat Kumar, Roll No : T21021

23-Nov-2021

1 Summary

<u>Component Estimation in Images using Principal Component Analysis (PCA)</u>

PCA is a statistical technique which has applications in various fields such as image compression, facial recognition, and is a common technique for finding patterns in high-dimensional data. It is a tool for analyzing data. In general, PCA is the Eigen decomposition of covariance matrix. Eigen vector with the highest Eigen value is the principal component of the data set. PCA is mainly used to reduce noise and reduce dimensionality. It does not eliminate noise completely, but it can reduce noise. The main idea underlying is that, any components with variance much larger than the effect of noise should be relatively unaffected by noise. So that if we reconstruct the data using just the largest subset of principal components, we should be able to preferentially keep the signal or data and throw out the noise. Here in this project we have created the images of English alphabets. These images are 4 x 4 pixels, hence their dimensionality in the pixel space is 16. Then multiple noisy images are generated (say 1000) for each image, through the addition of Additive White Gaussian Noise to them. Two approached are used for reduction of noise from the sample images. The first method is through the implementation of PCA and the other one is through NMF (Non-Negative Matrix Factorization). An efficient data analysis tool, sklearn is used for decomposition of PCA and NMF and finally, tried to reconstruct a clean image from the samples of noisy images.

2 Introduction

This project is designed to provide basic understanding of PCA. In general, PCA is the Eigen decomposition of covariance matrix. The Eigen vectors and Eigen values obtained from the covariance matrix are quite different. In fact, it turns out that the Eigen vector with the highest eigenvalue is the principle component of the data set. Once Eigen vectors are found from the covariance matrix, the next step is to order them by Eigen value, highest to lowest. This gives you the components in order of significance. Now, we can decide to ignore the components of lesser significance. We do lose some information, but if the eigenvalues are small, we don't lose much. If we leave out some components, the final data set will have less dimensions than the original. Once we have chosen the components (eigenvectors) that we wish to keep in our data we can suitably reconstruct the data from the bulk of elements in the dataset. Hence we find PCA having many applications in various fields such as face recognition, image compression and many other. Here in this project we are using PCA for noise filtering. It does not eliminate noise completely, but it can reduce noise.

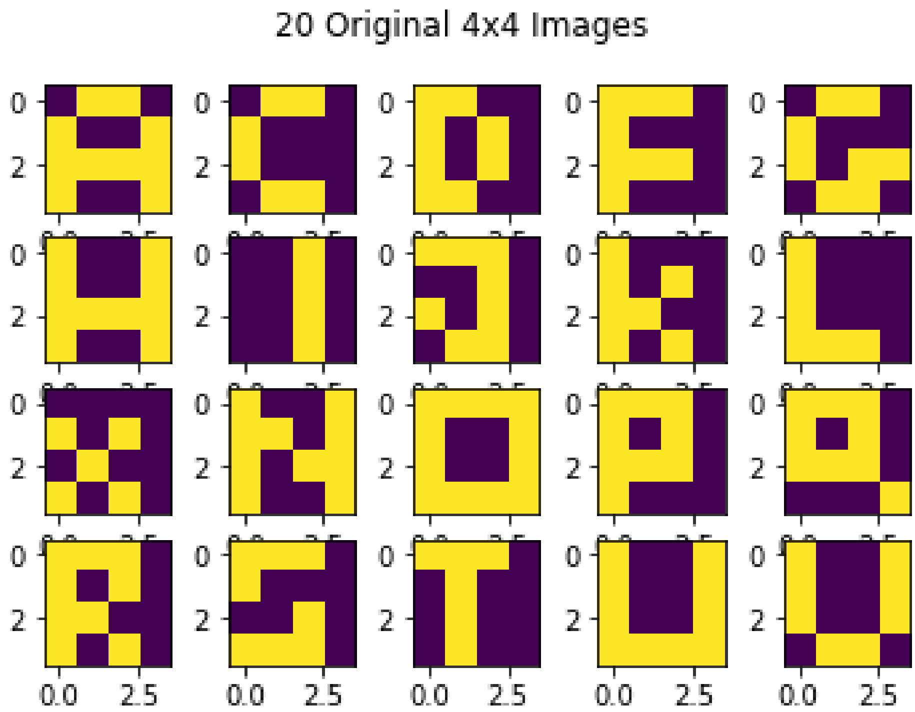
3 Solution

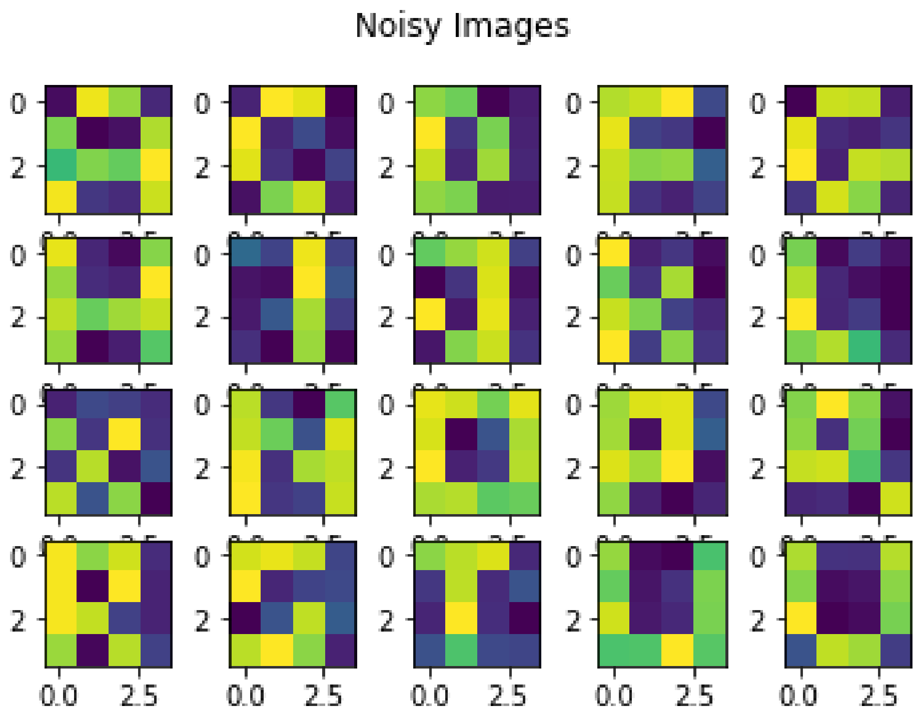
The basic principal used is that, any components with variance much larger than the effect of noise should be relatively unaffected by noise. So that if we reconstruct the data using just the largest subset of principal components, we should be able to preferentially keep the signal or data and throw out the noise.

3.1 Algorithms used

We have implemented PCA through the following steps. Step 1. Get the dataset Step 2. Subtract the mean of each variable from the data set Step 3. Calculate the Covariance Matrix Step 4. Compute Eigen values and Eigen vectors Step 5. Sort Eigen values and Eigen vectors Step 6. Choose the number of components and transform the data We also made use of an efficient data analysis tool sklearn for decomposition of PCA and NMF to perform PCA and NMF and transform the data as needed.

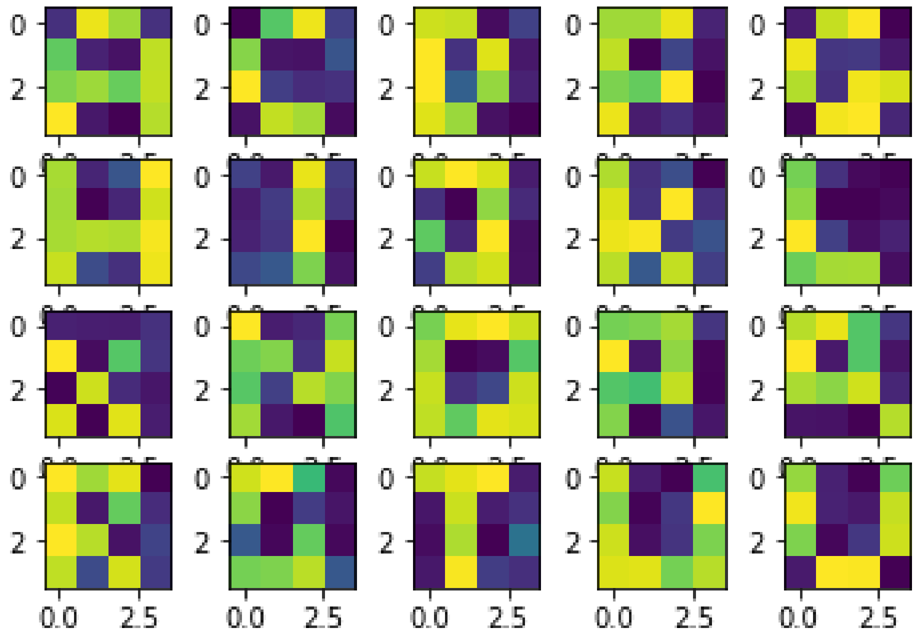
4 Results and analysis



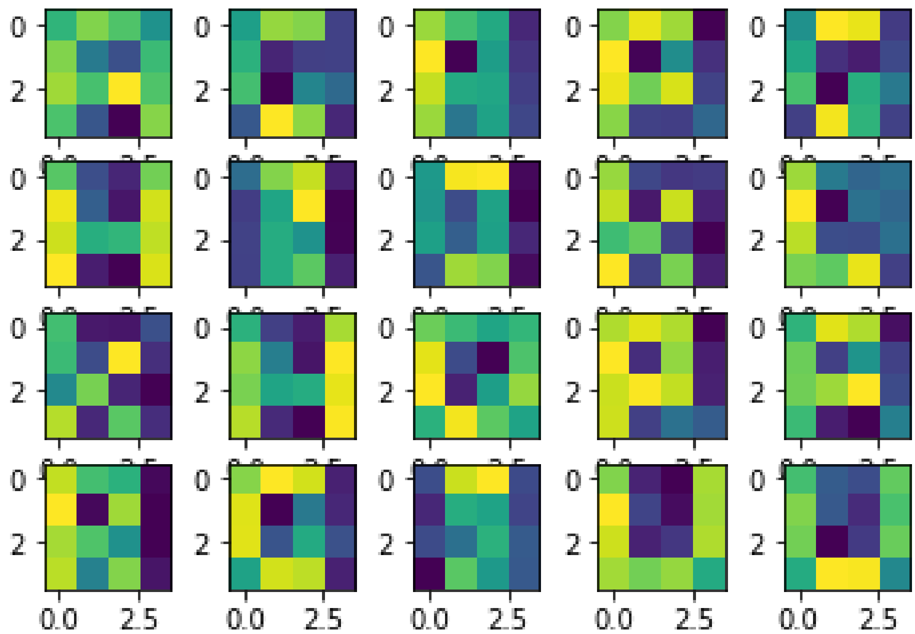

Noisy Images
0
2
0.0
2.5

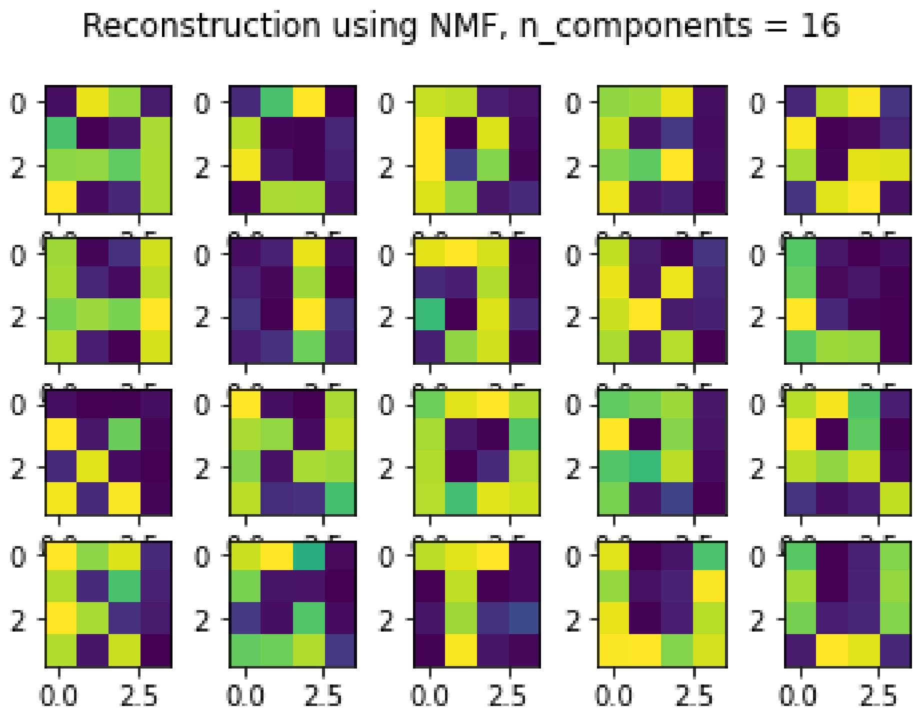

Reconstruction using PCA
0
2
0.0
2.5

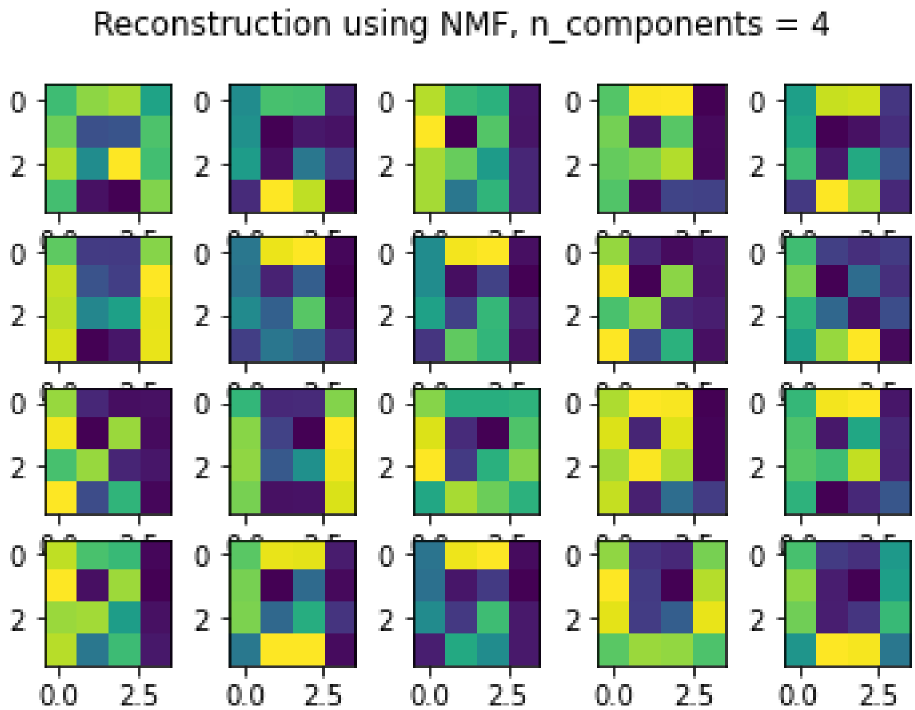
Reconstruction using PCA, n_components = 16



Reconstruction using PCA, n_components = 4







The above plots indicate that PCA can reduce noise but cannot completely eliminate noise. Since each image we generated is 4x4 pixels, hence their dimensionality in the pixel space is 16. Therefore, we cannot get more than 16 Eigen vectors for reconstruction of the dataset. Hence, PCA will not be able to get back all the images correctly and the images will be more blurry.

5 Conclusion

In this project we learned to implement PCA for filtering the image. We have seen that PCA cannot eliminate noise completely. Another limitation is that PCA will not be able to get back all the images correctly. Since each image we generated is 4x4 pixels, hence their dimensionality in the pixel space is 16. Therefore, we cannot get more than 16 Eigen vectors for reconstruction of the dataset. As the dataset increases PCA becomes less efficient. Also we can clearly see that as the number of components decreases image becomes more blurry.

6 Project Github page

Link to Github page : *https://github.com/Bharat-2k21/CS571-Project6*

7 References

[1] "Dictionary Learning", by Tosic et al, IEEE Sig Proc Mag, March 2011.