EE636 Matrix Computations

Assignment 5:SVD based image compression

Vyomkesh Chaudhary | 203070028

Result

I applied the SVD alrogithm and plotted singular values in BAR chart and then plotted the cumulative singular values in BAR chart and observed that around first 90 to 120 singular values captures most of the energy. Link for python.ipynb file: https://drive.google.com/drive/folders/1tuMmk4VzbV8kTrBptzrtp-R1ASx4tAcQ?usp=sharing

Singular Values

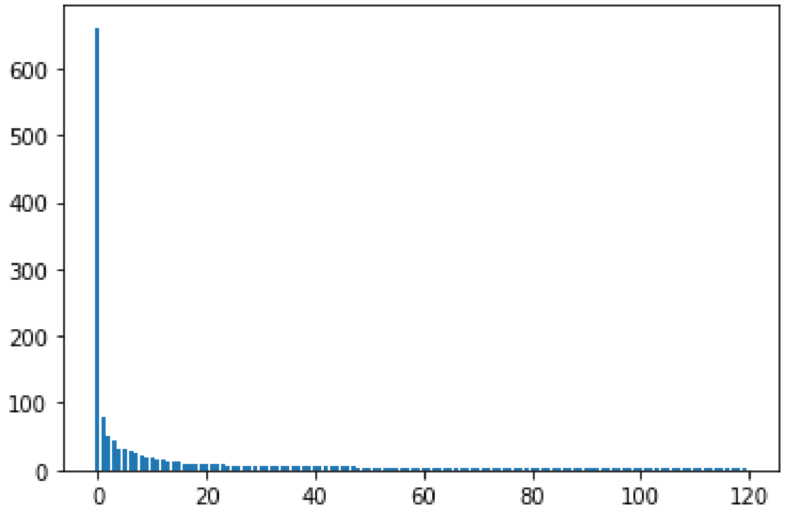

Figure 1: Singular values

Cumulative Singular Values

Around first 90 to 120 singular values captures most of the energy.

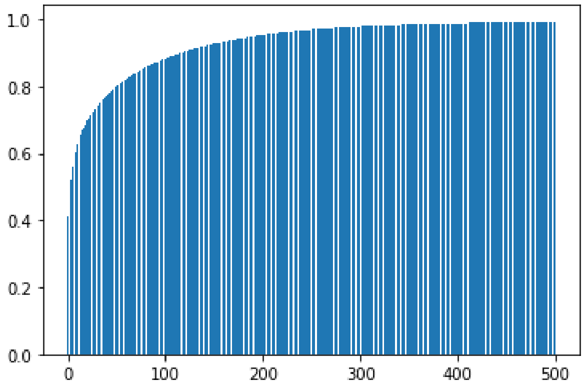

Figure 2: Cumulative Singular values

Plotting approx image considering only first rth columns

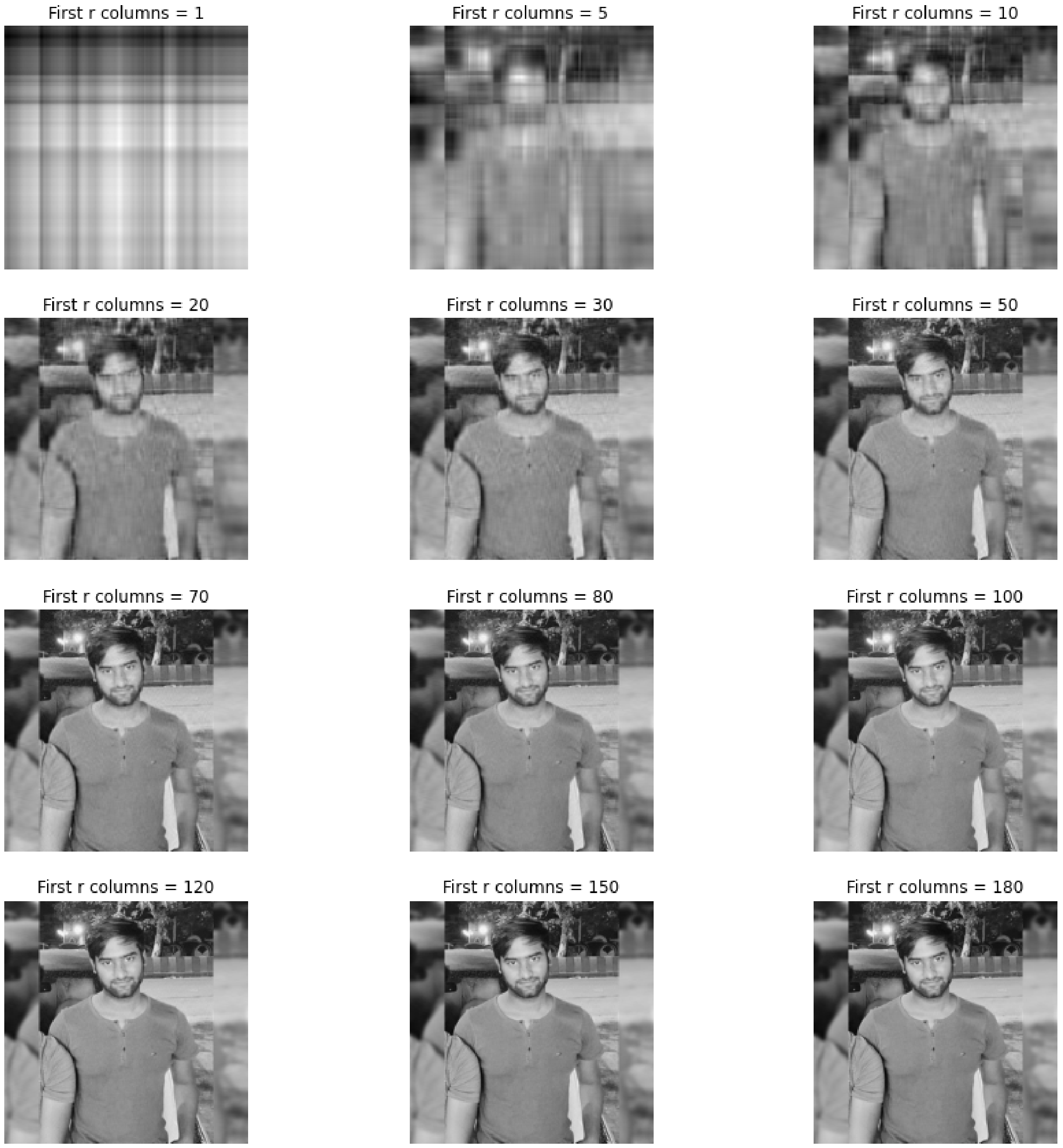

Figure 3: Approx images

Pseudocode

```
A = read(image)
U , S , VT = SVD(A)
diagonal(S)
i = min(row(A),col(A))
barplot(CumulativeSum(S[i,i])/total(S))
r = max i which captures most of the power
approx_image = U[:,:r] * S[:r,:r] * VT[:r,:]
save(approx_image)
```

MY CODE(Python)

```
from matplotlib.image import imread         ##importing libraries
import matplotlib.pyplot as plt
import numpy as np

A = imread('/content/greyscale.png')         ##reading image file
X = np.mean(A,-1)                            ##Convert RGB to grayscale
img = plt.imshow(X)                          ##displaying data as image
img.set_cmap('gray')                         ##setting the color for image
plt.axis('off')
plt.show()

U, S , VT = np.linalg.svd(X,full_matrices=False) ##Running economy SVD using numpy
S = np.diag(S)                       ##extacting S matrix in diagonal form

k = np.arange(0,500)           ##Checking for singular values
plt.bar(k,S[k,k])

plt.bar(k,(np.cumsum(S[k,k]/np.sum(S))))            ##Cumulative Singular value


from matplotlib.pyplot import imshow            ##importing library

fig = plt.figure(figsize=(15, 15))
rows=4
columns=3
for i in range(12):
    matrix = S[i]*np.outer(U[:,i],VT[i])    ## 1 rank matrix corresponding to ith singular val
    fig.add_subplot(rows, columns, i+1)     ## plotting images in rows and columns
    image1 = plt.imshow(matrix)
    image1.set_cmap('gray')
    plt.axis('off')
    plt.title('singular value = ' + str(S[i,i]))
plt.show()


fig = plt.figure(figsize=(15, 15))
rows=4
columns=3
```

```
i=0
for r  in (1,5,10,20,30,50,70,80,100,120,150,180):  ## considering only first rth singular val
  Xapprox = U[:,:r] @ S[:r,:r] @ VT[:r,:]       ## computing approx matrix for A
  fig.add_subplot(rows, columns, i+1)
  i += 1
  image = plt.imshow(Xapprox)
  image.set_cmap('gray')
  plt.axis('off')
  plt.title('First r columns = ' + str(r))
plt.show()


new= U[:,:120] @ S[:120,:120] @ VT[:120,:]  ## saving image considering first 120 singular val
plt.imshow(new,cmap='gray')
plt.axis('off')
plt.savefig('reconstructed_image_with_5_SVs.png',dpi=150, figsize=(15,15))
```