Problem 2 [15 marks]

You are one of the scientists working at NASA's Goddard Space Flight Center in Greenbelt, Maryland and have been researching Wide-Field Slitless Spectroscopy to capture galaxy spectra of the distant universe. With the help of NASA's James Webb Space Telescope, you have successfully captured the deepest and sharpest infrared image of the distant universe to date. It is an image of the galaxy cluster SMACS 0723 and has been named Webb's First Deep Field.

Unfortunately, due to some technical difficulties, the space telescope has not been able to transmit full-resolution images to Earth. However, an onboard computer can be programmed remotely from Earth to transmit the image in a compressed format until the difficulties are resolved. The control station on Earth has decided to use SVD to compress the image. As a scientist tasked with programming the onboard computer, think about the following:

(a) How many singular values are required to approximate the image i.e., make it look indistinguishable from the original image? (Hint: Load the image in Python or Matlab or Octave or Julia and the matrix representation of the image will be accessible to you.

For $r \times r$ pixel image, the image will have $r \times r \times 3$ matrix entries with the number 3 corresponding to color depth of the image representing Red, Blue, Green.)

(b) Based on your observation in (a), how many entries need to be transmitted to earth to reconstruct the approximate image as opposed to sending the original image?

[Perform the tasks in a programming environment comfortable to you like Matlab/Octave/Python/Julia. You can use inbuilt functions for computing SVD.]

(c) What is the 2-Norm and Frobenius-Norm error between the matrix representation of the original image and the approximate image obtained for different number of singular values. Check if the following theorems hold for these errors:

For the matrix $\mathbf{A}$ of rank r, with singular values $\sigma_1,\ \sigma_2,\ \sigma_3,\ \ldots\ \sigma_r$, $\mathbf{A}_v$ is the v-rank approximation of $\mathbf{A}$. ($\mathbf{A}_v = \sum_{i=1}^{v} \sigma_i \mathbf{u}_i \mathbf{v}_i$) such that $1 < v < r$, then: $\|\mathbf{A} - \mathbf{A}_v\|_2 = \sigma_{v+1}$, $\|\mathbf{A} - \mathbf{A}_v\|_F = \sqrt{\sigma_{v+1}^2 + \sigma_{v+2}^2 + \ldots + \sigma_r^2}$

The image Webb's First Deep Field is as below and also downloadable from Teams assignment page as a PNG file.

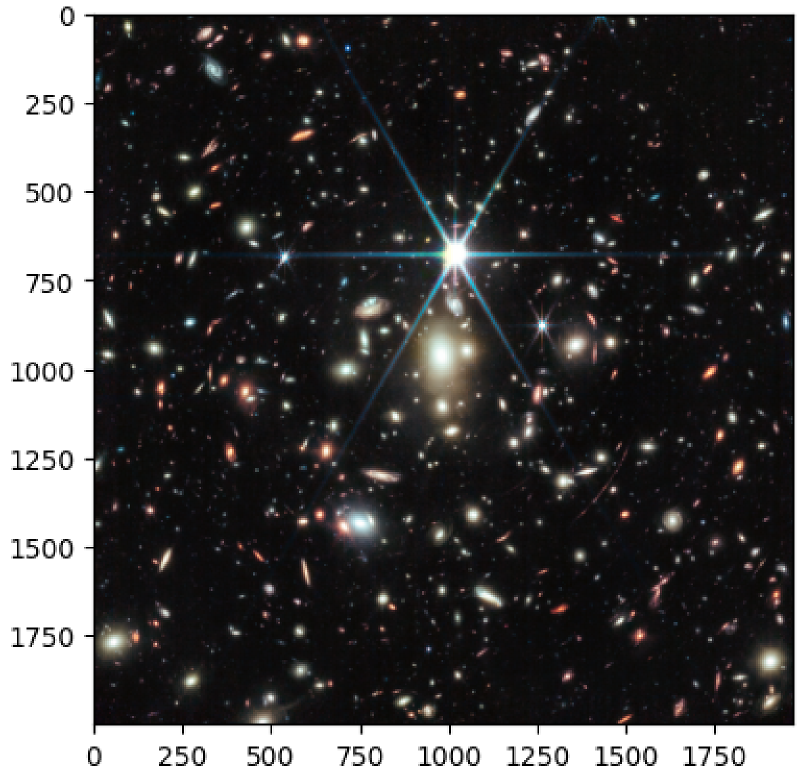
de8uv5b4q

October 5, 2023

[ ]:
```python
import matplotlib.pyplot as plt
import numpy as np
#Reading the image
painting=plt.imread("WebbFirstDeepField.png")
#Printing the corresponding tensor size
print(f"The image has matrix form of size {painting.shape}")
#Printing the 3D image matrix
print(painting)
print(f"The image consists of {painting.shape[0] * painting.shape[1]} pixels")
#Plotting the image
plt.imshow(painting);
#print(painting)
```

```
The image has matrix form of size (2000, 1968, 3)
[[[0.03529412 0.02352941 0.01568628]
  [0.03137255 0.04705882 0.07843138]
  [0.06666667 0.02745098 0.03137255]
  …
  [0.07058824 0.05098039 0.03137255]
  [0.05882353 0.07058824 0.12156863]
  [0.10588235 0.05098039 0.04313726]]

 [[0.06666667 0.03137255 0.01960784]
  [0.09803922 0.05490196 0.09803922]
  [0.05490196 0.02745098 0.01960784]
  …
  [0.02745098 0.02745098 0.01176471]
  [0.03921569 0.03921569 0.04313726]
  [0.10588235 0.13333334 0.09411765]]

 [[0.05098039 0.04313726 0.08235294]
  [0.04705882 0.05882353 0.10588235]
  [0.07843138 0.0627451  0.09411765]
  …
  [0.02745098 0.01568628 0.00392157]
  [0.04705882 0.03529412 0.02745098]
  [0.10196079 0.10196079 0.08627451]]
```

```
 …

 [[0.03529412 0.01960784 0.02352941]
  [0.02352941 0.01568628 0.00784314]
  [0.01960784 0.01960784 0.04313726]
  …
  [0.04705882 0.03529412 0.07843138]
  [0.09019608 0.04705882 0.10196079]
  [0.03529412 0.03921569 0.06666667]]

 [[0.05098039 0.01568628 0.07843138]
  [0.02745098 0.01568628 0.01568628]
  [0.05882353 0.01960784 0.05098039]
  …
  [0.10980392 0.05490196 0.13333334]
  [0.18039216 0.03137255 0.10588235]
  [0.06666667 0.02352941 0.11764706]]

 [[0.04313726 0.02745098 0.09411765]
  [0.02745098 0.03529412 0.05098039]
  [0.03137255 0.02745098 0.02745098]
  …
  [0.10196079 0.0627451  0.12156863]
  [0.08627451 0.0627451  0.11372549]
  [0.03921569 0.03921569 0.05490196]]]
The image consists of 3936000 pixels
```



Q2:a)

The number of singular values required to approximate an image to the point where it looks indistinguishable from the original image depends on the desired level of similarity or the acceptable level of error in the approximation. We will use Singular value decomposition (SVD)for image compression, where we keep only the largest singular values(i.e. first few singular values at the time of approximation) and corresponding singular vectors to achieve a certain level of image quality.

To be honest: the more singular values we retain, the closer the approximation will be to the original image. However, the number of singular values required for an indistinguishable approximation can vary depending on the image content, the viewer's perception, and the specific application.

To determine the number of singular values required for our specific image and quality threshold, we will follow an iterative approach:

1)We will start with a small number of singular values (e.g., 10) and use them to reconstruct the image.

2)Then we will evaluate the quality of the reconstructed image compared to the original one. We will use metrics namely Mean Squared Error (MSE) to measure the difference.

3)Then we will increase the number of retained singular values and repeat the reconstruction and quality evaluation.

We will continue this process until we achieve the desired level of indistinguishability.

1)Mean Squared Error (MSE): 0.008659213781356812 for approximation with 10 singular values.
2)Mean Squared Error (MSE): 0.00129564362578094 for approximation with 80 singular values.
3)Mean Squared Error (MSE): 0.001196396304294467 for approximation with 85 singular values.
4)Mean Squared Error (MSE): 0.001109389471821487 for approximation with 90 singular values.
5)Mean Squared Error (MSE): 0.0009647480910643935 for approximation with 100 singular values.
6)Mean Squared Error (MSE): 0.0008494030334986746 for approximation with 110 singular values.
7)Mean Squared Error (MSE): 0.0007564733386971056 for approximation with 120 singular values.
8)Mean Squared Error (MSE): 0.00068035873118788 for approximation with 130 singular values.
9)Mean Squared Error (MSE): 0.0005637842696160078 for approximation with 150 singular values.
10)Mean Squared Error (MSE): 0.0004449457919690758 for approximation with 180 singular values.
11)Mean Squared Error (MSE): 0.00038890799623914063 for approximation with 200 singular values.
12)Mean Squared Error (MSE): 0.0003448514617048204 for approximation with 220 singular values.
13)Mean Squared Error (MSE): 0.00029451819136738777 for approximation with 250 singular values.
14)Mean Squared Error (MSE): 0.0002566260809544474 for approximation with 280 singular values.
15)Mean Squared Error (MSE): 0.00023598670668434352 for approximation with 300 singular values.
16)Mean Squared Error (MSE): 0.00019526730466168374 for approximation with 350 singular values.
17)Mean Squared Error (MSE): 0.0001646571181481704 for approximation with 400 singular values.
18)Mean Squared Error (MSE): 0.0001406257360940799 for approximation with 450 singular values.
19)Mean Squared Error (MSE): 0.0001209731271956116 for approximation with 500 singular values.
20)Mean Squared Error (MSE): 0.00010451355774421245 for approximation with 550 singular values.
21)Mean Squared Error (MSE): $9.04864355106838 * 10^{-05}$ for approximation with 600 singular values.
22)MSE is $7.840691250748932 * 10^{-05}$ for approximation with 650 singular values.
23)MSE is $6.79036311339587 * 10^{-05}$ for approximation with 700 singular values.
24)MSE is $5.0658083637245 * 10^{-05}$ for approximation with 800 singular values.
25)MSE is $3.732395271072164 * 10^{-05}$ for approximation with 900 singular values.
26)MSE is $2.6982319468515925 * 10^{-05}$ for approximation with 1000 singular values.
27)MSE is $1.900247298181057 * 10^{-05}$ for approximation with 1100 singular values.
28)MSE is $1.2944567970407661 * 10^{-05}$ for approximation with 1200 singular values.
29)MSE is $1.0511334949114826 * 10^{-05}$ for approximation with 1250 singular values.
30)MSE is $8.425383384746965 * 10^{-06}$ for approximation with 1300 singular values.
31)MSE is $6.650524028373184e - 06 * 10^{-06}$ for approximation with 1350 singular values.
32)MSE is $5.15498140885029 * 10^{-06}$ for approximation with 1400 singular values.
33)MSE is $3.911697604053188 * 10^{-06}$ for approximation with 1450 singular values.
34)MSE is $2.89022 * 10^{-06}$ for approximation with 1500 singular values.
35)MSE is $1.4240572454582434 * 10^{-06}$ for approximation with 1600 singular values.

After comparing MSE's for different number of singular values and also comapring their corresponding approximated images[as shown in the output of below code], if we store only 180 singular values (for each color channel), we will get an "indistinguishable" image.In that case we are getting MSE as $< 5*10^{-4}$. we compared the images upto 1600 singular values and saw that there is not any distinguishable improvement in the corresponding image quality with the improvement of number of singular values, after the stage of storing 180 singular values for each channel.

2b)

As per our discussion in (2a),for each channel we want to store only 180 singular values.And thus, to store the reduced SVD of each channel(with the help of first 180 singular values only), we need to store 2000*180 entries for corresponding U of reduced SVD, 180 entries for [later from this 180 non-zero singular values we can make the $180 * 180$ diagonal matrix Σ for reduced SVD] corresponding Σ of reduces SVD, and $180 * 1968$ entries for corresponding V^T of reduced SVD[We used reduced SVD of a matrix A=UΣV^T, with proper dimensions each].Thus in total we need to store $(2000 * 180) + (180) + (180 * 1968)$ entries for each channel.Thus for 3 color channels, total:$3 * ((2000 * 180) + (180) + (180 * 1968)) = 714420$ entries (where as for actual image $2000 * 1968 * 3 = 11808000$ entries needed to be transmitted) needs to be transmitted to reconstruct the approximated image and we will get a good approximated image(with mean squared error $< 5 * 10^{-4}$).

[ ]:
```
#Slicing the image in 3 color channel
import numpy as np

# We have the image matrix, painting of size 2000x1968x3

# We will split the image matrix into its color channels
A1 = painting[:, :, 0]  # Red channel
B1 = painting[:, :, 1]  # Green channel
C1 = painting[:, :, 2]  # Blue channel

# Now A1, B1, and C1 contain the individual color channels as 2D matrices
```

1st color channel=Red color channel, 2nd color channel=Green color channel, 3rd color channel=Blue color channel.

[ ]:
```
import numpy as np
import math
from scipy.linalg import svd
import matplotlib.pyplot as plt

# We have the image matrix, painting of size 2000x1968x3
painting=plt.imread("WebbFirstDeepField.png")
# We will split the image matrix into its color channels
A1 = painting[:, :, 0]  # Red channel
B1 = painting[:, :, 1]  # Green channel
C1 = painting[:, :, 2]  # Blue channel

# Now A1, B1, and C1 contain the individual color channels as 2D matrices
```

```
S=[10,80,85,90,100,110,120,130,150,180,200,220,250,280,300,350,400,450,500,550,600,650,700
    ,800,900,1000,1100,1200,1250,1300,1350,1400,1450,1500,1600]
# Perform SVD on each color channel and retain only the top n singular values,␣
 ↪for each  n in above S.
for n in S:
    print(f"CASE:Number of singular values for approximation is {n}")
    num_singular_values_to_retain = n


    # For the red channel (A1)
    U_A, S_A, V_A = np.linalg.svd(A1, full_matrices=False)
    U_A_truncated = U_A[:, :num_singular_values_to_retain]
    p=len(S_A)
    #print(p)
    a=S_A[n]
    S_A_truncated = np.diag(S_A[:num_singular_values_to_retain])
    V_A_truncated = V_A[:num_singular_values_to_retain, :]
    A2 = np.dot(U_A_truncated, np.dot(S_A_truncated, V_A_truncated))

    sum1=0
    for t in range(n,p):
      sum1=sum1+(S_A[t])**2
    sum1=math.sqrt(sum1)
    print(f"Error in 2 norm and Frobenius norm for  first color channel is:␣
 ↪{a}, and {sum1} respectively.(Using singular values as per the mathematical␣
 ↪expression given in question) ")
    error1=A1-A2
    matrix_norm1 = np.linalg.norm(error1, ord=2)
    fro1=np.mean(error1**2)
    fro1=math.sqrt(fro1*2000*1968)
    print(f"Error in 2 norm and Frobenius norm for  first color channel is:␣
 ↪{matrix_norm1}, and {fro1} respectively.")
    # For the Green channel (B1)
    U_B, S_B, V_B = np.linalg.svd(B1, full_matrices=False)
    U_B_truncated = U_B[:, :num_singular_values_to_retain]
    b=S_B[n]
    q=len(S_B)
    S_B_truncated = np.diag(S_B[:num_singular_values_to_retain])
    V_B_truncated = V_B[:num_singular_values_to_retain, :]
    B2 = np.dot(U_B_truncated, np.dot(S_B_truncated, V_B_truncated))

    sum2=0
    for t in range(n,q):
      sum2=sum2+(S_B[t])**2
```

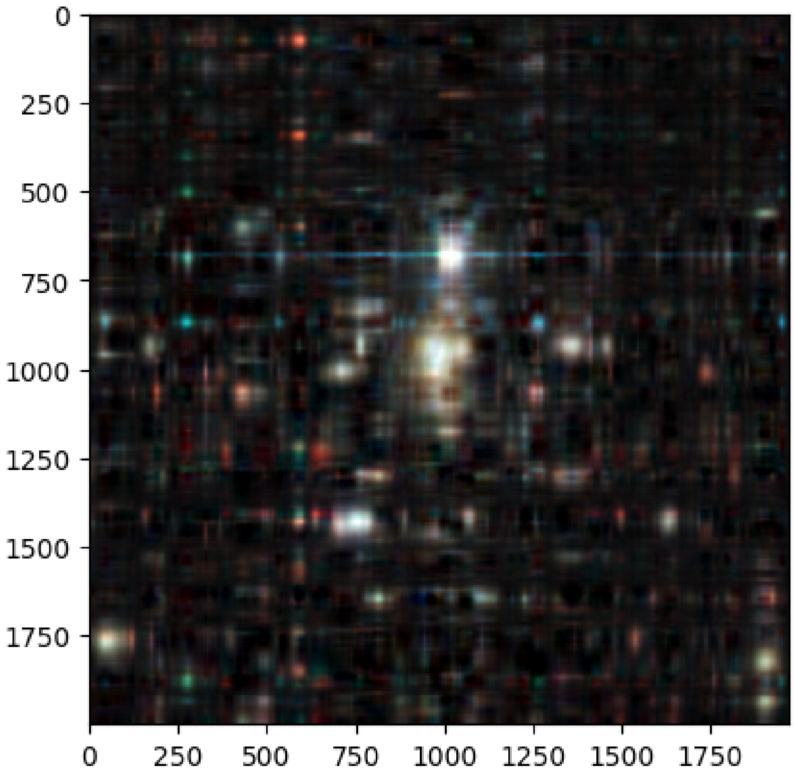
```python
    sum2=math.sqrt(sum2)
    print(f"Error in 2 norm and Frobenius norm for second color channel is: {b}, and {sum2} respectively.(Using singular values as per the mathematical expression given in question) ")
    error2=B1-B2
    matrix_norm2 = np.linalg.norm(error2, ord=2)
    fro2=np.mean(error2**2)
    fro2=math.sqrt(fro2*2000*1968)
    print(f"Error in 2 norm and Frobenius norm for  second color channel is: {matrix_norm2}, and {fro2} respectively.")

    # For the Blue channel (C1)
    U_C, S_C, V_C = np.linalg.svd(C1, full_matrices=False)
    U_C_truncated = U_C[:, :num_singular_values_to_retain]
    c=S_C[n]
    r=len(S_C)
    S_C_truncated = np.diag(S_C[:num_singular_values_to_retain])
    V_C_truncated = V_C[:num_singular_values_to_retain, :]
    C2 = np.dot(U_C_truncated, np.dot(S_C_truncated, V_C_truncated))

    sum3=0
    for t in range(n,r):
      sum3=sum3+(S_C[t])**2
    sum3=math.sqrt(sum3)
    print(f"Error in 2 norm and Frobenius norm for  third color channel is: {c}, and {sum3} respectively.(Using singular values as per the mathematical expression given in question) ")
    error3=C1-C2
    matrix_norm3 = np.linalg.norm(error3, ord=2)
    fro3=np.mean(error3**2)
    fro3=math.sqrt(fro3*2000*1968)
    print(f"Error in 2 norm and Frobenius norm for  third color channel is: {matrix_norm3}, and {fro3} respectively.")
    # Combining the reconstructed channels and creating the approximated image
    approximated_image_matrix = np.stack((A2, B2, C2), axis=-1)
    # Convert the approximated_image_matrix to an image for comparison purpose
    image = plt.imshow(approximated_image_matrix)

    # Save the image as "image1.png"
    plt.savefig(f"image{n}.png")

    # Show the image
    plt.show()
```

```
    # Calculate the Mean Squared Error (MSE) between the original image and the␣
↪approximated image
    mse = np.mean((painting - approximated_image_matrix) ** 2)
    #Calculating Frobenius norm of the error-tensor
    error_matrix = painting - approximated_image_matrix
    sum=0
    for i in range(2000):
      for j in range(1968):
        for k in range(3):
          sum=sum+(error_matrix[i][j][k]*error_matrix[i][j][k])
    frobenius_norm_error=math.sqrt(sum)
    # Display the approximated image


    # Print the Mean Squared Error
    print("FOR THE 3D ERROR MATRIX:")
    print(f"Error in 2-norm is {max(matrix_norm1,matrix_norm2,matrix_norm3)}␣
↪and error in 2-norm using the theorem is : {max(a,b,c)}")
    print(f"Error in Frobenius norm is {frobenius_norm_error} and error in␣
↪Frobenius norm using theorem is : {math.
↪sqrt((sum1**2)+(sum2**2)+(sum3**2))}")
    print(f"For the final approximated 3D tensor Mean Squared Error (MSE):␣
↪{mse},   for number of singular_values {num_singular_values_to_retain} ")
    print(f"*************************************")
```

```
CASE:Number of singular values for approximation is 10
Error in 2 norm and Frobenius norm for  first color channel is:
42.79275894165039, and 196.11081258551232 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
42.79275894165039, and 196.11085756306952 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
38.845558166503906, and 180.3746381905226 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
38.845558166503906, and 180.3746054678744 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
36.642818450927734, and 176.78660276563465 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

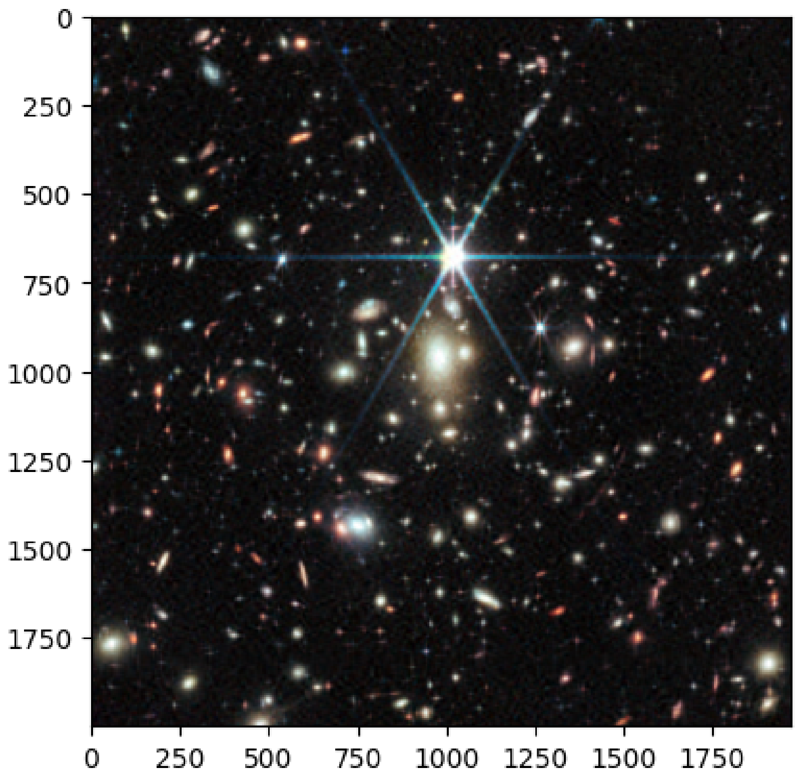
```
Error in 2 norm and Frobenius norm for  third color channel is:
36.642818450927734, and 176.78658486230356 respectively.
```




```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 42.79275894165039 and error in 2-norm using the theorem is :
42.79275894165039
Error in Frobenius norm is 319.762355381848 and error in Frobenius norm using
theorem is : 319.76235524640197
For the final approximated 3D tensor Mean Squared Error (MSE):
0.008659213781356812,   for number of singular_values 10
**************************************
CASE:Number of singular values for approximation is 80
Error in 2 norm and Frobenius norm for  first color channel is:
9.47665023803711, and 76.43781914176665 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
9.47665023803711, and 76.43781505829361 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
8.78646183013916, and 64.87529351446322 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
8.78646183013916, and 64.87528654542385 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
8.887383460998535, and 72.43908037949492 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
8.887383460998535, and 72.43907364694428 respectively.
```

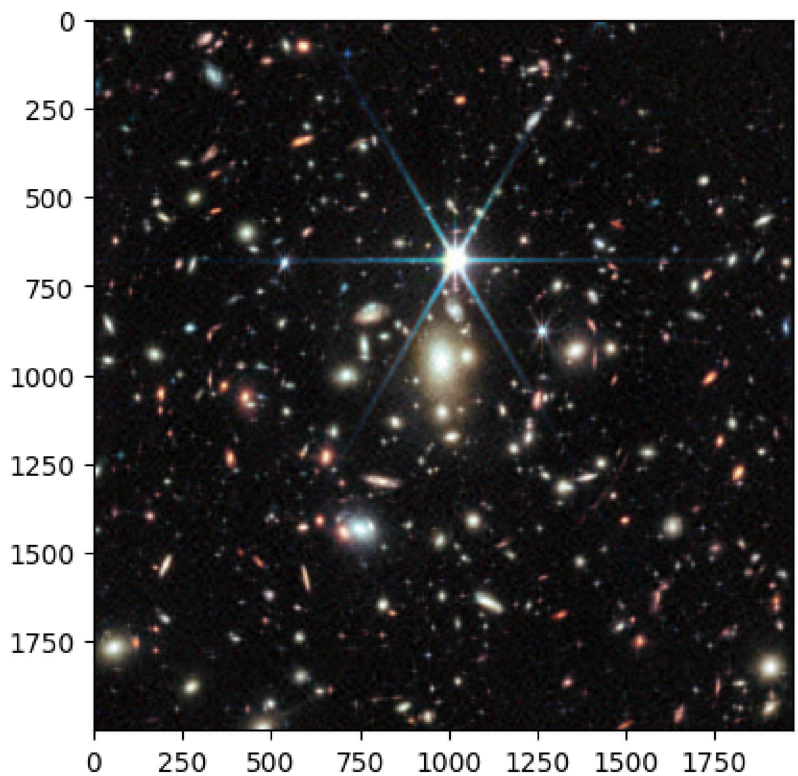



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 9.47665023803711 and error in 2-norm using the theorem is :
9.47665023803711
Error in Frobenius norm is 123.68898219857314 and error in Frobenius norm using
theorem is : 123.6889820071461
For the final approximated 3D tensor Mean Squared Error (MSE):
0.00129564362578094,   for number of singular_values 80
**************************************
CASE:Number of singular values for approximation is 85
Error in 2 norm and Frobenius norm for  first color channel is:
8.956530570983887, and 73.6003906711915 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
8.95652961730957, and 73.60036800477326 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
8.351887702941895, and 61.98284252691174 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
Error in 2 norm and Frobenius norm for  second color channel is:
8.351887702941895, and 61.982843915392905 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
8.354188919067383, and 69.7722536064626 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
8.354188919067383, and 69.7722483275091 respectively.
```

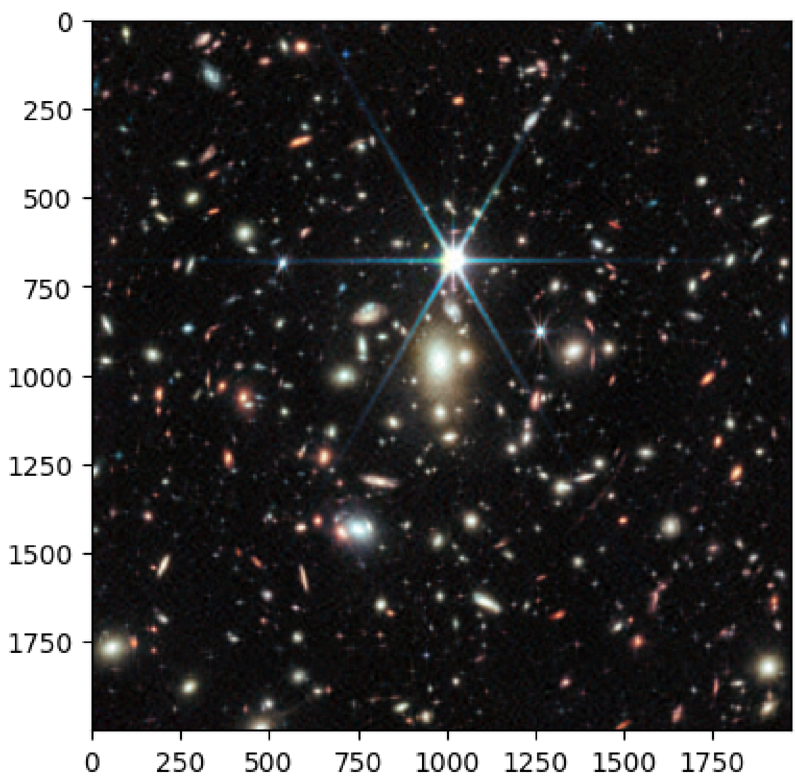


```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 8.95652961730957 and error in 2-norm using the theorem is :
8.956530570983887
Error in Frobenius norm is 118.85729963500921 and error in Frobenius norm using
theorem is : 118.85729951497504
For the final approximated 3D tensor Mean Squared Error (MSE):
0.001196396304294467,   for number of singular_values 85
**************************************
CASE:Number of singular values for approximation is 90
Error in 2 norm and Frobenius norm for  first color channel is:
8.324447631835938, and 71.01475657921611 respectively.(Using singular values as
```

```
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
8.324447631835938, and 71.01473577968737 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
7.760326862335205, and 59.32780951123109 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
7.760326862335205, and 59.32780835755618 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
7.875154972076416, and 67.35570057976918 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
7.875154972076416, and 67.35571482263393 respectively.
```

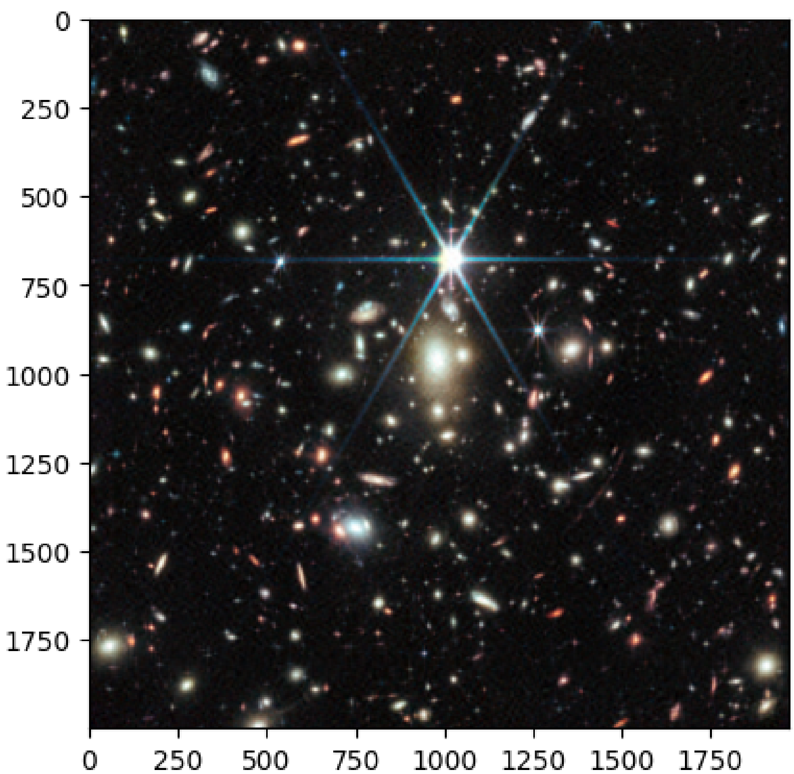



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 8.324447631835938 and error in 2-norm using the theorem is :
8.324447631835938
Error in Frobenius norm is 114.4538120349973 and error in Frobenius norm using
theorem is : 114.45381179321971
```

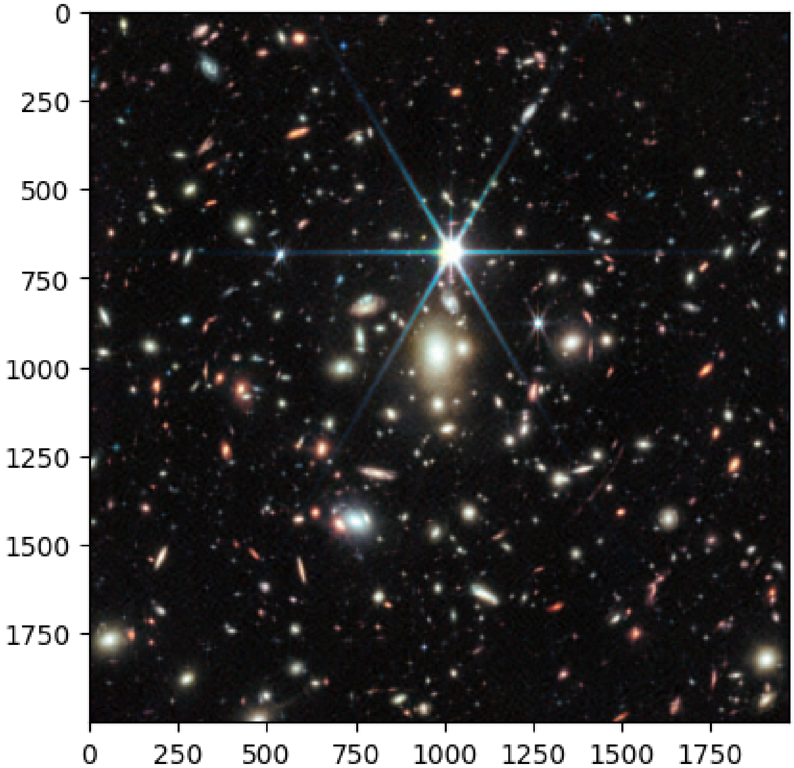
```
For the final approximated 3D tensor Mean Squared Error (MSE):
0.001109389471821487,   for number of singular_values 90
***************************************
CASE:Number of singular values for approximation is 100
Error in 2 norm and Frobenius norm for  first color channel is:
7.420379161834717, and 66.4871180035732 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
7.420379161834717, and 66.48712887646953 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
6.79449462890625, and 54.66123207182551 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
6.79449462890625, and 54.6612207759828 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
7.021533489227295, and 63.11387721403478 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
7.021533489227295, and 63.1138684235477 respectively.
```



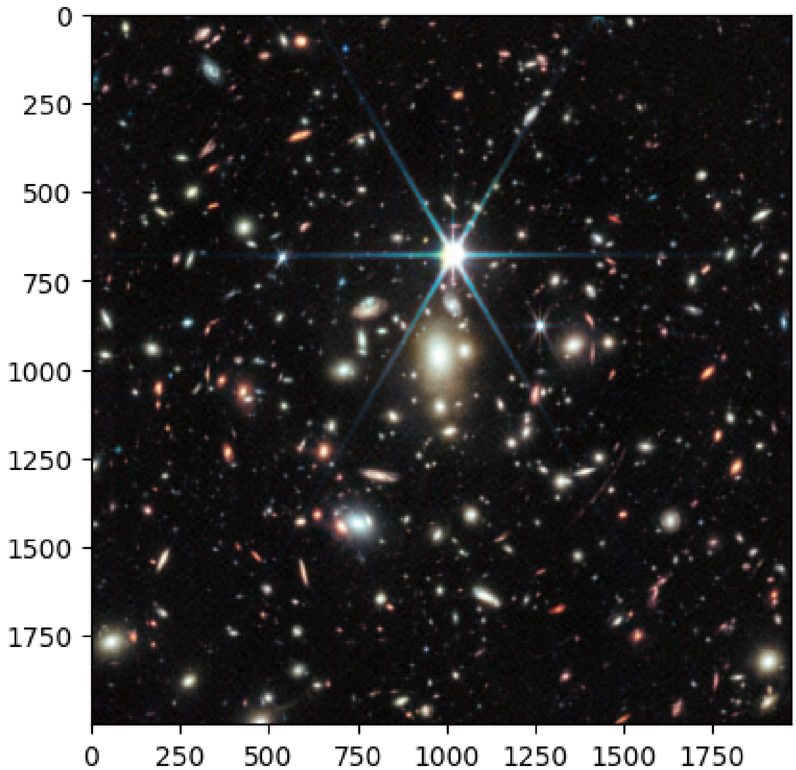
```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 7.420379161834717 and error in 2-norm using the theorem is :
7.420379161834717
Error in Frobenius norm is 106.73213520599646 and error in Frobenius norm using
theorem is : 106.73213503448383
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0009647480910643935,   for number of singular_values 100
**************************************
CASE:Number of singular values for approximation is 110
Error in 2 norm and Frobenius norm for  first color channel is:
6.644186496734619, and 62.644137844349665 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
6.644186496734619, and 62.64415864500036 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
6.087077617645264, and 50.67603289139996 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
6.087077617645264, and 50.67602310528738 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
6.312387943267822, and 59.475988600557926 respectively.(Using singular values as
per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
6.312387943267822, and 59.47599312389369 respectively.
```



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 6.644186496734619 and error in 2-norm using the theorem is :
6.644186496734619
Error in Frobenius norm is 100.14859742553966 and error in Frobenius norm using
theorem is : 100.14859727368042
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0008494030334986746,   for number of singular_values 110
**************************************
CASE:Number of singular values for approximation is 120
Error in 2 norm and Frobenius norm for  first color channel is:
5.971610069274902, and 59.377309330039 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
5.971610069274902, and 59.37729645981113 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
5.419720649719238, and 47.24579799443542 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
5.419720649719238, and 47.24578598698093 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
5.703122615814209, and 56.34368241008198 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
5.703122615814209, and 56.34367563477621 respectively.
```

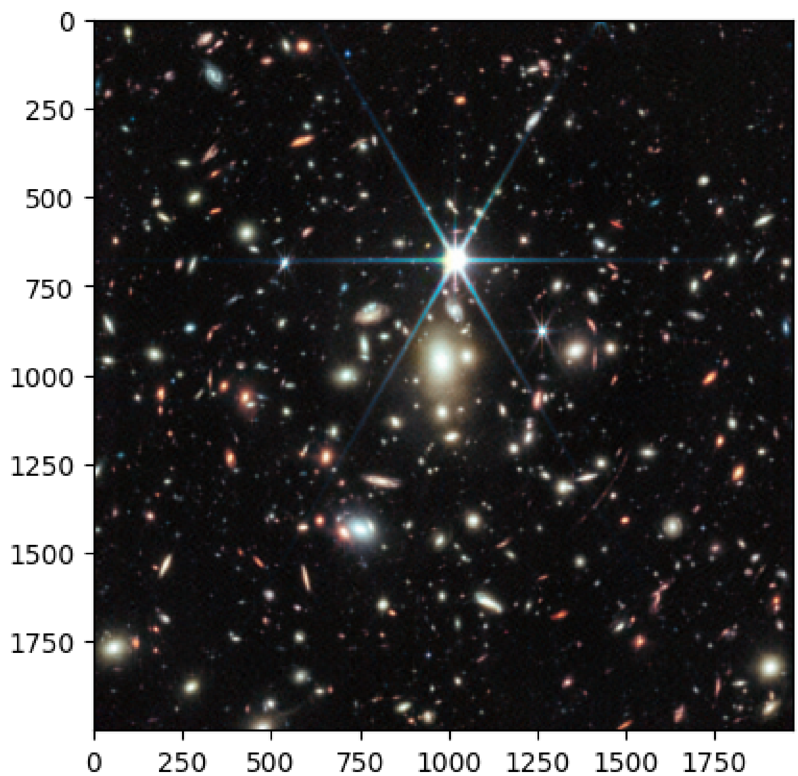


```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 5.971610069274902 and error in 2-norm using the theorem is :
5.971610069274902
Error in Frobenius norm is 94.51159124823194 and error in Frobenius norm using
theorem is : 94.51159102953623
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0007564733386971056,   for number of singular_values 120
**************************************
CASE:Number of singular values for approximation is 130
Error in 2 norm and Frobenius norm for  first color channel is:
5.486472129821777, and 56.52114167892452 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
5.486472129821777, and 56.52114249251624 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
4.902612686157227, and 44.2755145786999 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
Error in 2 norm and Frobenius norm for  second color channel is:
4.902612686157227, and 44.27551950635111 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
5.161121368408203, and 53.65368638929892 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
5.161121368408203, and 53.653697950439096 respectively.
```

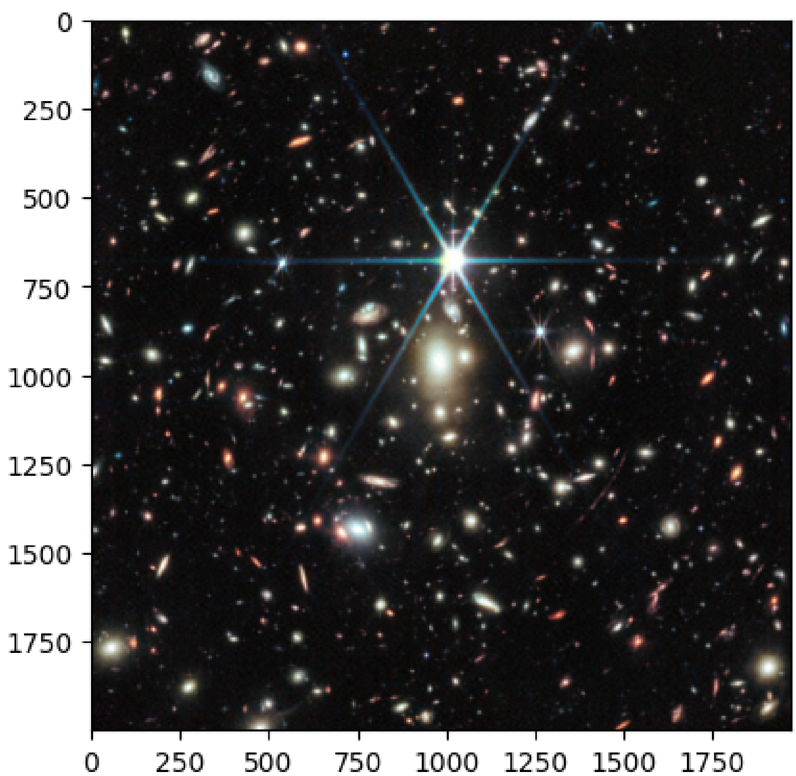



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 5.486472129821777 and error in 2-norm using the theorem is :
5.486472129821777
Error in Frobenius norm is 89.63079119593733 and error in Frobenius norm using
theorem is : 89.63079108799032
For the final approximated 3D tensor Mean Squared Error (MSE):
0.00068035873118788,   for number of singular_values 130
**************************************
CASE:Number of singular values for approximation is 150
Error in 2 norm and Frobenius norm for  first color channel is:
4.655424118041992, and 51.80594812062383 respectively.(Using singular values as
```

```
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
4.655424118041992, and 51.805959224264576 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
4.162152290344238, and 39.37060479887957 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
4.162152290344238, and 39.37060688551182 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
4.390741348266602, and 49.22660173814771 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
4.390741348266602, and 49.22659243819031 respectively.
```

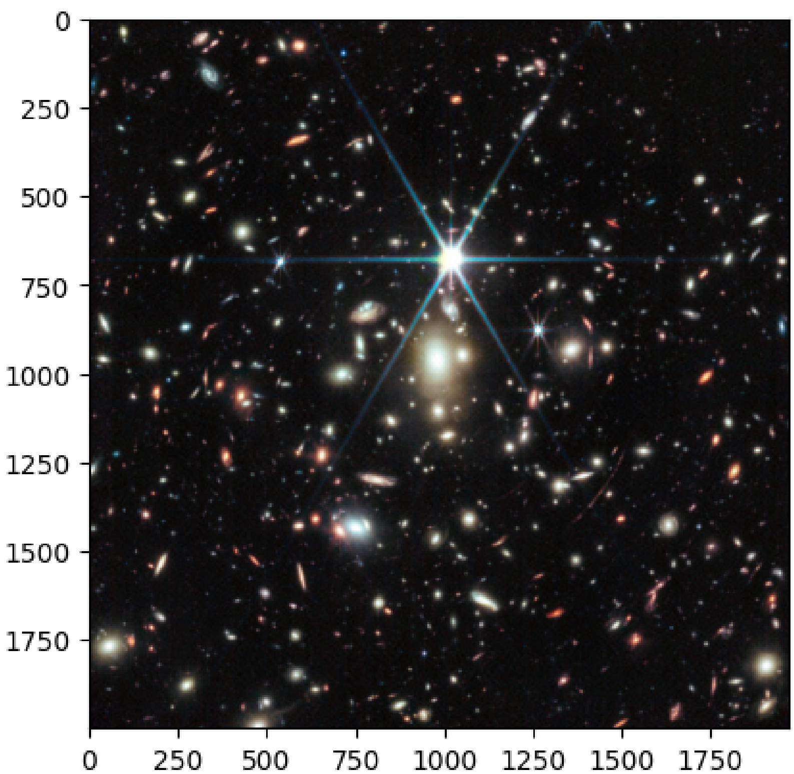



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 4.655424118041992 and error in 2-norm using the theorem is :
4.655424118041992
Error in Frobenius norm is 81.59141575243572 and error in Frobenius norm using
theorem is : 81.59141561213737
```

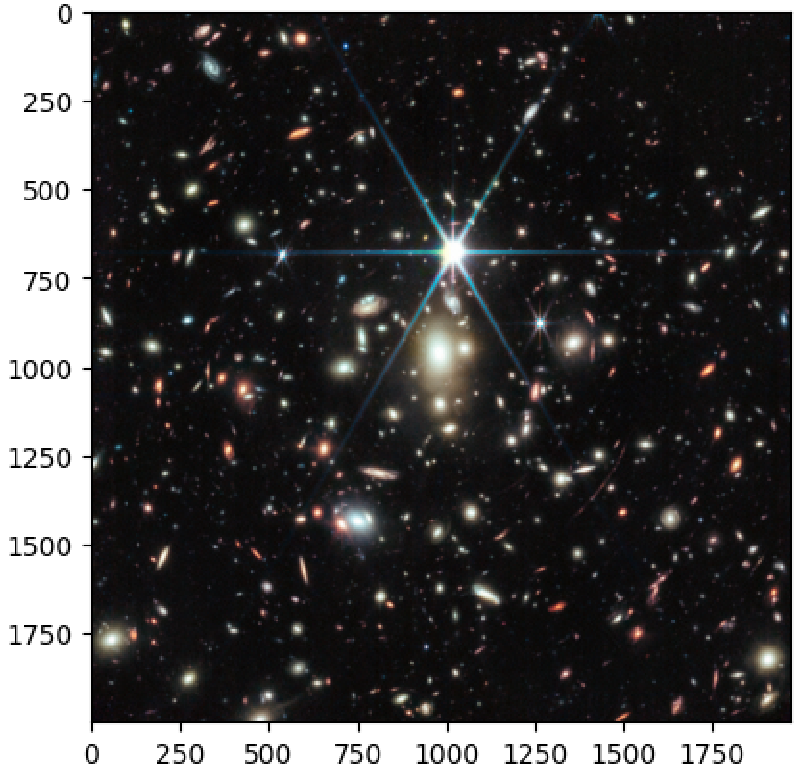
```
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0005637842696160078,   for number of singular_values 150
**************************************
CASE:Number of singular values for approximation is 180
Error in 2 norm and Frobenius norm for  first color channel is:
3.7444865703582764, and 46.48454291806275 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
3.7444865703582764, and 46.48454193277598 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
3.237455129623413, and 33.818432322781284 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
3.237455368041992, and 33.81842722267706 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
3.559849739074707, and 44.15227221114205 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
3.559849739074707, and 44.15227417702869 respectively.
```



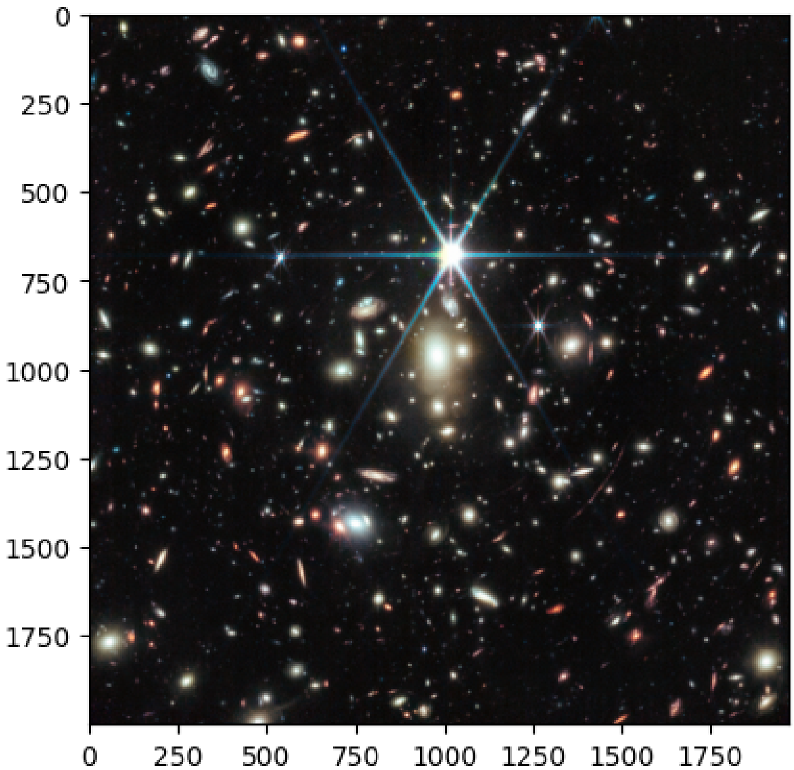
```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 3.7444865703582764 and error in 2-norm using the theorem is :
3.7444865703582764
Error in Frobenius norm is 72.4839447958198 and error in Frobenius norm using
theorem is : 72.48394468072597
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0004449457919690758,   for number of singular_values 180
**************************************
CASE:Number of singular values for approximation is 200
Error in 2 norm and Frobenius norm for  first color channel is:
3.3313539028167725, and 43.689289681499524 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
3.3313536643981934, and 43.68929742141339 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
2.803365707397461, and 30.979926270112106 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
2.803365707397461, and 30.979921650676257 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
3.144458770751953, and 41.51767131401518 respectively.(Using singular values as
per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
3.144458770751953, and 41.517671938864346 respectively.
```




```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 3.3313536643981934 and error in 2-norm using the theorem is :
3.3313539028167725
Error in Frobenius norm is 67.76597164088703 and error in Frobenius norm using
theorem is : 67.7659715190018
For the final approximated 3D tensor Mean Squared Error (MSE):
0.00038890799623914063,   for number of singular_values 200
**************************************
CASE:Number of singular values for approximation is 220
Error in 2 norm and Frobenius norm for  first color channel is:
2.978577136993408, and 41.330231030230976 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
2.978577136993408, and 41.330229423050966 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
2.469458818435669, and 28.634099067349485 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
2.469458818435669, and 28.634092294524933 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
2.850019693374634, and 39.292558541366496 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
2.850019693374634, and 39.29256246329988 respectively.
```

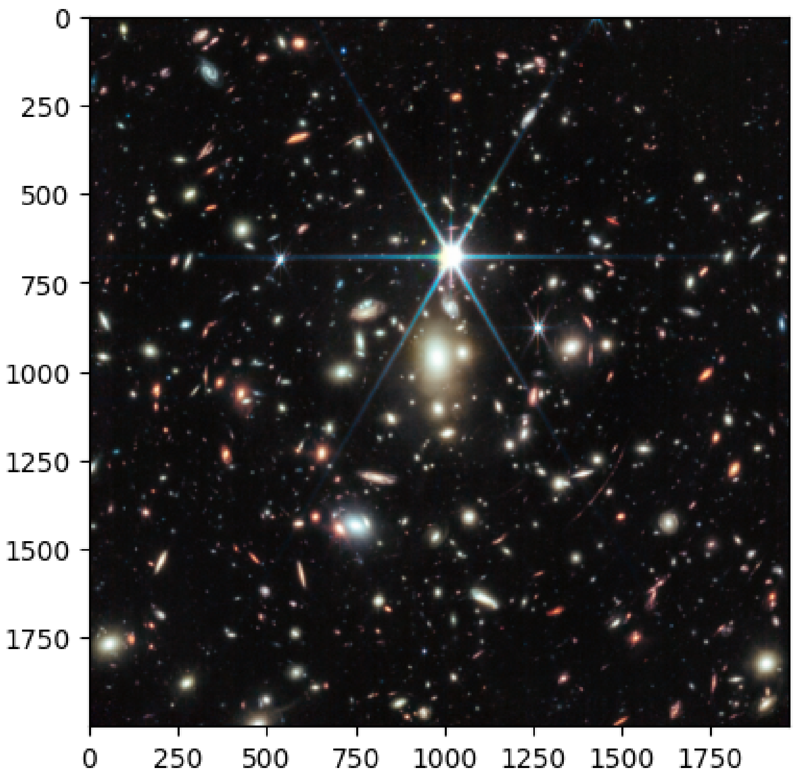



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 2.978577136993408 and error in 2-norm using the theorem is :
2.978577136993408
Error in Frobenius norm is 63.81226212395045 and error in Frobenius norm using
theorem is : 63.81226201238885
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0003448514617048204,   for number of singular_values 220
**************************************
CASE:Number of singular values for approximation is 250
Error in 2 norm and Frobenius norm for  first color channel is:
2.5744588375091553, and 38.421468793906904 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
2.5744588375091553, and 38.421467775418385 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
2.070561647415161, and 25.82443651084648 respectively.(Using singular values as
per the mathematical expression given in question)
```

Error in 2 norm and Frobenius norm for second color channel is:
2.070561647415161, and 25.824431918194435 respectively.
Error in 2 norm and Frobenius norm for third color channel is:
2.4573025703430176, and 36.53163756959382 respectively.(Using singular values as per the mathematical expression given in question)

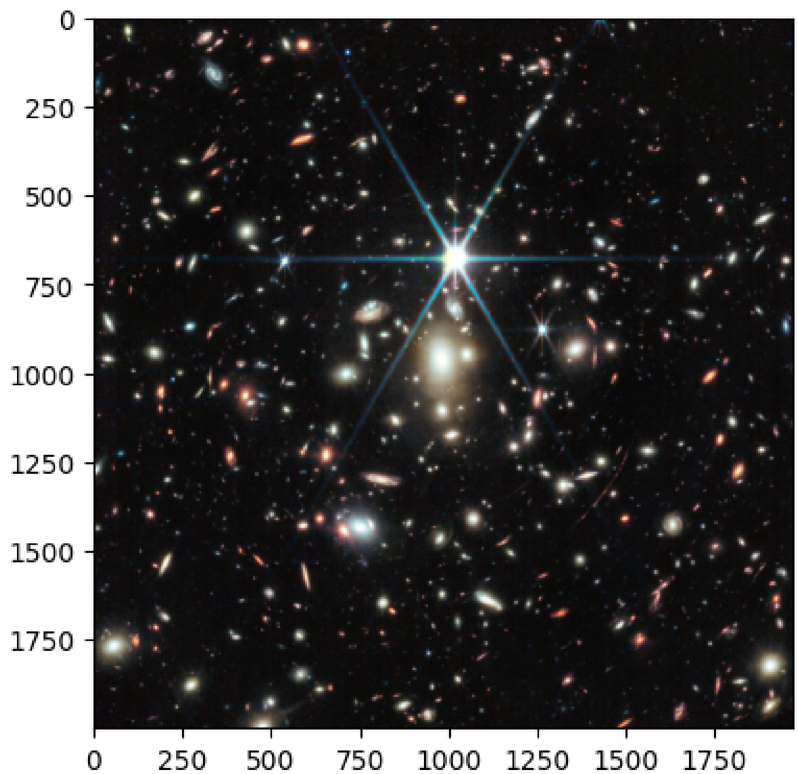
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for third color channel is:
2.4573025703430176, and 36.53164097001186 respectively.



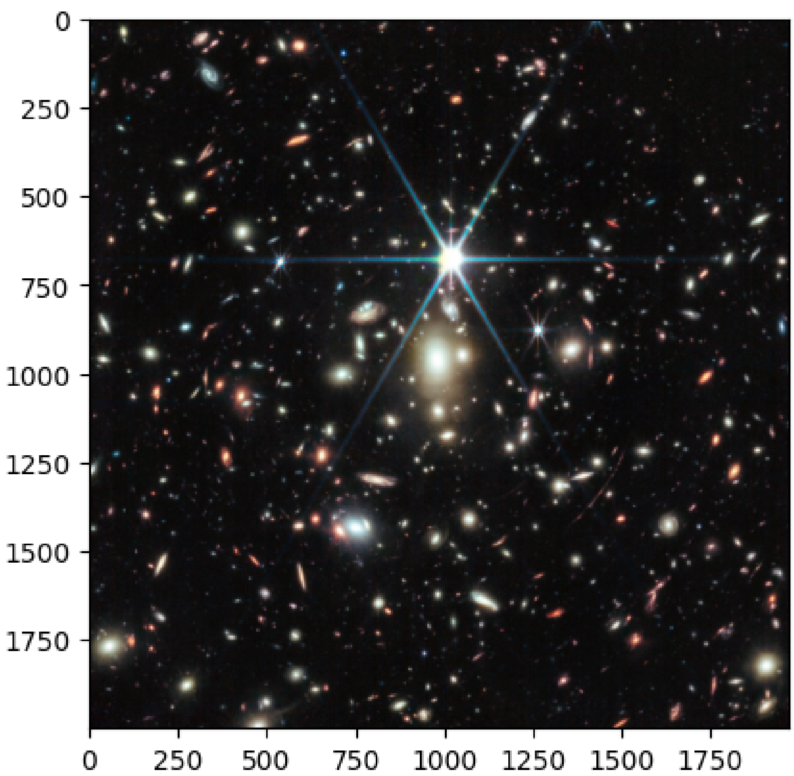
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 2.5744588375091553 and error in 2-norm using the theorem is :
2.5744588375091553
Error in Frobenius norm is 58.97178430170088 and error in Frobenius norm using theorem is : 58.97178417599438
For the final approximated 3D tensor Mean Squared Error (MSE):
0.00029451819136738777,   for number of singular_values 250
****
CASE:Number of singular values for approximation is 280
Error in 2 norm and Frobenius norm for first color channel is:
2.283099889755249, and 36.038571427090226 respectively.(Using singular values as

```
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
2.283099889755249, and 36.03857024233089 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
1.765142560005188, and 23.609241353500153 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
1.7651424407958984, and 23.609238484377368 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
2.1791226863861084, and 34.26461811242187 respectively.(Using singular values as
per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
2.1791226863861084, and 34.2646258961293 respectively.
```

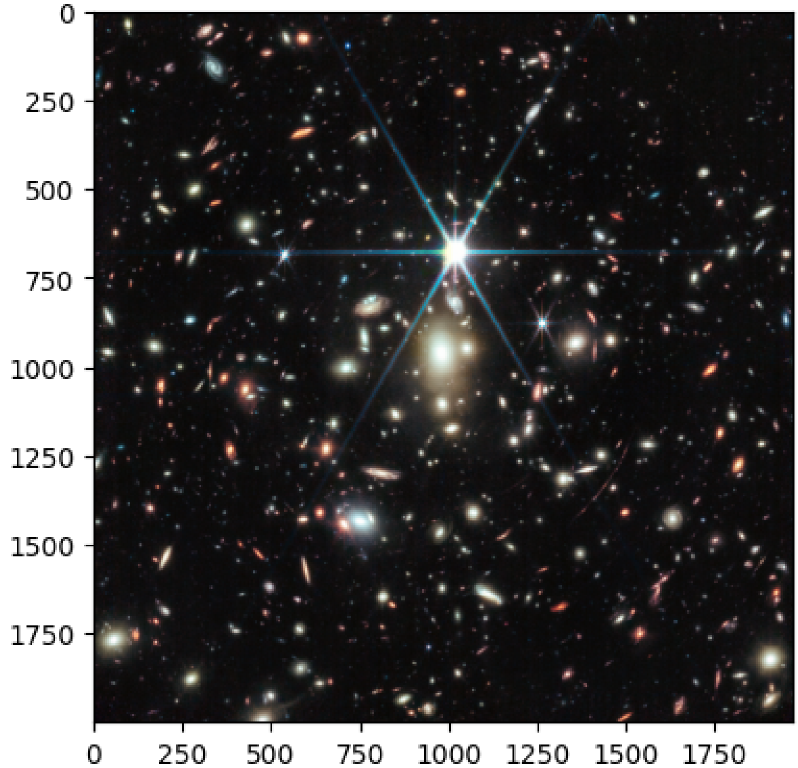



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 2.283099889755249 and error in 2-norm using the theorem is :
2.283099889755249
Error in Frobenius norm is 55.04760646446934 and error in Frobenius norm using
theorem is : 55.047606325647024
```

```
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0002566260809544474,   for number of singular_values 280
**************************************
CASE:Number of singular values for approximation is 300
Error in 2 norm and Frobenius norm for  first color channel is:
2.123112440109253, and 34.667255519189474 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
2.123112678527832, and 34.667259316506055 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
1.61300528049469, and 22.35855824948115 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
1.6130053997039795, and 22.358562198546544 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
2.03509259223938, and 32.93639564654318 respectively.(Using singular values as
per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
2.03509259223938, and 32.93639065608872 respectively.
```



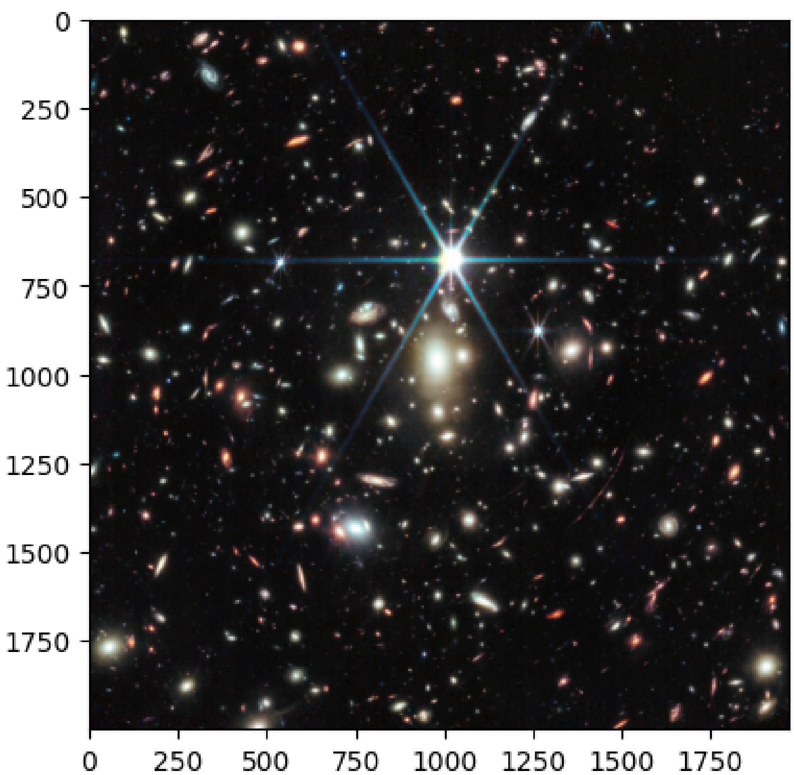
```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 2.123112678527832 and error in 2-norm using the theorem is :
2.123112440109253
Error in Frobenius norm is 52.78759233517079 and error in Frobenius norm using
theorem is : 52.78759220132932
For the final approximated 3D tensor Mean Squared Error (MSE):
0.00023598670668434352,   for number of singular_values 300
**************************************
CASE:Number of singular values for approximation is 350
Error in 2 norm and Frobenius norm for  first color channel is:
1.8313281536102295, and 31.750214652710373 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
1.83132803440094, and 31.75021865536648 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
1.3230299949645996, and 19.803166678069175 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
1.3230299949645996, and 19.803163258878055 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
1.7609654664993286, and 30.091057692598305 respectively.(Using singular values
as per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
1.760965347290039, and 30.09106006476662 respectively.
```




```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 1.83132803440094 and error in 2-norm using the theorem is :
1.8313281536102295
Error in Frobenius norm is 48.01784351390424 and error in Frobenius norm using
theorem is : 48.01784349626557
For the final approximated 3D tensor Mean Squared Error (MSE):
0.00019526730466168374,   for number of singular_values 350
**************************************
CASE:Number of singular values for approximation is 400
Error in 2 norm and Frobenius norm for  first color channel is:
1.6294286251068115, and 29.302792400716363 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
1.629428505897522, and 29.30278851908941 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
1.1178545951843262, and 17.81488366104214 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
1.1178544759750366, and 17.8148820815184 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
1.5534909963607788, and 27.717271973391266 respectively.(Using singular values
as per the mathematical expression given in question)
```

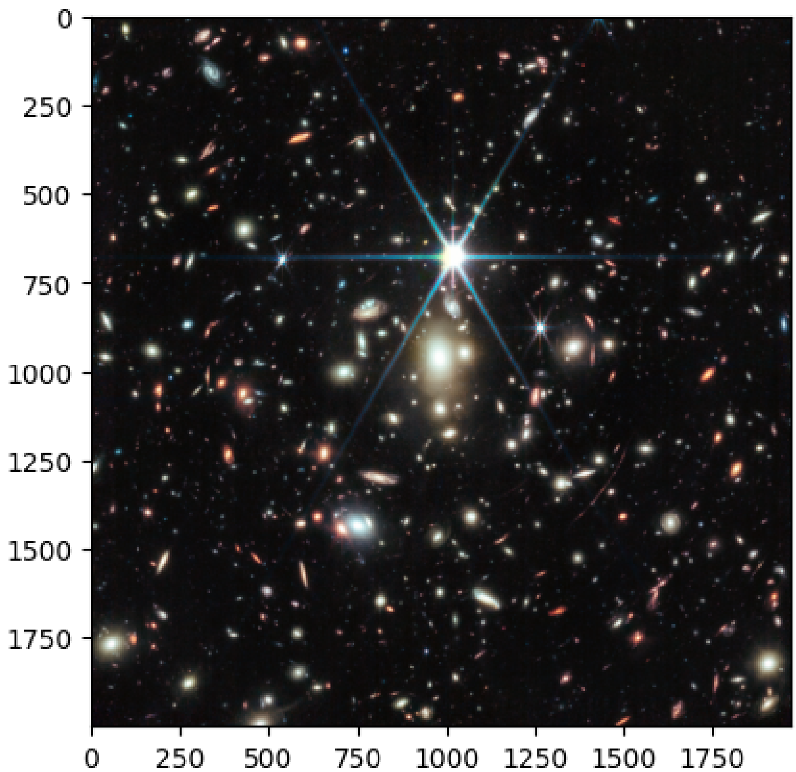
```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
1.5534909963607788, and 27.71727227706563 respectively.
```




```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 1.629428505897522 and error in 2-norm using the theorem is :
1.6294286251068115
Error in Frobenius norm is 44.093887211011804 and error in Frobenius norm using
theorem is : 44.09388719519848
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0001646571181481704,   for number of singular_values 400
**************************************
CASE:Number of singular values for approximation is 450
Error in 2 norm and Frobenius norm for  first color channel is:
1.4767993688583374, and 27.174015398826544 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
1.4767993688583374, and 27.174011241710147 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.9693855047225952, and 16.21973329278432 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
Error in 2 norm and Frobenius norm for  second color channel is:
0.9693855047225952, and 16.21973096528339 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
1.4025064706802368, and 25.671027857735403 respectively.(Using singular values
as per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
1.4025065898895264, and 25.671021218085514 respectively.
```

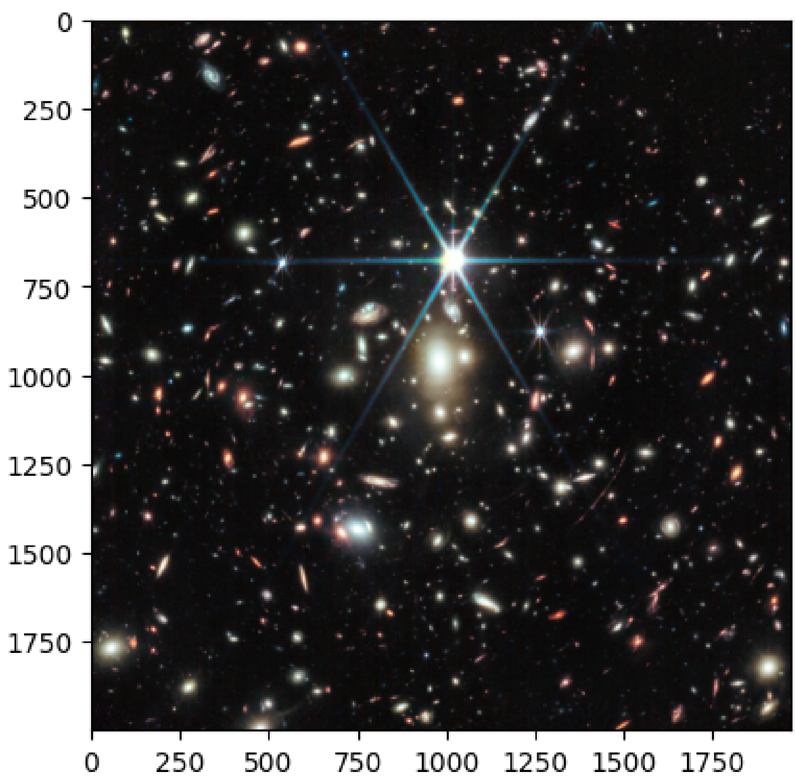



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 1.4767993688583374 and error in 2-norm using the theorem is :
1.4767993688583374
Error in Frobenius norm is 40.74933782930991 and error in Frobenius norm using
theorem is : 40.74933781372828
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0001406257360940799,   for number of singular_values 450
***************************************
CASE:Number of singular values for approximation is 500
Error in 2 norm and Frobenius norm for  first color channel is:
1.3519465923309326, and 25.267315762714894 respectively.(Using singular values
```

```
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
1.3519465923309326, and 25.267316677338773 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.8632491230964661, and 14.87359514929287 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.8632489442825317, and 14.87359163973562 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
1.2849781513214111, and 23.849319483144377 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
1.2849782705307007, and 23.84932244476332 respectively.
```

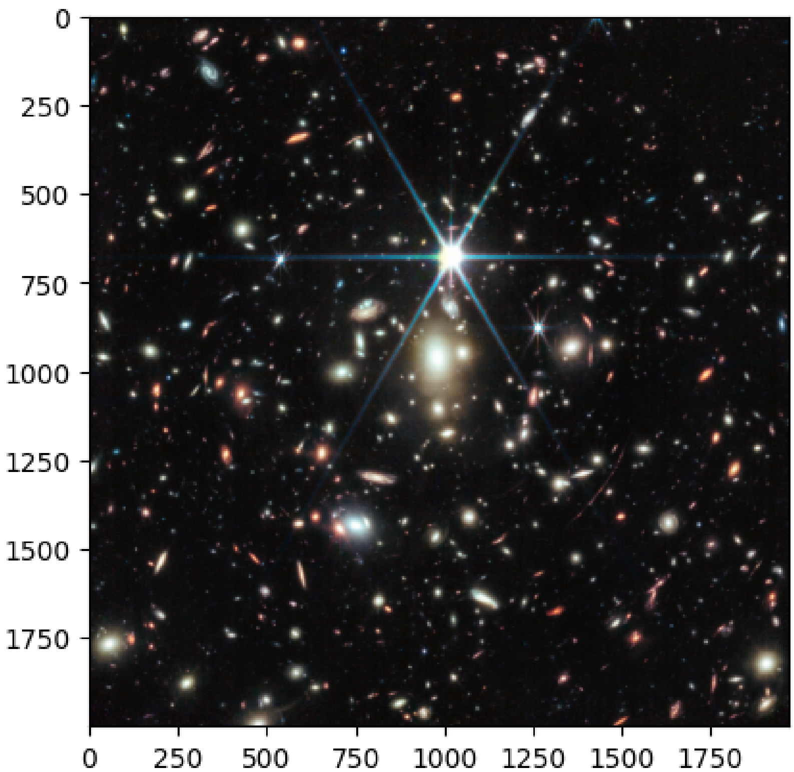



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 1.3519465923309326 and error in 2-norm using the theorem is :
1.3519465923309326
Error in Frobenius norm is 37.79485574062294 and error in Frobenius norm using
theorem is : 37.79485571247625
```

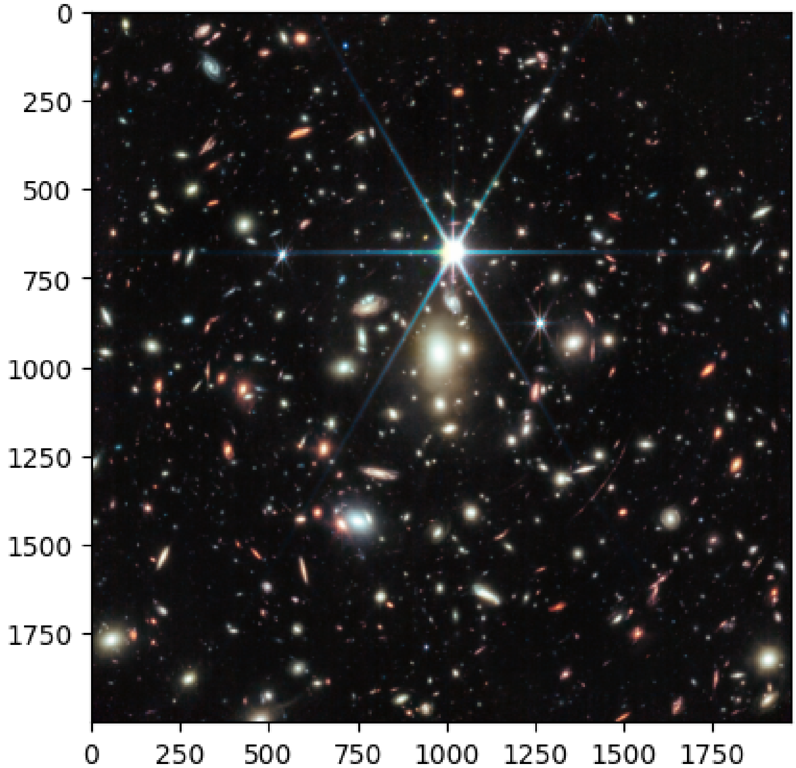
```
For the final approximated 3D tensor Mean Squared Error (MSE):
0.0001209731271956116,   for number of singular_values 500
**************************************
CASE:Number of singular values for approximation is 550
Error in 2 norm and Frobenius norm for  first color channel is:
1.2518247365951538, and 23.528869624615538 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
1.2518247365951538, and 23.528869776626728 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.7769363522529602, and 13.701090000286403 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.7769363522529602, and 13.701089308510998 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
1.1849969625473022, and 22.198405158520107 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
1.1849969625473022, and 22.19840506244308 respectively.
```



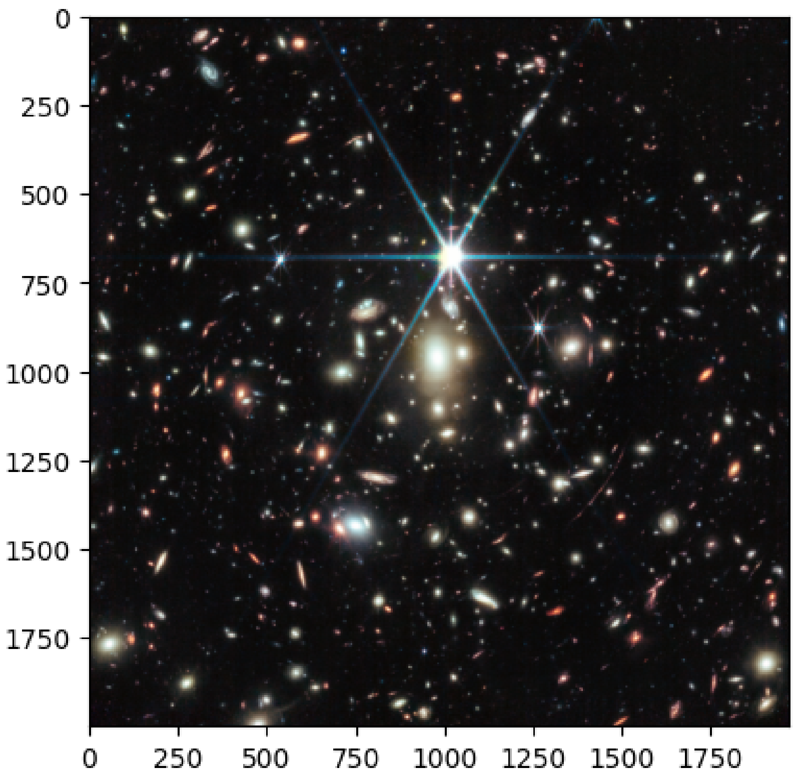
```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 1.2518247365951538 and error in 2-norm using the theorem is :
1.2518247365951538
Error in Frobenius norm is 35.129713471092195 and error in Frobenius norm using
theorem is : 35.12971341457137
For the final approximated 3D tensor Mean Squared Error (MSE):
0.00010451355774421245,   for number of singular_values 550
**************************************
CASE:Number of singular values for approximation is 600
Error in 2 norm and Frobenius norm for  first color channel is:
1.1659921407699585, and 21.920525780394453 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
1.165992021560669, and 21.920531530551724 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.7053496241569519, and 12.661345793992353 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.7053496241569519, and 12.661344352283436 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
1.0986214876174927, and 20.67955504797274 respectively.(Using singular values as
per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
1.0986214876174927, and 20.6795540888824 respectively.
```




```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 1.165992021560669 and error in 2-norm using the theorem is :
1.1659921407699585
Error in Frobenius norm is 32.68735422649164 and error in Frobenius norm using
theorem is : 32.687354202904224
For the final approximated 3D tensor Mean Squared Error (MSE):
9.04864355106838e-05,   for number of singular_values 600
**************************************
CASE:Number of singular values for approximation is 650
Error in 2 norm and Frobenius norm for  first color channel is:
1.0858559608459473, and 20.423138174328795 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
1.0858560800552368, and 20.42313241963868 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.6483625769615173, and 11.724521935557 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.6483625173568726, and 11.724522554689992 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
1.0242308378219604, and 19.26809953807487 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
1.0242308378219604, and 19.26809458250699 respectively.
```

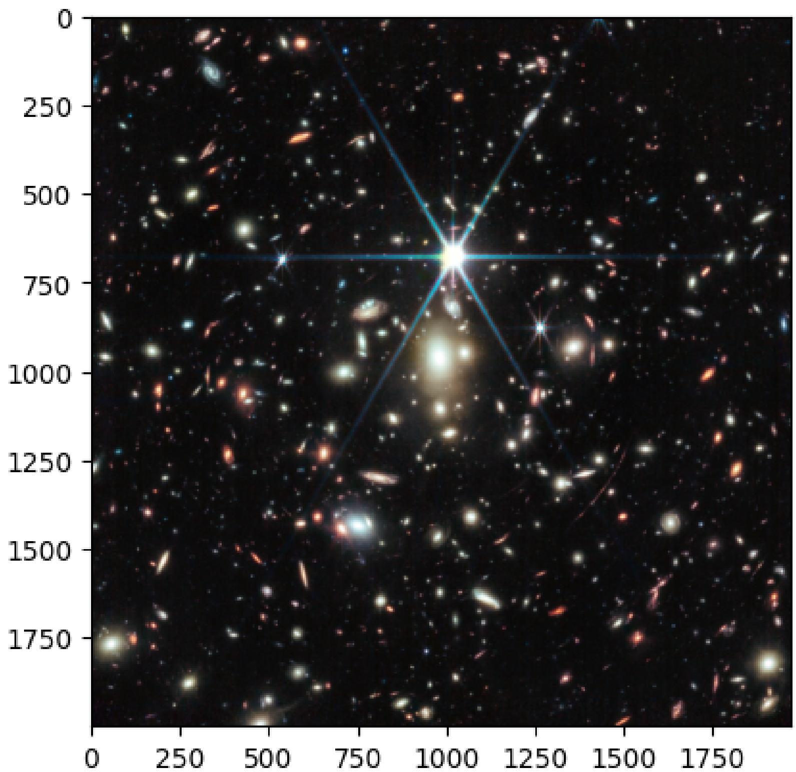


```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 1.0858560800552368 and error in 2-norm using the theorem is :
1.0858559608459473
Error in Frobenius norm is 30.42743257275703 and error in Frobenius norm using
theorem is : 30.42743247982393
For the final approximated 3D tensor Mean Squared Error (MSE):
7.840691250748932e-05,   for number of singular_values 650
**************************************
CASE:Number of singular values for approximation is 700
Error in 2 norm and Frobenius norm for  first color channel is:
1.0151889324188232, and 19.018957187439607 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
1.0151889324188232, and 19.018949454376532 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.5979960560798645, and 10.864730042537062 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
Error in 2 norm and Frobenius norm for  second color channel is:
0.597996175289154, and 10.864728413794149 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.9587940573692322, and 17.945563295506375 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
0.9587939977645874, and 17.945564519902398 respectively.
```

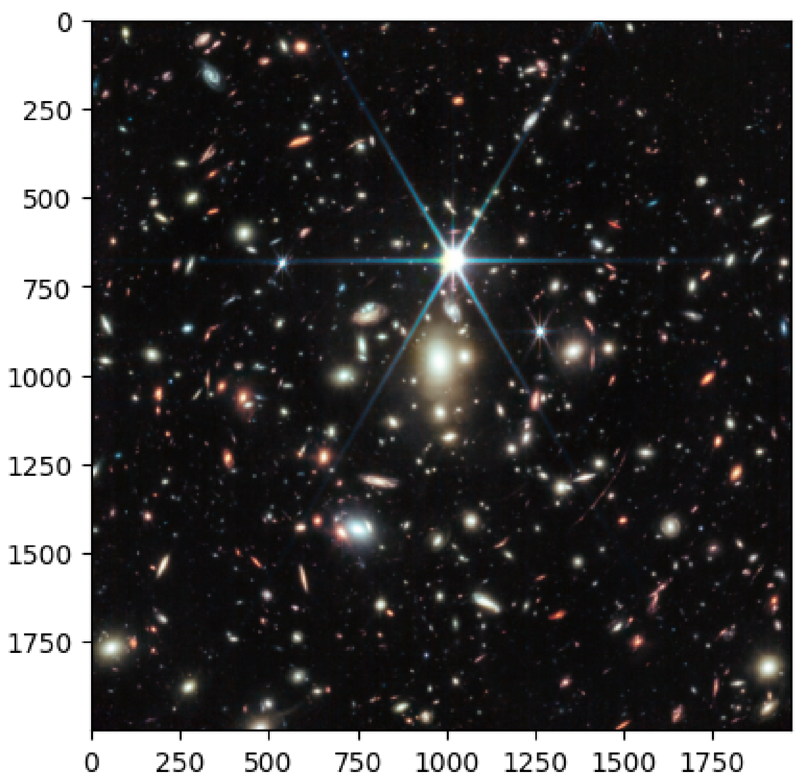



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 1.0151889324188232 and error in 2-norm using the theorem is :
1.0151889324188232
Error in Frobenius norm is 28.316185115640355 and error in Frobenius norm using
theorem is : 28.316185007657612
For the final approximated 3D tensor Mean Squared Error (MSE):
6.79036311339587e-05,   for number of singular_values 700
**************************************
CASE:Number of singular values for approximation is 800
Error in 2 norm and Frobenius norm for  first color channel is:
0.8950958847999573, and 16.438872843954595 respectively.(Using singular values
```

```
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.8950958847999573, and 16.438874054112276 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.516181230545044, and 9.326944674644631 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.5161811113357544, and 9.326945055622494 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.8430574536323547, and 15.522320109430066 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
0.8430574536323547, and 15.522320222651796 respectively.
```

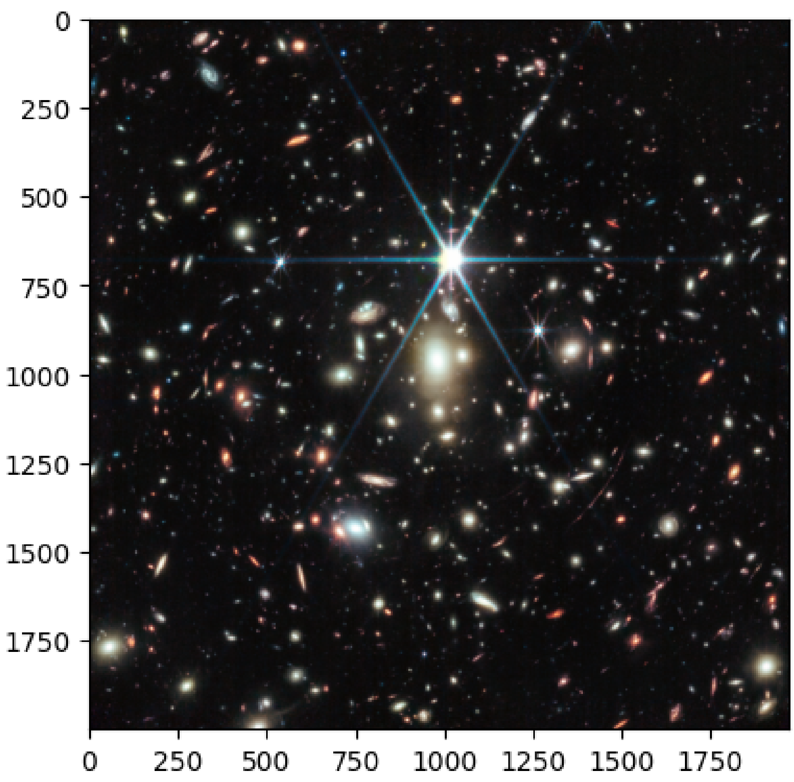



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.8950958847999573 and error in 2-norm using the theorem is :
0.8950958847999573
Error in Frobenius norm is 24.457531835459463 and error in Frobenius norm using
theorem is : 24.457531742250826
```

```
For the final approximated 3D tensor Mean Squared Error (MSE):
5.0658083637245e-05,   for number of singular_values 800
**************************************
CASE:Number of singular values for approximation is 900
Error in 2 norm and Frobenius norm for  first color channel is:
0.7895172834396362, and 14.114703482077216 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.7895174026489258, and 14.11470369534593 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.45059892535209656, and 7.972847369000911 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.4505990147590637, and 7.97284927623949 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.7442525029182434, and 13.339034278837222 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

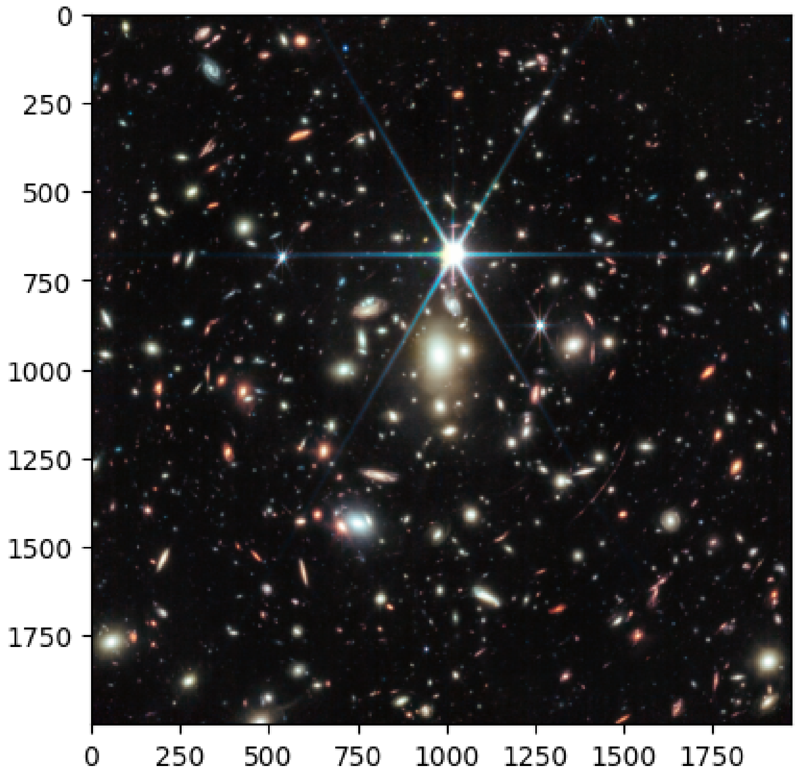
```
Error in 2 norm and Frobenius norm for  third color channel is:
0.7442524433135986, and 13.339029348536815 respectively.
```



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.7895174026489258 and error in 2-norm using the theorem is :
0.7895172834396362
Error in Frobenius norm is 20.993355865647782 and error in Frobenius norm using
theorem is : 20.99335573576416
For the final approximated 3D tensor Mean Squared Error (MSE):
3.732395271072164e-05,   for number of singular_values 900
**************************************
CASE:Number of singular values for approximation is 1000
Error in 2 norm and Frobenius norm for  first color channel is:
0.6951292157173157, and 12.00142276485655 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.6951293349266052, and 12.001424192687761 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.3943375051021576, and 6.761193722176726 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.39433753490448, and 6.761193827466924 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.6565554141998291, and 11.351628963159534 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

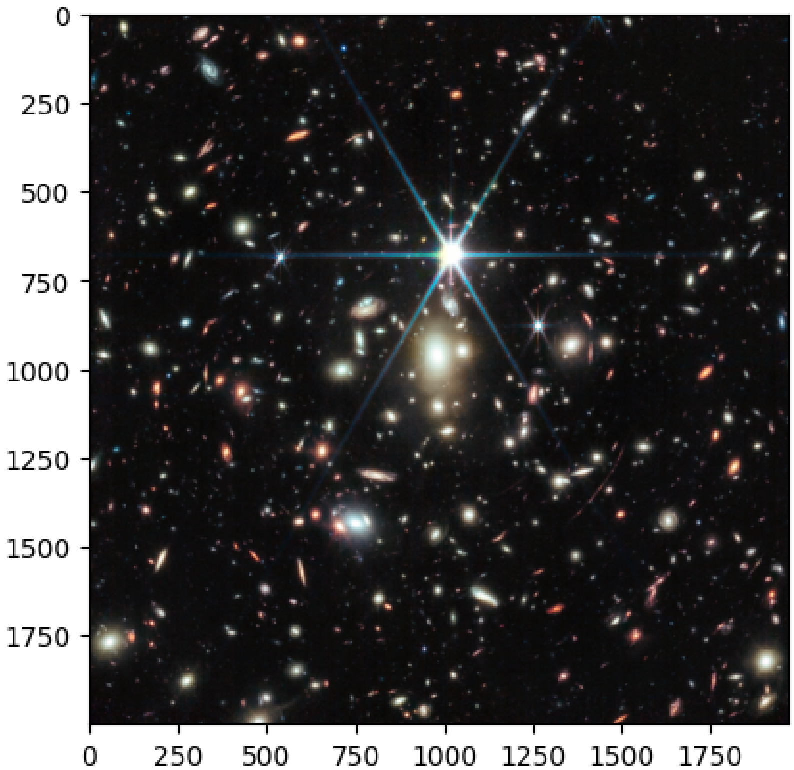
```
Error in 2 norm and Frobenius norm for  third color channel is:
0.6565553545951843, and 11.351627497838095 respectively.
```



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.6951293349266052 and error in 2-norm using the theorem is :
0.6951292157173157
Error in Frobenius norm is 17.849576270515957 and error in Frobenius norm using
theorem is : 17.84957615874566
For the final approximated 3D tensor Mean Squared Error (MSE):
2.6982319468515925e-05,   for number of singular_values 1000
**************************************
CASE:Number of singular values for approximation is 1100
Error in 2 norm and Frobenius norm for  first color channel is:
0.6099511384963989, and 10.069401174978461 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.6099511981010437, and 10.069399224635431 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.34380173683166504, and 5.664666348669683 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.3438017964363098, and 5.664665790196282 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.5747400522232056, and 9.534150282178135 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
0.5747401118278503, and 9.534148267576786 respectively.
```

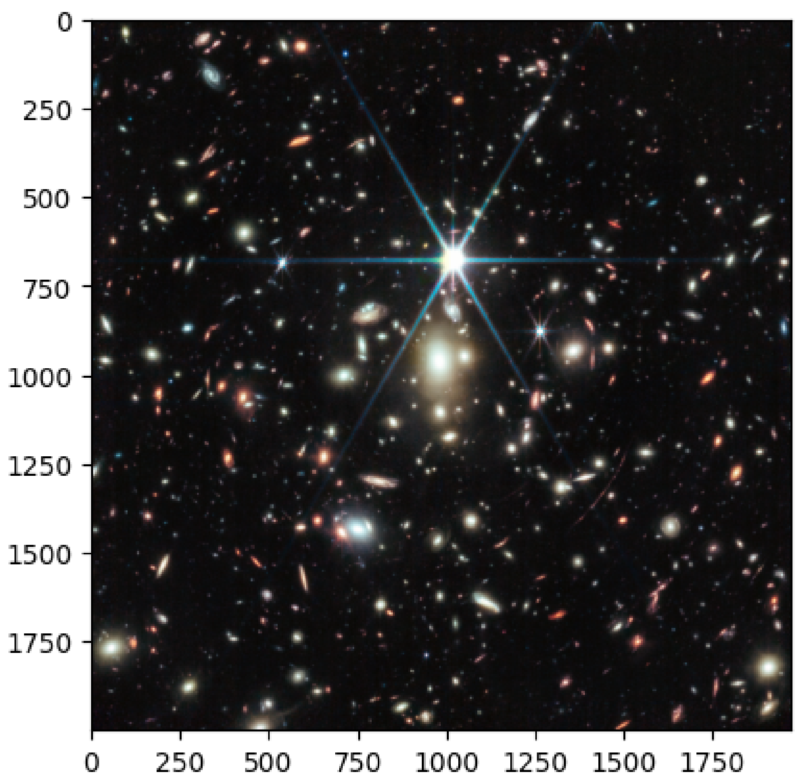



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.6099511981010437 and error in 2-norm using the theorem is :
0.6099511384963989
Error in Frobenius norm is 14.97936279038315 and error in Frobenius norm using
theorem is : 14.979362685627375
For the final approximated 3D tensor Mean Squared Error (MSE):
1.900247298181057e-05,   for number of singular_values 1100
**************************************
CASE:Number of singular values for approximation is 1200
Error in 2 norm and Frobenius norm for  first color channel is:
0.5287599563598633, and 8.308593115746634 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.5287599563598633, and 8.308595418845817 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.29719552397727966, and 4.668174981511943 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
Error in 2 norm and Frobenius norm for  second color channel is:
0.29719555377960205, and 4.668174128373724 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.5000472068786621, and 7.875584513442753 respectively.(Using singular values as
per the mathematical expression given in question)
```

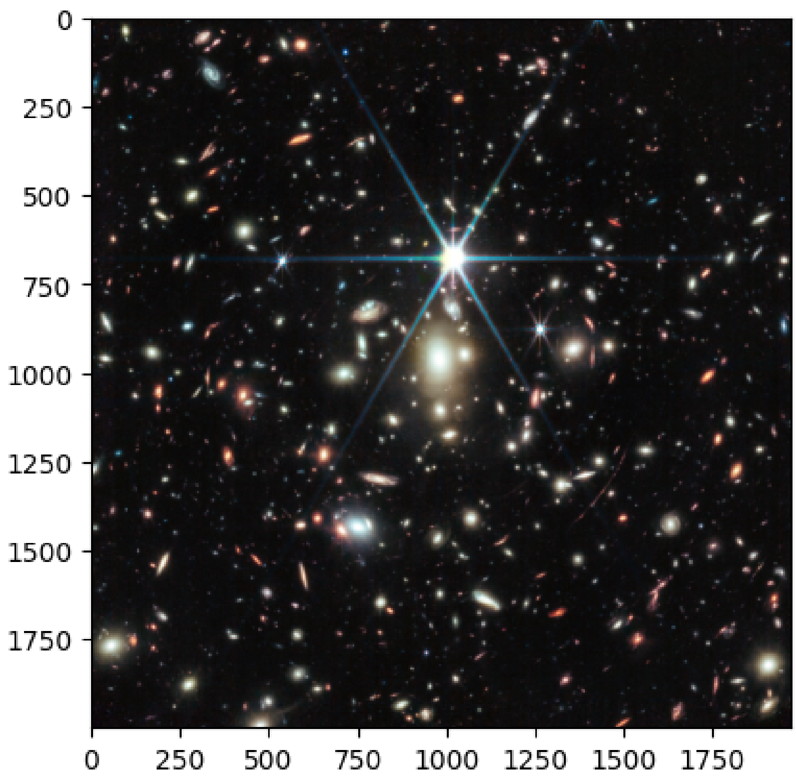
```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
0.5000470280647278, and 7.875585533020325 respectively.
```



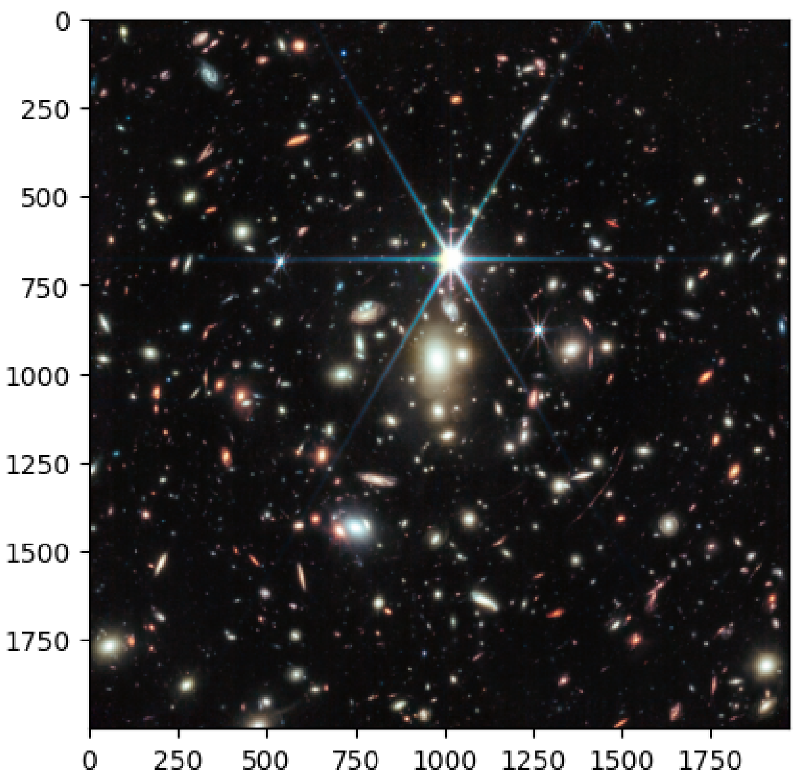

```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.5287599563598633 and error in 2-norm using the theorem is :
0.5287599563598633
Error in Frobenius norm is 12.36322815626749 and error in Frobenius norm using
theorem is : 12.363228083693421
For the final approximated 3D tensor Mean Squared Error (MSE):
1.2944567970407661e-05,   for number of singular_values 1200
***************************************
CASE:Number of singular values for approximation is 1250
Error in 2 norm and Frobenius norm for  first color channel is:
0.49042198061943054, and 7.485119259839002 respectively.(Using singular values
```

```
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.49042201042175293, and 7.485120602019451 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.27603182196617126, and 4.205058317220913 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.2760317921638489, and 4.205058150793818 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.463296502828598, and 7.09987362780056 respectively.(Using singular values as
per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
0.463296502828598, and 7.09987324936916 respectively.
```

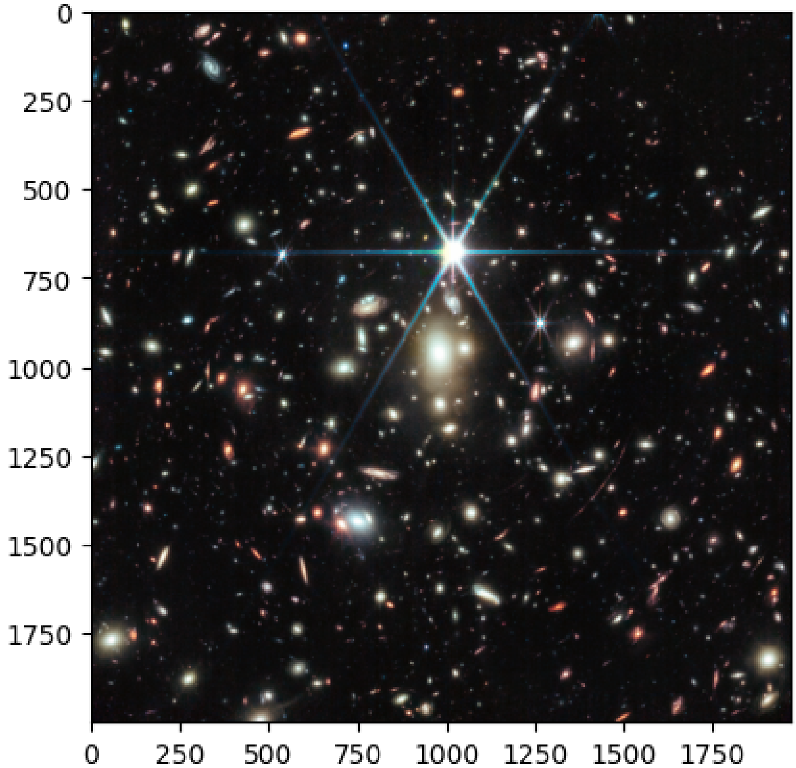



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.49042201042175293 and error in 2-norm using the theorem is
: 0.49042198061943054
Error in Frobenius norm is 11.140813842005846 and error in Frobenius norm using
theorem is : 11.14081376363412
```

```
For the final approximated 3D tensor Mean Squared Error (MSE):
1.0511334949114826e-05,   for number of singular_values 1250
**************************************
CASE:Number of singular values for approximation is 1300
Error in 2 norm and Frobenius norm for  first color channel is:
0.4522330164909363, and 6.699667208657451 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.45223307609558105, and 6.699670661779317 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.2547501027584076, and 3.7635552532498697 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.2547500729560852, and 3.763554769390967 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.4292741119861603, and 6.359013645705349 respectively.(Using singular values as
per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
0.42927423119544983, and 6.359013156945389 respectively.
```



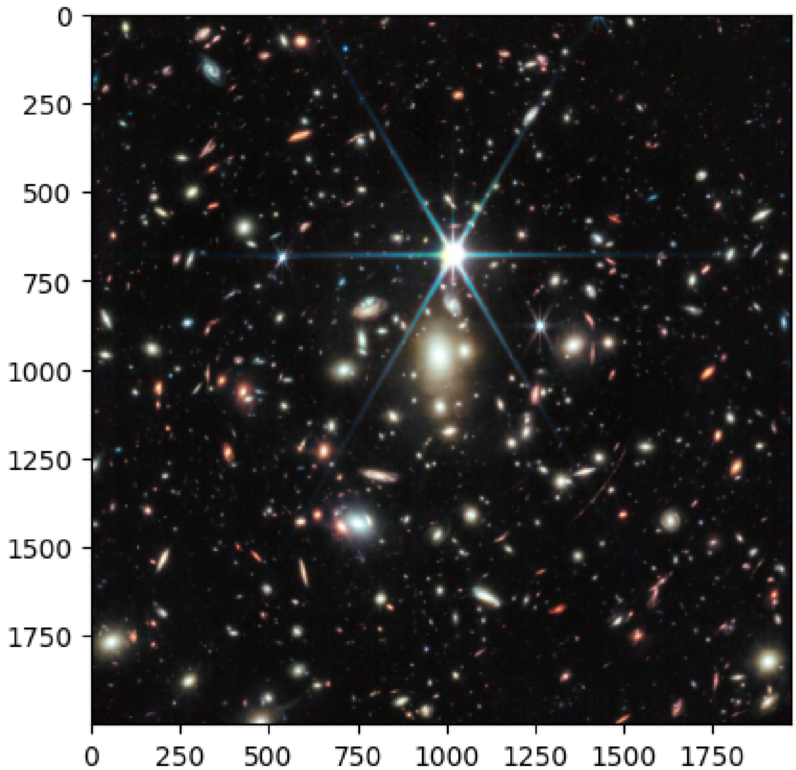
```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.45223307609558105 and error in 2-norm using the theorem is
: 0.4522330164909363
Error in Frobenius norm is 9.97431425959314 and error in Frobenius norm using
theorem is : 9.974314181801747
For the final approximated 3D tensor Mean Squared Error (MSE):
8.425383384746965e-06,   for number of singular_values 1300
**************************************
CASE:Number of singular values for approximation is 1350
Error in 2 norm and Frobenius norm for  first color channel is:
0.4164079427719116, and 5.9518790552688685 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.4164077937602997, and 5.951879029165782 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.23422709107398987, and 3.343301133306052 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.23422709107398987, and 3.3433017003786714 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.39562520384788513, and 5.6503908852133105 respectively.(Using singular values
as per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
0.3956252634525299, and 5.650392020012235 respectively.
```




```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.4164077937602997 and error in 2-norm using the theorem is :
0.4164079427719116
Error in Frobenius norm is 8.861684039741373 and error in Frobenius norm using
theorem is : 8.8616840336482
For the final approximated 3D tensor Mean Squared Error (MSE):
6.650524028373184e-06,   for number of singular_values 1350
**************************************
CASE:Number of singular values for approximation is 1400
Error in 2 norm and Frobenius norm for  first color channel is:
0.38038814067840576, and 5.24137940052198 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.3803882598876953, and 5.241379979377103 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.2145383656024933, and 2.942454104645005 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.21453839540481567, and 2.9424545822936055 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.36134982109069824, and 4.973925964543567 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
0.36134982109069824, and 4.973924554425162 respectively.
```

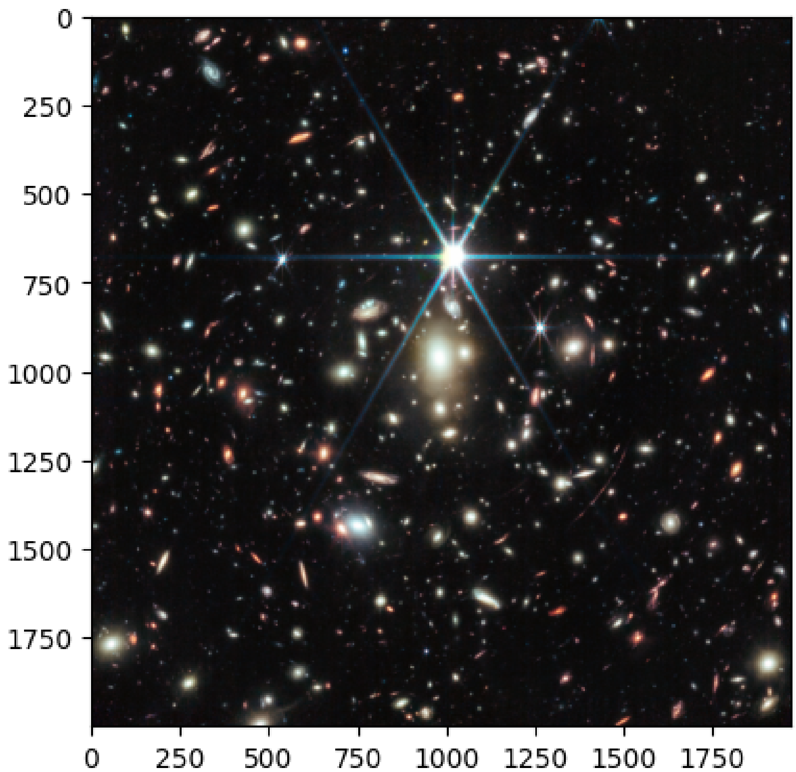



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.3803882598876953 and error in 2-norm using the theorem is :
0.38038814067840576
Error in Frobenius norm is 7.80192502939811 and error in Frobenius norm using
theorem is : 7.801924998288502
For the final approximated 3D tensor Mean Squared Error (MSE):
5.15498140885029e-06,   for number of singular_values 1400
**************************************
CASE:Number of singular values for approximation is 1450
Error in 2 norm and Frobenius norm for  first color channel is:
0.34610381722450256, and 4.566532812651046 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.3461037278175354, and 4.566533709337827 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.1947459578514099, and 2.5622415890190147 respectively.(Using singular values
as per the mathematical expression given in question)
```

```
Error in 2 norm and Frobenius norm for  second color channel is:
0.19474616646766663, and 2.5622413183845127 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.3286561071872711, and 4.332552467953613 respectively.(Using singular values as
per the mathematical expression given in question)
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
0.3286561965942383, and 4.332552949933323 respectively.
```

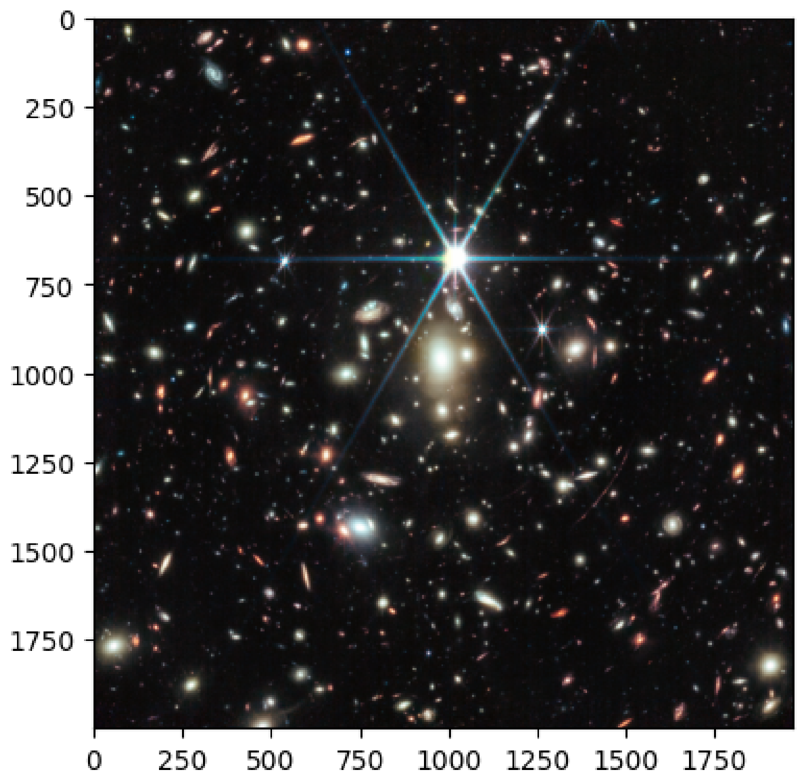



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.3461037278175354 and error in 2-norm using the theorem is :
0.34610381722450256
Error in Frobenius norm is 6.796272182690076 and error in Frobenius norm using
theorem is : 6.796272123531274
For the final approximated 3D tensor Mean Squared Error (MSE):
3.911697604053188e-06,   for number of singular_values 1450
**************************************
CASE:Number of singular values for approximation is 1500
Error in 2 norm and Frobenius norm for  first color channel is:
0.31338974833488464, and 3.9249627213442855 respectively.(Using singular values
```

```
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.31338974833488464, and 3.9249633757337916 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.17555226385593414, and 2.20186028768294 respectively.(Using singular values as
per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.17555221915245056, and 2.2018606027606 3 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.29652801156044006, and 3.7248002604614836 respectively.(Using singular values
as per the mathematical expression given in question)
```

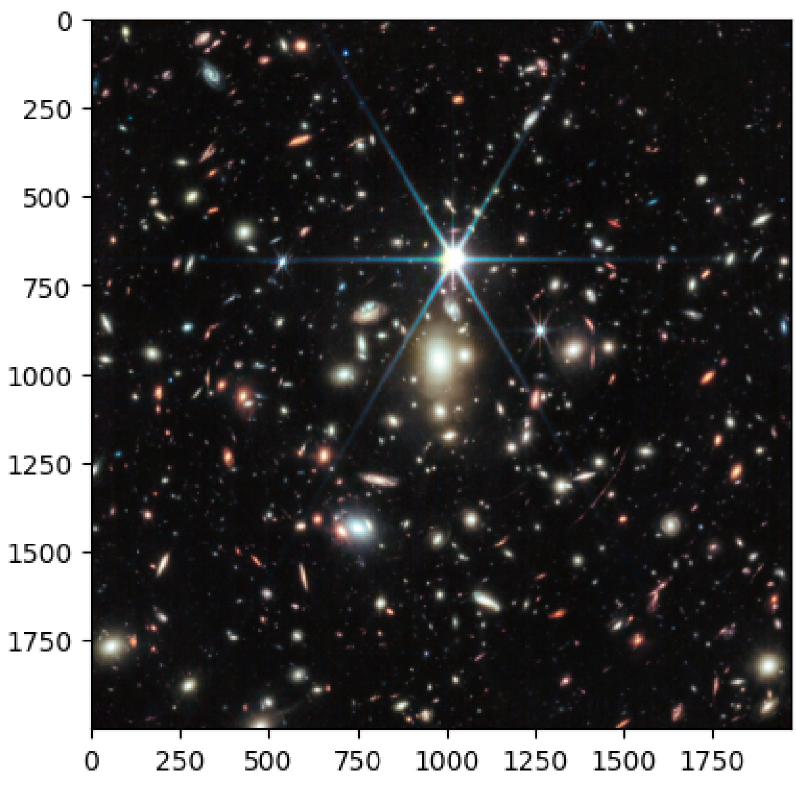
```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Error in 2 norm and Frobenius norm for  third color channel is:
0.29652804136276245, and 3.7248007342500755 respectively.
```




```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.31338974833488464 and error in 2-norm using the theorem is
: 0.31338974833488464
Error in Frobenius norm is 5.841888238200081 and error in Frobenius norm using
theorem is : 5.84188822819741
```

```
For the final approximated 3D tensor Mean Squared Error (MSE):
2.8902165922772838e-06,   for number of singular_values 1500
**************************************
CASE:Number of singular values for approximation is 1600
Error in 2 norm and Frobenius norm for  first color channel is:
0.24493606388568878, and 2.7540774532054986 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  first color channel is:
0.24493607878684998, and 2.7540781009982553 respectively.
Error in 2 norm and Frobenius norm for second color channel is:
0.13762377202510834, and 1.5457517352460848 respectively.(Using singular values
as per the mathematical expression given in question)
Error in 2 norm and Frobenius norm for  second color channel is:
0.13762390613555908, and 1.5457520340528432 respectively.
Error in 2 norm and Frobenius norm for  third color channel is:
0.23342539370059967, and 2.615523481235506 respectively.(Using singular values
as per the mathematical expression given in question)

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Error in 2 norm and Frobenius norm for  third color channel is:
0.2334253340959549, and 2.6155233413387715 respectively.
```



```
FOR THE 3D ERROR MATRIX:
Error in 2-norm is 0.24493607878684998 and error in 2-norm using the theorem is
: 0.24493606388568878
Error in Frobenius norm is 4.100640667485104 and error in Frobenius norm using
theorem is : 4.100640697033266
For the final approximated 3D tensor Mean Squared Error (MSE):
1.4240572454582434e-06,   for number of singular_values 1600
**************************************
```

[7]:
```python
import numpy as np
import math
from scipy.linalg import svd
import matplotlib.pyplot as plt

# We have the image matrix, painting of size 2000x1968x3
painting=plt.imread("WebbFirstDeepField.png")
# We will split the image matrix into its color channels
A1 = painting[:, :, 0]  # Red channel
B1 = painting[:, :, 1]  # Green channel
C1 = painting[:, :, 2]  # Blue channel

# Now A1, B1, and C1 contain the individual color channels as 2D matrices



S=[10,80,85,90,100,110,120,130,150,180,200,220,250,280,300,350,400,450,500,550,600,650,700
   ,800 ,900,1000,1100,1200,1250,1300,1350,1400,1450,1500,1600]
# Perform SVD on each color channel and retain only the top n singular values,␣
 ↪for each  n in above S.
for n in S:
    print(f"CASE:Number of singular values for approximation is {n}")
    num_singular_values_to_retain = n


    # For the red channel (A1)
    U_A, S_A, V_A = np.linalg.svd(A1, full_matrices=False)
    U_A_truncated = U_A[:, :num_singular_values_to_retain]
    p=len(S_A)
    #print(p)
    a=S_A[n]
    S_A_truncated = np.diag(S_A[:num_singular_values_to_retain])
    V_A_truncated = V_A[:num_singular_values_to_retain, :]
    A2 = np.dot(U_A_truncated, np.dot(S_A_truncated, V_A_truncated))

    sum1=0
    for t in range(n,p):
      sum1=sum1+(S_A[t])**2
```

```
    sum1=math.sqrt(sum1)
    #print(f"Error in 2 norm and Frobenius norm for  first color channel is:␣
↪{a}, and {sum1} respectively.(Using singular values as per the mathematical␣
↪expression given in question) ")
    error1=A1-A2
    matrix_norm1 = np.linalg.norm(error1, ord=2)
    fro1=np.mean(error1**2)
    fro1=math.sqrt(fro1*2000*1968)
    #print(f"Error in 2 norm and Frobenius norm for  first color channel is:␣
↪{matrix_norm1}, and {fro1} respectively.")
    if abs(a-matrix_norm1)<0.0001 and abs(sum1-fro1)<0.0001:
      print("Theorem satisfied for the first color channel for both Frobenius␣
↪as well as 2-norm")
    else:
      print("Theorem not satisfied")
    # For the Green channel (B1)
    U_B, S_B, V_B = np.linalg.svd(B1, full_matrices=False)
    U_B_truncated = U_B[:, :num_singular_values_to_retain]
    b=S_B[n]
    q=len(S_B)
    S_B_truncated = np.diag(S_B[:num_singular_values_to_retain])
    V_B_truncated = V_B[:num_singular_values_to_retain, :]
    B2 = np.dot(U_B_truncated, np.dot(S_B_truncated, V_B_truncated))

    sum2=0
    for t in range(n,q):
      sum2=sum2+(S_B[t])**2
    sum2=math.sqrt(sum2)
    #print(f"Error in 2 norm and Frobenius norm for second color channel is:␣
↪{b}, and {sum2} respectively.(Using singular values as per the mathematical␣
↪expression given in question) ")
    error2=B1-B2
    matrix_norm2 = np.linalg.norm(error2, ord=2)
    fro2=np.mean(error2**2)
    fro2=math.sqrt(fro2*2000*1968)
    #print(f"Error in 2 norm and Frobenius norm for  second color channel is:␣
↪{matrix_norm2}, and {fro2} respectively.")
    if abs(b-matrix_norm2)<0.0001 and abs(sum2-fro2)<0.0001:
      print("Theorem satisfied for the second color channel for both Frobenius␣
↪as well as 2-norm")
    else:
      print("Theorem not satisfied")
    # For the Blue channel (C1)
    U_C, S_C, V_C = np.linalg.svd(C1, full_matrices=False)
    U_C_truncated = U_C[:, :num_singular_values_to_retain]
    c=S_C[n]
```

```
    r=len(S_C)
    S_C_truncated = np.diag(S_C[:num_singular_values_to_retain])
    V_C_truncated = V_C[:num_singular_values_to_retain, :]
    C2 = np.dot(U_C_truncated, np.dot(S_C_truncated, V_C_truncated))

    sum3=0
    for t in range(n,r):
      sum3=sum3+(S_C[t])**2
    sum3=math.sqrt(sum3)
    #print(f"Error in 2 norm and Frobenius norm for  third color channel is:␣
↪{c}, and {sum3} respectively.(Using singular values as per the mathematical␣
↪expression given in question) ")
    error3=C1-C2
    matrix_norm3 = np.linalg.norm(error3, ord=2)
    fro3=np.mean(error3**2)
    fro3=math.sqrt(fro3*2000*1968)
    #print(f"Error in 2 norm and Frobenius norm for  third color channel is:␣
↪{matrix_norm3}, and {fro3} respectively.")
    if abs(c-matrix_norm3)<0.0001 and abs(sum3-fro3)<0.0001:
      print("Theorem satisfied for the third color channel for both Frobenius␣
↪as well as 2-norm")
    else:
      print("Theorem not satisfied")
    # Combining the reconstructed channels and creating the approximated image
    approximated_image_matrix = np.stack((A2, B2, C2), axis=-1)
    # Convert the approximated_image_matrix to an image for comparison purpose
    image = plt.imshow(approximated_image_matrix)

    # Save the image as "image1.png"
    #plt.savefig(f"image{n}.png")

    # Show the image
    #plt.show()




    # Calculate the Mean Squared Error (MSE) between the original image and the␣
↪approximated image
    mse = np.mean((painting - approximated_image_matrix) ** 2)
    #Calculating Frobenius norm of the error-tensor
    error_matrix = painting - approximated_image_matrix
    sum=0
    for i in range(2000):
      for j in range(1968):
        for k in range(3):
          sum=sum+(error_matrix[i][j][k]*error_matrix[i][j][k])
    frobenius_norm_error=math.sqrt(sum)
```

```python
    # Display the approximated image


    # Print the Mean Squared Error
    print("FOR THE 3D ERROR MATRIX:")
    #print(f"Error in 2-norm is {max(matrix_norm1,matrix_norm2,matrix_norm3)}␣
↪and error in 2-norm using the theorem is : {max(a,b,c)}")
    #print(f"Error in Frobenius norm is {frobenius_norm_error} and error in␣
↪Frobenius norm using theorem is : {math.
↪sqrt((sum1**2)+(sum2**2)+(sum3**2))}")
    #print(f"For the final approximated 3D tensor Mean Squared Error (MSE):␣
↪{mse},   for number of singular_values {num_singular_values_to_retain} ")
    if abs(max(matrix_norm1,matrix_norm2,matrix_norm3)-max(a,b,c))<0.0001 and␣
↪abs(frobenius_norm_error-math.sqrt((sum1**2)+(sum2**2)+(sum3**2)))<0.0001:
      print("Theorem satisfied for the 3D error matrix  for both Frobenius as␣
↪well as 2-norm")
    else:
      print("Theorem is not satisfied")
    print(f"**************************************")
```

```
CASE:Number of singular values for approximation is 10
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 80
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 85
```

```
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 90
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 100
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 110
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
```

```
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 120
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 130
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 150
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 180
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 200
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 220
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 250
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 280
```

```
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 300
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 350
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 400
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
```

```
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 450
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 500
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 550
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 600
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 650
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 700
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 800
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 900
```

```
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1000
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1100
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

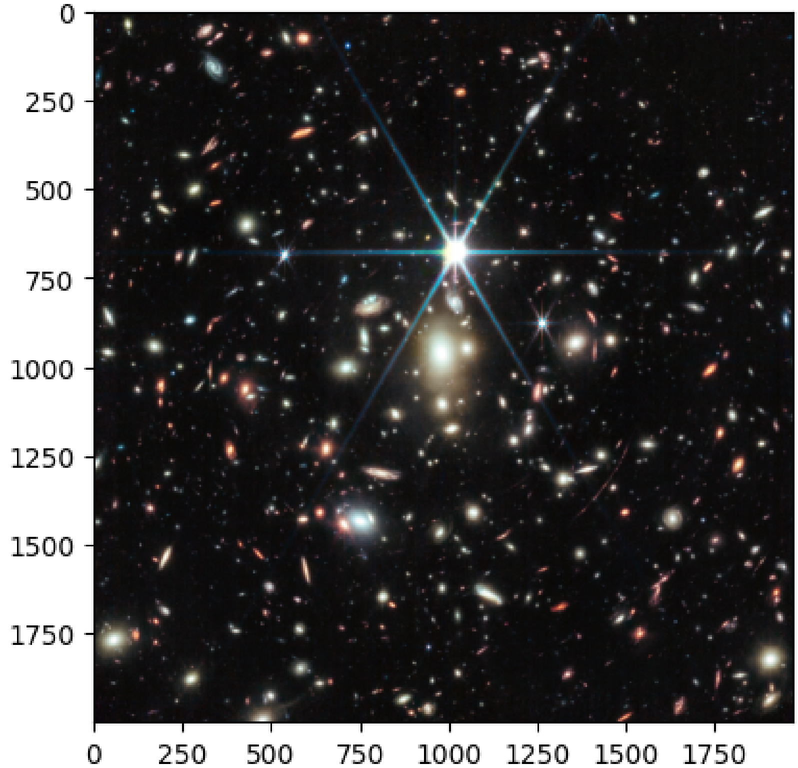
```
Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1200
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).
```

```
Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
```

```
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1250
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1300
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1350
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1400
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm
```

```
WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1450
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1500
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
CASE:Number of singular values for approximation is 1600
Theorem satisfied for the first color channel for both Frobenius as well as
2-norm
Theorem satisfied for the second color channel for both Frobenius as well as
2-norm

WARNING:matplotlib.image:Clipping input data to the valid range for imshow with
RGB data ([0..1] for floats or [0..255] for integers).

Theorem satisfied for the third color channel for both Frobenius as well as
2-norm
FOR THE 3D ERROR MATRIX:
Theorem satisfied for the 3D error matrix  for both Frobenius as well as 2-norm
**************************************
```



2.c)

For different number of singular values, we calculated 2-norms and Frobenius norms of the matrix representation of each color channel of the error tensor(Actual image tensor-approximated tensor) and from the outcome of the above code it is visible that for each color channel our calculated 2-norms and Frobenius norms are same(under absolute error bound of 10^{-4}.) as the 2-norms and Frobenius norms respectively, derived using singular values as per the given theorem in the question.

Later for different singular values, to find the 2-norm of the error tensor, we considered the maximum of 2-norms of the corresponding color channels.As for each channel, computed 2-norm agreeing with(under the above mentioned error bound) the given theorem, taking max value to find the same for the error tensor, will also agree(under the above mentioned error bound) with the norm value computed using given theorem.[Using:for real umbers a1,a2,a3,b1,b2,b3, if a1=b1,a2=b2,a3=b3, then max(a1,a2,a3)=max(b1,b2,b3)]

To find the Frobenius norm of the 3D error matrix A of dimension $2000 * 1968 * 3$, we used the formula:$\sqrt{\sum_{i=1}^{2000} \sum_{j=1}^{1968} \sum_{k=1}^{3} a_{ijk}^2}$, and as per the theorem it is(when we used n singular values for approximation and let k_i be the sum of the square of the singular values except first n singular values of the matrix corresponding to the i th color channel) $\sqrt{\sum_{i=1}^{3} k_i}$.From the output of the above code it is clear that, computed Frobenius norm using both the formula above are same(under the above mentioned error bound).

Also if someone use the maximum of Frobenius norms of each color channel to calculate the Frobenius norm of the 3D error matrix, then also the theorem will be satisfied(under the above mentioned error bound), using the argument as above in the case of 2-norm.

Thus given theorem holds for he 2-Norm and Frobenius-Norm error between the matrix representation of the original image and the approximate image obtained for different number of singular values.