AMATH301_Homework8_writeup

March 10, 2023

1 Homework 8 writeup solutions

1.1 Name: Aqua Karaman

1.2 Problem 1

[ ]:

```
import numpy as np
import matplotlib.pyplot as plt
import cv2
import scipy.linalg
```

1.3 Load in the image of Olive's perfect paws.

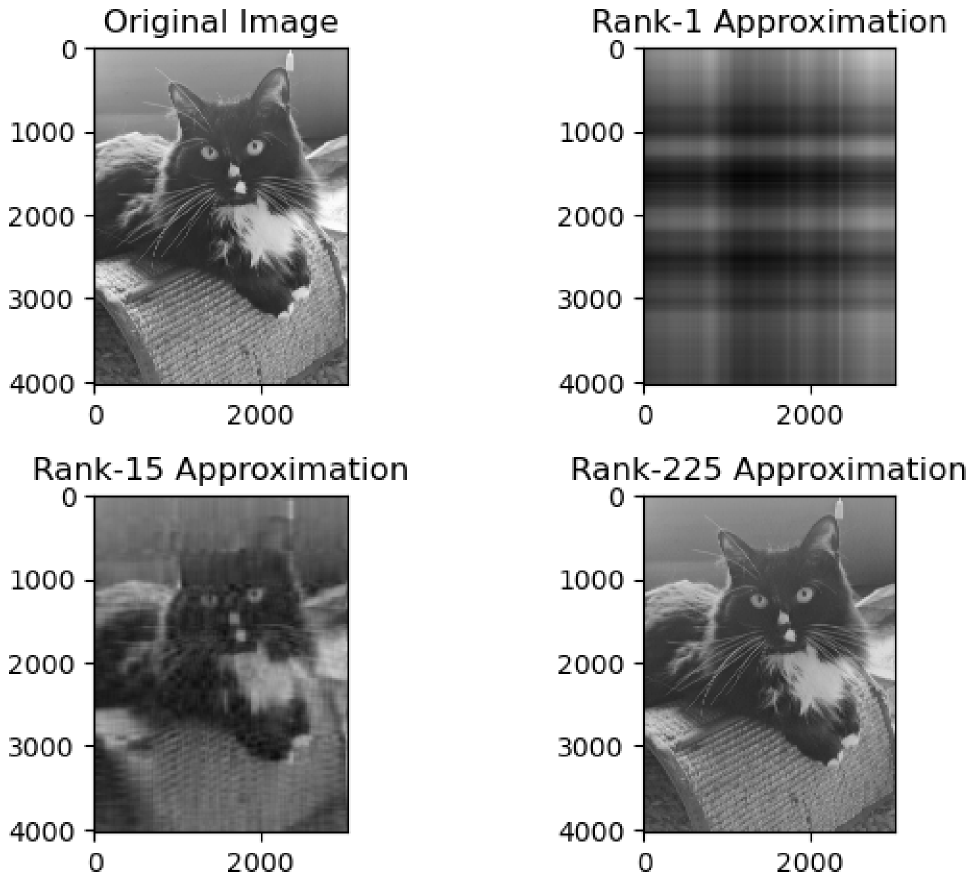
[ ]:

```
A = cv2.imread('olive.jpg', 0) # Remove the 0 if you want
                               # to see the color version!

MP = 4032 * 3024

U, S, Vt = np.linalg.svd(A, full_matrices=False) # Vt = V transpose

total_energy = np.sum(S)
S_mat = np.diag(S)
rank_1 = (U[:,0:1]@S_mat[0:1, 0:1])@Vt[0:1, :]
rank_15 = (U[:,0:15]@S_mat[0:15, 0:15])@Vt[0:15, :]

r_array = np.arange(3024)
r_true = np.where(np.cumsum(S[r_array])/total_energy >= 0.75, r_array, 0)
r_true_indices = np.nonzero(r_true)
weird_thing = r_true_indices[0] # no idea if this is the most efficient means of doing this but it worked. also i have weird_thing here bc it saves as int64 so i need to index it twice ?? weird. a weird thing even
r = weird_thing[0]

rank_r = (U[:,0:r]@S_mat[0:r, 0:r])@Vt[0:r, :]
```

1.4 Part (a) - 2x2 grid

[ ]:

```
fig, ax = plt.subplots(2, 2)

ax[0,0].imshow(A, cmap='gray')
ax[0,0].set_title('Original Image')

ax[0,1].imshow(rank_1, cmap='gray')
ax[0,1].set_title('Rank-1 Approximation')

ax[1,0].imshow(rank_15, cmap='gray')
ax[1,0].set_title('Rank-15 Approximation')

ax[1,1].imshow(rank_r, cmap='gray')
ax[1,1].set_title('Rank-225 Approximation')

fig.tight_layout(pad=1.0)
fig.savefig('rank_approximations.svg')
```



1.5 Part b - Calculate the total number of pixels for the image and its approximation.

[ ]:

```
print('The original image contains', MP, 'pixels, or', MP/10e6, 'megapixels␣
  ↪(MP).')

# U[:,0:r].shape
# (4032, 225)
# Vt[0:r,:].shape
# (225, 3024)
rMP = 4032*225+225+3024*225
print('The Rank-225 approximation only uses', rMP, 'values of data to be stored.
  ↪')
```

```
The original image contains 12192768 pixels, or 1.2192768 megapixels (MP).
The Rank-225 approximation only uses 1587825 values of data to be stored.
```

1.6 Part c - Discuss

[ ]:

```
ratio = rMP / MP
print('The ratio of data points between the rank-225 approximation and the␣
  ↪original image is', ratio, 'or', ratio*100, '%.')
```

```
The ratio of data points between the rank-225 approximation and the original
image is 0.13022678689531367 or 13.022678689531366 %.
```

The ratio is 0.1302 … . Based off the value of the ratio, it seems storing the rank-225 image is vastly more efficient than the original image, especially since the quality is mostly indistinguishable.