Mohanty_R_HW2_Computing_3

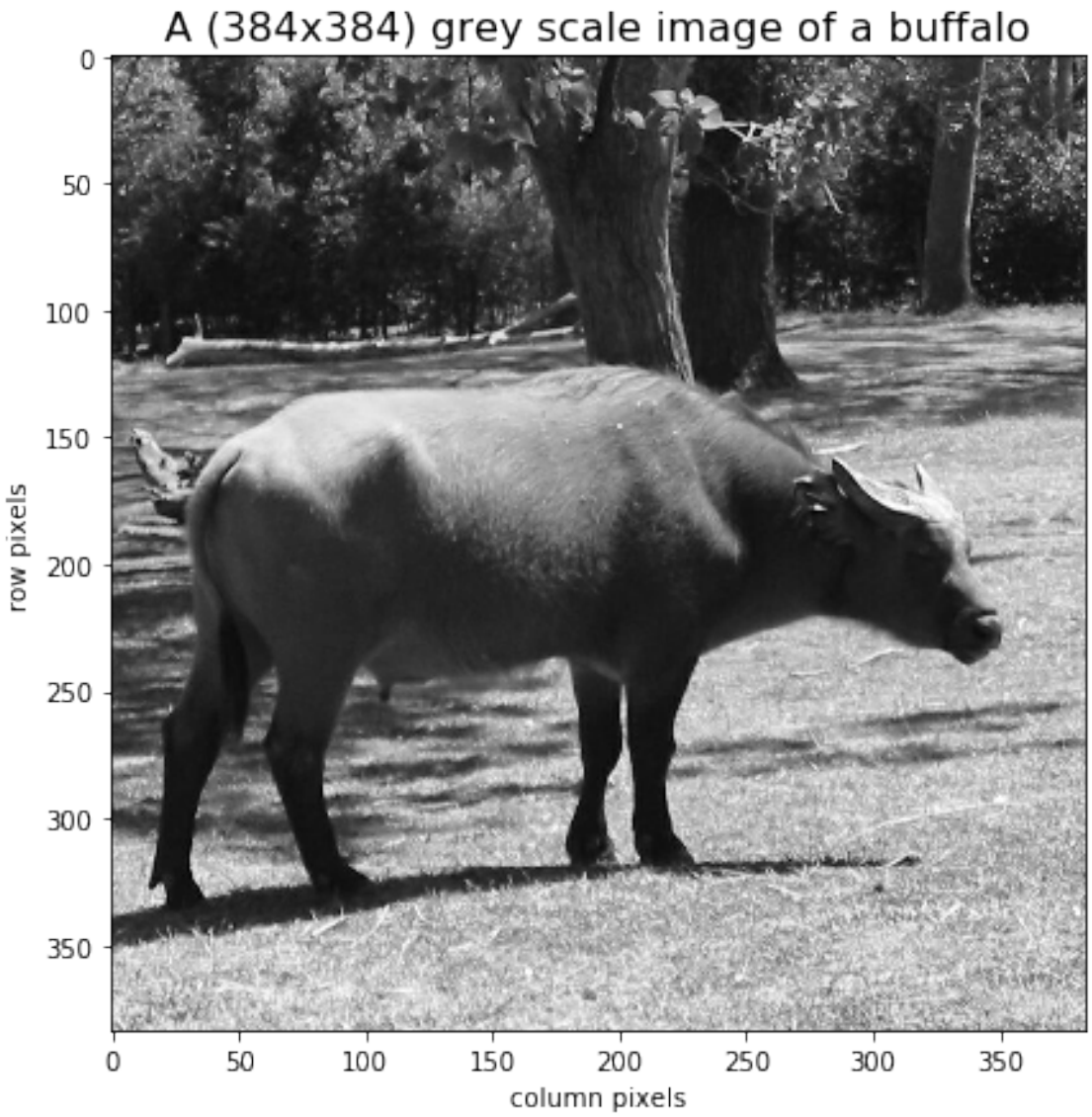
March 12, 2020

[2]:
```
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from  imageio import imread
import pandas as pd
```

[3]:
```
A = imread('buffalo.png')
```

[8]:
```
fig, ax = plt.subplots(1, 1,figsize=(7,7))
ax.set_title('A (384x384) grey scale image of a buffalo',fontsize=16)
print('\n')
ax.set_xlabel('column pixels')
ax.set_ylabel('row pixels')
ax.imshow(A,cmap='gray')
#plt.imshow(A,cmap='gray')
```

[8]:
```
<matplotlib.image.AxesImage at 0x11f0c6ba8>
```



[9]:
```
A.shape
```

[9]:
```
(384, 384)
```

[10]:
```
A
```

[10]:
```
Array([[ 95,  77, 107, ...,  96, 106, 110],
       [ 91,  57,  56, ..., 129, 116, 167],
       [ 97,  58,  58, ..., 136, 108,  73],
       ...,
       [177, 173, 148, ..., 181, 186, 208],
       [186, 187, 160, ..., 190, 192, 189],
       [166, 198, 224, ..., 124, 230, 183]], dtype=uint8)
```

[11]:
```
A=A.astype(np.float64)
```

[23]:
```
U,S,V=np.linalg.svd(A)
```

[13]:
```
#S=np.diag(S)
```

[14]:
```
k=10
```

[17]:
```
#A_k = np.dot(U[:,:k],np.dot(np.diag(S[:k]),V[:k,:]))
```

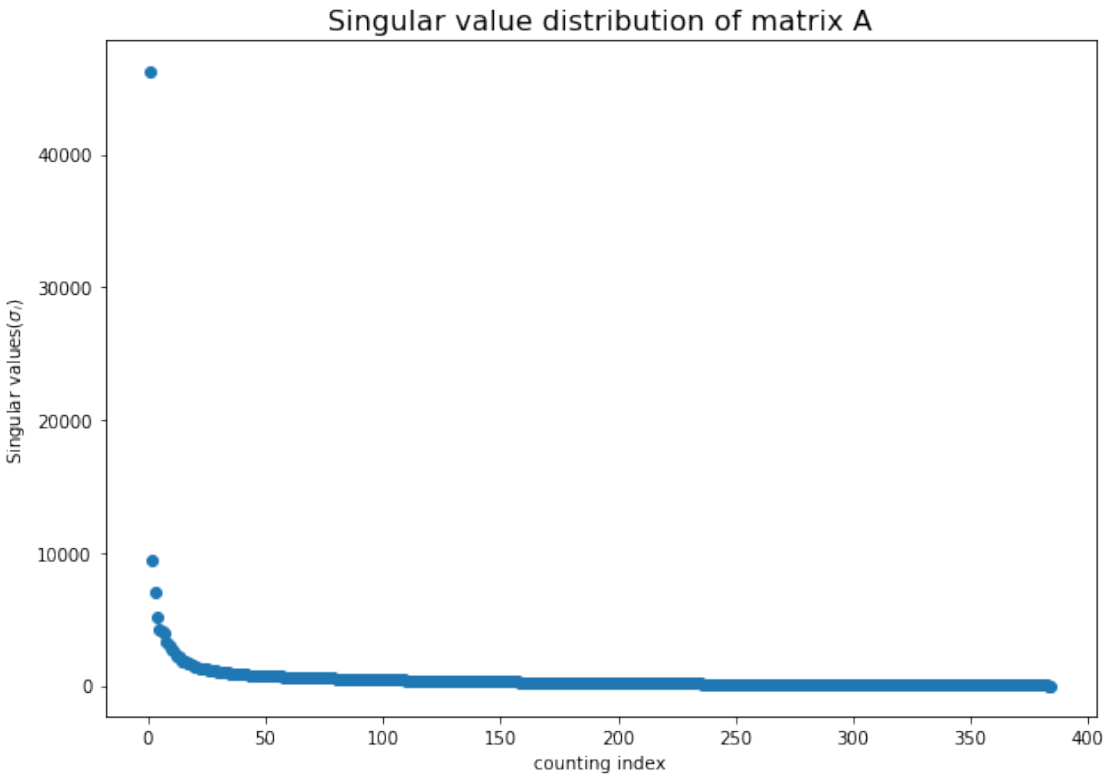
[18]:
```
#plt.imshow(A_k,cmap='gray')
```

[20]:
```
#A_k.shape
```

[24]:
```
x=list(range(1,len(S)+1))
singular_values=list(S)
```

[28]:
```
fig, ax = plt.subplots(1, 1,figsize=(10,7))
ax.set_title('Singular value distribution of matrix A',fontsize=16)
print('\n')
ax.set_xlabel('counting index')
ax.set_ylabel('Singular values($\sigma _i$)')
ax.scatter(x,singular_values)
#plt.scatter(x,singular_values)
```

[28]:
```
<matplotlib.collections.PathCollection at 0x11fad9e10>
```



[29]:
```
energy_cal=pd.DataFrame(list(zip(x, singular_values)),
               columns =['k', 'singular values'])
```

[30]:
```
energy_cal['sigma_sq']=energy_cal['singular values']**2
```

[31]:
```
energy_cal['sigma_sq_cusum']=energy_cal.sigma_sq.cumsum()
```

[32]:
```
energy_cal['E_k']=energy_cal['sigma_sq_cusum']/energy_cal.sigma_sq.sum()
```

[33]:
```
energy_cal
```

[33]:

	k	singular values	sigma_sq	sigma_sq_cusum	E_k
0	1	46281.329948	2.141962e+09	2.141962e+09	0.865386
1	2	9386.268303	8.810203e+07	2.230064e+09	0.900981
2	3	6968.066915	4.855396e+07	2.278617e+09	0.920598
3	4	5201.601380	2.705666e+07	2.305674e+09	0.931529
4	5	4252.804230	1.808634e+07	2.323760e+09	0.938836
5	6	4112.323483	1.691120e+07	2.340672e+09	0.945669
6	7	3931.462750	1.545640e+07	2.356128e+09	0.951913
7	8	3225.366003	1.040299e+07	2.366531e+09	0.956116
8	9	3047.430954	9.286835e+06	2.375818e+09	0.959868
9	10	2702.392059	7.302923e+06	2.383121e+09	0.962819
10	11	2580.168508	6.657270e+06	2.389778e+09	0.965508
11	12	2269.410181	5.150223e+06	2.394928e+09	0.967589
12	13	2146.176259	4.606073e+06	2.399534e+09	0.969450
13	14	2032.199076	4.129833e+06	2.403664e+09	0.971119
14	15	1833.597463	3.362080e+06	2.407026e+09	0.972477
15	16	1769.834836	3.132315e+06	2.410159e+09	0.973742
16	17	1647.304860	2.713613e+06	2.412872e+09	0.974839
17	18	1608.257274	2.586491e+06	2.415459e+09	0.975884
18	19	1498.530334	2.245593e+06	2.417704e+09	0.976791
19	20	1431.695072	2.049751e+06	2.419754e+09	0.977619
20	21	1358.543953	1.845642e+06	2.421600e+09	0.978365
21	22	1313.071624	1.724157e+06	2.423324e+09	0.979061
22	23	1257.289705	1.580777e+06	2.424905e+09	0.979700
23	24	1231.882274	1.517534e+06	2.426422e+09	0.980313
24	25	1206.347075	1.455273e+06	2.427877e+09	0.980901
25	26	1176.147332	1.383323e+06	2.429261e+09	0.981460
26	27	1125.977265	1.267825e+06	2.430529e+09	0.981972
27	28	1095.924042	1.201050e+06	2.431730e+09	0.982457
28	29	1074.006781	1.153491e+06	2.432883e+09	0.982923
29	30	1046.672063	1.095522e+06	2.433979e+09	0.983366
..	...	...	...	...	...
354	355	17.314320	2.997857e+02	2.475147e+09	0.999999
355	356	16.505158	2.724202e+02	2.475148e+09	0.999999
356	357	15.680552	2.458797e+02	2.475148e+09	0.999999
357	358	15.617818	2.439163e+02	2.475148e+09	0.999999

```
358  359        15.142702  2.293014e+02    2.475148e+09  0.999999
359  360        14.213009  2.020096e+02    2.475149e+09  0.999999
360  361        13.182723  1.737842e+02    2.475149e+09  0.999999
361  362        13.033572  1.698740e+02    2.475149e+09  1.000000
362  363        12.146748  1.475435e+02    2.475149e+09  1.000000
363  364        11.766328  1.384465e+02    2.475149e+09  1.000000
364  365        11.141653  1.241364e+02    2.475149e+09  1.000000
365  366        10.820558  1.170845e+02    2.475150e+09  1.000000
366  367        10.182809  1.036896e+02    2.475150e+09  1.000000
367  368         9.404634  8.844715e+01    2.475150e+09  1.000000
368  369         8.957195  8.023135e+01    2.475150e+09  1.000000
369  370         8.402091  7.059514e+01    2.475150e+09  1.000000
370  371         7.466453  5.574792e+01    2.475150e+09  1.000000
371  372         6.814635  4.643925e+01    2.475150e+09  1.000000
372  373         6.374152  4.062981e+01    2.475150e+09  1.000000
373  374         5.653246  3.195919e+01    2.475150e+09  1.000000
374  375         5.065388  2.565815e+01    2.475150e+09  1.000000
375  376         4.362624  1.903248e+01    2.475150e+09  1.000000
376  377         3.745859  1.403146e+01    2.475150e+09  1.000000
377  378         3.493954  1.220772e+01    2.475150e+09  1.000000
378  379         2.792784  7.799642e+00    2.475150e+09  1.000000
379  380         2.353781  5.540287e+00    2.475150e+09  1.000000
380  381         1.845766  3.406851e+00    2.475150e+09  1.000000
381  382         0.922918  8.517776e-01    2.475150e+09  1.000000
382  383         0.646929  4.185170e-01    2.475150e+09  1.000000
383  384         0.230497  5.312897e-02    2.475150e+09  1.000000

[384 rows x 5 columns]
```

[41]:
```
energy_cal[energy_cal.E_k>=0.95].head()
```

[41]:
```
     k  singular values      sigma_sq  sigma_sq_cusum       E_k
6    7      3931.462750  1.545640e+07    2.356128e+09  0.951913
7    8      3225.366003  1.040299e+07    2.366531e+09  0.956116
8    9      3047.430954  9.286835e+06    2.375818e+09  0.959868
9   10      2702.392059  7.302923e+06    2.383121e+09  0.962819
10  11      2580.168508  6.657270e+06    2.389778e+09  0.965508
```

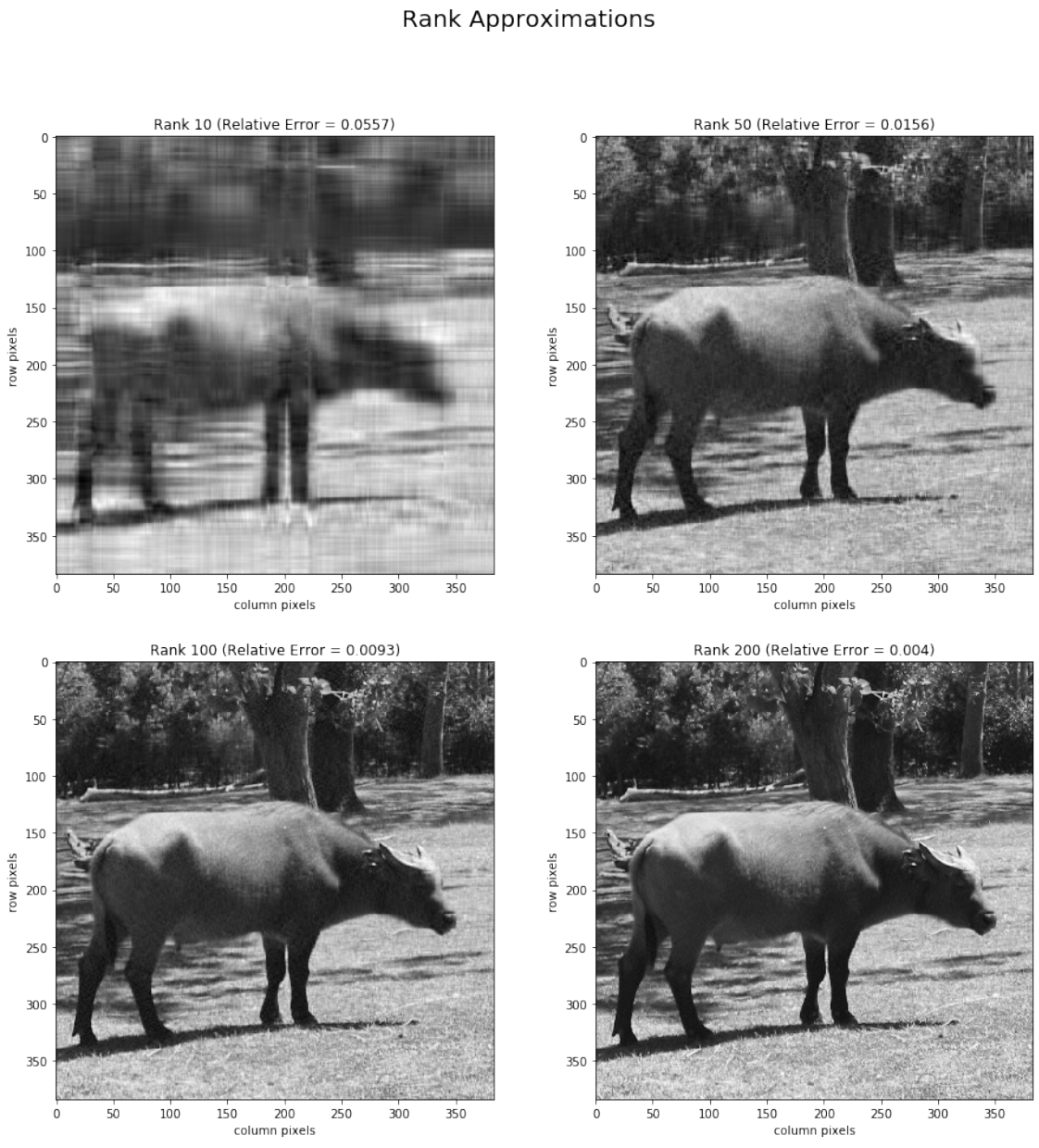
0.0.1 k = 7

[82]:
```
rank_k=[10,50,100,200]

position=[(0,0),(0,1),(1,0),(1,1)]

fig, ax = plt.subplots(2, 2,figsize=(15,15))
fig.suptitle('Rank Approximations', fontsize=20)
i=0
for k in rank_k:
```

```
    A_k = np.dot(U[:,:k],np.dot(np.diag(S[:k]),V[:k,:]))
    relative_error = round(S[k]/S[0],4)
    ax[position[i][0],position[i][1]].set_title('Rank '+str(k)+' (Relative␣
↪Error = '+str(relative_error)+')')
    ax[position[i][0],position[i][1]].set_xlabel('column pixels')
    ax[position[i][0],position[i][1]].set_ylabel('row pixels')
    ax[position[i][0],position[i][1]].imshow(A_k,cmap='gray')
    i+=1
```



[ ]: