To apply histogram equalization to improve contrast of different test images.

Preet Jha
B030
B1
B.Tech CE
28 JUL 2022

In [ ]:
```
from skimage import io
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
image = io.imread("pollen_very_dark.tif")
sh = image.shape
rows =sh[0]
cols = sh[1]
```

In [ ]:
```
numb_pix = rows*cols
numb_pix
```

Out[ ]:
```
250000
```

In [ ]:
```
imhist, bins = np.histogram(image.flatten(),256)
imhist
```

Out[ ]:
```
array([17718,     0,     0,    96,     0,     0,     0,   506,     0,
           0,  1750,     0,     0,     0,  9441,     0,     0,     0,
        1642,     0,     0, 16457,     0,     0,     0, 10381,     0,
           0,     0,  2945,     0,     0,  6149,     0,     0,     0,
        1038,     0,     0,     0,  4566,     0,     0,   961,     0,
           0,     0,  4562,     0,     0,     0,  2446,     0,     0,
        2641,     0,     0,     0,  5860,     0,     0,     0,  1861,
           0,     0,  3850,     0,     0,     0,  2016,     0,     0,
           0,  4290,     0,     0,  3081,     0,     0,     0,  6031,
           0,     0,     0,  2830,     0,     0,  4868,     0,     0,
           0,  7185,     0,     0,     0,  4197,     0,     0,  3696,
           0,     0,     0,  3911,     0,     0,     0,  6814,     0,
           0,  1379,     0,     0,     0,  7094,     0,     0,     0,
        1295,     0,     0,  6812,     0,     0,     0,  9944,     0,
           0,     0,  2898,     0,     0,  5615,     0,     0,     0,
        2791,     0,     0,  5118,     0,     0,     0,  3819,     0,
           0,     0,  3533,     0,     0,  3482,     0,     0,     0,
        5679,     0,     0,     0,  5933,     0,     0,  1008,     0,
           0,     0,  4599,     0,     0,     0,  2546,     0,     0,
        4007,     0,     0,     0,   752,     0,     0,     0,  3433,
           0,     0,   690,     0,     0,     0,  1939,     0,     0,
           0,  3544,     0,     0,  2668,     0,     0,     0,   533,
           0,     0,     0,  2482,     0,     0,  1359,     0,     0,
           0,  1336,     0,     0,     0,  1408,     0,     0,  1778,
           0,     0,     0,   884,     0,     0,     0,  2373,     0,
           0,  1419,     0,     0,     0,   830,     0,     0,     0,
         949,     0,     0,   116,     0,     0,     0,   150,     0,
           0,     0,     5,     0,     0,    10,     0,     0,     0,
           0,     0,     0,     1])
```

In [ ]:
```
bins
```

Out[ ]:
```
array([13.       , 13.2734375, 13.546875 , 13.8203125, 14.09375  ,
       14.3671875, 14.640625 , 14.9140625, 15.1875   , 15.4609375,
       15.734375 , 16.0078125, 16.28125  , 16.5546875, 16.828125 ,
       17.1015625, 17.375    , 17.6484375, 17.921875 , 18.1953125,
       18.46875  , 18.7421875, 19.015625 , 19.2890625, 19.5625   ,
       19.8359375, 20.109375 , 20.3828125, 20.65625  , 20.9296875,
       21.203125 , 21.4765625, 21.75     , 22.0234375, 22.296875 ,
       22.5703125, 22.84375  , 23.1171875, 23.390625 , 23.6640625,
       23.9375   , 24.2109375, 24.484375 , 24.7578125, 25.03125  ,
       25.3046875, 25.578125 , 25.8515625, 26.125    , 26.3984375,
       26.671875 , 26.9453125, 27.21875  , 27.4921875, 27.765625 ,
       28.0390625, 28.3125   , 28.5859375, 28.859375 , 29.1328125,
       29.40625  , 29.6796875, 29.953125 , 30.2265625, 30.5      ,
       30.7734375, 31.046875 , 31.3203125, 31.59375  , 31.8671875,
       32.140625 , 32.4140625, 32.6875   , 32.9609375, 33.234375 ,
       33.5078125, 33.78125  , 34.0546875, 34.328125 , 34.6015625,
       34.875    , 35.1484375, 35.421875 , 35.6953125, 35.96875  ,
       36.2421875, 36.515625 , 36.7890625, 37.0625   , 37.3359375,
       37.609375 , 37.8828125, 38.15625  , 38.4296875, 38.703125 ,
       38.9765625, 39.25     , 39.5234375, 39.796875 , 40.0703125,
       40.34375  , 40.6171875, 40.890625 , 41.1640625, 41.4375   ,
       41.7109375, 41.984375 , 42.2578125, 42.53125  , 42.8046875,
       43.078125 , 43.3515625, 43.625    , 43.8984375, 44.171875 ,
       44.4453125, 44.71875  , 44.9921875, 45.265625 , 45.5390625,
       45.8125   , 46.0859375, 46.359375 , 46.6328125, 46.90625  ,
       47.1796875, 47.453125 , 47.7265625, 48.       , 48.2734375,
       48.546875 , 48.8203125, 49.09375  , 49.3671875, 49.640625 ,
       49.9140625, 50.1875   , 50.4609375, 50.734375 , 51.0078125,
       51.28125  , 51.5546875, 51.828125 , 52.1015625, 52.375    ,
       52.6484375, 52.921875 , 53.1953125, 53.46875  , 53.7421875,
       54.015625 , 54.2890625, 54.5625   , 54.8359375, 55.109375 ,
       55.3828125, 55.65625  , 55.9296875, 56.203125 , 56.4765625,
       56.75     , 57.0234375, 57.296875 , 57.5703125, 57.84375  ,
       58.1171875, 58.390625 , 58.6640625, 58.9375   , 59.2109375,
       59.484375 , 59.7578125, 60.03125  , 60.3046875, 60.578125 ,
       60.8515625, 61.125    , 61.3984375, 61.671875 , 61.9453125,
       62.21875  , 62.4921875, 62.765625 , 63.0390625, 63.3125   ,
       63.5859375, 63.859375 , 64.1328125, 64.40625  , 64.6796875,
       64.953125 , 65.2265625, 65.5      , 65.7734375, 66.046875 ,
       66.3203125, 66.59375  , 66.8671875, 67.140625 , 67.4140625,
       67.6875   , 67.9609375, 68.234375 , 68.5078125, 68.78125  ,
       69.0546875, 69.328125 , 69.6015625, 69.875    , 70.1484375,
       70.421875 , 70.6953125, 70.96875  , 71.2421875, 71.515625 ,
       71.7890625, 72.0625   , 72.3359375, 72.609375 , 72.8828125,
       73.15625  , 73.4296875, 73.703125 , 73.9765625, 74.25     ,
       74.5234375, 74.796875 , 75.0703125, 75.34375  , 75.6171875,
       75.890625 , 76.1640625, 76.4375   , 76.7109375, 76.984375 ,
       77.2578125, 77.53125  , 77.8046875, 78.078125 , 78.3515625,
       78.625    , 78.8984375, 79.171875 , 79.4453125, 79.71875  ,
       79.9921875, 80.265625 , 80.5390625, 80.8125   , 81.0859375,
       81.359375 , 81.6328125, 81.90625  , 82.1796875, 82.453125 ,
       82.7265625, 83.       ])
```

In [ ]:
```
cf = imhist.cumsum()
cf.shape
cf
```

Out[ ]:
```
array([ 17718,  17718,  17718,  17814,  17814,  17814,  17814,  18320,
        18320,  18320,  20070,  20070,  20070,  20070,  29511,  29511,
        29511,  29511,  31153,  31153,  31153,  47610,  47610,  47610,
        47610,  57991,  57991,  57991,  57991,  60936,  60936,  60936,
        67085,  67085,  67085,  67085,  68123,  68123,  68123,  68123,
        72689,  72689,  72689,  73650,  73650,  73650,  73650,  78212,
        78212,  78212,  78212,  80658,  80658,  80658,  83299,  83299,
        83299,  83299,  89159,  89159,  89159,  89159,  91020,  91020,
        91020,  94870,  94870,  94870,  94870,  96886,  96886,  96886,
        96886, 101176, 101176, 101176, 104257, 104257, 104257, 104257,
       110288, 110288, 110288, 110288, 113118, 113118, 113118, 117986,
       117986, 117986, 117986, 125171, 125171, 125171, 125171, 129368,
       129368, 129368, 133064, 133064, 133064, 133064, 136975, 136975,
       136975, 136975, 143789, 143789, 143789, 145168, 145168, 145168,
       145168, 152262, 152262, 152262, 152262, 153557, 153557, 153557,
       160369, 160369, 160369, 160369, 170313, 170313, 170313, 170313,
       173211, 173211, 173211, 178826, 178826, 178826, 178826, 181617,
       181617, 181617, 186735, 186735, 186735, 186735, 190554, 190554,
       190554, 190554, 194087, 194087, 194087, 197569, 197569, 197569,
       197569, 203248, 203248, 203248, 203248, 209181, 209181, 209181,
       210189, 210189, 210189, 210189, 214788, 214788, 214788, 214788,
       217334, 217334, 217334, 221341, 221341, 221341, 221341, 222093,
       222093, 222093, 222093, 225526, 225526, 225526, 226216, 226216,
       226216, 226216, 228155, 228155, 228155, 228155, 231699, 231699,
       231699, 234367, 234367, 234367, 234367, 234900, 234900, 234900,
       234900, 237382, 237382, 237382, 238741, 238741, 238741, 238741,
       240077, 240077, 240077, 240077, 241485, 241485, 241485, 243263,
       243263, 243263, 243263, 244147, 244147, 244147, 244147, 246520,
       246520, 246520, 247939, 247939, 247939, 247939, 248769, 248769,
       248769, 248769, 249718, 249718, 249718, 249834, 249834, 249834,
       249834, 249984, 249984, 249984, 249984, 249989, 249989, 249989,
       249999, 249999, 249999, 249999, 249999, 249999, 249999, 250000])
```

In [ ]:
```
cdf = 255 * cf/numb_pix
s = cdf.astype(int)
s
```

Out[ ]:
```
array([ 18,  18,  18,  18,  18,  18,  18,  18,  18,  18,  20,  20,  20,
        20,  30,  30,  30,  30,  31,  31,  31,  48,  48,  48,  48,  59,
        59,  59,  59,  62,  62,  62,  68,  68,  68,  68,  69,  69,  69,
        69,  74,  74,  74,  75,  75,  75,  75,  79,  79,  79,  79,  82,
        82,  82,  84,  84,  84,  84,  90,  90,  90,  90,  92,  92,  92,
        96,  96,  96,  96,  98,  98,  98,  98, 103, 103, 103, 106, 106,
       106, 106, 112, 112, 112, 112, 115, 115, 115, 120, 120, 120, 120,
       127, 127, 127, 127, 131, 131, 131, 135, 135, 135, 135, 139, 139,
       139, 139, 146, 146, 146, 148, 148, 148, 148, 155, 155, 155, 155,
       156, 156, 156, 163, 163, 163, 163, 173, 173, 173, 173, 176, 176,
       176, 182, 182, 182, 182, 185, 185, 185, 190, 190, 190, 190, 194,
       194, 194, 194, 197, 197, 197, 201, 201, 201, 201, 207, 207, 207,
       207, 213, 213, 213, 214, 214, 214, 214, 219, 219, 219, 219, 221,
       221, 221, 225, 225, 225, 225, 226, 226, 226, 226, 230, 230, 230,
       230, 230, 230, 230, 232, 232, 232, 232, 236, 236, 236, 239, 239,
       239, 239, 239, 239, 239, 239, 242, 242, 242, 243, 243, 243, 243,
       244, 244, 244, 244, 246, 246, 246, 248, 248, 248, 248, 249, 249,
       249, 249, 251, 251, 251, 252, 252, 252, 252, 253, 253, 253, 253,
       254, 254, 254, 254, 254, 254, 254, 254, 254, 254, 254, 254, 254,
       254, 254, 254, 254, 254, 254, 254, 254, 255])
```

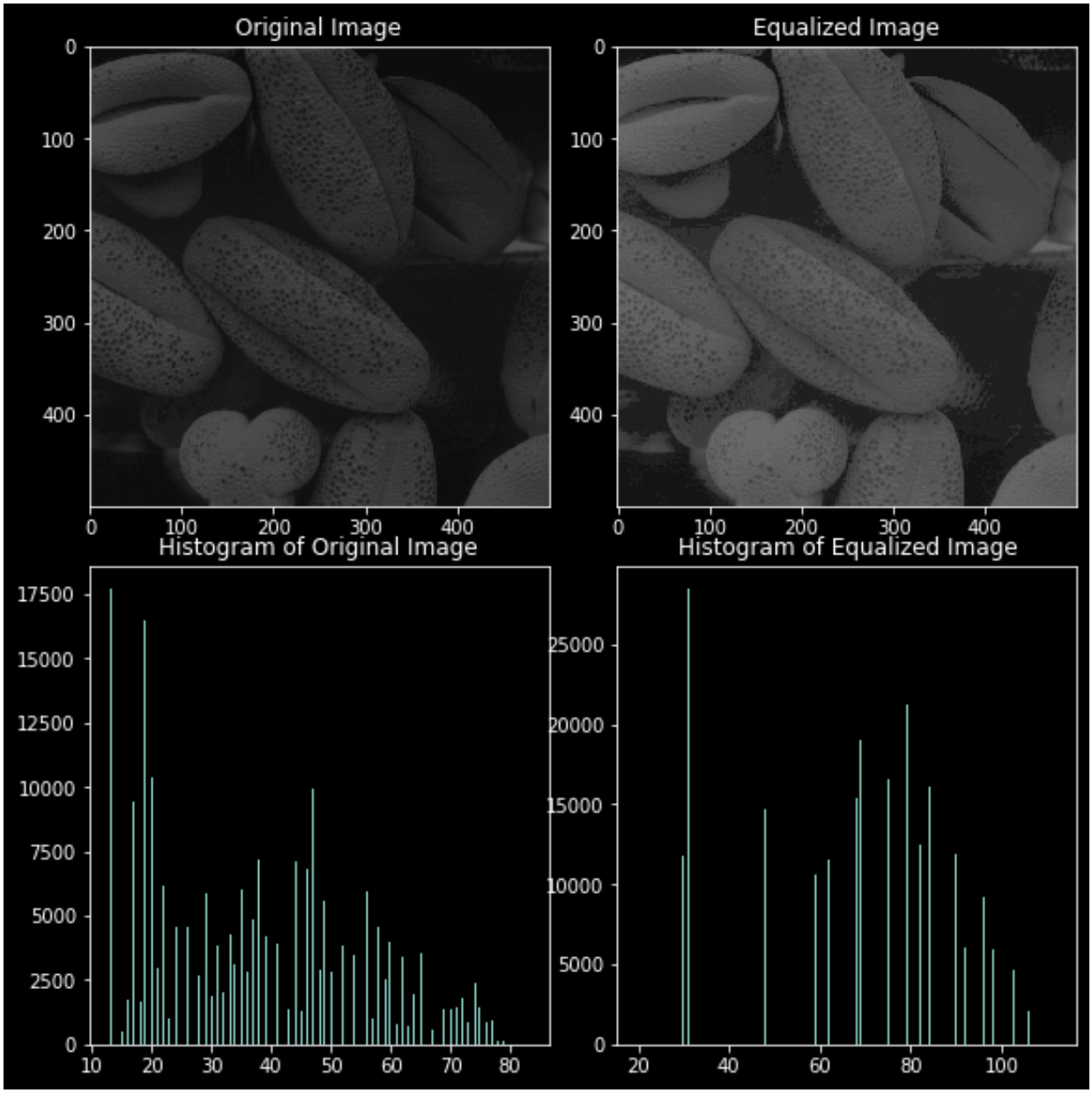
In [ ]:
```
image_eq=image.copy()
```

In [ ]:
```
for rw in range(rows):
    for cl in range(cols):
        r=image[rw,cl]
        s_new=s[r]
        image_eq[rw][cl]=s_new
s.shape
```

Out[ ]:
```
(256,)
```

In [ ]:
```
plt.figure(figsize=(8,8))

plt.subplot(2,2,1)
io.imshow(image)
plt.title("Original Image")

plt.subplot(2,2,2)
io.imshow(image_eq)
plt.title("Equalized Image")

plt.subplot(2,2,3)
plt.hist(image.flatten(),bins=256)
plt.title("Histogram of Original Image")

plt.subplot(2,2,4)
plt.hist(image_eq.flatten(),bins=256)
plt.title("Histogram of Equalized Image")
```

Out[ ]:
```
Text(0.5, 1.0, 'Histogram of Equalized Image')
```



CONCLUSION

- Histogram equalization in applied to various contrast levels.
- For dark images like pollen_dark.tif and pollen_very_dark.tif, histogram is concentrated in the lower range of pixel intensities.
- After applying histogram equalization, histogram is redistributed and contrast of the image improves.
- For very bright images, histogram is concentrated in higher range of intensity.
- After applying equalization contrast of the image improves.
- If original image is having good contrast levels and histogram equalization is applied on it, there is no substantial improvement in the contrast.