Histogram equalisation

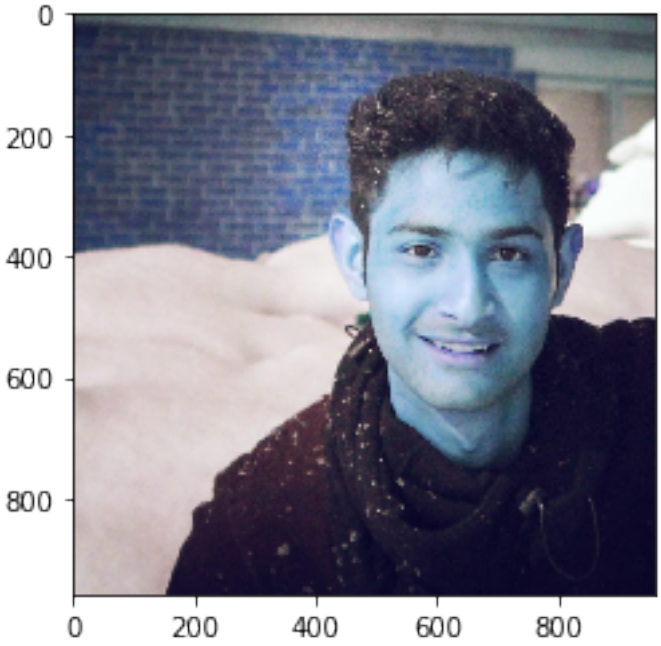
April 28, 2019

In [2]:
```
import cv2
import numpy as np
import matplotlib.pyplot as plt
```

In [3]:
```
image = cv2.imread('sample7.jpg')
```

In [4]:
```
plt.imshow(image)
```

Out[4]:
```
<matplotlib.image.AxesImage at 0x123bbe5f8>
```

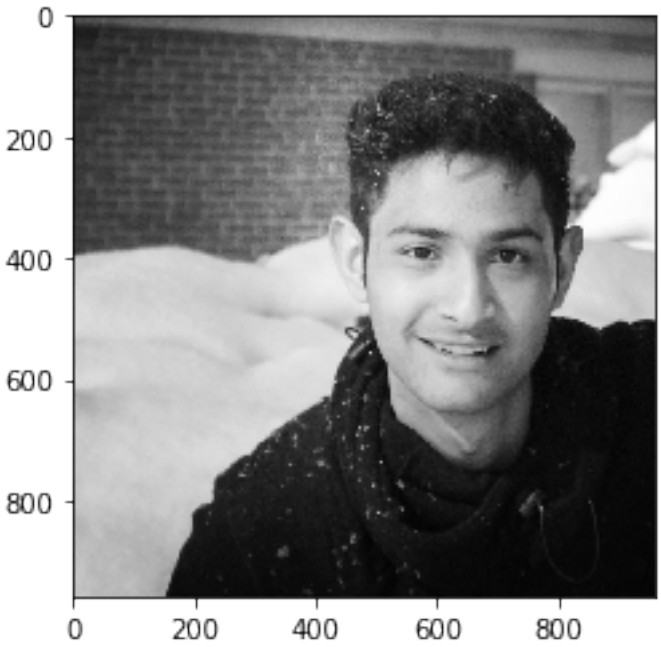


In [5]:
```
image.shape
```

Out[5]:
```
(959, 960, 3)
```

In [6]:
```
gray = cv2.cvtColor(image,cv2.COLOR_BGR2GRAY)
```

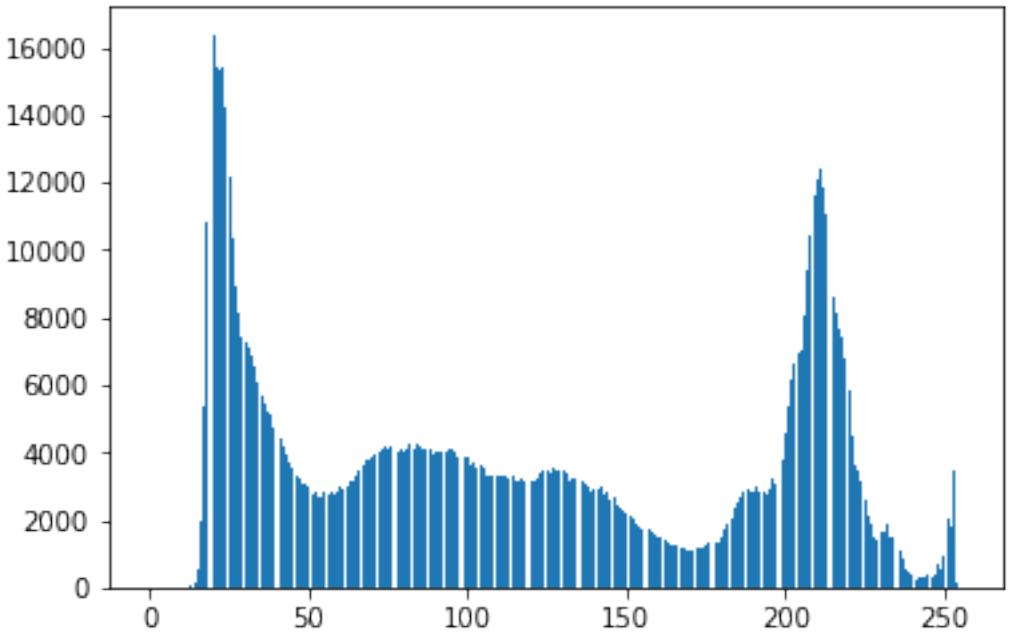
```
In [7]: gray.shape

Out[7]: (959, 960)

In [8]: plt.imshow(gray,cmap='gray')

Out[8]: <matplotlib.image.AxesImage at 0x12499f7f0>
```




```
In [9]: gray.shape

Out[9]: (959, 960)

In [10]: hist = np.zeros(256)

         for i in range(gray.shape[0]):
             for j in range(gray.shape[1]):
                 hist[gray[i][j]] = hist[gray[i][j]]+1

In [11]: hist

Out[11]: array([0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
                0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00, 0.0000e+00,
                0.0000e+00, 4.0000e+00, 1.4000e+01, 4.1000e+01, 1.3900e+02,
                5.3000e+02, 1.9360e+03, 5.4010e+03, 1.0821e+04, 1.4391e+04,
                1.6384e+04, 1.5396e+04, 1.5370e+04, 1.5431e+04, 1.4197e+04,
                1.2152e+04, 1.0335e+04, 8.9400e+03, 8.0950e+03, 7.4570e+03,
                7.2320e+03, 7.0980e+03, 6.8880e+03, 6.5340e+03, 6.1070e+03,
```

```
       5.6790e+03, 5.4360e+03, 5.1730e+03, 5.0900e+03, 4.7480e+03,
       4.4260e+03, 4.3900e+03, 4.1620e+03, 3.9660e+03, 3.6680e+03,
       3.5500e+03, 3.2820e+03, 3.2280e+03, 3.0660e+03, 3.0590e+03,
       2.9810e+03, 2.7740e+03, 2.8010e+03, 2.7070e+03, 2.6940e+03,
       2.8070e+03, 2.7780e+03, 2.7930e+03, 2.7750e+03, 2.8080e+03,
       2.9620e+03, 2.9490e+03, 2.9780e+03, 3.1340e+03, 3.1530e+03,
       3.3280e+03, 3.4790e+03, 3.6340e+03, 3.8070e+03, 3.8160e+03,
       3.8480e+03, 3.9350e+03, 3.9990e+03, 4.0850e+03, 4.1750e+03,
       4.0950e+03, 4.1620e+03, 4.0940e+03, 3.9930e+03, 4.0690e+03,
       3.9880e+03, 4.1040e+03, 4.2280e+03, 4.1020e+03, 4.2210e+03,
       4.1830e+03, 4.0910e+03, 4.1350e+03, 4.0990e+03, 3.9790e+03,
       4.0110e+03, 4.0540e+03, 4.0350e+03, 4.0300e+03, 4.1130e+03,
       4.1030e+03, 3.9810e+03, 3.8950e+03, 3.8410e+03, 3.8920e+03,
       3.8270e+03, 3.6620e+03, 3.7050e+03, 3.5640e+03, 3.6170e+03,
       3.5370e+03, 3.3260e+03, 3.3210e+03, 3.3270e+03, 3.3310e+03,
       3.2930e+03, 3.3090e+03, 3.2780e+03, 3.2510e+03, 3.2690e+03,
       3.1190e+03, 3.1670e+03, 3.2130e+03, 3.1790e+03, 3.2070e+03,
       3.1680e+03, 3.1680e+03, 3.2270e+03, 3.3870e+03, 3.4450e+03,
       3.4750e+03, 3.3830e+03, 3.5060e+03, 3.4310e+03, 3.4800e+03,
       3.4670e+03, 3.4230e+03, 3.1820e+03, 3.2260e+03, 3.2200e+03,
       3.1090e+03, 3.1170e+03, 3.0980e+03, 2.9830e+03, 2.8660e+03,
       2.8860e+03, 2.9300e+03, 2.9830e+03, 2.7810e+03, 2.8090e+03,
       2.6030e+03, 2.6710e+03, 2.4640e+03, 2.3790e+03, 2.3130e+03,
       2.1720e+03, 2.1240e+03, 2.0690e+03, 1.8860e+03, 1.8370e+03,
       1.7160e+03, 1.7500e+03, 1.7540e+03, 1.6800e+03, 1.5720e+03,
       1.4550e+03, 1.4500e+03, 1.4180e+03, 1.3210e+03, 1.2840e+03,
       1.2180e+03, 1.2120e+03, 1.1840e+03, 1.1950e+03, 1.0960e+03,
       1.0740e+03, 1.0900e+03, 1.1620e+03, 1.1720e+03, 1.1790e+03,
       1.2720e+03, 1.2930e+03, 1.2640e+03, 1.3320e+03, 1.3540e+03,
       1.5240e+03, 1.6960e+03, 1.8470e+03, 2.0300e+03, 2.3390e+03,
       2.4810e+03, 2.7110e+03, 2.7930e+03, 2.8720e+03, 2.8440e+03,
       2.8310e+03, 3.0060e+03, 2.8070e+03, 2.8200e+03, 2.7230e+03,
       2.9290e+03, 3.1890e+03, 3.0830e+03, 3.3730e+03, 3.7540e+03,
       4.5340e+03, 5.3890e+03, 6.1430e+03, 6.5910e+03, 6.9450e+03,
       7.0540e+03, 8.0210e+03, 9.3930e+03, 1.0456e+04, 1.1643e+04,
       1.2068e+04, 1.2370e+04, 1.1875e+04, 1.1074e+04, 1.0362e+04,
       8.5860e+03, 8.1610e+03, 7.6920e+03, 7.4170e+03, 6.8190e+03,
       5.8080e+03, 4.5270e+03, 3.6060e+03, 3.4900e+03, 3.1600e+03,
       2.6170e+03, 2.1490e+03, 1.8970e+03, 1.4790e+03, 1.4330e+03,
       1.6260e+03, 1.6690e+03, 1.9050e+03, 1.4740e+03, 1.4610e+03,
       1.1510e+03, 1.0960e+03, 8.4300e+02, 5.2100e+02, 4.3000e+02,
       3.5700e+02, 2.4500e+02, 2.7500e+02, 2.6800e+02, 3.3800e+02,
       3.5000e+02, 3.4000e+02, 4.0700e+02, 7.1200e+02, 5.7700e+02,
       9.2300e+02, 2.0710e+03, 1.8130e+03, 3.4500e+03, 1.3800e+02,
       8.0000e+00])
```

In [12]:
```
index = np.arange(len(hist))
plot1 = plt.bar(index,hist)
```




```
In [13]: pdf = [np.around(i/(gray.shape[0]*gray.shape[1]),decimals=5) for i in hist];pdf

Out[13]: [0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          2e-05,
          4e-05,
          0.00015,
          0.00058,
          0.0021,
          0.00587,
          0.01175,
          0.01563,
          0.0178,
          0.01672,
          0.01669,
          0.01676,
```

0.01542,
0.0132,
0.01123,
0.00971,
0.00879,
0.0081,
0.00786,
0.00771,
0.00748,
0.0071,
0.00663,
0.00617,
0.0059,
0.00562,
0.00553,
0.00516,
0.00481,
0.00477,
0.00452,
0.00431,
0.00398,
0.00386,
0.00356,
0.00351,
0.00333,
0.00332,
0.00324,
0.00301,
0.00304,
0.00294,
0.00293,
0.00305,
0.00302,
0.00303,
0.00301,
0.00305,
0.00322,
0.0032,
0.00323,
0.0034,
0.00342,
0.00361,
0.00378,
0.00395,
0.00414,
0.00414,
0.00418,
0.00427,

0.00434,
0.00444,
0.00453,
0.00445,
0.00452,
0.00445,
0.00434,
0.00442,
0.00433,
0.00446,
0.00459,
0.00446,
0.00458,
0.00454,
0.00444,
0.00449,
0.00445,
0.00432,
0.00436,
0.0044,
0.00438,
0.00438,
0.00447,
0.00446,
0.00432,
0.00423,
0.00417,
0.00423,
0.00416,
0.00398,
0.00402,
0.00387,
0.00393,
0.00384,
0.00361,
0.00361,
0.00361,
0.00362,
0.00358,
0.00359,
0.00356,
0.00353,
0.00355,
0.00339,
0.00344,
0.00349,
0.00345,
0.00348,

0.00344,
0.00344,
0.00351,
0.00368,
0.00374,
0.00377,
0.00367,
0.00381,
0.00373,
0.00378,
0.00377,
0.00372,
0.00346,
0.0035,
0.0035,
0.00338,
0.00339,
0.00337,
0.00324,
0.00311,
0.00313,
0.00318,
0.00324,
0.00302,
0.00305,
0.00283,
0.0029,
0.00268,
0.00258,
0.00251,
0.00236,
0.00231,
0.00225,
0.00205,
0.002,
0.00186,
0.0019,
0.00191,
0.00182,
0.00171,
0.00158,
0.00157,
0.00154,
0.00143,
0.00139,
0.00132,
0.00132,
0.00129,

```
0.0013,
0.00119,
0.00117,
0.00118,
0.00126,
0.00127,
0.00128,
0.00138,
0.0014,
0.00137,
0.00145,
0.00147,
0.00166,
0.00184,
0.00201,
0.0022,
0.00254,
0.00269,
0.00294,
0.00303,
0.00312,
0.00309,
0.00308,
0.00327,
0.00305,
0.00306,
0.00296,
0.00318,
0.00346,
0.00335,
0.00366,
0.00408,
0.00492,
0.00585,
0.00667,
0.00716,
0.00754,
0.00766,
0.00871,
0.0102,
0.01136,
0.01265,
0.01311,
0.01344,
0.0129,
0.01203,
0.01126,
0.00933,
```

```
 0.00886,
 0.00836,
 0.00806,
 0.00741,
 0.00631,
 0.00492,
 0.00392,
 0.00379,
 0.00343,
 0.00284,
 0.00233,
 0.00206,
 0.00161,
 0.00156,
 0.00177,
 0.00181,
 0.00207,
 0.0016,
 0.00159,
 0.00125,
 0.00119,
 0.00092,
 0.00057,
 0.00047,
 0.00039,
 0.00027,
 0.0003,
 0.00029,
 0.00037,
 0.00038,
 0.00037,
 0.00044,
 0.00077,
 0.00063,
 0.001,
 0.00225,
 0.00197,
 0.00375,
 0.00015,
 1e-05]
```

In [14]:
```
cdf = []
cdf.append(pdf[0])

for i in range(1,len(pdf)):
    cdf.append(np.around(cdf[i-1]+pdf[i],decimals=6))

cdf
```

```
Out[14]: [0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          0.0,
          2e-05,
          6e-05,
          0.00021,
          0.00079,
          0.00289,
          0.00876,
          0.02051,
          0.03614,
          0.05394,
          0.07066,
          0.08735,
          0.10411,
          0.11953,
          0.13273,
          0.14396,
          0.15367,
          0.16246,
          0.17056,
          0.17842,
          0.18613,
          0.19361,
          0.20071,
          0.20734,
          0.21351,
          0.21941,
          0.22503,
          0.23056,
          0.23572,
          0.24053,
          0.2453,
          0.24982,
          0.25413,
          0.25811,
          0.26197,
          0.26553,
          0.26904,
```

```
0.27237,
0.27569,
0.27893,
0.28194,
0.28498,
0.28792,
0.29085,
0.2939,
0.29692,
0.29995,
0.30296,
0.30601,
0.30923,
0.31243,
0.31566,
0.31906,
0.32248,
0.32609,
0.32987,
0.33382,
0.33796,
0.3421,
0.34628,
0.35055,
0.35489,
0.35933,
0.36386,
0.36831,
0.37283,
0.37728,
0.38162,
0.38604,
0.39037,
0.39483,
0.39942,
0.40388,
0.40846,
0.413,
0.41744,
0.42193,
0.42638,
0.4307,
0.43506,
0.43946,
0.44384,
0.44822,
0.45269,
0.45715,
```

```
0.46147,
0.4657,
0.46987,
0.4741,
0.47826,
0.48224,
0.48626,
0.49013,
0.49406,
0.4979,
0.50151,
0.50512,
0.50873,
0.51235,
0.51593,
0.51952,
0.52308,
0.52661,
0.53016,
0.53355,
0.53699,
0.54048,
0.54393,
0.54741,
0.55085,
0.55429,
0.5578,
0.56148,
0.56522,
0.56899,
0.57266,
0.57647,
0.5802,
0.58398,
0.58775,
0.59147,
0.59493,
0.59843,
0.60193,
0.60531,
0.6087,
0.61207,
0.61531,
0.61842,
0.62155,
0.62473,
0.62797,
0.63099,
```

```
0.63404,
0.63687,
0.63977,
0.64245,
0.64503,
0.64754,
0.6499,
0.65221,
0.65446,
0.65651,
0.65851,
0.66037,
0.66227,
0.66418,
0.666,
0.66771,
0.66929,
0.67086,
0.6724,
0.67383,
0.67522,
0.67654,
0.67786,
0.67915,
0.68045,
0.68164,
0.68281,
0.68399,
0.68525,
0.68652,
0.6878,
0.68918,
0.69058,
0.69195,
0.6934,
0.69487,
0.69653,
0.69837,
0.70038,
0.70258,
0.70512,
0.70781,
0.71075,
0.71378,
0.7169,
0.71999,
0.72307,
0.72634,
```

```
0.72939,
0.73245,
0.73541,
0.73859,
0.74205,
0.7454,
0.74906,
0.75314,
0.75806,
0.76391,
0.77058,
0.77774,
0.78528,
0.79294,
0.80165,
0.81185,
0.82321,
0.83586,
0.84897,
0.86241,
0.87531,
0.88734,
0.8986,
0.90793,
0.91679,
0.92515,
0.93321,
0.94062,
0.94693,
0.95185,
0.95577,
0.95956,
0.96299,
0.96583,
0.96816,
0.97022,
0.97183,
0.97339,
0.97516,
0.97697,
0.97904,
0.98064,
0.98223,
0.98348,
0.98467,
0.98559,
0.98616,
0.98663,
```

```
          0.98702,
          0.98729,
          0.98759,
          0.98788,
          0.98825,
          0.98863,
          0.989,
          0.98944,
          0.99021,
          0.99084,
          0.99184,
          0.99409,
          0.99606,
          0.99981,
          0.99996,
          0.99997]
```

```
In [15]: eq_levels = np.around(np.multiply(cdf,255))
```

```
In [16]: eq_levels
```

```
Out[16]: array([  0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,   0.,
                  0.,   0.,   0.,   0.,   0.,   1.,   2.,   5.,   9.,  14.,  18.,
                 22.,  27.,  30.,  34.,  37.,  39.,  41.,  43.,  45.,  47.,  49.,
                 51.,  53.,  54.,  56.,  57.,  59.,  60.,  61.,  63.,  64.,  65.,
                 66.,  67.,  68.,  69.,  69.,  70.,  71.,  72.,  73.,  73.,  74.,
                 75.,  76.,  76.,  77.,  78.,  79.,  80.,  80.,  81.,  82.,  83.,
                 84.,  85.,  86.,  87.,  88.,  89.,  90.,  92.,  93.,  94.,  95.,
                 96.,  97.,  98., 100., 101., 102., 103., 104., 105., 106., 108.,
                109., 110., 111., 112., 113., 114., 115., 117., 118., 119., 120.,
                121., 122., 123., 124., 125., 126., 127., 128., 129., 130., 131.,
                132., 132., 133., 134., 135., 136., 137., 138., 139., 140., 140.,
                141., 142., 143., 144., 145., 146., 147., 148., 149., 150., 151.,
                152., 153., 153., 154., 155., 156., 157., 158., 158., 159., 160.,
                161., 162., 162., 163., 164., 164., 165., 166., 166., 167., 167.,
                168., 168., 169., 169., 170., 170., 171., 171., 171., 172., 172.,
                173., 173., 173., 174., 174., 174., 174., 175., 175., 175., 176.,
                176., 176., 177., 177., 178., 178., 179., 179., 180., 180., 181.,
                182., 183., 184., 184., 185., 186., 187., 188., 188., 189., 190.,
                191., 192., 193., 195., 196., 198., 200., 202., 204., 207., 210.,
                213., 216., 220., 223., 226., 229., 232., 234., 236., 238., 240.,
                241., 243., 244., 245., 246., 246., 247., 247., 248., 248., 249.,
                249., 250., 250., 250., 251., 251., 251., 251., 252., 252., 252.,
                252., 252., 252., 252., 252., 252., 253., 253., 253., 253., 254.,
                255., 255., 255.])
```

```
In [27]: eq_hist = np.zeros(hist.shape)
```

```
         for i in range(len(eq_hist)):
```

```
            eq_hist[i] = eq_levels[i]

        eq_hist = np.array(eq_hist,dtype=np.uint8)
```

In [28]:
```
eq_hist
```

Out[28]:
```
array([  0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,
         0,   0,   0,   1,   2,   5,   9,  14,  18,  22,  27,  30,  34,
        37,  39,  41,  43,  45,  47,  49,  51,  53,  54,  56,  57,  59,
        60,  61,  63,  64,  65,  66,  67,  68,  69,  69,  70,  71,  72,
        73,  73,  74,  75,  76,  76,  77,  78,  79,  80,  80,  81,  82,
        83,  84,  85,  86,  87,  88,  89,  90,  92,  93,  94,  95,  96,
        97,  98, 100, 101, 102, 103, 104, 105, 106, 108, 109, 110, 111,
       112, 113, 114, 115, 117, 118, 119, 120, 121, 122, 123, 124, 125,
       126, 127, 128, 129, 130, 131, 132, 132, 133, 134, 135, 136, 137,
       138, 139, 140, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149,
       150, 151, 152, 153, 153, 154, 155, 156, 157, 158, 158, 159, 160,
       161, 162, 162, 163, 164, 164, 165, 166, 166, 167, 167, 168, 168,
       169, 169, 170, 170, 171, 171, 171, 172, 172, 173, 173, 173, 174,
       174, 174, 174, 175, 175, 175, 176, 176, 176, 177, 177, 178, 178,
       179, 179, 180, 180, 181, 182, 183, 184, 184, 185, 186, 187, 188,
       188, 189, 190, 191, 192, 193, 195, 196, 198, 200, 202, 204, 207,
       210, 213, 216, 220, 223, 226, 229, 232, 234, 236, 238, 240, 241,
       243, 244, 245, 246, 246, 247, 247, 248, 248, 249, 249, 250, 250,
       250, 251, 251, 251, 251, 252, 252, 252, 252, 252, 252, 252, 252,
       252, 253, 253, 253, 253, 254, 255, 255, 255], dtype=uint8)
```

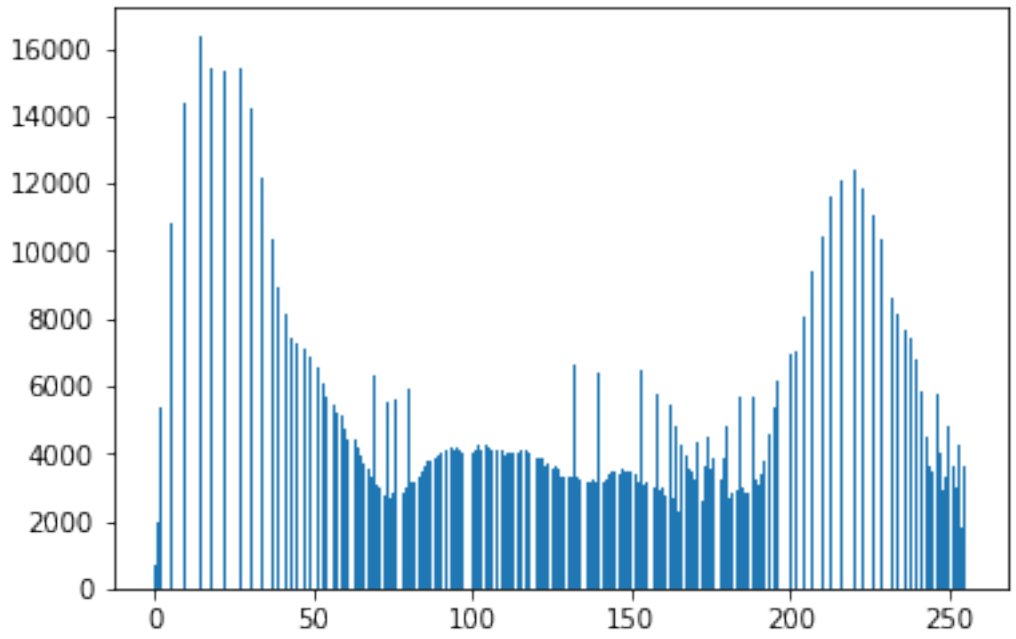
In [29]:
```
final_hist = np.zeros(256)

for i in range(len(final_hist)):
    final_hist[eq_hist[i]]+=hist[i]

final_hist
```

Out[29]:
```
array([  728.,  1936.,  5401.,     0.,     0., 10821.,     0.,     0.,
            0., 14391.,     0.,     0.,     0.,     0., 16384.,     0.,
            0.,     0., 15396.,     0.,     0.,     0., 15370.,     0.,
            0.,     0.,     0., 15431.,     0.,     0., 14197.,     0.,
            0.,     0., 12152.,     0.,     0., 10335.,     0.,  8940.,
            0.,  8095.,     0.,  7457.,     0.,  7232.,     0.,  7098.,
            0.,  6888.,     0.,  6534.,     0.,  6107.,  5679.,     0.,
         5436.,  5173.,     0.,  5090.,  4748.,  4426.,     0.,  4390.,
         4162.,  3966.,  3668.,  3550.,  3282.,  6294.,  3059.,  2981.,
         2774.,  5508.,  2694.,  2807.,  5571.,  2775.,  2808.,  2962.,
         5927.,  3134.,  3153.,  3328.,  3479.,  3634.,  3807.,  3816.,
         3848.,  3935.,  3999.,     0.,  4085.,  4175.,  4095.,  4162.,
         4094.,  3993.,  4069.,     0.,  3988.,  4104.,  4228.,  4102.,
         4221.,  4183.,  4091.,     0.,  4135.,  4099.,  3979.,  4011.,
         4054.,  4035.,  4030.,  4113.,     0.,  4103.,  3981.,  3895.,
```

```
       3841.,  3892.,  3827.,  3662.,  3705.,  3564.,  3617.,  3537.,
       3326.,  3321.,  3327.,  3331.,  6602.,  3278.,  3251.,  3269.,
       3119.,  3167.,  3213.,  3179.,  6375.,  3168.,  3227.,  3387.,
       3445.,  3475.,  3383.,  3506.,  3431.,  3480.,  3467.,  3423.,
       3182.,  6446.,  3109.,  3117.,  3098.,  2983.,  5752.,  2930.,
       2983.,  2781.,  5412.,  2671.,  4843.,  2313.,  4296.,  3955.,
       3553.,  3504.,  3252.,  4323.,  2605.,  3614.,  4455.,  3513.,
       3829.,  2686.,  3220.,  3877.,  4820.,  2711.,  2793.,  2872.,
       5675.,  3006.,  2807.,  2820.,  5652.,  3189.,  3083.,  3373.,
       3754.,  4534.,     0.,  5389.,  6143.,     0.,  6591.,     0.,
       6945.,     0.,  7054.,     0.,  8021.,     0.,     0.,  9393.,
          0.,     0., 10456.,     0.,     0., 11643.,     0.,     0.,
      12068.,     0.,     0.,     0., 12370.,     0.,     0., 11875.,
          0.,     0., 11074.,     0.,     0., 10362.,     0.,     0.,
       8586.,     0.,  8161.,     0.,  7692.,     0.,  7417.,     0.,
       6819.,  5808.,     0.,  4527.,  3606.,  3490.,  5777.,  4046.,
       2912.,  3295.,  4840.,  3611.,  3010.,  4283.,  1813.,  3596.])
```

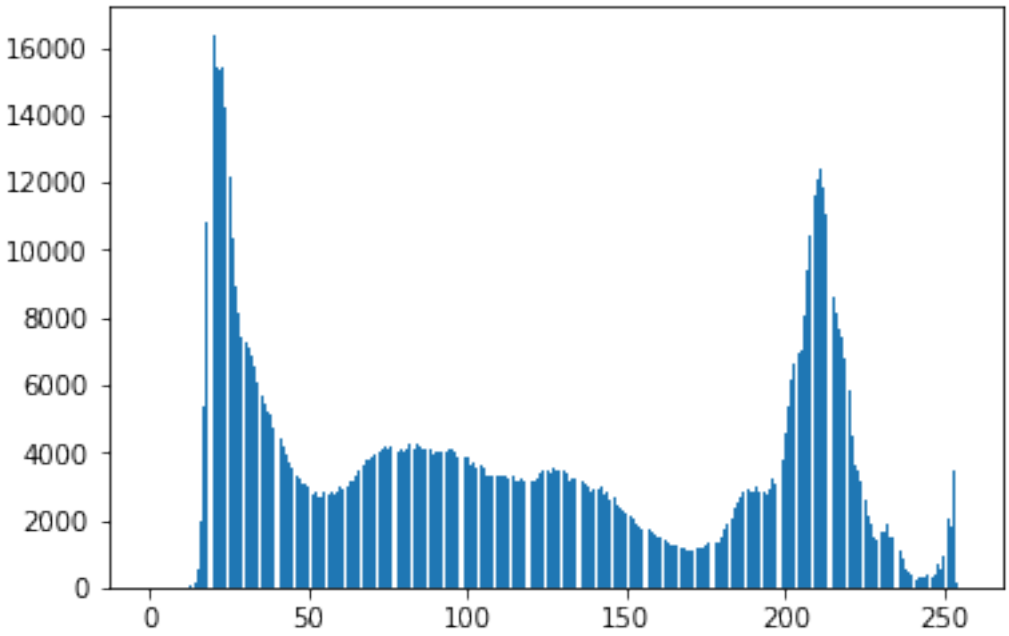
In [30]: `plt.bar(index,final_hist)`

Out[30]: `<BarContainer object of 256 artists>`

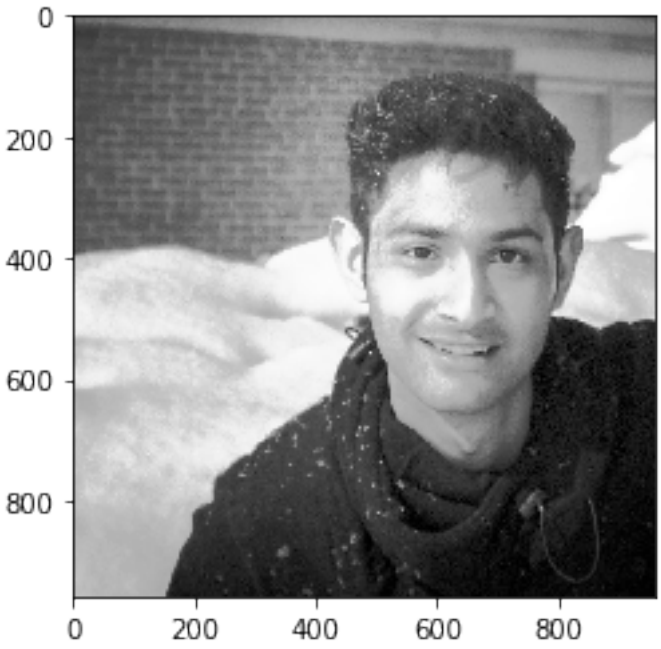


In [22]: `plt.bar(index,hist)`

Out[22]: `<BarContainer object of 256 artists>`

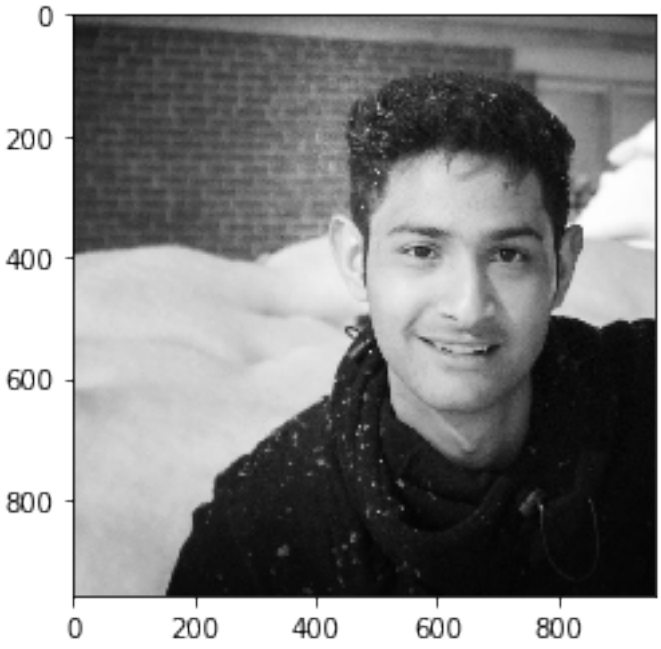



```
In [46]: out=np.zeros(gray.shape)
         for i in range(gray.shape[0]):
             for j in range(gray.shape[1]):
                 out[i][j]=eq_levels[gray[i][j]]
         out = np.array(out,dtype=np.uint8)

In [47]: out

Out[47]: array([[ 92, 100, 105, ...,  80,  86,  96],
                [ 98, 102, 102, ...,  80,  83,  90],
                [106, 105, 100, ...,  80,  81,  85],
                ...,
                [193, 193, 193, ...,   1,   2,   5],
                [195, 195, 195, ...,   1,   2,   5],
                [195, 195, 195, ...,   1,   2,   5]], dtype=uint8)

In [48]: plt.imshow(out,cmap='gray')

Out[48]: <matplotlib.image.AxesImage at 0x1286f5358>
```

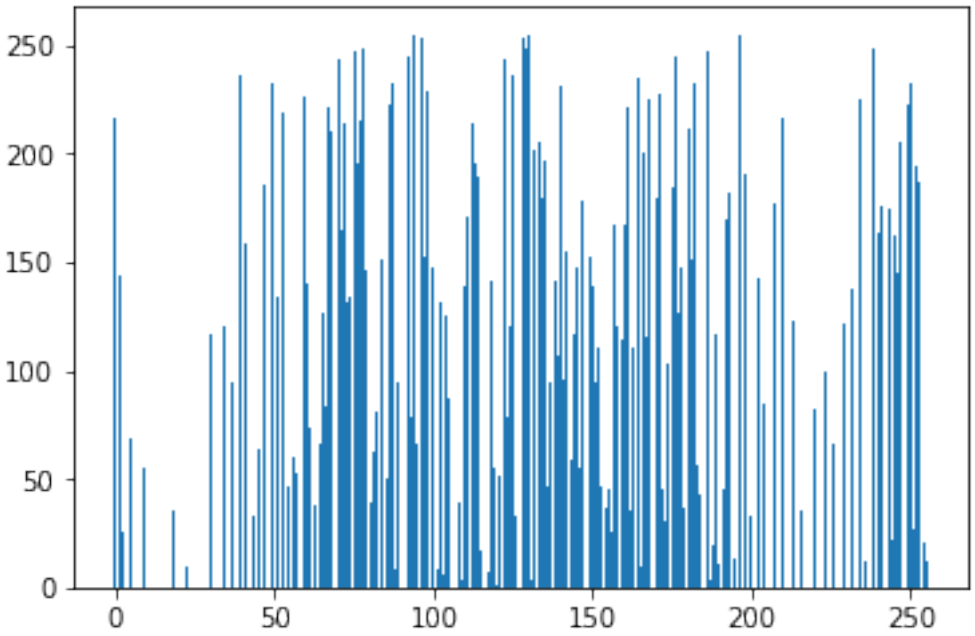



```
In [49]: plt.imshow(gray,cmap='gray')
```

```
Out[49]: <matplotlib.image.AxesImage at 0x1287c15f8>
```

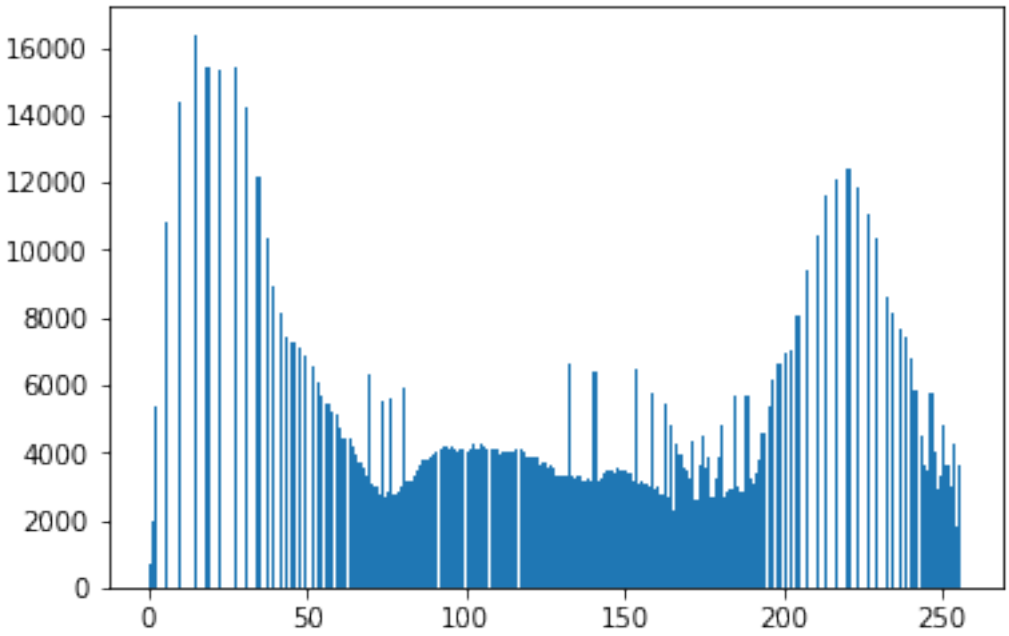


```
In [50]: out_hist = np.zeros(256)
         for i in range(out.shape[0]):
             for j in range(out.shape[1]):
                 out_hist[out[i,j]] = out_hist[out[i,j]]+1

In [51]: out_hist = np.array(out_hist,dtype=np.uint8)
         out_hist

Out[51]: array([216, 144,  25,   0,   0,  69,   0,   0,   0,  55,   0,   0,   0,
                  0,   0,   0,   0,   0,  36,   0,   0,   0,  10,   0,   0,   0,
                  0,  71,   0,   0, 117,   0,   0,   0, 120,   0,   0,  95,   0,
                236,   0, 159,   0,  33,   0,  64,   0, 186,   0, 232,   0, 134,
                  0, 219,  47,   0,  60,  53,   0, 226, 140,  74,   0,  38,  66,
                126,  84, 222, 210, 150, 243, 165, 214, 132, 134, 247, 195, 215,
                248, 146,  39,  62,  81,   0, 151,  50, 223, 232,   8,  95, 159,
                  0, 245,  79, 255,  66, 254, 153, 229,   0, 148,   8, 132,   6,
                125,  87, 251,   0,  39,   3, 139, 171, 214, 195, 190,  17,   0,
                  7, 141,  55,   1,  52, 243,  78, 121, 236,  33, 209, 254, 249,
                255,   3, 202, 206, 179, 197,  47,  95, 141, 107, 231,  96, 155,
                 59, 117, 147,  55, 178, 103, 152, 139,  95, 110,  46,  37,  45,
                 26, 167, 120, 114, 167, 221,  36, 111, 235,   9, 200, 115, 225,
                176, 180, 227,  45,  30, 103, 185, 245, 126, 148,  37, 212, 151,
                233,  56,  43, 190, 247,   4,  20, 117,  11,  45, 170, 182,   0,
                 13, 255,   0, 191,   0,  33,   0, 142,   0,  85,   0,   0, 177,
                  0,   0, 216,   0,   0, 123,   0,   0,  36,   0,   0,   0,  82,
                  0,   0,  99,   0,   0,  66,   0,   0, 122,   0,   0, 138,   0,
                225,   0,  12,   0, 249,   0, 163, 176,   0, 175,  22, 162, 145,
                206,  96, 223, 232,  27, 194, 187,  21,  12], dtype=uint8)

In [52]: plt.bar(index,out_hist)

Out[52]: <BarContainer object of 256 artists>
```




```
In [53]: plt.hist(out.ravel(),256,[0,256])

Out[53]: (array([  728.,  1936.,  5401.,     0.,     0., 10821.,     0.,     0.,
                      0., 14391.,     0.,     0.,     0.,     0., 16384.,     0.,
                      0.,     0., 15396.,     0.,     0.,     0., 15370.,     0.,
                      0.,     0.,     0., 15431.,     0.,     0., 14197.,     0.,
                      0.,     0., 12152.,     0.,     0., 10335.,     0.,  8940.,
                      0.,  8095.,     0.,  7457.,     0.,  7232.,     0.,  7098.,
                      0.,  6888.,     0.,  6534.,     0.,  6107.,  5679.,     0.,
                   5436.,  5173.,     0.,  5090.,  4748.,  4426.,     0.,  4390.,
                   4162.,  3966.,  3668.,  3550.,  3282.,  6294.,  3059.,  2981.,
                   2774.,  5508.,  2694.,  2807.,  5571.,  2775.,  2808.,  2962.,
                   5927.,  3134.,  3153.,  3328.,  3479.,  3634.,  3807.,  3816.,
                   3848.,  3935.,  3999.,     0.,  4085.,  4175.,  4095.,  4162.,
                   4094.,  3993.,  4069.,     0.,  3988.,  4104.,  4228.,  4102.,
                   4221.,  4183.,  4091.,     0.,  4135.,  4099.,  3979.,  4011.,
                   4054.,  4035.,  4030.,  4113.,     0.,  4103.,  3981.,  3895.,
                   3841.,  3892.,  3827.,  3662.,  3705.,  3564.,  3617.,  3537.,
                   3326.,  3321.,  3327.,  3331.,  6602.,  3278.,  3251.,  3269.,
                   3119.,  3167.,  3213.,  3179.,  6375.,  3168.,  3227.,  3387.,
                   3445.,  3475.,  3383.,  3506.,  3431.,  3480.,  3467.,  3423.,
                   3182.,  6446.,  3109.,  3117.,  3098.,  2983.,  5752.,  2930.,
                   2983.,  2781.,  5412.,  2671.,  4843.,  2313.,  4296.,  3955.,
                   3553.,  3504.,  3252.,  4323.,  2605.,  3614.,  4455.,  3513.,
                   3829.,  2686.,  3220.,  3877.,  4820.,  2711.,  2793.,  2872.,
                   5675.,  3006.,  2807.,  2820.,  5652.,  3189.,  3083.,  3373.,
```

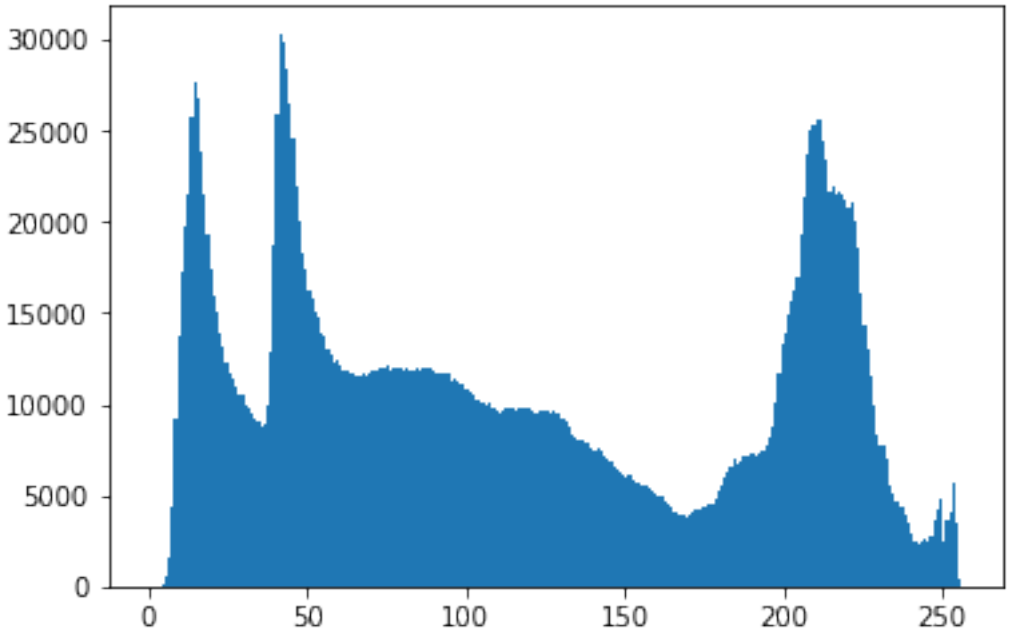
```
        3754.,  4534.,     0.,  5389.,  6143.,     0.,  6591.,     0.,
        6945.,     0.,  7054.,     0.,  8021.,     0.,     0.,  9393.,
           0.,     0., 10456.,     0.,     0., 11643.,     0.,     0.,
       12068.,     0.,     0.,     0., 12370.,     0.,     0., 11875.,
           0.,     0., 11074.,     0.,     0., 10362.,     0.,     0.,
        8586.,     0.,  8161.,     0.,  7692.,     0.,  7417.,     0.,
        6819.,  5808.,     0.,  4527.,  3606.,  3490.,  5777.,  4046.,
        2912.,  3295.,  4840.,  3611.,  3010.,  4283.,  1813.,  3596.]),
array([  0.,   1.,   2.,   3.,   4.,   5.,   6.,   7.,   8.,   9.,  10.,
        11.,  12.,  13.,  14.,  15.,  16.,  17.,  18.,  19.,  20.,  21.,
        22.,  23.,  24.,  25.,  26.,  27.,  28.,  29.,  30.,  31.,  32.,
        33.,  34.,  35.,  36.,  37.,  38.,  39.,  40.,  41.,  42.,  43.,
        44.,  45.,  46.,  47.,  48.,  49.,  50.,  51.,  52.,  53.,  54.,
        55.,  56.,  57.,  58.,  59.,  60.,  61.,  62.,  63.,  64.,  65.,
        66.,  67.,  68.,  69.,  70.,  71.,  72.,  73.,  74.,  75.,  76.,
        77.,  78.,  79.,  80.,  81.,  82.,  83.,  84.,  85.,  86.,  87.,
        88.,  89.,  90.,  91.,  92.,  93.,  94.,  95.,  96.,  97.,  98.,
        99., 100., 101., 102., 103., 104., 105., 106., 107., 108., 109.,
       110., 111., 112., 113., 114., 115., 116., 117., 118., 119., 120.,
       121., 122., 123., 124., 125., 126., 127., 128., 129., 130., 131.,
       132., 133., 134., 135., 136., 137., 138., 139., 140., 141., 142.,
       143., 144., 145., 146., 147., 148., 149., 150., 151., 152., 153.,
       154., 155., 156., 157., 158., 159., 160., 161., 162., 163., 164.,
       165., 166., 167., 168., 169., 170., 171., 172., 173., 174., 175.,
       176., 177., 178., 179., 180., 181., 182., 183., 184., 185., 186.,
       187., 188., 189., 190., 191., 192., 193., 194., 195., 196., 197.,
       198., 199., 200., 201., 202., 203., 204., 205., 206., 207., 208.,
       209., 210., 211., 212., 213., 214., 215., 216., 217., 218., 219.,
       220., 221., 222., 223., 224., 225., 226., 227., 228., 229., 230.,
       231., 232., 233., 234., 235., 236., 237., 238., 239., 240., 241.,
       242., 243., 244., 245., 246., 247., 248., 249., 250., 251., 252.,
       253., 254., 255., 256.]),
<a list of 256 Patch objects>)
```




```
In [54]: plt.hist(image.ravel(),256,[0,256])

Out[54]: (array([1.0000e+00, 3.0000e+00, 1.1000e+01, 3.6000e+01, 1.3600e+02,
                 5.6400e+02, 1.6230e+03, 4.3590e+03, 9.2610e+03, 1.3725e+04,
                 1.7203e+04, 1.9697e+04, 2.1563e+04, 2.5752e+04, 2.7741e+04,
                 2.6778e+04, 2.3856e+04, 2.1448e+04, 1.9279e+04, 1.7449e+04,
                 1.6002e+04, 1.5037e+04, 1.3903e+04, 1.3218e+04, 1.2298e+04,
                 1.1694e+04, 1.1451e+04, 1.0985e+04, 1.0499e+04, 1.0584e+04,
                 9.9930e+03, 9.7670e+03, 9.4280e+03, 9.2190e+03, 9.0150e+03,
                 8.7210e+03, 8.9560e+03, 9.9120e+03, 1.2814e+04, 1.8721e+04,
                 2.5932e+04, 3.0355e+04, 2.9881e+04, 2.8446e+04, 2.6528e+04,
                 2.4577e+04, 2.1939e+04, 2.0000e+04, 1.8323e+04, 1.7394e+04,
                 1.6267e+04, 1.5848e+04, 1.5051e+04, 1.4769e+04, 1.3840e+04,
                 1.3717e+04, 1.3040e+04, 1.2698e+04, 1.2313e+04, 1.2416e+04,
                 1.2076e+04, 1.1903e+04, 1.1871e+04, 1.1764e+04, 1.1654e+04,
                 1.1616e+04, 1.1490e+04, 1.1664e+04, 1.1596e+04, 1.1751e+04,
                 1.1830e+04, 1.1907e+04, 1.1918e+04, 1.1971e+04, 1.2051e+04,
                 1.2086e+04, 1.1887e+04, 1.2054e+04, 1.2045e+04, 1.1950e+04,
                 1.1877e+04, 1.1963e+04, 1.1916e+04, 1.1820e+04, 1.2002e+04,
                 1.1802e+04, 1.1939e+04, 1.2020e+04, 1.2028e+04, 1.1790e+04,
                 1.1703e+04, 1.1687e+04, 1.1759e+04, 1.1672e+04, 1.1714e+04,
                 1.1325e+04, 1.1396e+04, 1.1267e+04, 1.1069e+04, 1.0861e+04,
                 1.0786e+04, 1.0725e+04, 1.0469e+04, 1.0190e+04, 1.0117e+04,
                 1.0044e+04, 9.8820e+03, 1.0126e+04, 9.8470e+03, 9.6230e+03,
                 9.4930e+03, 9.6140e+03, 9.8550e+03, 9.7420e+03, 9.7550e+03,
                 9.7080e+03, 9.7800e+03, 9.7520e+03, 9.8470e+03, 9.7300e+03,
```

```
       9.7080e+03, 9.5240e+03, 9.5490e+03, 9.6560e+03, 9.6250e+03,
       9.5850e+03, 9.4810e+03, 9.6010e+03, 9.5480e+03, 9.4350e+03,
       9.2060e+03, 9.0970e+03, 8.7810e+03, 8.3940e+03, 8.2500e+03,
       7.9640e+03, 7.9960e+03, 7.8940e+03, 7.8460e+03, 7.6050e+03,
       7.5130e+03, 7.5610e+03, 7.3850e+03, 7.1510e+03, 6.9760e+03,
       6.8790e+03, 6.5170e+03, 6.4170e+03, 6.2190e+03, 6.1560e+03,
       6.0380e+03, 6.0940e+03, 5.8900e+03, 5.6710e+03, 5.7190e+03,
       5.5430e+03, 5.6000e+03, 5.4570e+03, 5.2950e+03, 5.1310e+03,
       4.9750e+03, 4.8830e+03, 4.7140e+03, 4.4980e+03, 4.3420e+03,
       4.0420e+03, 4.0760e+03, 3.9840e+03, 3.9940e+03, 3.8130e+03,
       3.9550e+03, 4.0100e+03, 4.1760e+03, 4.1590e+03, 4.3050e+03,
       4.4060e+03, 4.4710e+03, 4.5170e+03, 4.8100e+03, 5.2020e+03,
       5.5220e+03, 5.9230e+03, 6.2150e+03, 6.5960e+03, 6.9390e+03,
       6.7770e+03, 6.8780e+03, 7.2110e+03, 7.1220e+03, 7.3160e+03,
       7.2600e+03, 7.1840e+03, 7.3630e+03, 7.4980e+03, 7.7910e+03,
       8.2060e+03, 8.8030e+03, 1.0022e+04, 1.1720e+04, 1.3329e+04,
       1.3956e+04, 1.4900e+04, 1.5681e+04, 1.6289e+04, 1.7004e+04,
       1.9289e+04, 2.1375e+04, 2.3741e+04, 2.4990e+04, 2.5335e+04,
       2.5610e+04, 2.5646e+04, 2.4449e+04, 2.3451e+04, 2.1688e+04,
       2.1891e+04, 2.1551e+04, 2.1633e+04, 2.1568e+04, 2.1183e+04,
       2.0715e+04, 2.1077e+04, 2.0071e+04, 1.8596e+04, 1.6092e+04,
       1.4393e+04, 1.3026e+04, 1.1503e+04, 9.9660e+03, 8.2790e+03,
       7.7190e+03, 7.7200e+03, 6.9430e+03, 5.5660e+03, 5.0940e+03,
       4.7120e+03, 4.3880e+03, 4.3420e+03, 3.8980e+03, 3.5120e+03,
       2.8970e+03, 2.5100e+03, 2.3760e+03, 2.4630e+03, 2.5780e+03,
       2.4980e+03, 2.6890e+03, 3.6340e+03, 4.2020e+03, 4.7830e+03,
       2.4340e+03, 3.6320e+03, 4.1280e+03, 5.6280e+03, 3.4350e+03,
       4.1200e+02]),
array([  0.,   1.,   2.,   3.,   4.,   5.,   6.,   7.,   8.,   9.,  10.,
        11.,  12.,  13.,  14.,  15.,  16.,  17.,  18.,  19.,  20.,  21.,
        22.,  23.,  24.,  25.,  26.,  27.,  28.,  29.,  30.,  31.,  32.,
        33.,  34.,  35.,  36.,  37.,  38.,  39.,  40.,  41.,  42.,  43.,
        44.,  45.,  46.,  47.,  48.,  49.,  50.,  51.,  52.,  53.,  54.,
        55.,  56.,  57.,  58.,  59.,  60.,  61.,  62.,  63.,  64.,  65.,
        66.,  67.,  68.,  69.,  70.,  71.,  72.,  73.,  74.,  75.,  76.,
        77.,  78.,  79.,  80.,  81.,  82.,  83.,  84.,  85.,  86.,  87.,
        88.,  89.,  90.,  91.,  92.,  93.,  94.,  95.,  96.,  97.,  98.,
        99., 100., 101., 102., 103., 104., 105., 106., 107., 108., 109.,
       110., 111., 112., 113., 114., 115., 116., 117., 118., 119., 120.,
       121., 122., 123., 124., 125., 126., 127., 128., 129., 130., 131.,
       132., 133., 134., 135., 136., 137., 138., 139., 140., 141., 142.,
       143., 144., 145., 146., 147., 148., 149., 150., 151., 152., 153.,
       154., 155., 156., 157., 158., 159., 160., 161., 162., 163., 164.,
       165., 166., 167., 168., 169., 170., 171., 172., 173., 174., 175.,
       176., 177., 178., 179., 180., 181., 182., 183., 184., 185., 186.,
       187., 188., 189., 190., 191., 192., 193., 194., 195., 196., 197.,
       198., 199., 200., 201., 202., 203., 204., 205., 206., 207., 208.,
       209., 210., 211., 212., 213., 214., 215., 216., 217., 218., 219.,
```

```
        220., 221., 222., 223., 224., 225., 226., 227., 228., 229., 230.,
        231., 232., 233., 234., 235., 236., 237., 238., 239., 240., 241.,
        242., 243., 244., 245., 246., 247., 248., 249., 250., 251., 252.,
        253., 254., 255., 256.]),
 <a list of 256 Patch objects>)
```



In [ ]: