

# Histogram specification

April 28, 2019

```
In [1]: import cv2
import numpy as np
import matplotlib.pyplot as plt

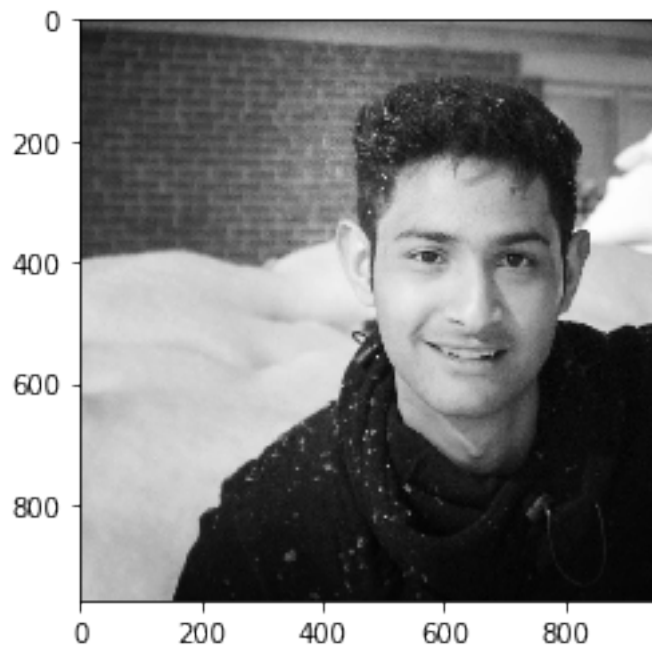
In [12]: image1 = cv2.imread('sample7.jpg')
image1 = cv2.cvtColor(image1, cv2.COLOR_BGR2GRAY)

In [16]: image1.shape

Out[16]: (959, 960)

In [13]: plt.imshow(image1, cmap='gray')

Out[13]: <matplotlib.image.AxesImage at 0x13195c780>
```



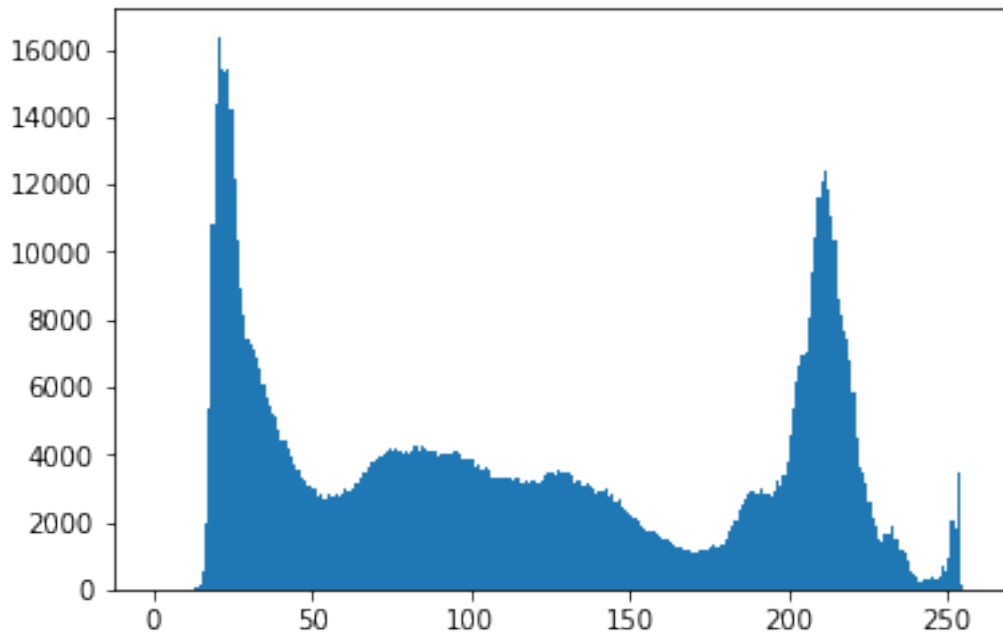
```
In [18]: plt.hist(image1.ravel(), 256, [0, 256])
```

```
Out[18]: (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]),
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>)

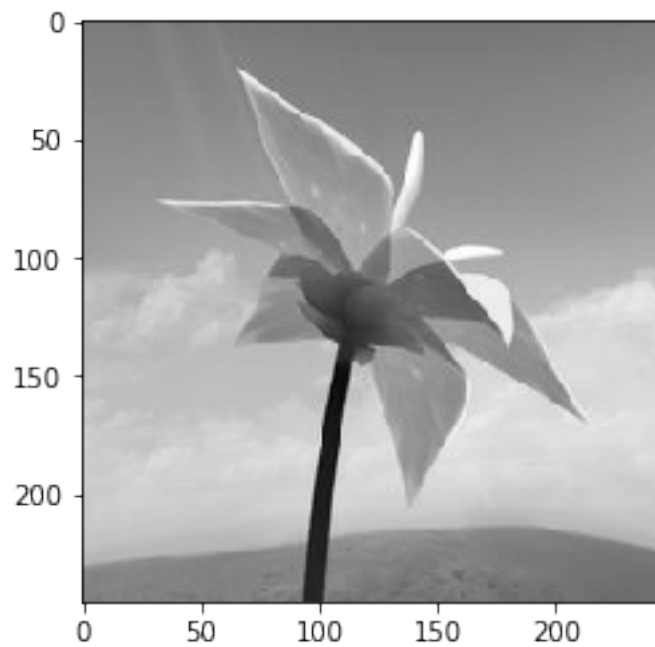
```



```
In [14]: image2 = cv2.imread('flower1.jpg')  
         image2 = cv2.cvtColor(image2,cv2.COLOR_BGR2GRAY)
```

```
In [15]: plt.imshow(image2,cmap='gray')
```

```
Out[15]: <matplotlib.image.AxesImage at 0x131d64a58>
```



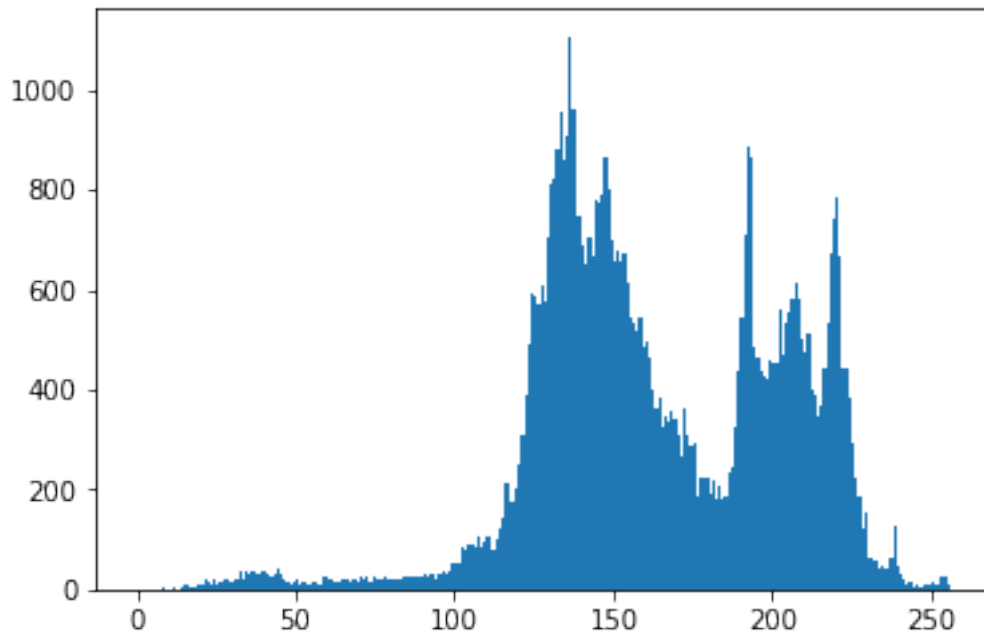
```
In [19]: plt.hist(image2.ravel(),256,[0,256])
```

```
Out[19]: (array([0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
0.000e+00, 0.000e+00, 2.000e+00, 0.000e+00, 1.000e+00, 3.000e+00,
1.000e+00, 1.000e+00, 6.000e+00, 7.000e+00, 3.000e+00, 4.000e+00,
5.000e+00, 8.000e+00, 9.000e+00, 1.800e+01, 1.300e+01, 1.000e+01,
2.000e+01, 1.100e+01, 1.400e+01, 2.100e+01, 2.100e+01, 1.700e+01,
1.700e+01, 2.200e+01, 3.400e+01, 1.800e+01, 3.400e+01, 3.200e+01,
3.700e+01, 2.900e+01, 3.300e+01, 3.800e+01, 3.500e+01, 3.100e+01,
2.600e+01, 3.200e+01, 4.200e+01, 2.900e+01, 1.900e+01, 1.600e+01,
1.100e+01, 1.700e+01, 1.900e+01, 9.000e+00, 1.700e+01, 1.200e+01,
1.200e+01, 1.500e+01, 1.600e+01, 7.000e+00, 2.300e+01, 2.600e+01,
1.900e+01, 2.100e+01, 1.500e+01, 1.700e+01, 2.200e+01, 2.000e+01,
2.000e+01, 1.600e+01, 1.900e+01, 1.300e+01, 2.400e+01, 1.800e+01,
2.500e+01, 1.600e+01, 2.300e+01, 1.800e+01, 2.200e+01, 2.100e+01,
2.400e+01, 2.100e+01, 2.000e+01, 2.100e+01, 1.800e+01, 1.900e+01,
2.400e+01, 2.600e+01, 2.500e+01, 2.300e+01, 2.800e+01, 2.500e+01,
3.300e+01, 2.300e+01, 3.300e+01, 3.000e+01, 2.000e+01, 3.100e+01,
3.400e+01, 3.000e+01, 3.900e+01, 5.200e+01, 5.100e+01, 5.200e+01,
8.200e+01, 8.000e+01, 9.000e+01, 9.100e+01, 8.500e+01, 1.040e+02,
8.500e+01, 9.400e+01, 1.040e+02, 8.000e+01, 7.900e+01, 9.800e+01,
1.240e+02, 1.420e+02, 2.140e+02, 1.770e+02, 1.750e+02, 2.030e+02,
2.490e+02, 3.080e+02, 3.900e+02, 4.910e+02, 5.900e+02, 5.850e+02,
5.710e+02, 6.090e+02, 5.760e+02, 7.050e+02, 8.130e+02, 8.210e+02,
8.790e+02, 9.540e+02, 8.570e+02, 9.090e+02, 1.107e+03, 9.590e+02,
7.480e+02, 7.460e+02, 6.900e+02, 6.500e+02, 7.040e+02, 6.690e+02,
7.820e+02, 7.760e+02, 7.920e+02, 8.630e+02, 7.990e+02, 7.000e+02,
6.560e+02, 6.790e+02, 6.590e+02, 6.720e+02, 6.160e+02, 5.430e+02,
5.350e+02, 5.150e+02, 5.440e+02, 4.860e+02, 4.940e+02, 4.660e+02,
4.000e+02, 3.600e+02, 3.820e+02, 3.250e+02, 3.490e+02, 3.350e+02,
3.590e+02, 3.410e+02, 3.080e+02, 2.660e+02, 3.600e+02, 3.090e+02,
2.850e+02, 2.950e+02, 1.880e+02, 2.260e+02, 2.220e+02, 2.240e+02,
1.920e+02, 2.160e+02, 1.790e+02, 2.080e+02, 1.800e+02, 1.870e+02,
2.320e+02, 2.460e+02, 3.260e+02, 4.380e+02, 5.450e+02, 7.080e+02,
8.850e+02, 8.630e+02, 4.840e+02, 4.650e+02, 4.360e+02, 4.250e+02,
4.220e+02, 4.580e+02, 4.510e+02, 4.520e+02, 5.580e+02, 4.720e+02,
5.320e+02, 5.530e+02, 5.840e+02, 6.120e+02, 5.790e+02, 4.990e+02,
4.760e+02, 5.110e+02, 3.990e+02, 3.870e+02, 3.460e+02, 3.690e+02,
4.420e+02, 5.350e+02, 6.700e+02, 7.430e+02, 7.870e+02, 6.650e+02,
4.450e+02, 4.400e+02, 3.820e+02, 2.920e+02, 2.240e+02, 1.860e+02,
1.210e+02, 1.550e+02, 6.100e+01, 6.400e+01, 5.900e+01, 4.400e+01,
4.500e+01, 4.200e+01, 4.100e+01, 6.400e+01, 1.280e+02, 4.800e+01,
3.000e+01, 2.200e+01, 7.000e+00, 1.600e+01, 6.000e+00, 8.000e+00,
5.000e+00, 3.000e+00, 7.000e+00, 7.000e+00, 1.400e+01, 1.100e+01,
1.200e+01, 2.600e+01, 2.400e+01, 8.000e+00])),
```

```

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 [36]: def equalisation(image):
          hist = np.zeros(256)

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

          pdf = [np.around(i/(image.shape[0]*image.shape[1]),decimals=5) for i in hist]
          cdf = []
          cdf.append(pdf[0])

          for i in range(1,len(pdf1)):
              cdf.append(np.around(cdf[i-1]+pdf[i],decimals=6))

          eq_levels = np.around(np.multiply(cdf,255))

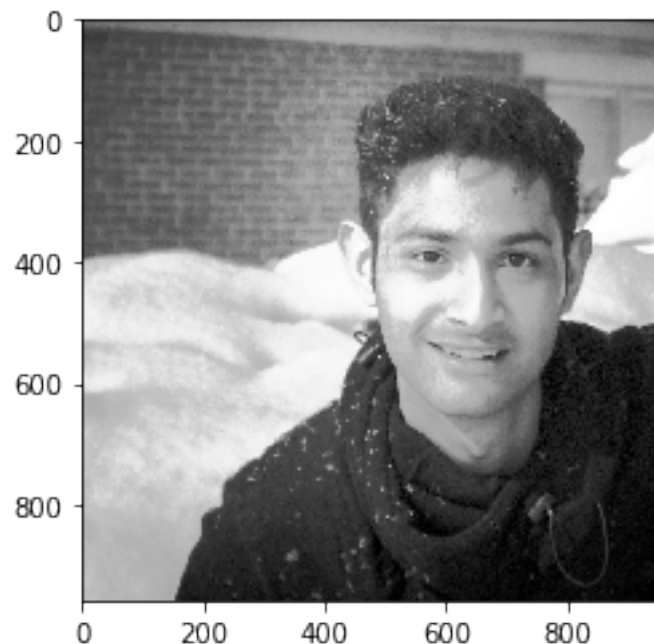
          out=np.zeros(image.shape)
          for i in range(image.shape[0]):
              for j in range(image.shape[1]):
                  out[i][j]=eq_levels[image[i][j]]
          out = np.array(out,dtype=np.uint8)
          return out,eq_levels

In [37]: out1,levels1 = equalisation(image1)

In [38]: plt.imshow(out1,cmap='gray')

Out[38]: <matplotlib.image.AxesImage at 0x128f4ce48>

```



```
In [39]: levels1
```

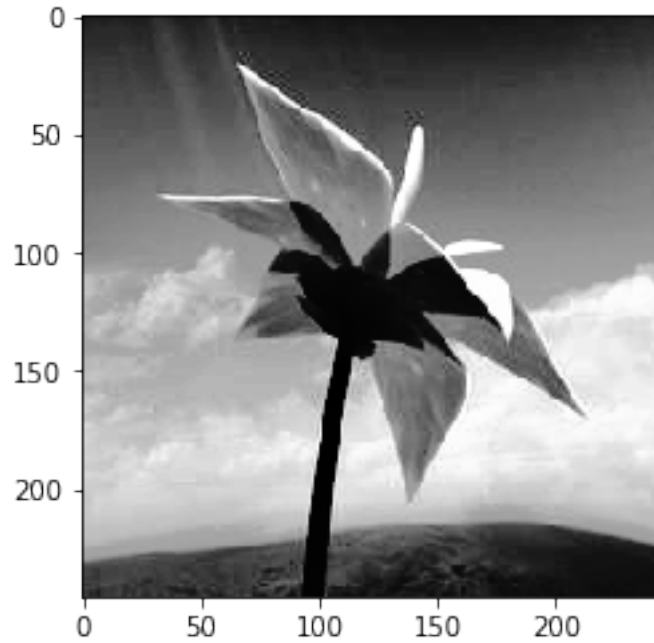
```
Out[39]: array([[ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
                  0.,  0.,  0.,  0.,  0.,  1.,  2.,  5.,  9., 14., 18.,
                  22., 27., 30., 34., 37., 39., 41., 43., 45., 47., 49.,
                  51., 53., 54., 56., 57., 59., 60., 61., 63., 64., 65.,
                  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., 253., 253., 253., 253., 254.,
                  255., 255., 255.]])
```

```
In [40]: out2, levels2 = equalisation(image2)
```

```
In [41]: plt.imshow(out2, cmap='gray')
```

```
Out[41]: <matplotlib.image.AxesImage at 0x129004278>
```





```
In [42]: levels2
```

```
Out[42]: array([[ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
  0.,  0.,  0.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,  1.,
  1.,  1.,  1.,  2.,  2.,  2.,  2.,  2.,  2.,  2.,  2.,  3.,
  3.,  3.,  3.,  3.,  3.,  3.,  3.,  3.,  3.,  3.,  3.,  3.,
  4.,  4.,  4.,  4.,  4.,  4.,  4.,  4.,  4.,  4.,  4.,  4.,
  4.,  4.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,  5.,
  5.,  5.,  5.,  6.,  6.,  6.,  6.,  6.,  6.,  6.,  6.,  6.,
  6.,  6.,  7.,  7.,  7.,  7.,  7.,  7.,  7.,  7.,  7.,  8.,
  8.,  8.,  8.,  9.,  9.,  9., 10., 10., 10., 11., 11.,
 12., 12., 12., 13., 13., 14., 15., 16., 16., 17., 18.,
 19., 21., 23., 26., 28., 31., 33., 36., 38., 42., 45.,
 49., 53., 57., 61., 65., 69., 72., 76., 78., 81., 84.,
 87., 90., 94., 97., 101., 104., 107., 110., 112., 115., 118.,
121., 123., 125., 127., 130., 132., 134., 136., 137., 139., 141.,
142., 143., 145., 146., 148., 149., 150., 152., 153., 154., 155.,
156., 157., 158., 159., 160., 161., 162., 162., 163., 164., 165.,
166., 167., 169., 171., 174., 178., 182., 184., 186., 188., 189.,
191., 193., 195., 197., 199., 201., 204., 206., 208., 211., 213.,
215., 217., 220., 221., 223., 224., 226., 228., 230., 233., 236.,
239., 242., 244., 246., 247., 249., 250., 250., 251., 252., 252.,
252., 252., 253., 253., 253., 253., 253., 254., 254., 254., 254.,
254., 254., 254., 254., 255., 255., 255., 255., 255., 255.,
255., 255., 255.]])
```

```
In [51]: final_mapping = np.zeros(256)
```

```
j = 0
i = 0
while i<256:
    if levels2[j]>=levels1[i]:
        final_mapping[i]=j
        i+=1
        continue
    else:
        j+=1
```

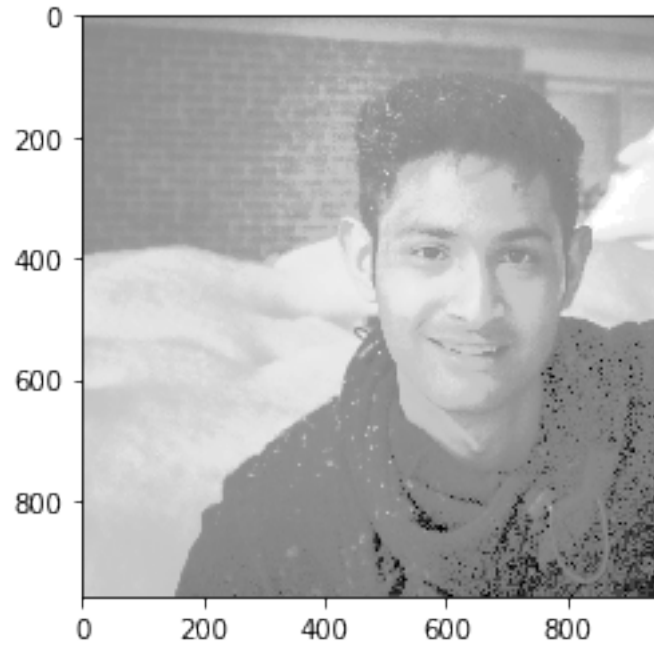
```
In [52]: final_mapping
```

```
Out[52]: array([ 0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,  0.,
                0.,  0.,  0.,  0.,  0., 25., 36., 68., 102., 115., 120.,
                123., 125., 126., 128., 129., 130., 130., 131., 131., 132., 132.,
                133., 133., 134., 134., 134., 135., 135., 135., 136., 136., 136.,
                137., 137., 137., 137., 137., 138., 138., 138., 139., 139., 139.,
                139., 139., 139., 140., 140., 141., 141., 141., 141., 142., 142.,
                142., 143., 143., 143., 144., 144., 144., 145., 145., 145., 146.,
                146., 146., 147., 147., 147., 148., 148., 148., 149., 149., 150.,
                150., 150., 151., 151., 152., 152., 152., 153., 153., 154., 154.,
                154., 155., 155., 156., 156., 157., 157., 158., 158., 158., 159.,
                159., 159., 160., 160., 161., 161., 162., 163., 163., 164., 164.,
                164., 165., 166., 167., 167., 168., 169., 169., 170., 171., 172.,
                172., 173., 173., 174., 175., 176., 177., 178., 178., 179., 180.,
                181., 182., 182., 184., 185., 185., 186., 187., 187., 188., 188.,
                189., 189., 189., 189., 190., 190., 190., 190., 190., 191., 191.,
                191., 191., 191., 191., 191., 192., 192., 192., 192.,
                192., 192., 193., 193., 193., 193., 193.,
                193., 194., 194., 194., 195., 195., 196., 196., 196., 197., 198.,
                198., 199., 199., 200., 201., 202., 203., 204., 204., 206., 207.,
                208., 210., 211., 213., 215., 217., 218., 219., 219., 220., 221.,
                221., 222., 222., 223., 223., 223., 224., 224., 225., 225., 225.,
                225., 226., 226., 226., 228., 228., 228., 228., 229., 229., 229.,
                229., 229., 229., 229., 229., 229., 233., 233., 233., 233., 238.,
                246., 246., 246.] )
```

```
In [53]: final_out=np.zeros(image1.shape)
for i in range(image1.shape[0]):
    for j in range(image1.shape[1]):
        final_out[i][j]=final_mapping[image1[i][j]]
final_out = np.array(final_out,dtype=np.uint8)
```

```
In [55]: plt.imshow(final_out,cmap='gray')
```

```
Out[55]: <matplotlib.image.AxesImage at 0x12945eeb8>
```



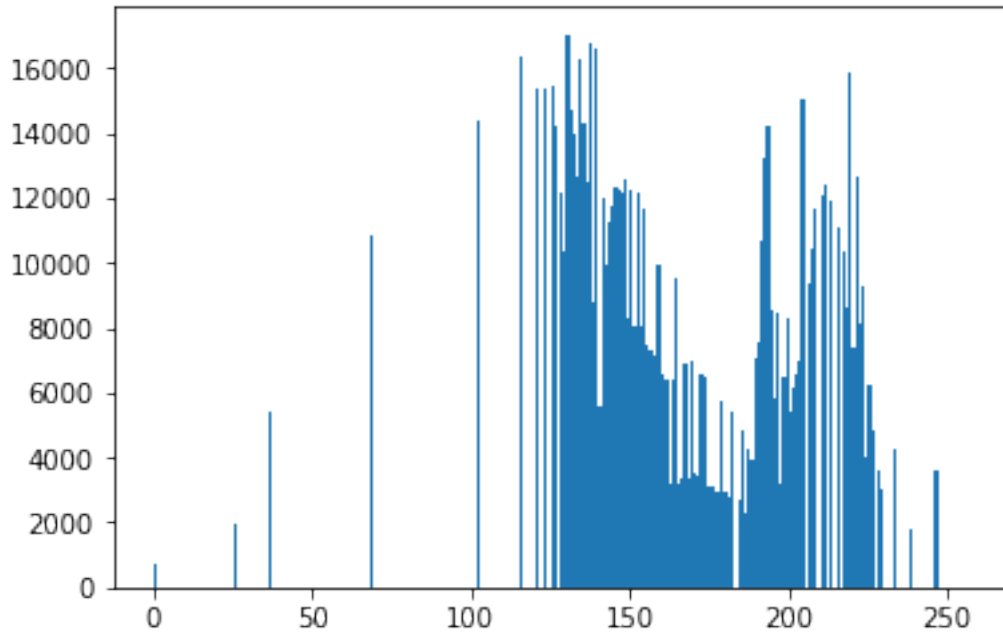
```
In [56]: plt.hist(final_out.ravel(),256,[0,256])
```

```
Out[56]: (array([ 728.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
    0., 1936.,    0.,    0.,    0.,    0.,    0.,    0.,
    0.,    0.,    0.,    0., 5401.,    0.,    0.,    0.,
    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
    0.,    0.,    0.,    0., 10821.,    0.,    0.,    0.,
    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
    0.,    0.,    0.,    0.,    0.,    0., 14391.,    0.,
    0.,    0.,    0.,    0.,    0.,    0.,    0.,    0.,
    0.,    0.,    0., 16384.,    0.,    0.,    0.,    0.,
15396.,    0.,    0., 15370.,    0., 15431., 14197.,    0.,
12152., 10335., 17035., 14689., 13986., 12641., 16288., 14264.,
12518., 16794.,  8814., 16580.,  5583., 12023.,  9960., 11257.,
11782., 12355., 12249., 12161., 12551.,  8274., 12213.,  8065.,
12178.,  8084., 11628.,  7489.,  7269.,  7154.,  9974.,  9933.,
 6529.,  6388.,  3167.,  6392.,  9543.,  3227.,  3387.,  6920.,
 3383.,  6937.,  3480.,  3467.,  6605.,  6446.,  3109.,  3117.,
 3098.,  2983.,  5752.,  2930.,  2983.,  2781.,  5412.,    0.,
 2671.,  4843.,  2313.,  4296.,  3955.,  7057.,  7575., 10674.,
```

```

13248., 14201., 8547., 5813., 8472., 3189., 6456., 8288.,
5389., 6143., 6591., 6945., 15075., 0., 9393., 10456.,
11643., 0., 12068., 12370., 0., 11875., 0., 11074.,
0., 10362., 8586., 15853., 7417., 12627., 8133., 9267.,
4046., 6207., 4840., 0., 3611., 3010., 0., 0.,
0., 4283., 0., 0., 0., 0., 1813., 0.,
0., 0., 0., 0., 0., 0., 3596., 0.,
0., 0., 0., 0., 0., 0., 0., 0.]),
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 [57]: plt.hist(image1.ravel(),256,[0,256])
```

```
Out[57]: (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,
```

```

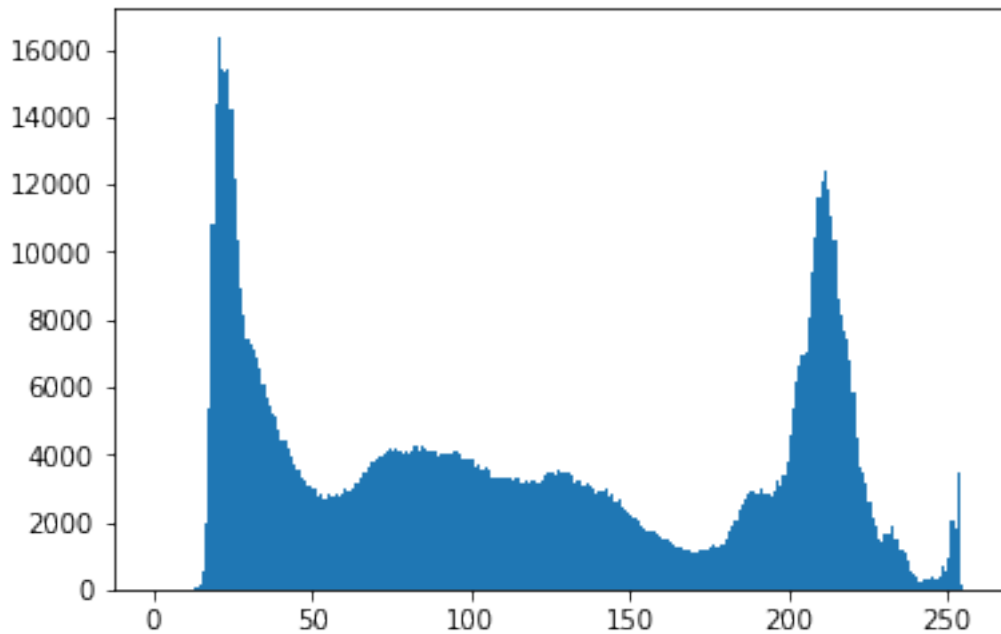
3.1680e+03, 3.1680e+03, 3.2270e+03, 3.3870e+03, 3.4450e+03,
3.4750e+03, 3.3830e+03, 3.5060e+03, 3.4310e+03, 3.4800e+03,
3.4670e+03, 3.4230e+03, 3.1820e+03, 3.2260e+03, 3.2200e+03,
3.1090e+03, 3.1170e+03, 3.0980e+03, 2.9830e+03, 2.8660e+03,
2.8860e+03, 2.9300e+03, 2.9830e+03, 2.7810e+03, 2.8090e+03,
2.6030e+03, 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]),
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 [58]: plt.hist(image2.ravel(),256,[0,256])
```

```

Out[58]: (array([0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00, 0.000e+00,
0.000e+00, 0.000e+00, 2.000e+00, 0.000e+00, 1.000e+00, 3.000e+00,
1.000e+00, 1.000e+00, 6.000e+00, 7.000e+00, 3.000e+00, 4.000e+00,
5.000e+00, 8.000e+00, 9.000e+00, 1.800e+01, 1.300e+01, 1.000e+01,
2.000e+01, 1.100e+01, 1.400e+01, 2.100e+01, 2.100e+01, 1.700e+01,
1.700e+01, 2.200e+01, 3.400e+01, 1.800e+01, 3.400e+01, 3.200e+01,
3.700e+01, 2.900e+01, 3.300e+01, 3.800e+01, 3.500e+01, 3.100e+01,
2.600e+01, 3.200e+01, 4.200e+01, 2.900e+01, 1.900e+01, 1.600e+01,
1.100e+01, 1.700e+01, 1.900e+01, 9.000e+00, 1.700e+01, 1.200e+01,
1.200e+01, 1.500e+01, 1.600e+01, 7.000e+00, 2.300e+01, 2.600e+01,
1.900e+01, 2.100e+01, 1.500e+01, 1.700e+01, 2.200e+01, 2.000e+01,
2.000e+01, 1.600e+01, 1.900e+01, 1.300e+01, 2.400e+01, 1.800e+01,
2.500e+01, 1.600e+01, 2.300e+01, 1.800e+01, 2.200e+01, 2.100e+01,
2.400e+01, 2.100e+01, 2.000e+01, 2.100e+01, 1.800e+01, 1.900e+01,
2.400e+01, 2.600e+01, 2.500e+01, 2.300e+01, 2.800e+01, 2.500e+01,
3.300e+01, 2.300e+01, 3.300e+01, 3.000e+01, 2.000e+01, 3.100e+01,
3.400e+01, 3.000e+01, 3.900e+01, 5.200e+01, 5.100e+01, 5.200e+01,
8.200e+01, 8.000e+01, 9.000e+01, 9.100e+01, 8.500e+01, 1.040e+02,

```

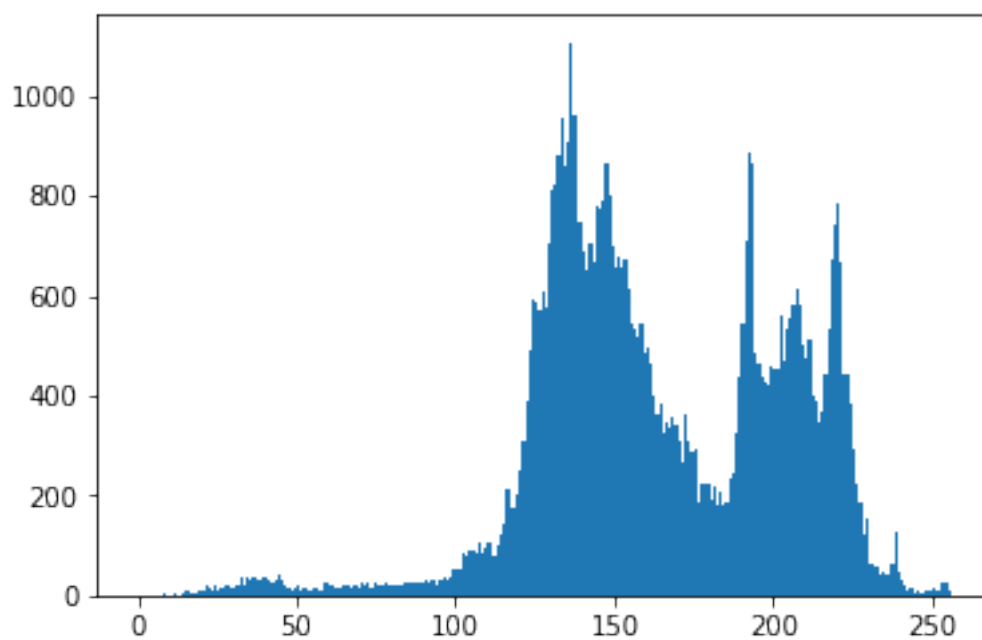
```

8.500e+01, 9.400e+01, 1.040e+02, 8.000e+01, 7.900e+01, 9.800e+01,
1.240e+02, 1.420e+02, 2.140e+02, 1.770e+02, 1.750e+02, 2.030e+02,
2.490e+02, 3.080e+02, 3.900e+02, 4.910e+02, 5.900e+02, 5.850e+02,
5.710e+02, 6.090e+02, 5.760e+02, 7.050e+02, 8.130e+02, 8.210e+02,
8.790e+02, 9.540e+02, 8.570e+02, 9.090e+02, 1.107e+03, 9.590e+02,
7.480e+02, 7.460e+02, 6.900e+02, 6.500e+02, 7.040e+02, 6.690e+02,
7.820e+02, 7.760e+02, 7.920e+02, 8.630e+02, 7.990e+02, 7.000e+02,
6.560e+02, 6.790e+02, 6.590e+02, 6.720e+02, 6.160e+02, 5.430e+02,
5.350e+02, 5.150e+02, 5.440e+02, 4.860e+02, 4.940e+02, 4.660e+02,
4.000e+02, 3.600e+02, 3.820e+02, 3.250e+02, 3.490e+02, 3.350e+02,
3.590e+02, 3.410e+02, 3.080e+02, 2.660e+02, 3.600e+02, 3.090e+02,
2.850e+02, 2.950e+02, 1.880e+02, 2.260e+02, 2.220e+02, 2.240e+02,
1.920e+02, 2.160e+02, 1.790e+02, 2.080e+02, 1.800e+02, 1.870e+02,
2.320e+02, 2.460e+02, 3.260e+02, 4.380e+02, 5.450e+02, 7.080e+02,
8.850e+02, 8.630e+02, 4.840e+02, 4.650e+02, 4.360e+02, 4.250e+02,
4.220e+02, 4.580e+02, 4.510e+02, 4.520e+02, 5.580e+02, 4.720e+02,
5.320e+02, 5.530e+02, 5.840e+02, 6.120e+02, 5.790e+02, 4.990e+02,
4.760e+02, 5.110e+02, 3.990e+02, 3.870e+02, 3.460e+02, 3.690e+02,
4.420e+02, 5.350e+02, 6.700e+02, 7.430e+02, 7.870e+02, 6.650e+02,
4.450e+02, 4.400e+02, 3.820e+02, 2.920e+02, 2.240e+02, 1.860e+02,
1.210e+02, 1.550e+02, 6.100e+01, 6.400e+01, 5.900e+01, 4.400e+01,
4.500e+01, 4.200e+01, 4.100e+01, 6.400e+01, 1.280e+02, 4.800e+01,
3.000e+01, 2.200e+01, 7.000e+00, 1.600e+01, 6.000e+00, 8.000e+00,
5.000e+00, 3.000e+00, 7.000e+00, 7.000e+00, 1.400e+01, 1.100e+01,
1.200e+01, 2.600e+01, 2.400e+01, 8.000e+00]),
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 [ ]:
```