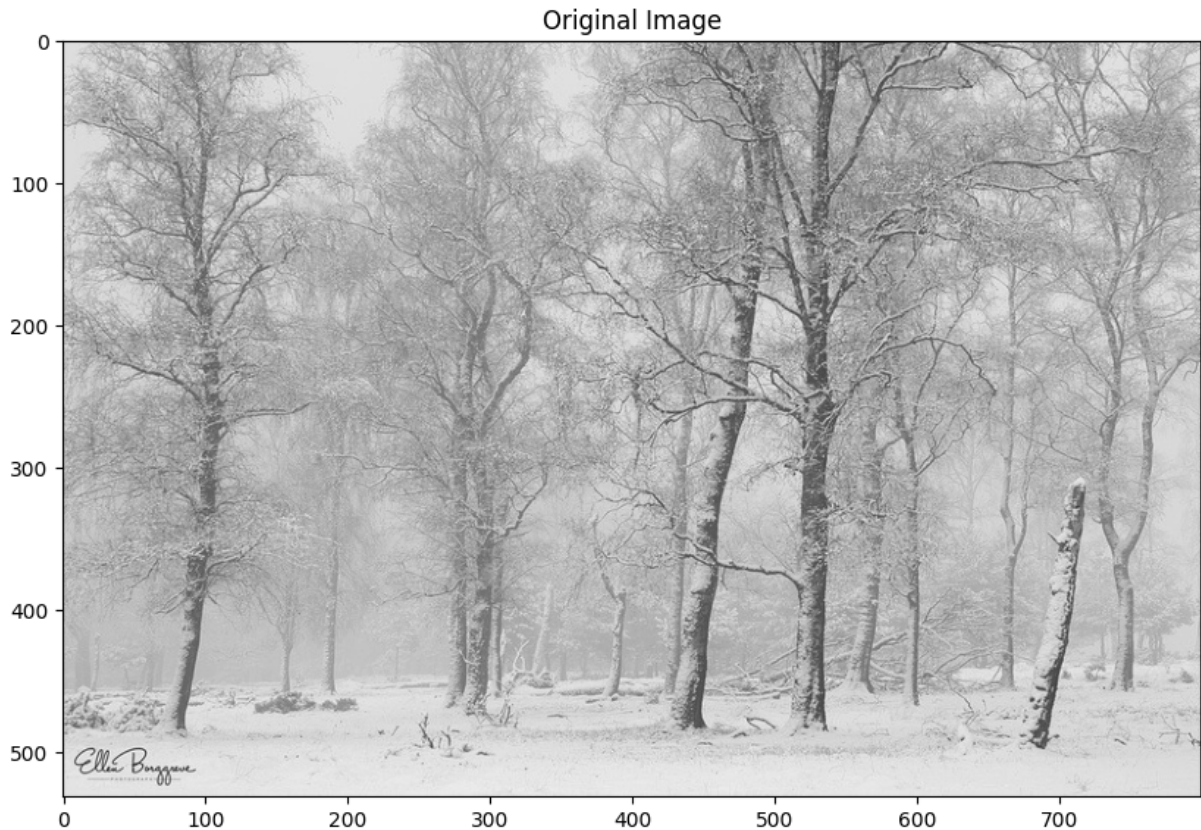


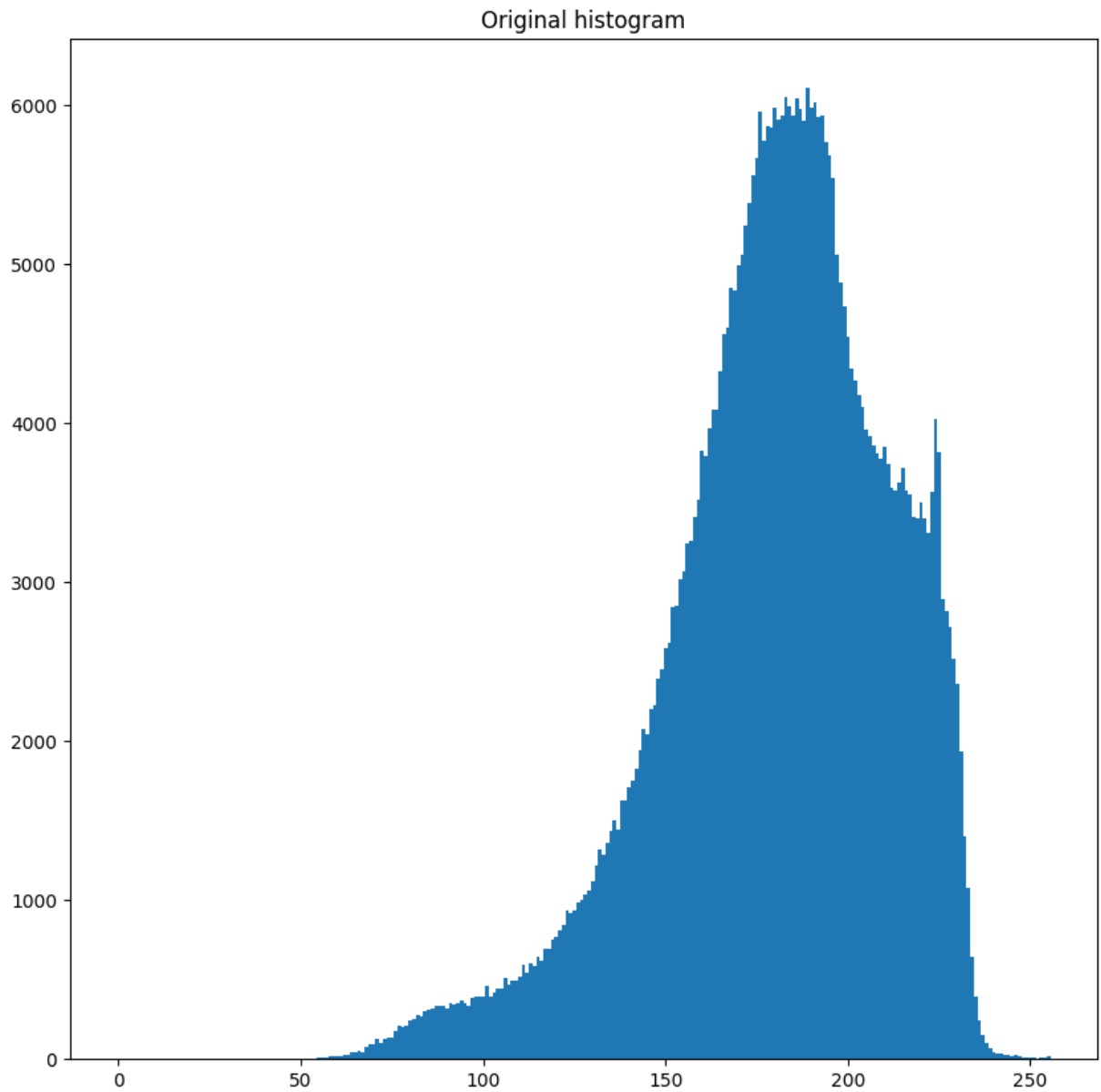
```
In [1]: import cv2
import matplotlib.pyplot as plt
import numpy as np
```

```
In [2]: figsize = (10, 10)
I = cv2.imread('forest.jpeg', 0)
plt.figure(figsize=figsize)
plt.imshow(I, cmap='gray', vmin=0, vmax=255)
plt.title('Original Image')
plt.show()
```

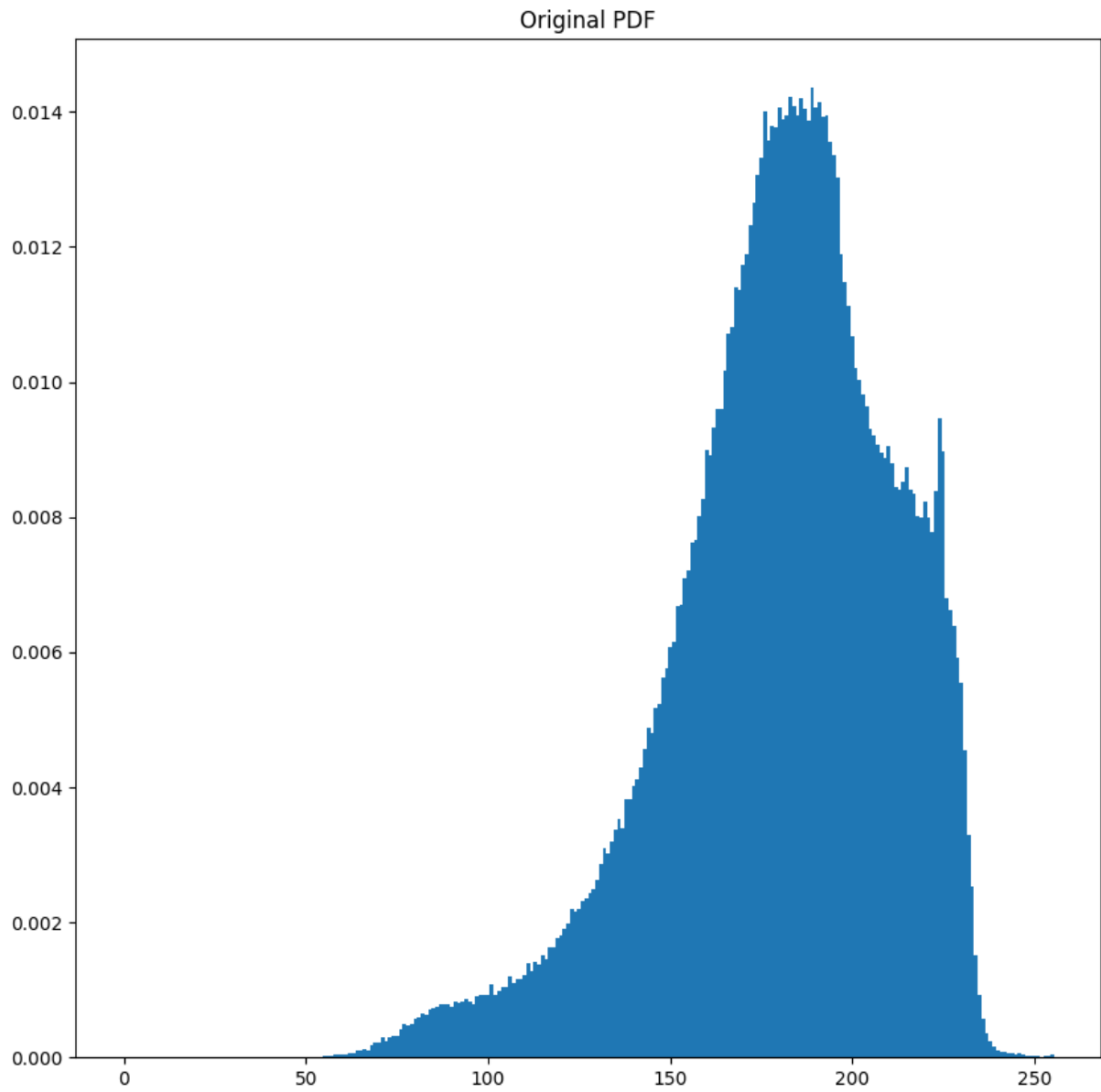


```
In [3]: bins_edges_min_max = [0, 256]
num_bins = 256
bin_count, bins_edges = np.histogram(I, num_bins, bins_edges_min_max)
bins_start = bins_edges[:-1]
```

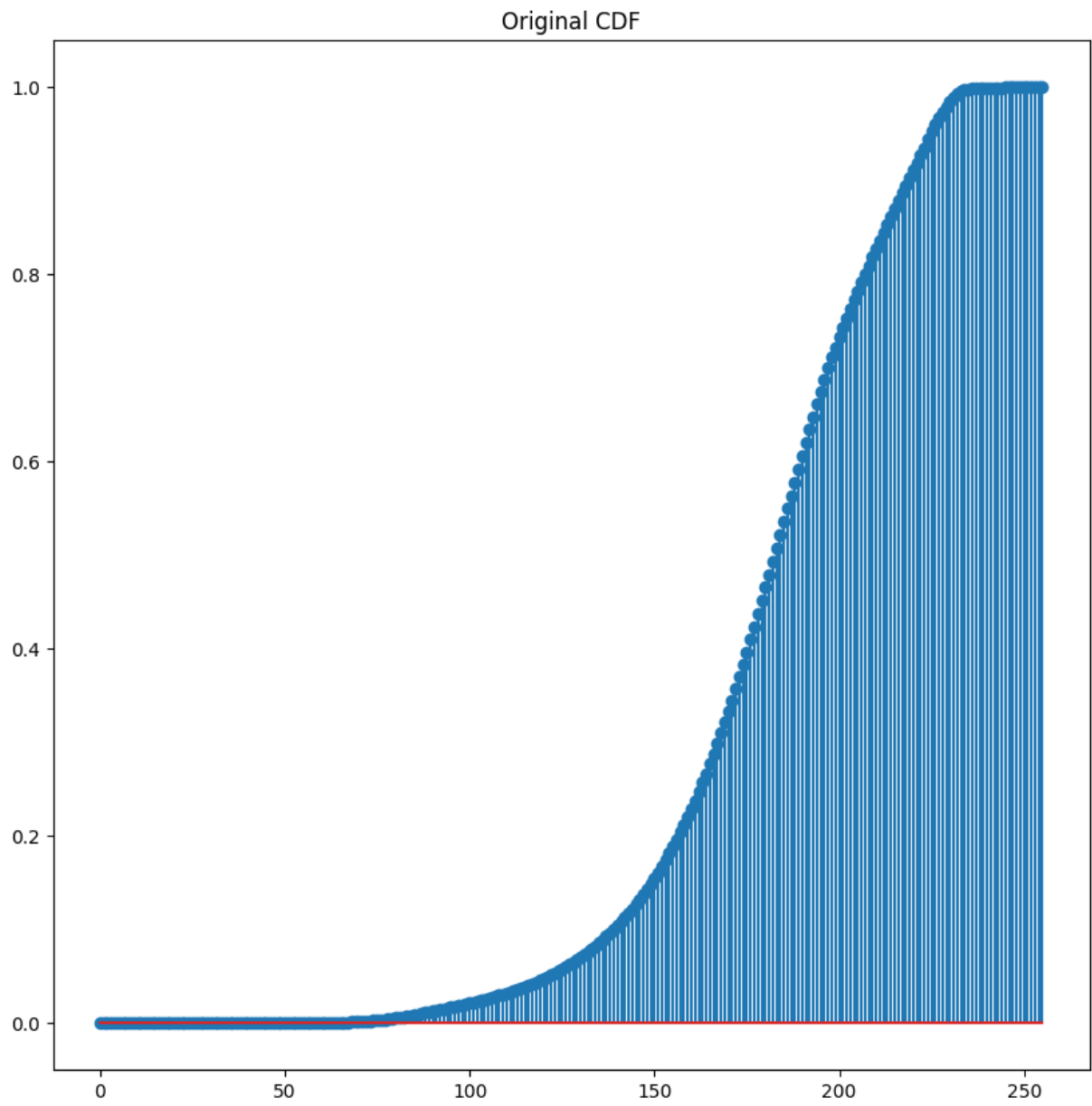
```
In [4]: def draw_hist(x_axis, input):
fig, ax = plt.subplots(figsize=figsize)
plt.bar(x_axis, input, width=input.shape[0] / (x_axis[-1] - x_axis[0] + 1))
return fig, ax
draw_hist(bins_start, bin_count)
plt.title("Original histogram")
plt.show()
```



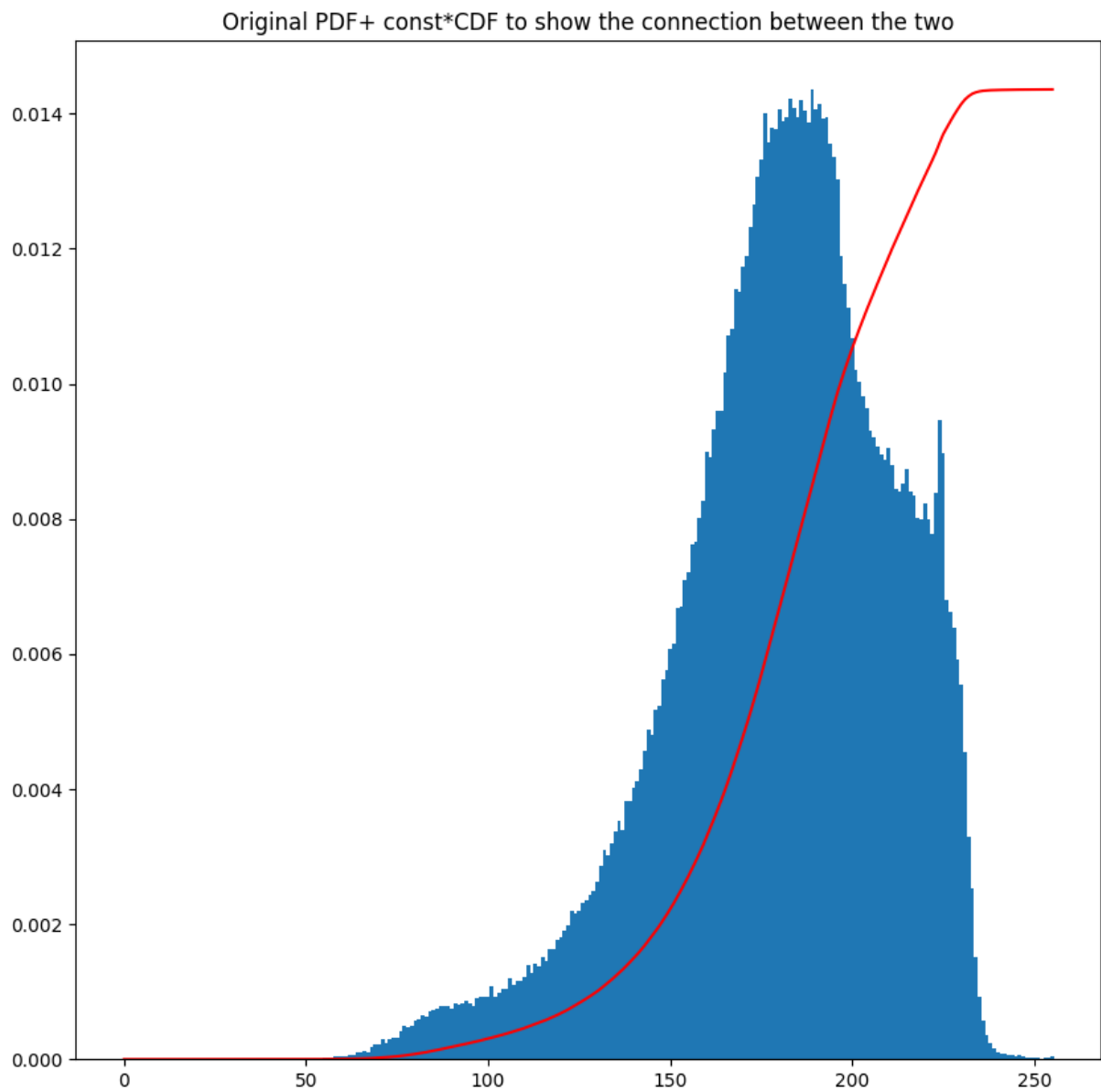
```
In [5]: pdf = bin_count / np.sum(bin_count)
draw_hist(bins_start, pdf)
plt.title("Original PDF")
plt.show()
```



```
In [6]: cdf = np.cumsum(pdf)
plt.figure(figsize=figsize)
plt.stem(cdf)
plt.title("Original CDF")
plt.show()
```



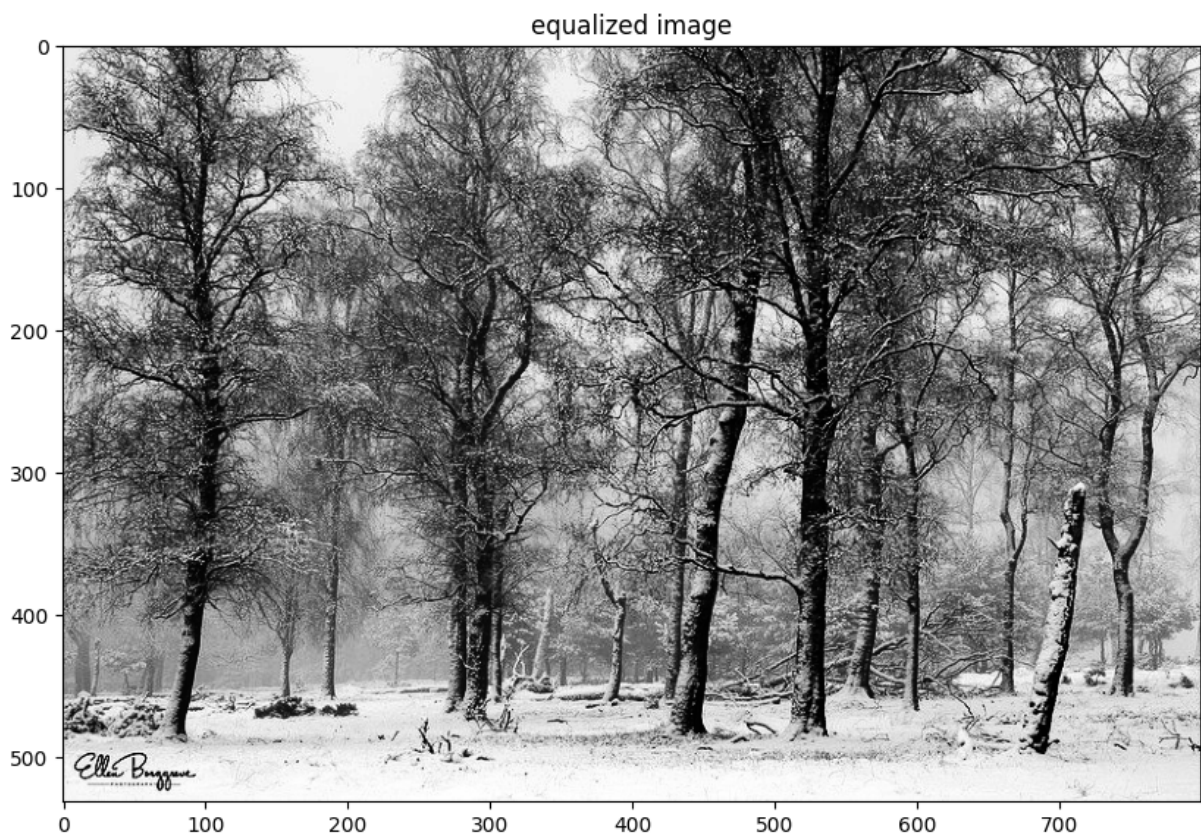
```
In [7]: fig, ax = draw_hist(bins_start, pdf)
ax.plot(cdf * np.max(pdf), "r")
plt.title("Original PDF+ const*CDF to show the connection between the two")
plt.show()
```



```
In [8]: f_eq = np.round(cdf * 255).astype(int)
f_eq
```

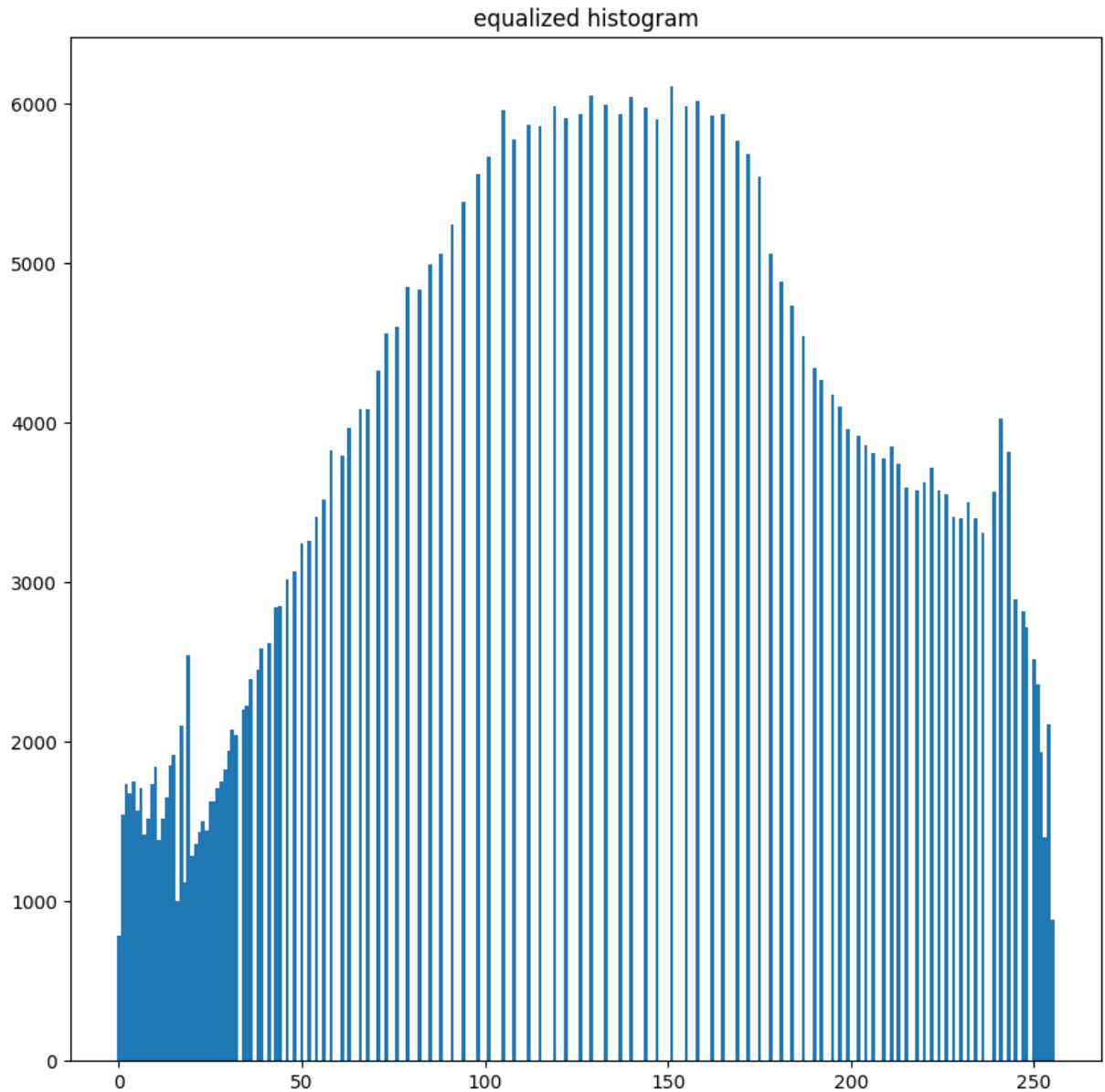
```
Out[8]: array([[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
  0,  0,  0,  0,  0,  0,  0,  1,  1,  1,  1,  1,  1,
  1,  1,  1,  2,  2,  2,  2,  2,  2,  3,  3,  3,  3,
  3,  4,  4,  4,  4,  4,  5,  5,  5,  5,  6,  6,  6,
  6,  7,  7,  7,  8,  8,  8,  9,  9,  9, 10, 10, 10,
11, 11, 12, 12, 13, 13, 14, 14, 15, 15, 16, 17, 17,
18, 19, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29,
30, 31, 32, 34, 35, 36, 38, 39, 41, 43, 44, 46, 48,
50, 52, 54, 56, 58, 61, 63, 66, 68, 71, 73, 76, 79,
82, 85, 88, 91, 94, 98, 101, 105, 108, 112, 115, 119, 122,
126, 129, 133, 137, 140, 144, 147, 151, 155, 158, 162, 165, 169,
172, 175, 178, 181, 184, 187, 190, 192, 195, 197, 199, 202, 204,
206, 209, 211, 213, 215, 218, 220, 222, 224, 226, 228, 230, 232,
234, 236, 239, 241, 243, 245, 247, 248, 250, 251, 252, 253, 254,
254, 254, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
255, 255, 255, 255, 255, 255, 255, 255])
```

```
In [9]: I_eq = f_eq[I]
plt.figure(figsize=figsize)
plt.imshow(I_eq, cmap="gray", vmin=0, vmax=255)
plt.title("equalized image")
plt.show()
```



```
In [10]: bin_count, bins_edges = np.histogram(I_eq, num_bins, bins_edges_min_max)
bins_start = bins_edges[:-1]
draw_hist(bins_start, bin_count)
```

```
plt.title("equalized histogram")  
plt.show()
```

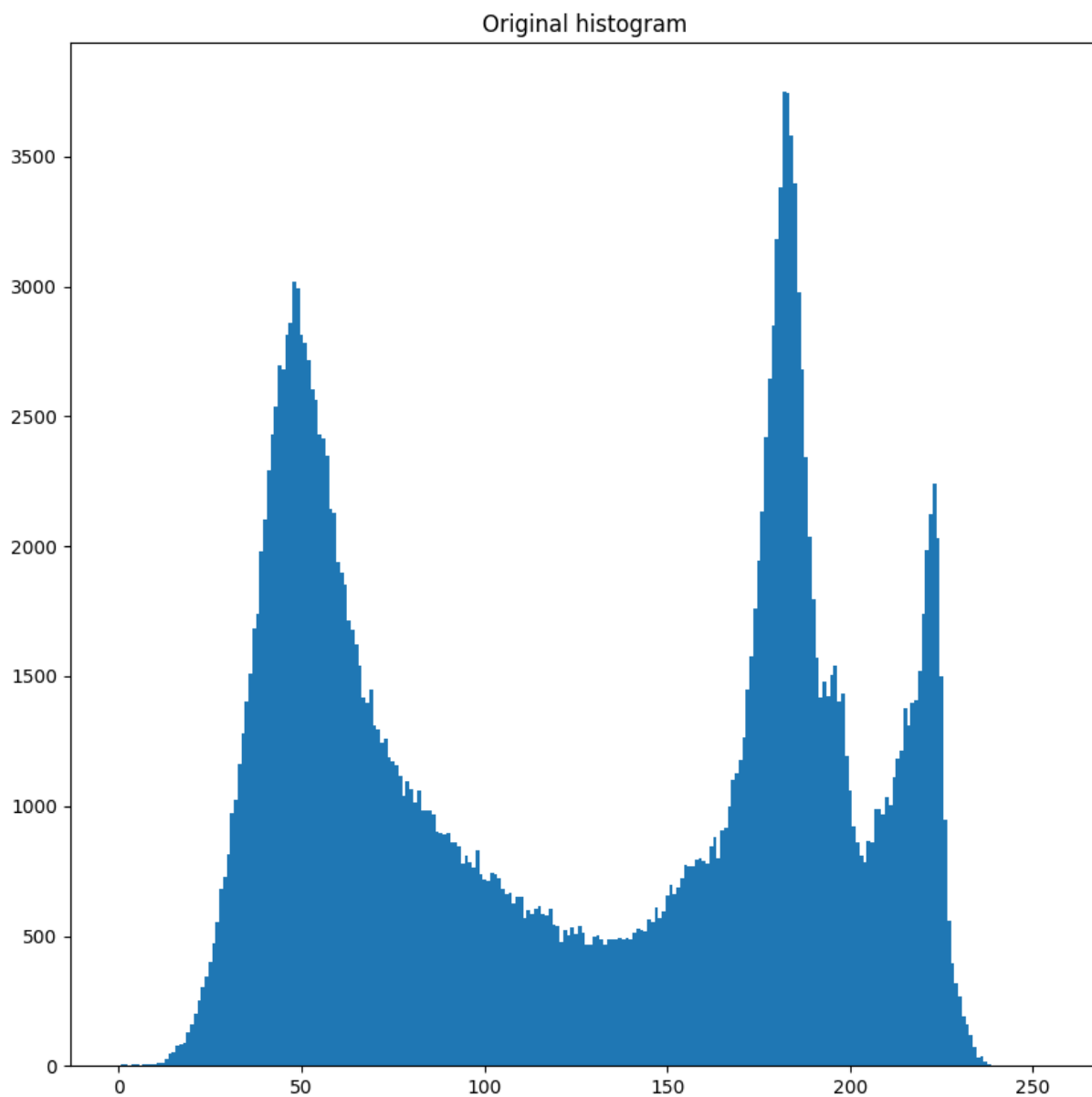


```
In [11]: figsize = (10, 10)  
I = cv2.imread('lake.tif', 0)  
plt.figure(figsize=figsize)  
plt.imshow(I, cmap='gray', vmin=0, vmax=255)  
plt.title('Original Image')  
plt.show()
```

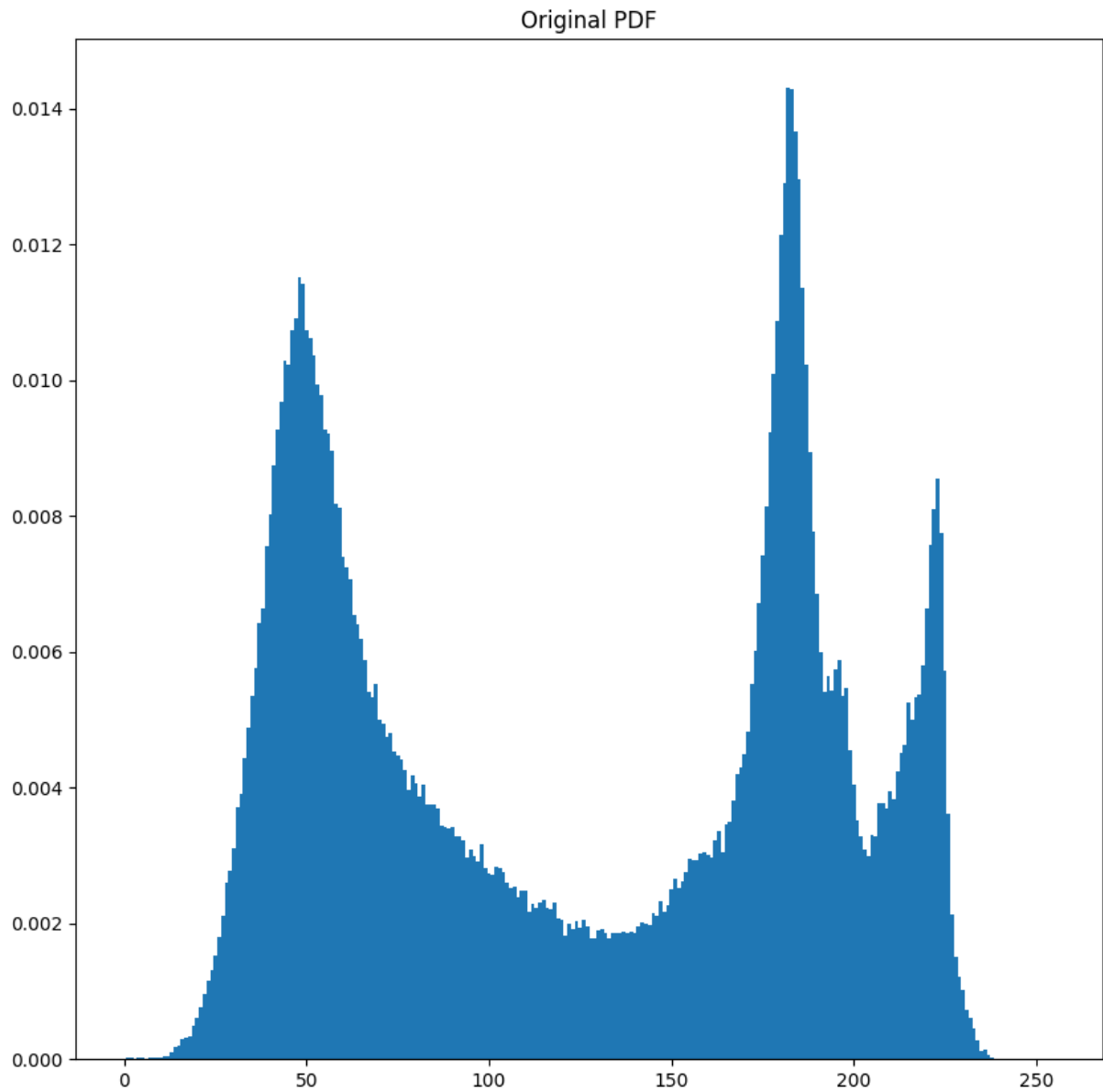



```
In [12]: bins_edges_min_max = [0, 256]
num_bins = 256
bin_count, bins_edges = np.histogram(I, num_bins, bins_edges_min_max)
bins_start = bins_edges[:-1]
```

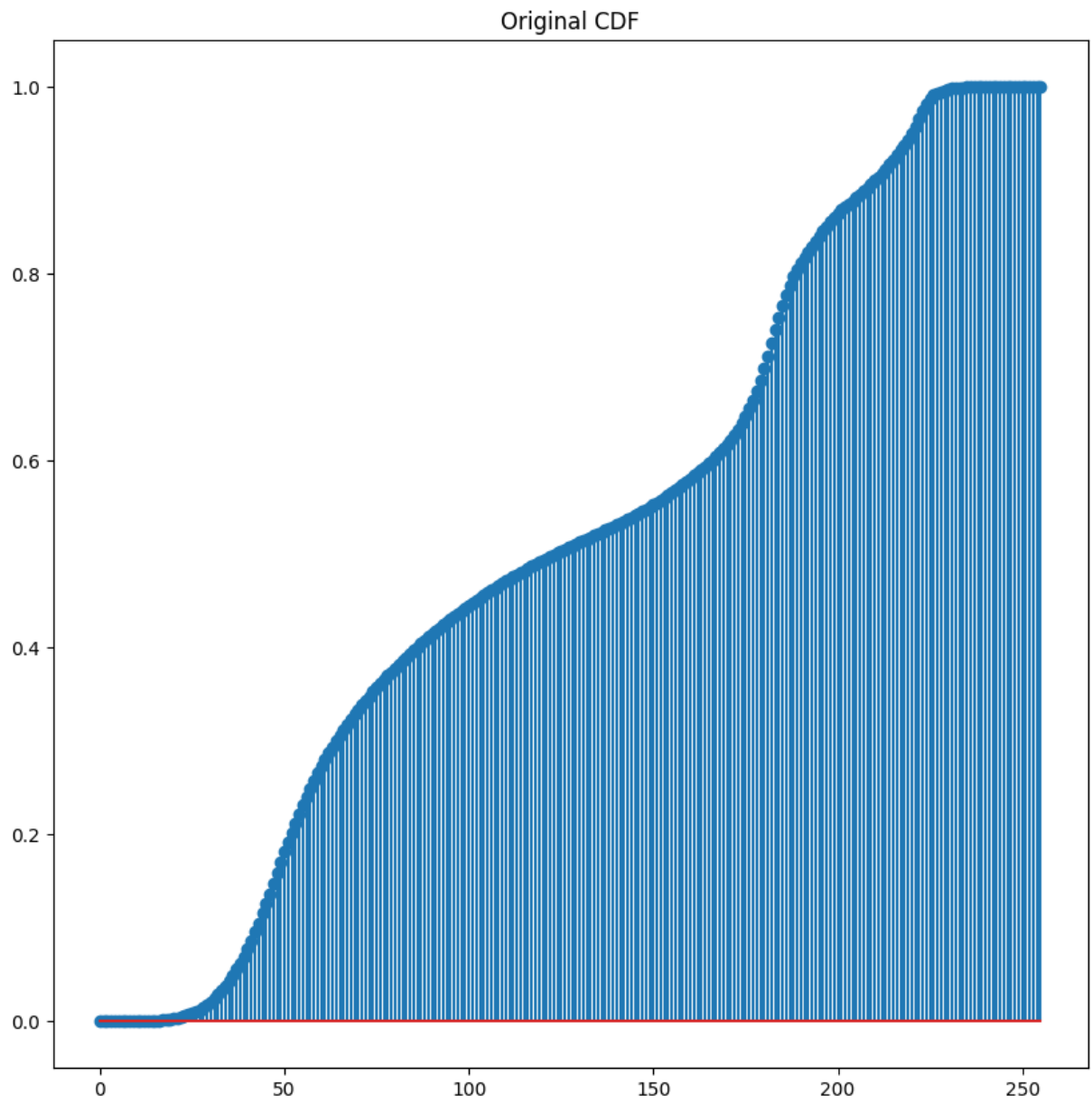
```
In [13]: def draw_hist(x_axis, input):
fig, ax = plt.subplots(figsize=figsize)
plt.bar(x_axis, input, width=input.shape[0] / (x_axis[-1] - x_axis[0] + 1))
return fig, ax
draw_hist(bins_start, bin_count)
plt.title("Original histogram")
plt.show()
```

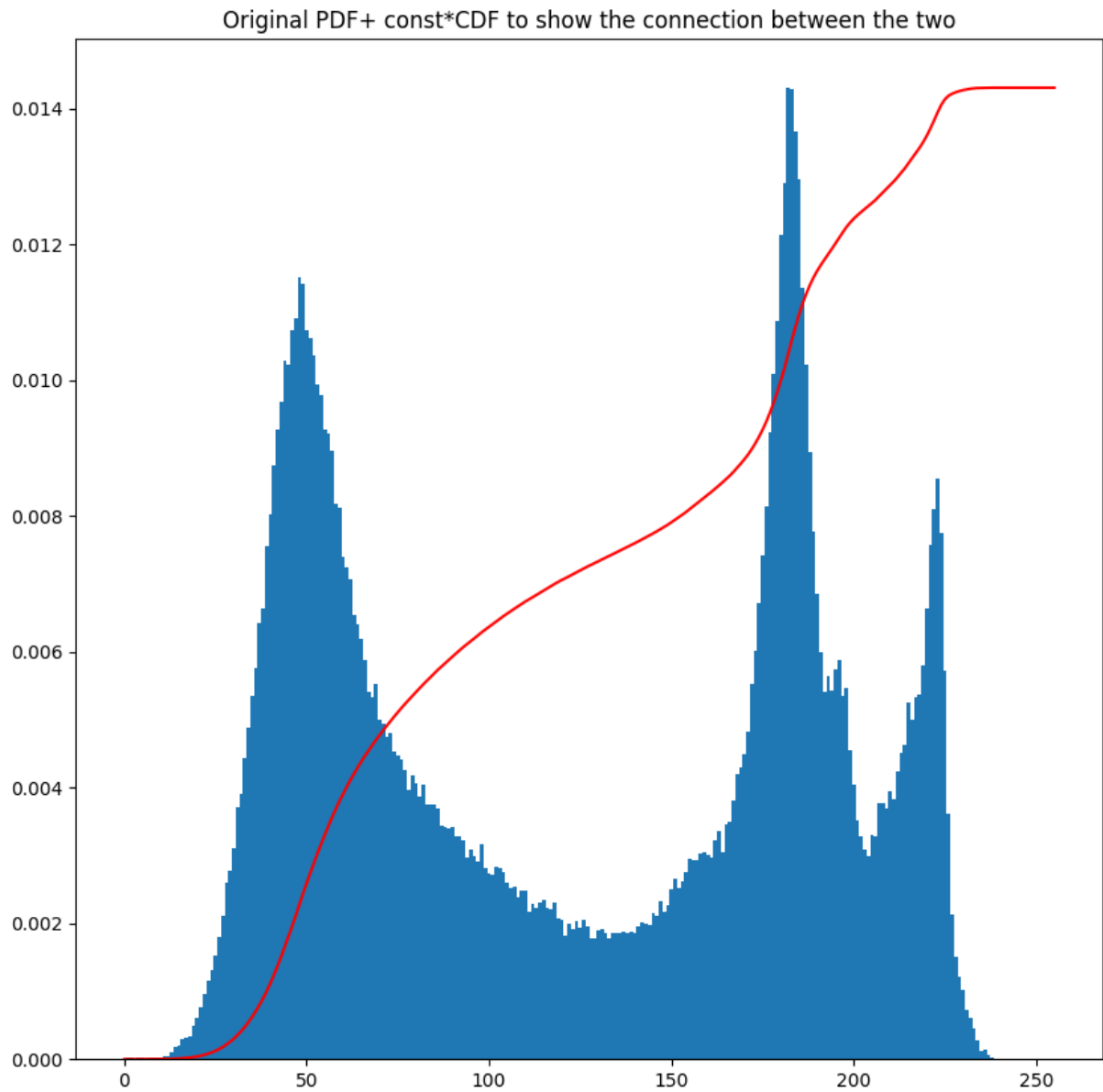
```
In [14]: pdf = bin_count / np.sum(bin_count)
draw_hist(bins_start, pdf)
plt.title("Original PDF")
plt.show()
```



```
In [15]: cdf = np.cumsum(pdf)
plt.figure(figsize=figsize)
plt.stem(cdf)
plt.title("Original CDF")
plt.show()
```



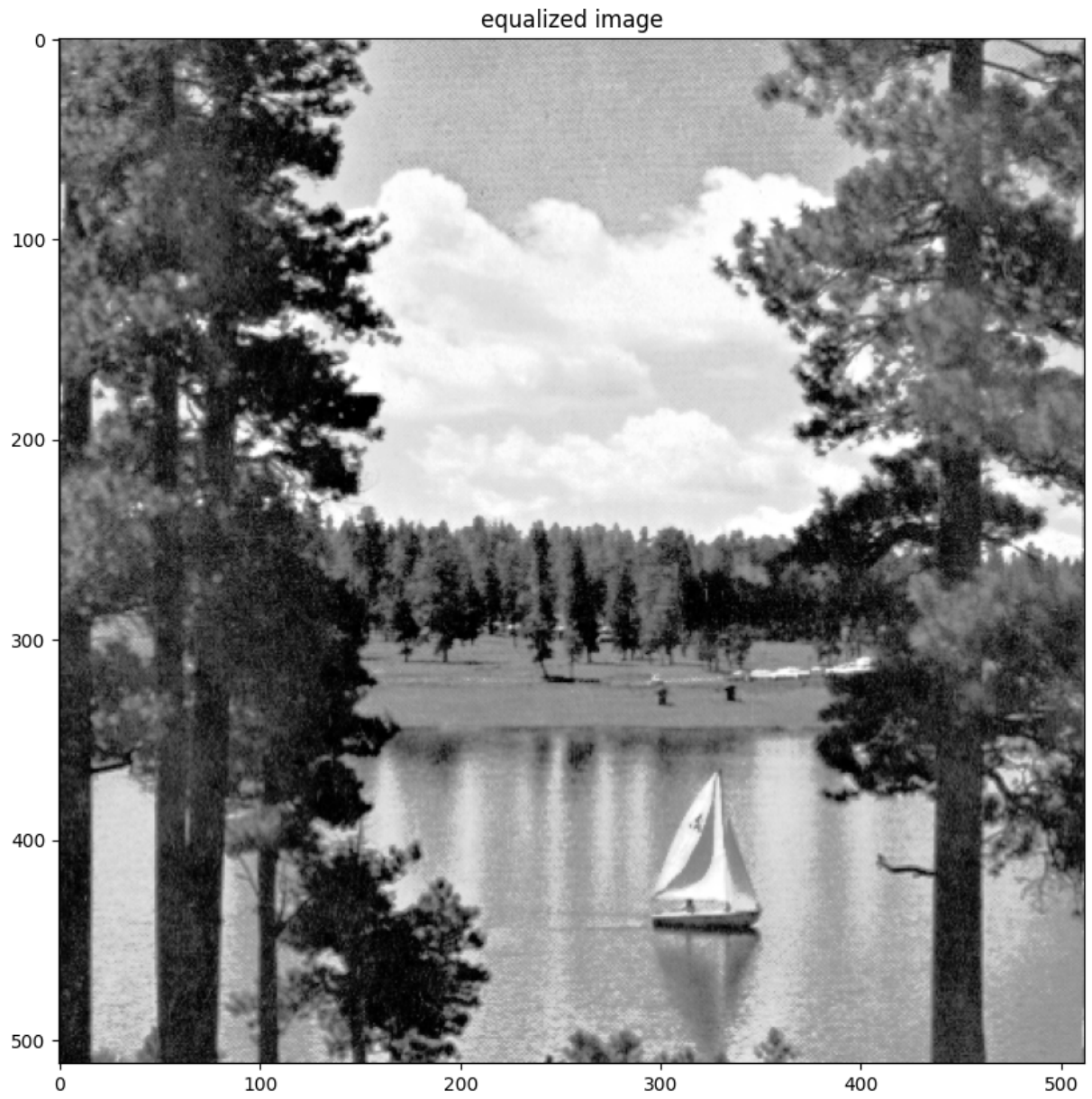
```
In [16]: fig, ax = draw_hist(bins_start, pdf)
ax.plot(cdf * np.max(pdf), "r")
plt.title("Original PDF+ const*CDF to show the connection between the two")
plt.show()
```



```
In [17]: f_eq = np.round(cdf * 255).astype(int)
         f_eq
```

```
Out[17]: array([[ 0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,  0,
                  0,  0,  0,  0,  0,  0,  1,  1,  1,  1,  1,  2,  2,
                  3,  3,  4,  5,  5,  6,  7,  8, 10, 11, 13, 14, 16,
                  18, 20, 22, 24, 27, 29, 32, 35, 38, 41, 43, 46, 49,
                  52, 54, 57, 59, 61, 64, 66, 68, 70, 71, 73, 75, 77,
                  78, 80, 81, 82, 84, 85, 86, 88, 89, 90, 91, 92, 93,
                  94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106,
                  107, 107, 108, 109, 110, 111, 111, 112, 113, 113, 114, 115, 116,
                  116, 117, 118, 118, 119, 120, 120, 121, 121, 122, 122, 123, 124,
                  124, 125, 125, 126, 126, 127, 127, 128, 128, 129, 129, 130, 130,
                  131, 131, 132, 132, 133, 133, 134, 134, 134, 135, 135, 136, 136,
                  137, 137, 138, 139, 139, 140, 140, 141, 142, 142, 143, 144, 144,
                  145, 146, 147, 147, 148, 149, 150, 151, 151, 152, 153, 154, 155,
                  156, 157, 159, 160, 162, 163, 165, 167, 170, 172, 175, 178, 181,
                  185, 189, 192, 195, 198, 201, 203, 205, 207, 208, 210, 211, 213,
                  214, 216, 217, 218, 220, 221, 222, 222, 223, 224, 225, 226, 227,
                  227, 228, 229, 230, 232, 233, 234, 235, 236, 238, 239, 241, 242,
                  244, 246, 249, 250, 252, 253, 253, 254, 254, 254, 255, 255, 255,
                  255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
                  255, 255, 255, 255, 255, 255, 255, 255, 255])
```

```
In [18]: I_eq = f_eq[I]
plt.figure(figsize=figsize)
plt.imshow(I_eq, cmap="gray", vmin=0, vmax=255)
plt.title("equalized image")
plt.show()
```



```
In [19]: bin_count, bins_edges = np.histogram(I_eq, num_bins, bins_edges_min_max)
        bins_start = bins_edges[:-1]
        draw_hist(bins_start, bin_count)
        plt.title("equalized histogram")
        plt.show()
```

