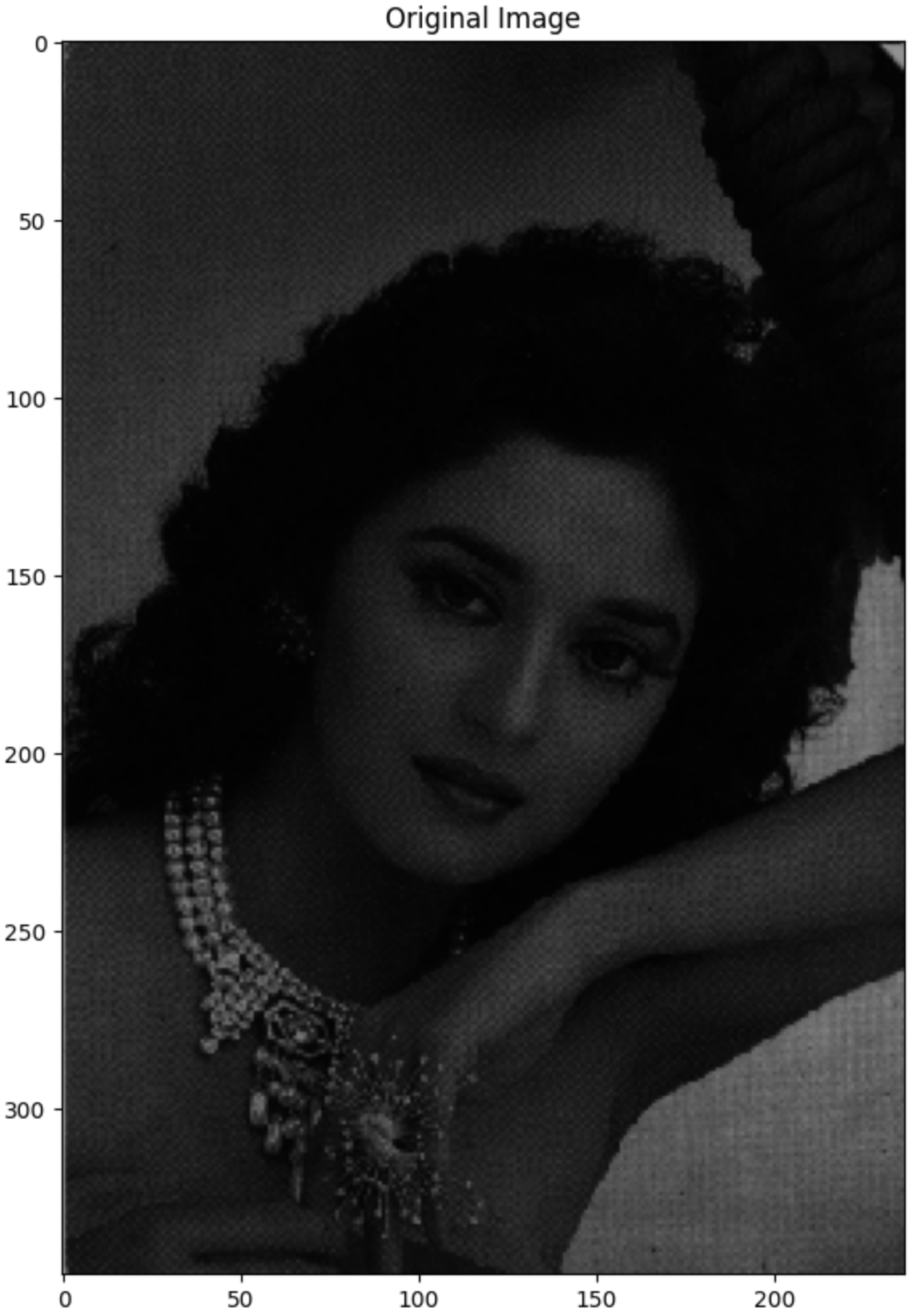
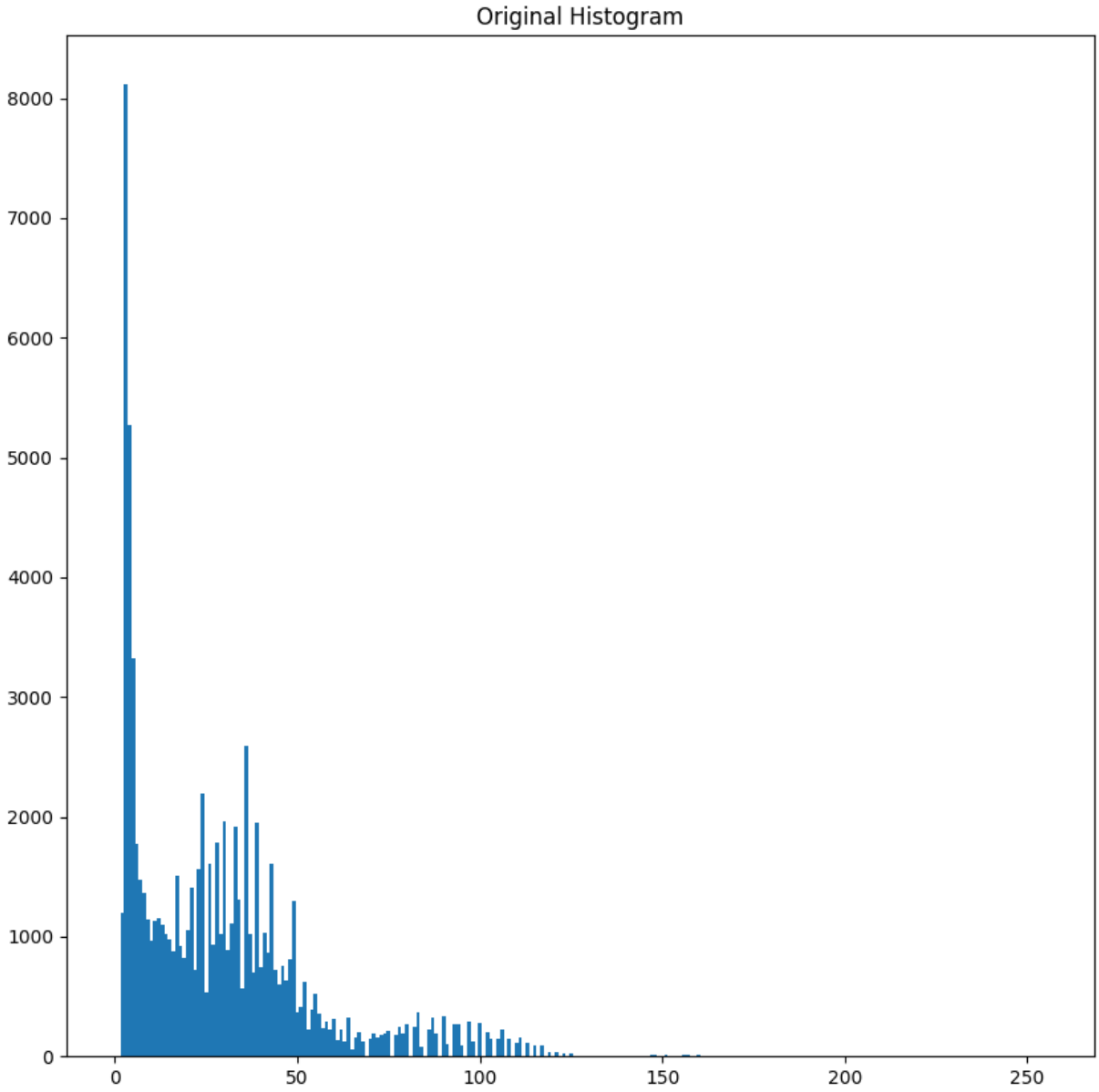
In [2]:
```
import cv2
import matplotlib.pyplot as plt
import numpy as np
```

In [3]:
```
figsize = (10, 10)
I = cv2.imread('MAD.TIF', 0)
plt.figure(figsize=figsize)
plt.imshow(I, cmap='gray', vmin=0, vmax=255)
plt.title('Original Image')
plt.show()
```

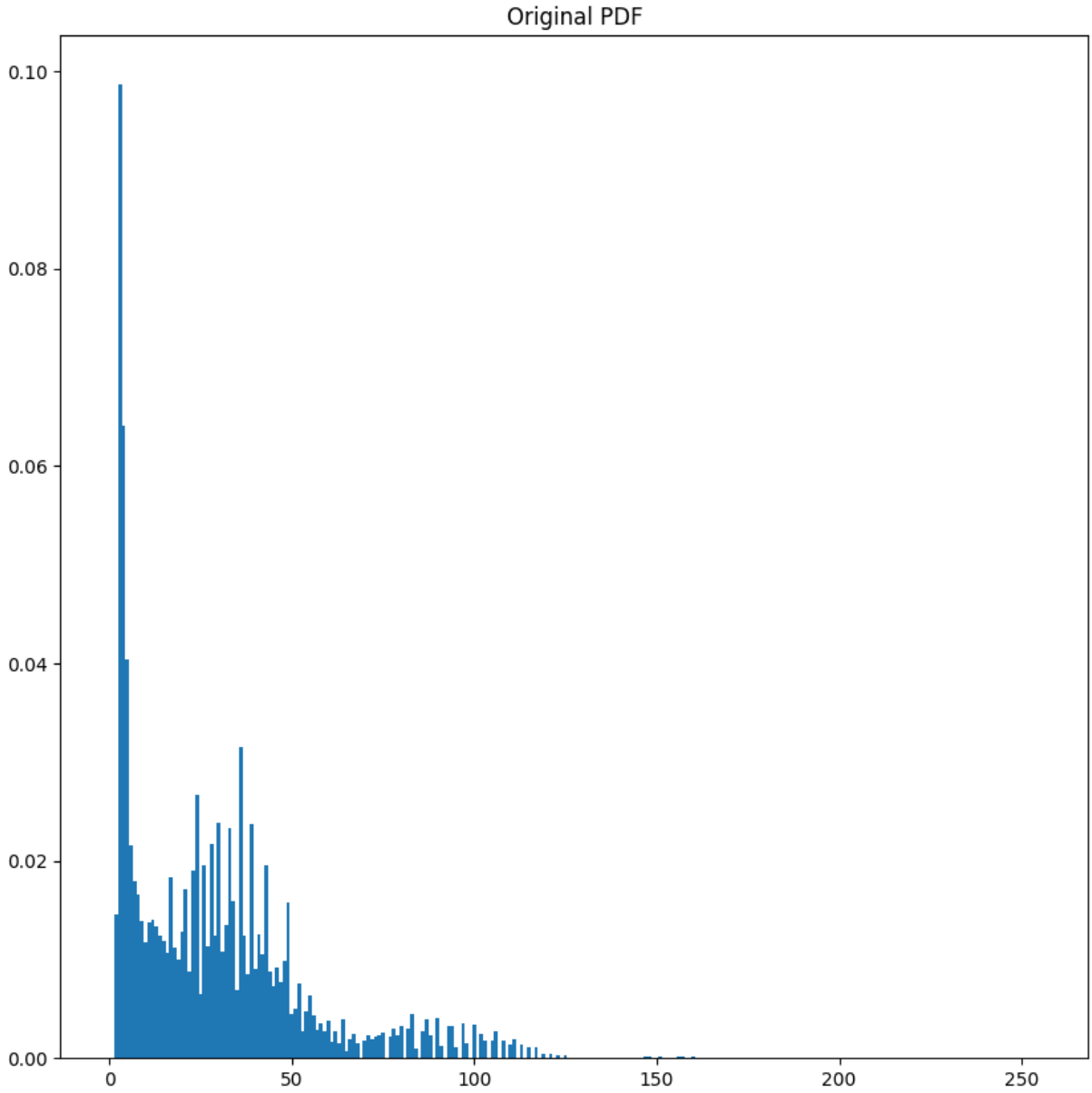


In [4]:
```
bins_edges_min_max = [0, 256]
num_bins = 256
bin_count, bins_edges = np.histogram(I, num_bins, bins_edges_min_max)
bins_start = bins_edges[:-1]
```

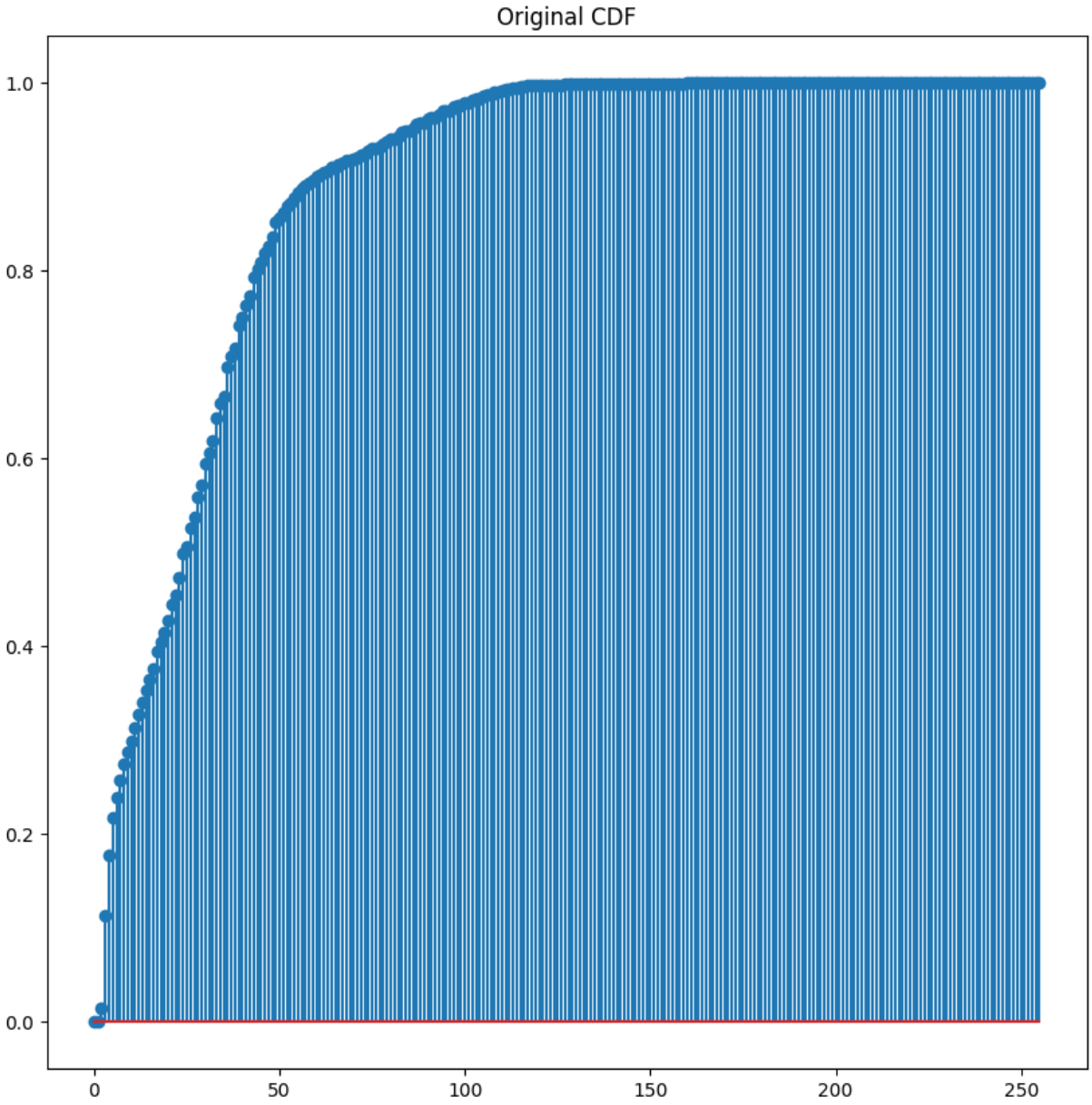
In [5]:
```
def draw_hist(x_axis, input):
    fig, ax = plt.subplots(figsize=figsize)
    plt.bar(x_axis, input, width=input.shape[0] / (x_axis[-1] - x_axis[0] + 1))
    return fig, ax

draw_hist(bins_start, bin_count)
plt.title('Original Histogram')
plt.show()
```



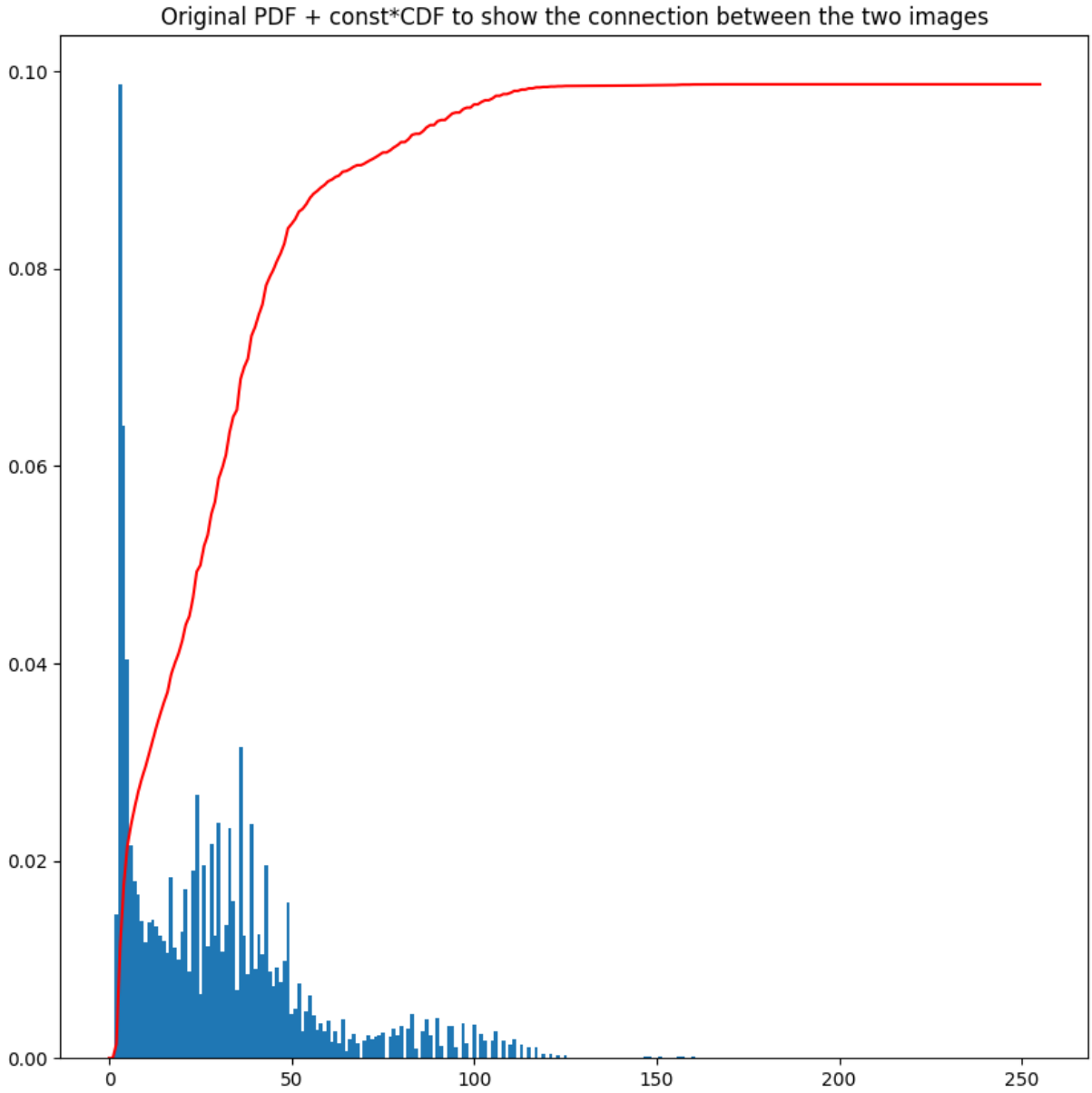
In [6]:
```
pdf = bin_count / np.sum(bin_count)
draw_hist(bins_start, pdf)
plt.title('Original PDF')
plt.show()
```



In [7]:
```
cdf = np.cumsum(pdf)
plt.figure(figsize=figsize)
plt.stem(cdf)
plt.title('Original CDF')
plt.show()
```



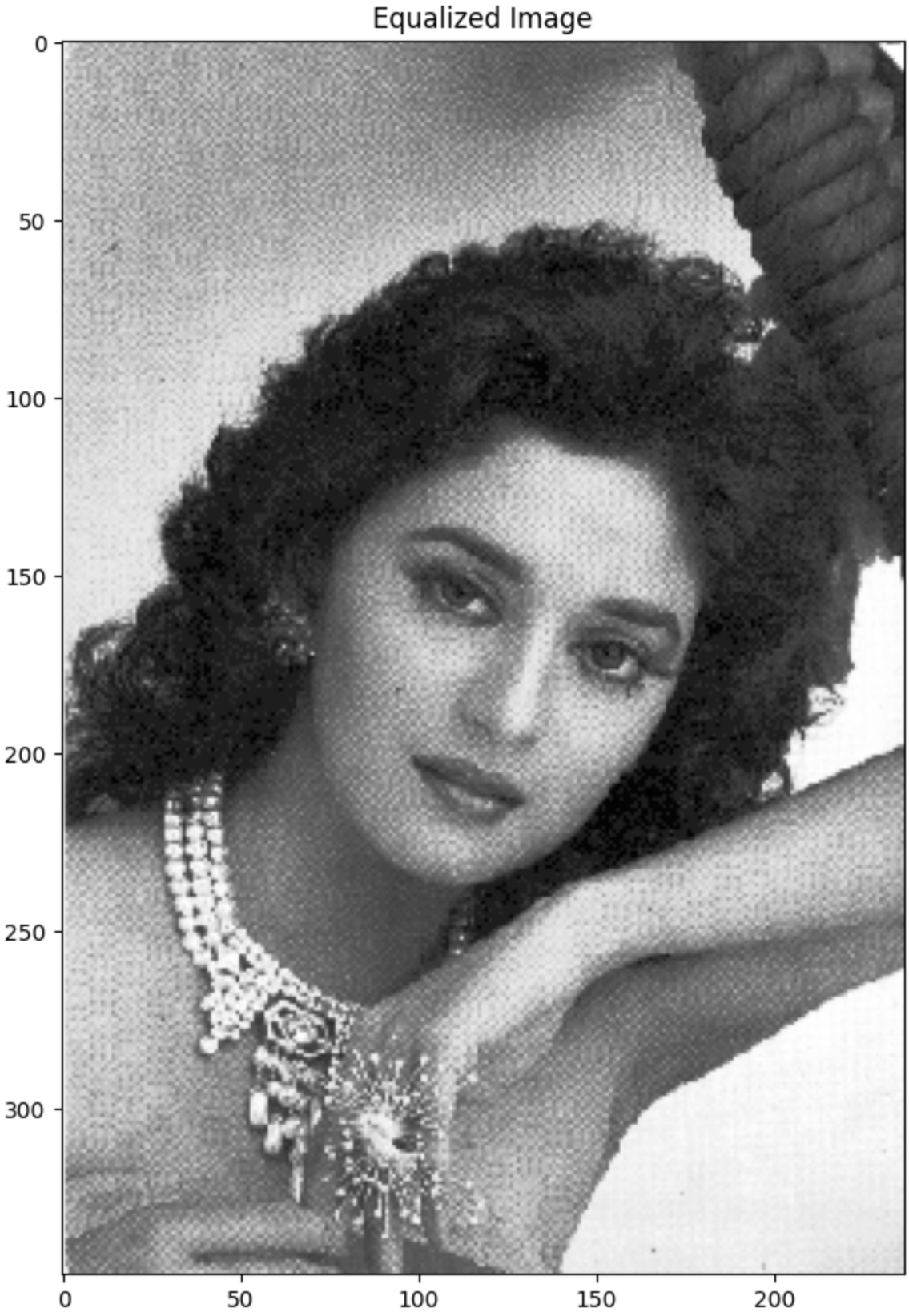
In [8]:
```
fig, ax = draw_hist(bins_start, pdf)
ax.plot(cdf * np.max(pdf), "r")
plt.title('Original PDF + const*CDF to show the connection between the two images')
plt.show()
```



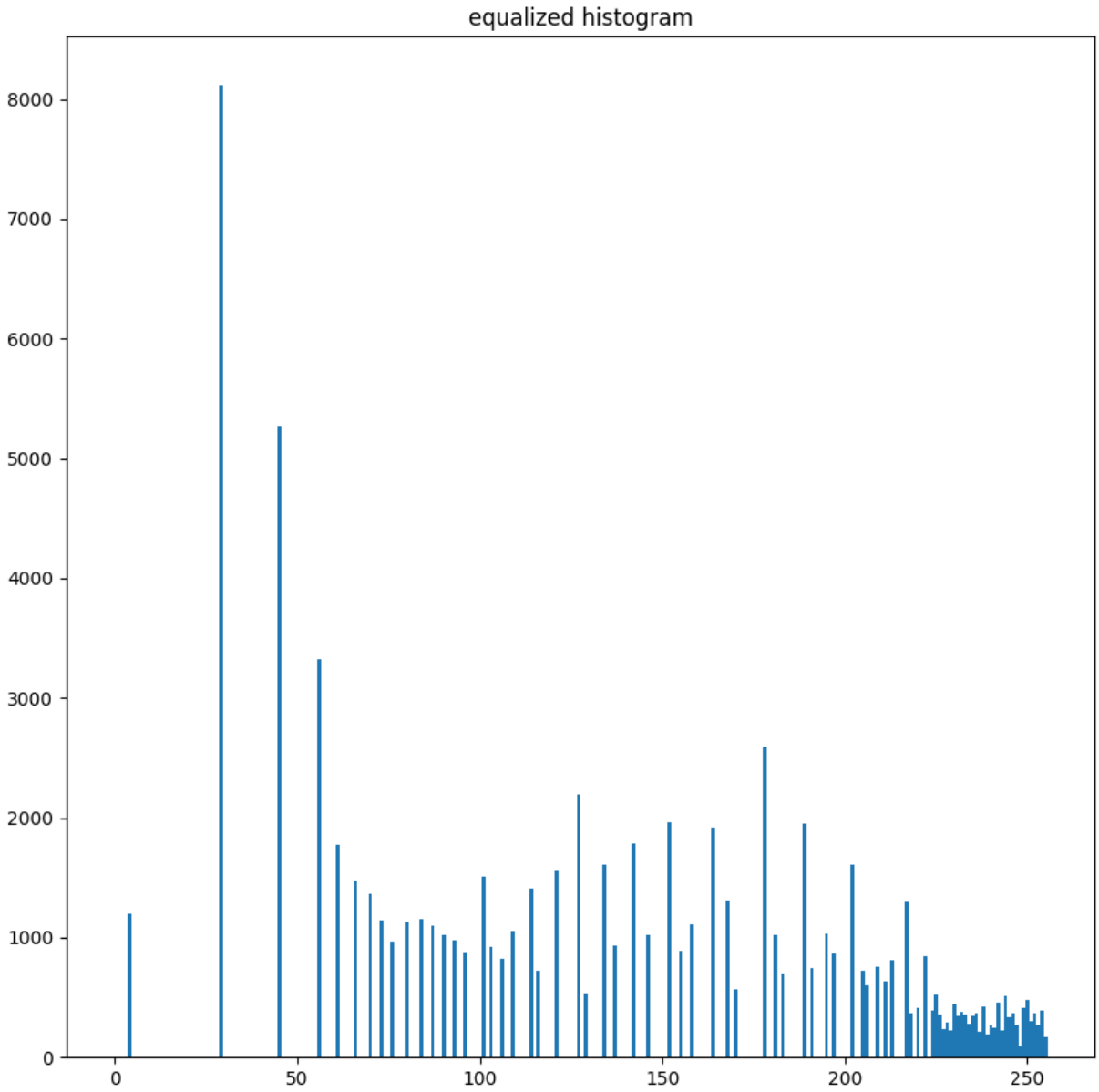
In [9]:
```
f_eq = np.round(cdf * 255).astype(int)
f_eq
```

Out[9]:
```
array([  0,   0,   4,  29,  45,  56,  61,  66,  70,  73,  76,  80,  84,
        87,  90,  93,  96, 101, 103, 106, 109, 114, 116, 121, 127, 129,
       134, 137, 142, 146, 152, 155, 158, 164, 168, 170, 178, 181, 183,
       189, 191, 195, 197, 202, 205, 206, 209, 211, 213, 217, 218, 220,
       222, 222, 224, 225, 226, 227, 228, 229, 230, 230, 231, 231, 232,
       232, 233, 233, 234, 234, 234, 235, 235, 236, 236, 237, 237, 238,
       238, 239, 240, 240, 241, 242, 242, 242, 243, 244, 244, 244, 245,
       246, 246, 246, 247, 248, 248, 249, 249, 249, 250, 250, 250, 251,
       251, 251, 252, 252, 252, 252, 253, 253, 253, 254, 254, 254, 254,
       254, 254, 254, 254, 254, 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, 255, 255, 255, 255, 255, 255, 255,
       255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
       255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
       255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
       255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
       255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
       255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
       255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255, 255,
       255, 255, 255, 255, 255, 255, 255, 255, 255])
```

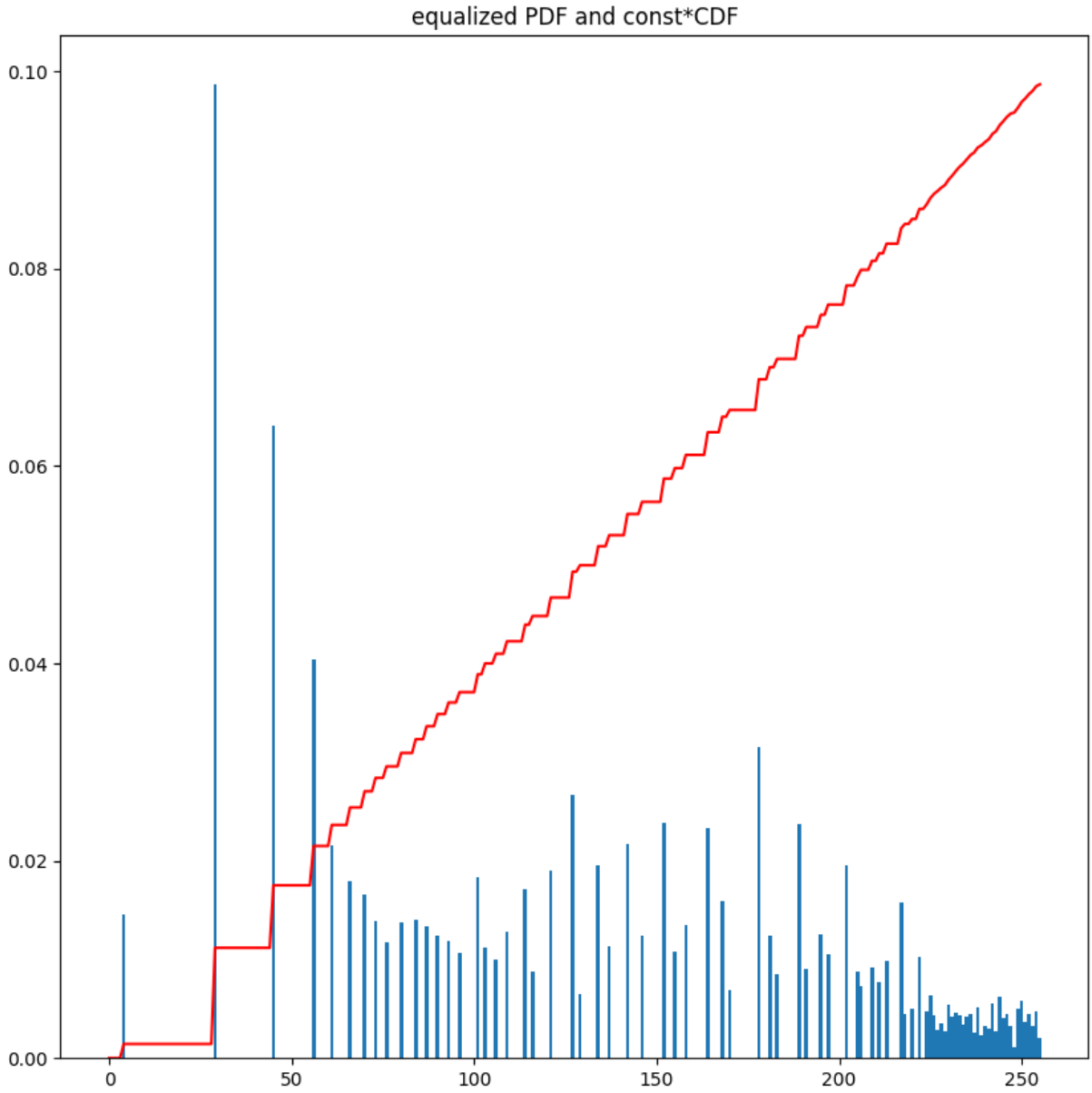
In [10]:
```
I_eq = f_eq[I]
plt.figure(figsize=figsize)
plt.imshow(I_eq, cmap='gray', vmin=0, vmax=255)
plt.title('Equalized Image')
plt.show()
```



In [11]:
```
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 [12]:
```
pdf = bin_count / np.sum(bin_count)
cdf = np.cumsum(pdf)
fig, ax = draw_hist(bins_start, pdf)
ax.plot(cdf * np.max(pdf), "r")
plt.title("equalized PDF and const*CDF")
plt.show()
```


equalized PDF and const*CDF
0.10
0.08
0.06
0.04
0.02
0.00
0
50
100
150
200
250

In [ ]: