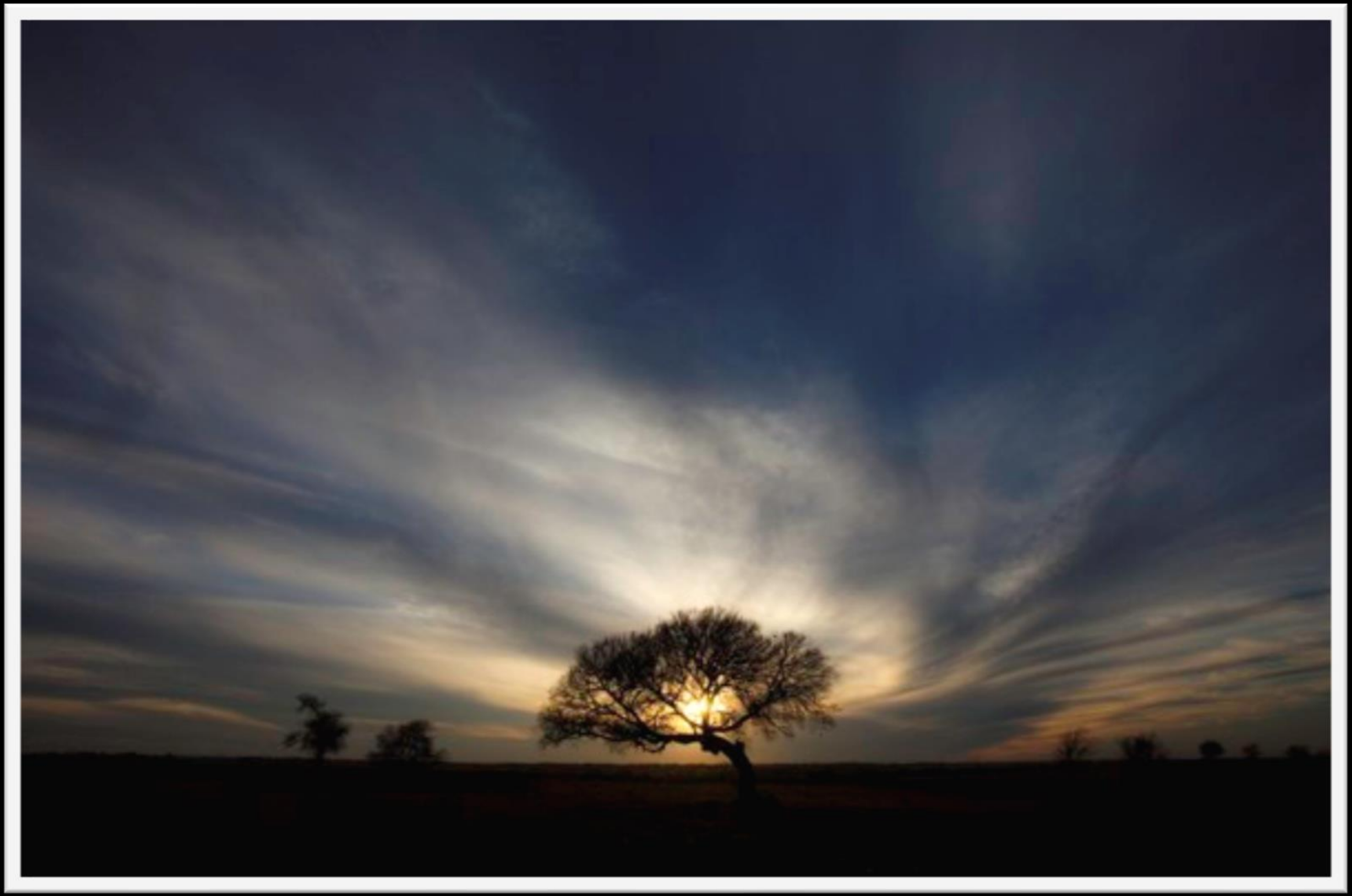
ENHANCEMENT BY HISTOGRAM MANIPULATION

ACTIVITY 5

Krishna Lyn Delima

2014-64503

Original Image



From: https://digital-photography-school.com/7-ways-to-handle-high-contrast-scenes/

Done using:

- Jupyter Notebook (Python)
  - Packages:

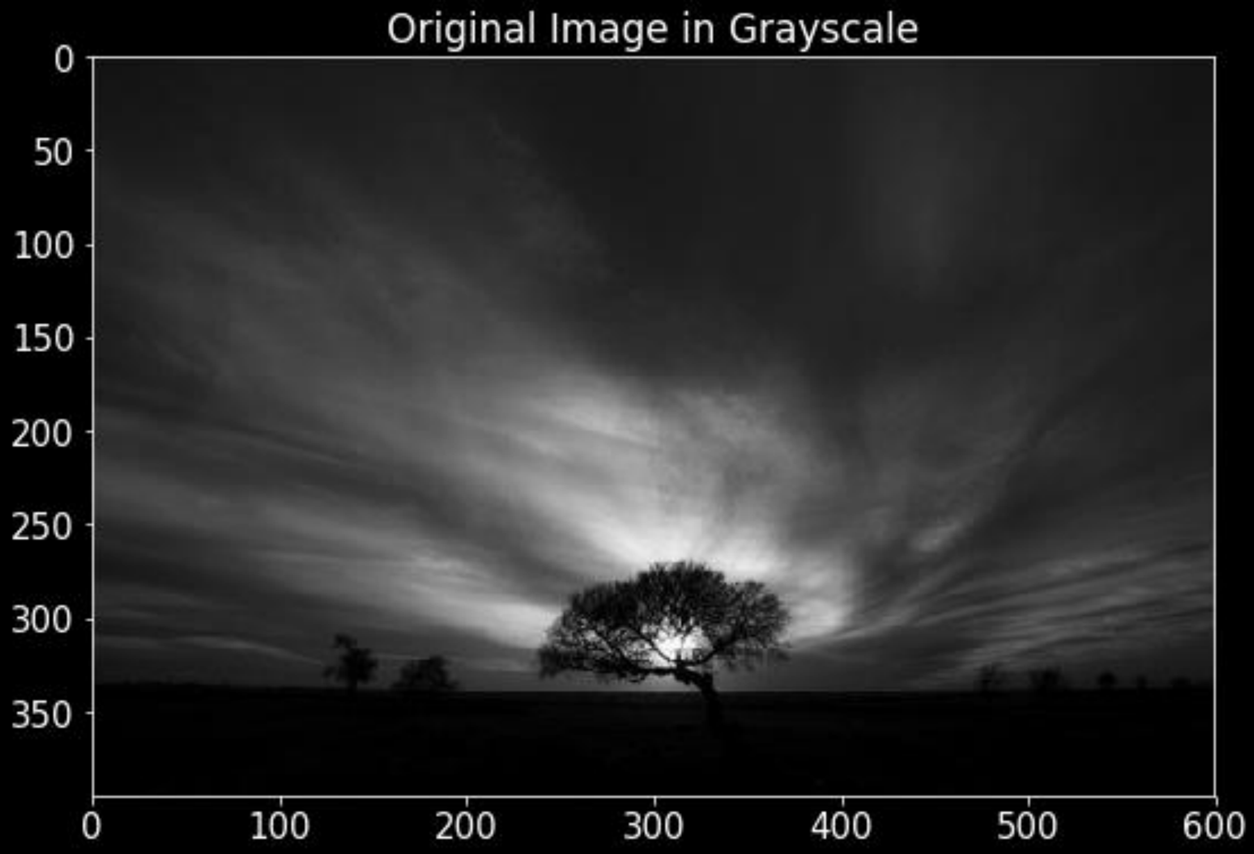
      matplotlib.pyplot

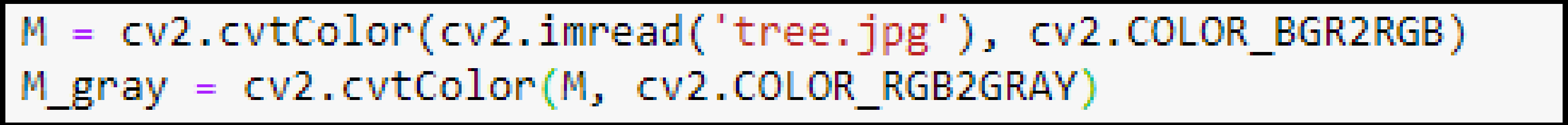
      numpy

      cv2

      skimage
- Photos taken from Google Images

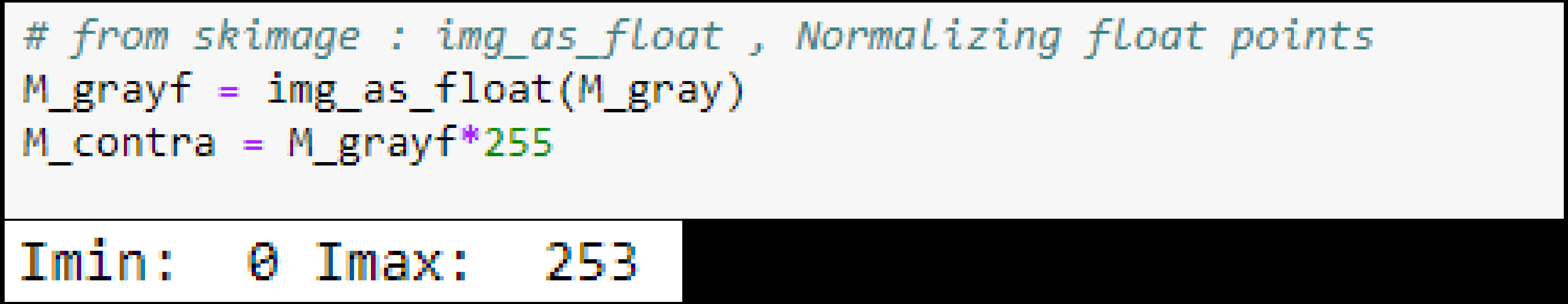
Step 1 Convert to Grayscale





```
M = cv2.cvtColor(cv2.imread('tree.jpg'), cv2.COLOR_BGR2RGB)
M_gray = cv2.cvtColor(M, cv2.COLOR_RGB2GRAY)
```

Step 2 Contrast Stretching

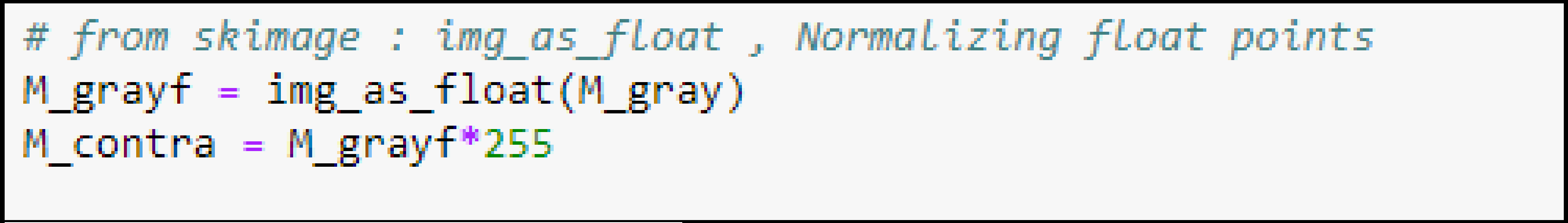


```
# from skimage : img_as_float , Normalizing float points
M_grayf = img_as_float(M_gray)
M_contra = M_grayf*255
```

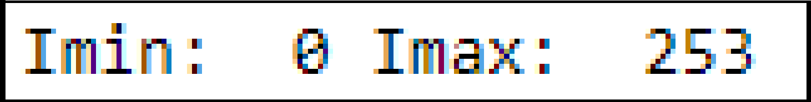
```
Imin:   0 Imax:   253
```

- Using img_as_float from skimage, the image’s pixel values were **NORMALIZED** and converted to **FLOAT**.
- 255 was multiplied since the image was 8-bit

Step 2 Contrast Stretching



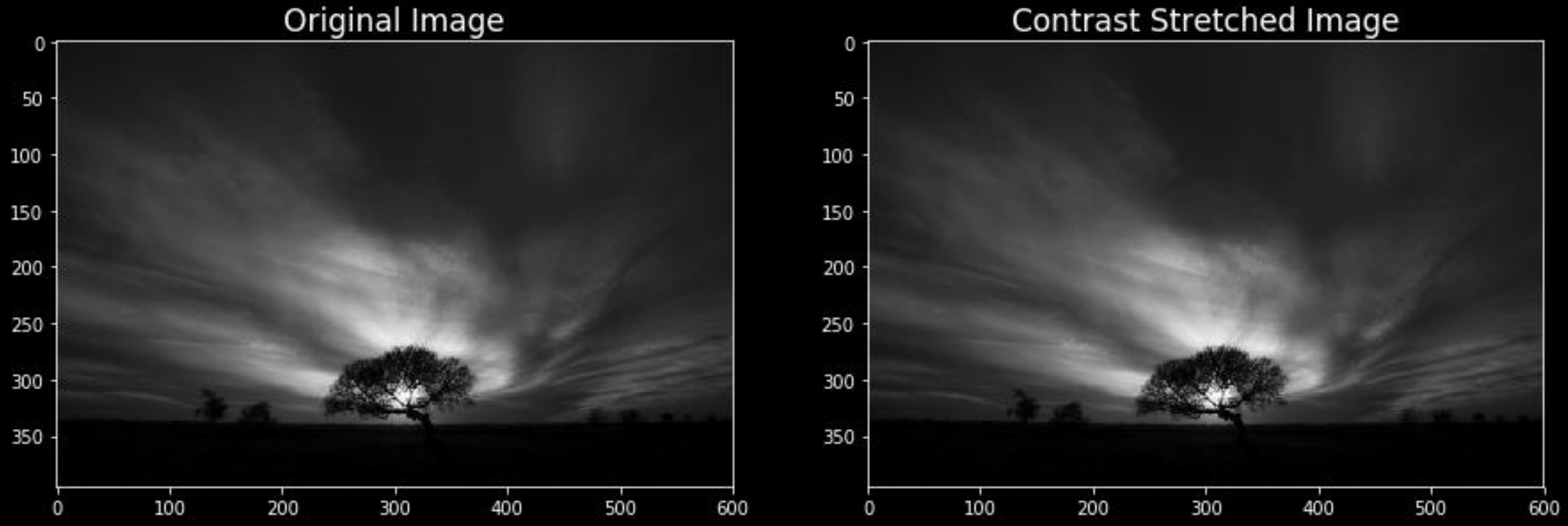
```
# from skimage : img_as_float , Normalizing float points
M_grayf = img_as_float(M_gray)
M_contra = M_grayf*255
```



```
Imin:   0 Imax:   253
```

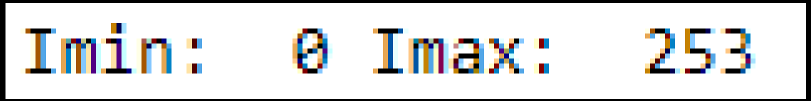
- Contrast stretching is used for low contrast photos
  - original maximum and minimum values were set to 0 and 1 after normalizing
  - results to a higher contrast image (darker on dark areas and brighter on bright areas)

Step 2 Contrast Stretching


Original Image

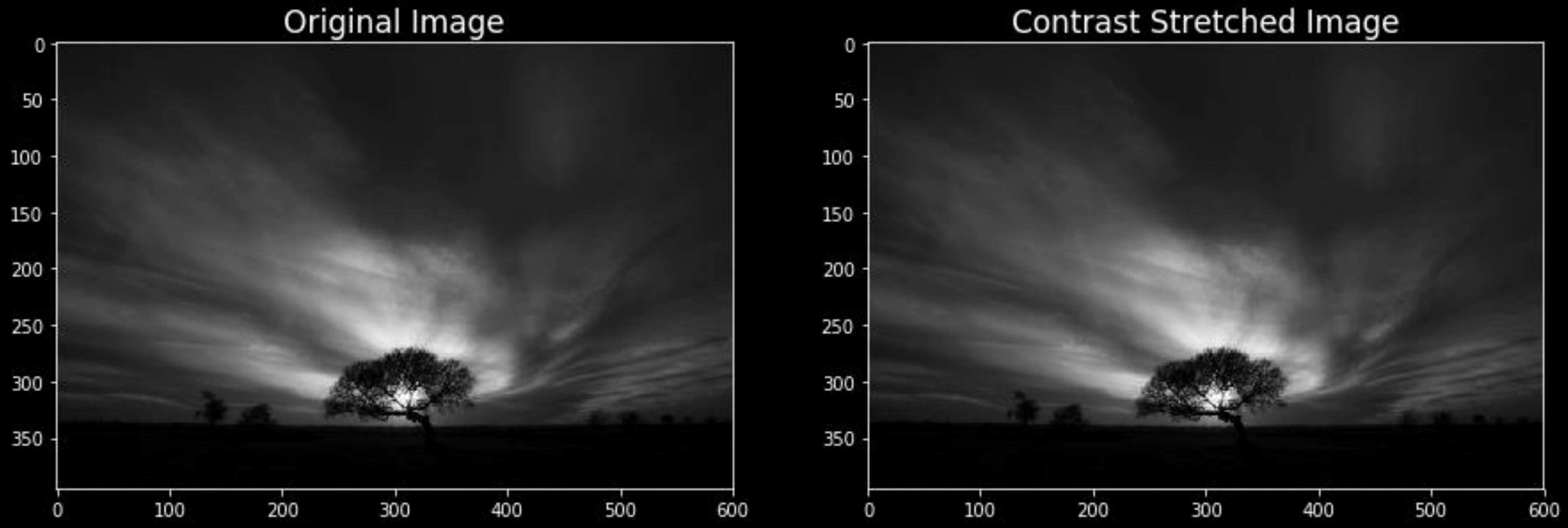
Contrast Stretched Image

Resulted to an image with slightly higher contrast (considering the original Imax (max pixel value)



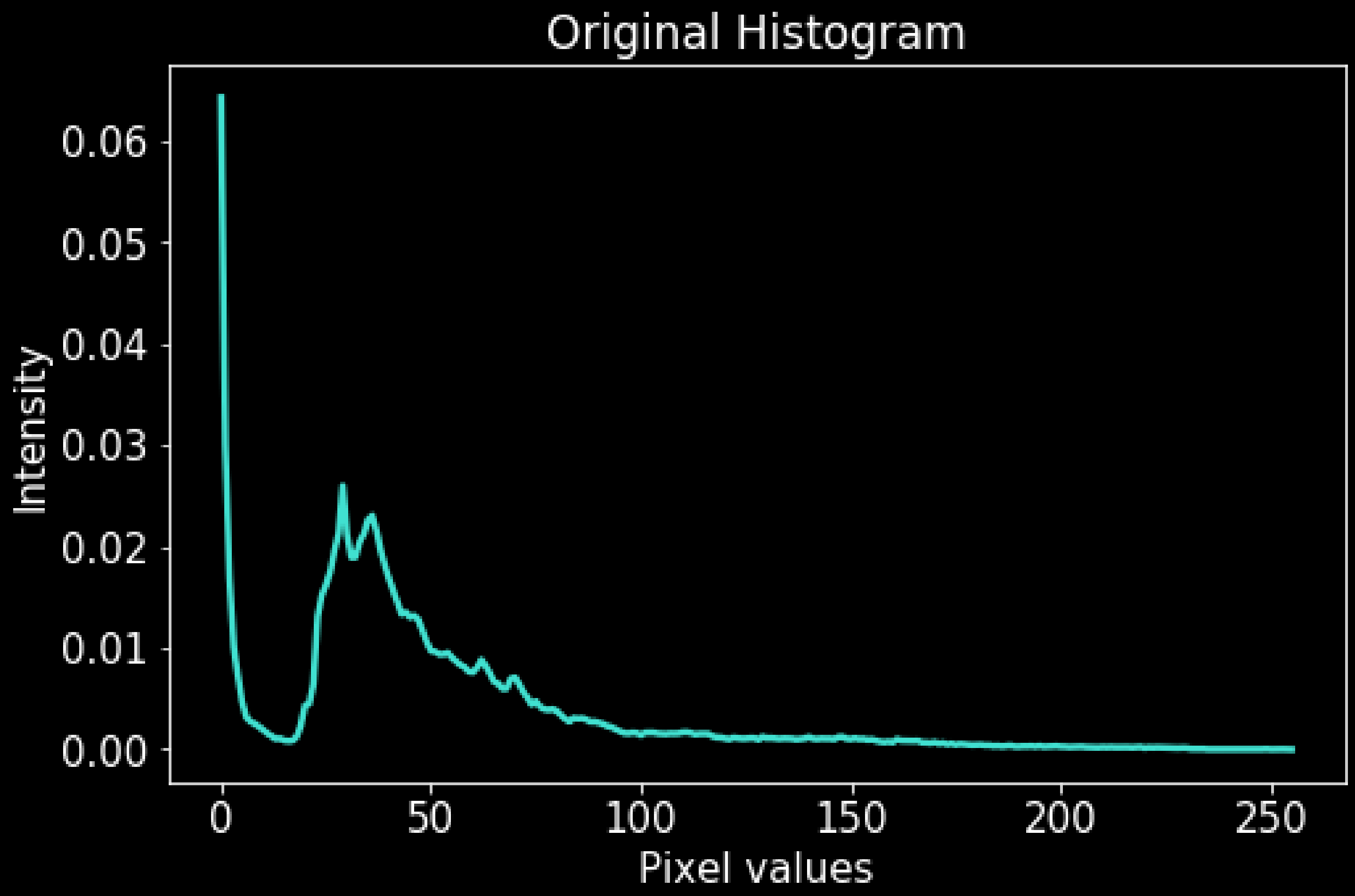
```
Imin:   0 Imax:   253
```

Step 2 Contrast Stretching

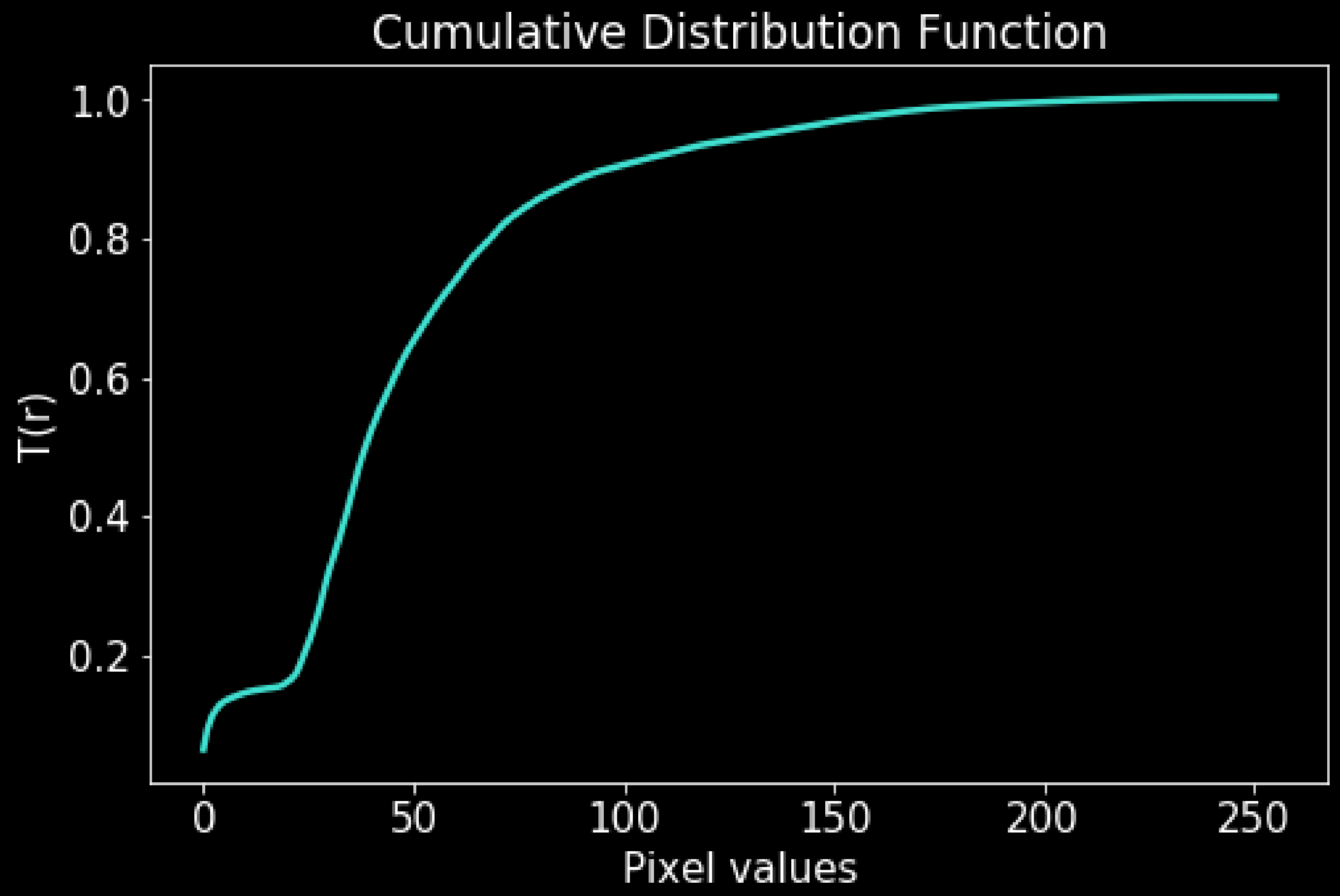


Contrast change isn’t that obvious

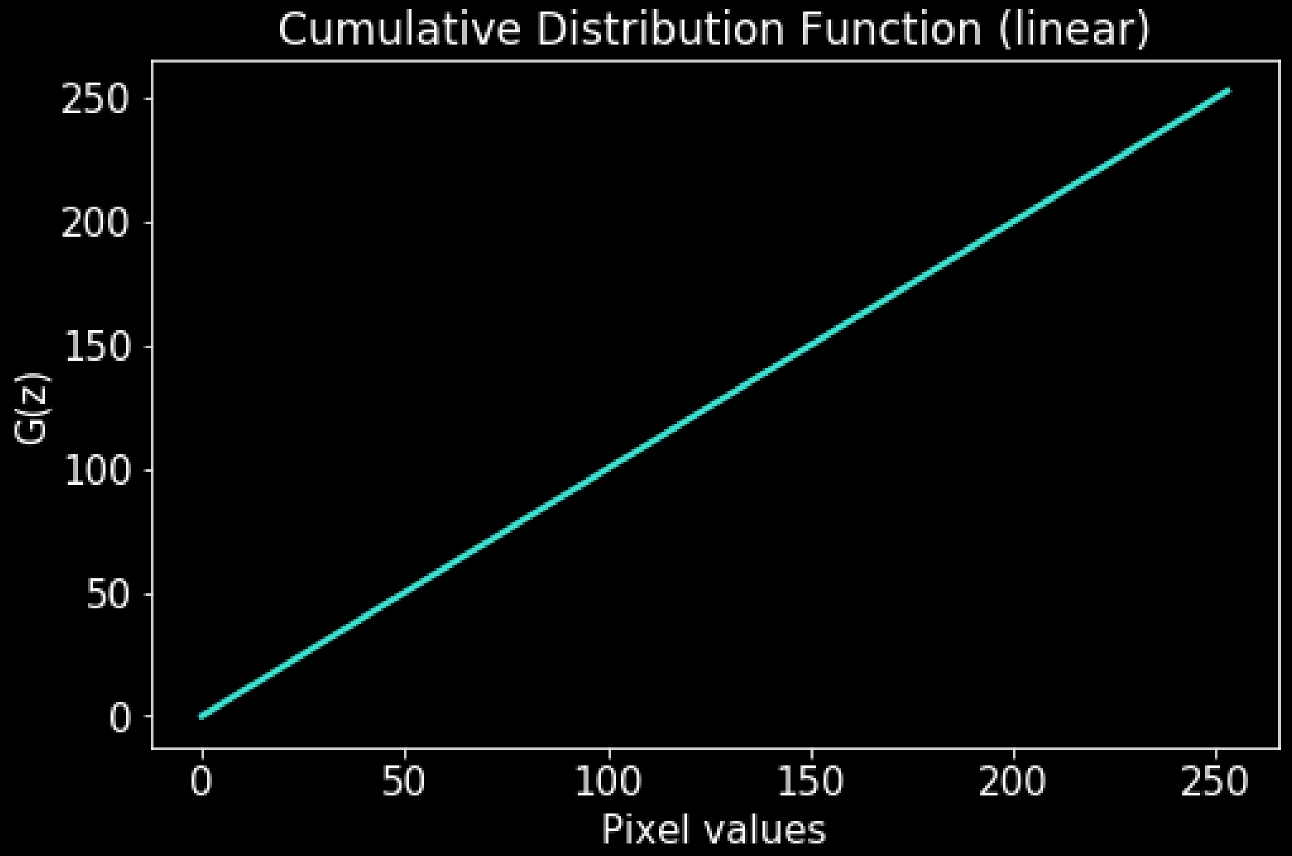
Step 3 Grayscale Histogram (PDF)



Step 4 Original CDF (from PDF)

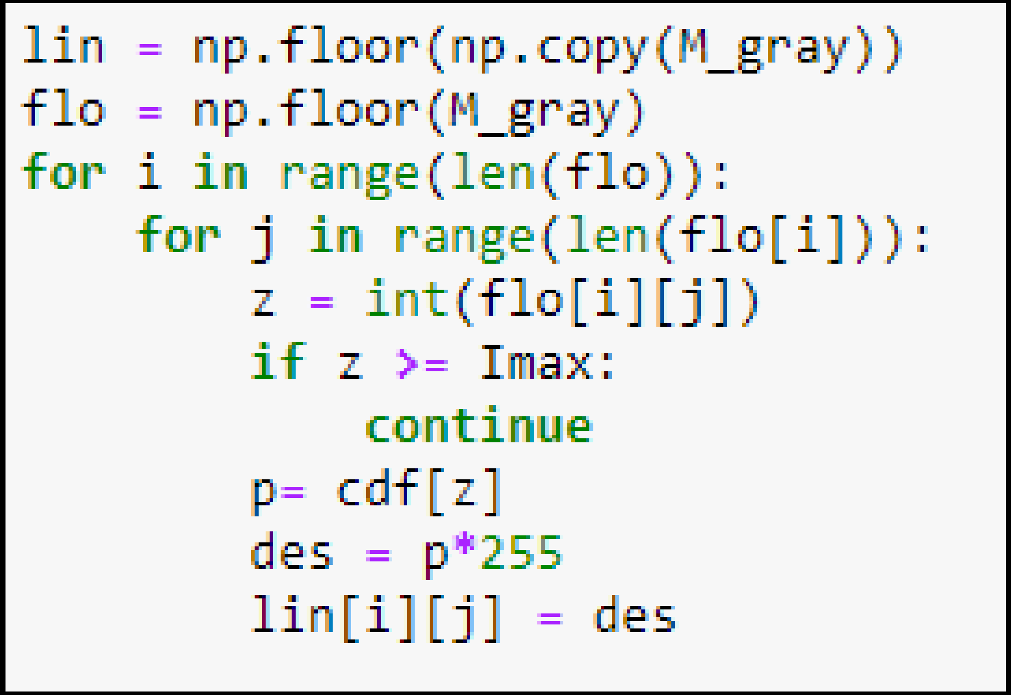


Step 5 Desired CDF (Linear)



$y = x$

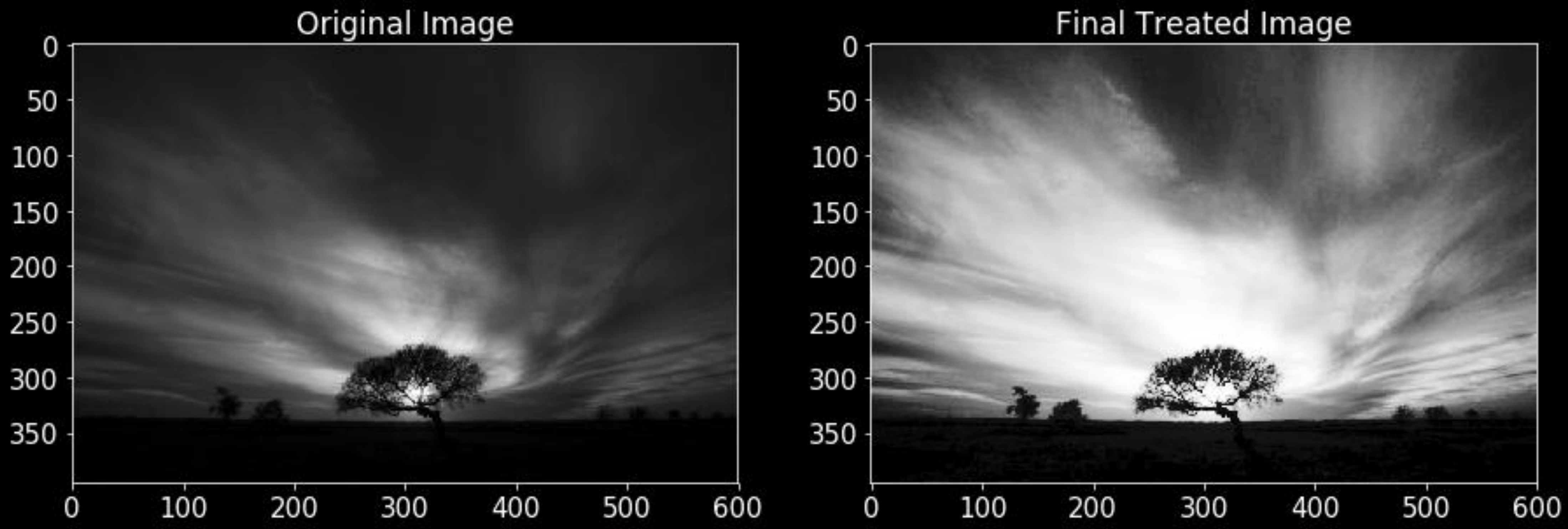
Step 6 Backprojection



```
lin = np.floor(np.copy(M_gray))
flo = np.floor(M_gray)
for i in range(len(flo)):
    for j in range(len(flo[i])):
        z = int(flo[i][j])
        if z >= Imax:
            continue
        p= cdf[z]
        des = p*255
        lin[i][j] = des
```

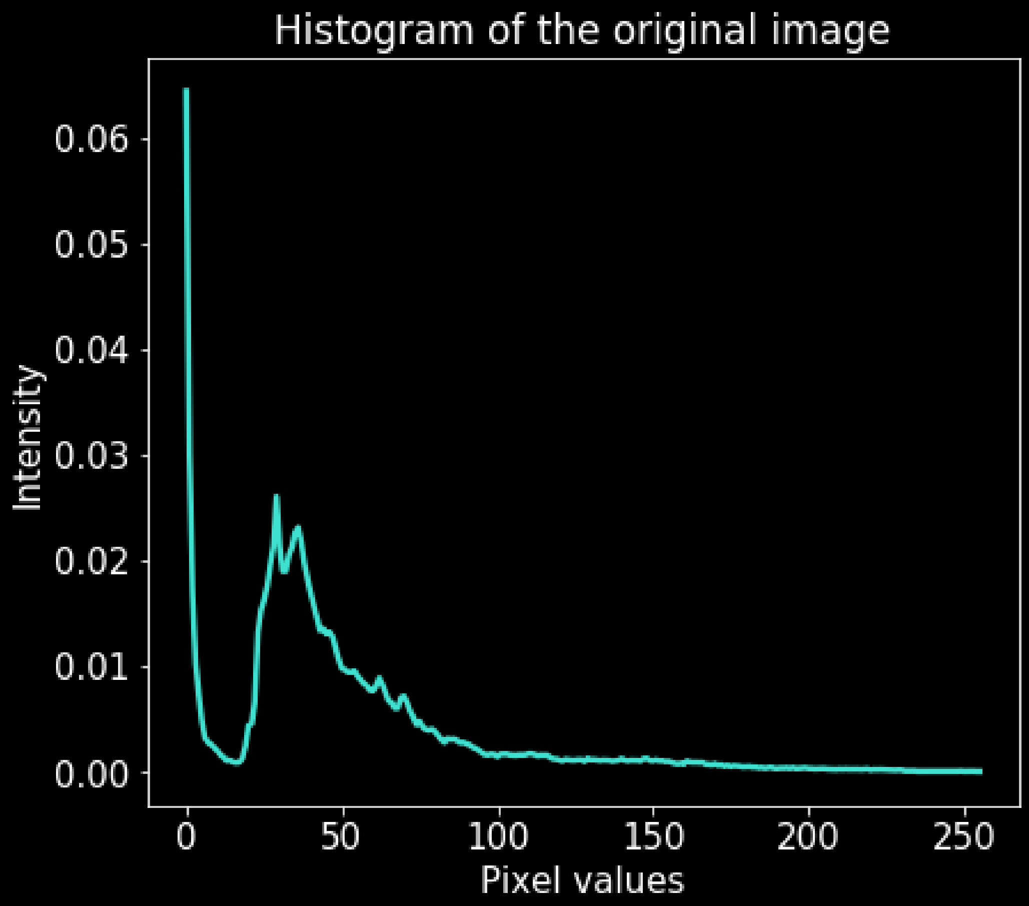
- Uses **for loop** for backprojection between Original CDF and Desired CDF

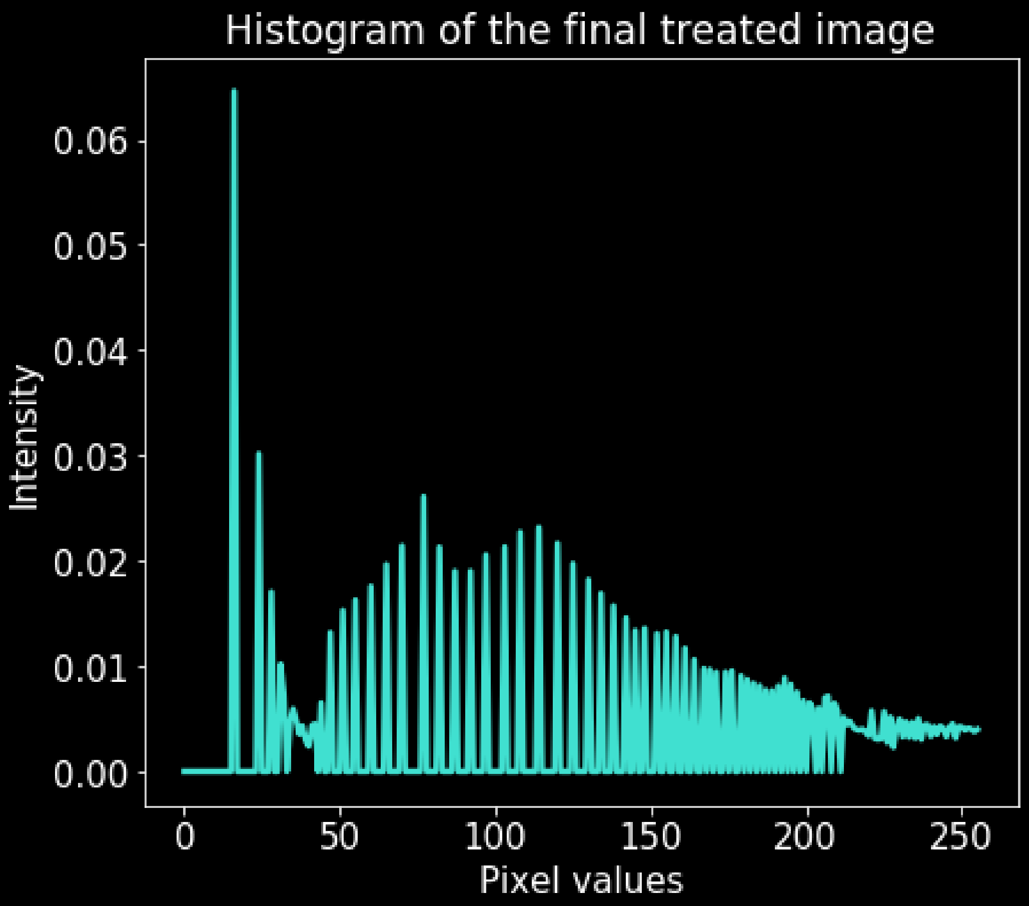
Step 7 Results



- Brighter : sky, terrain, horizon
- Clearer : tree outlines
- Limited : tree branches

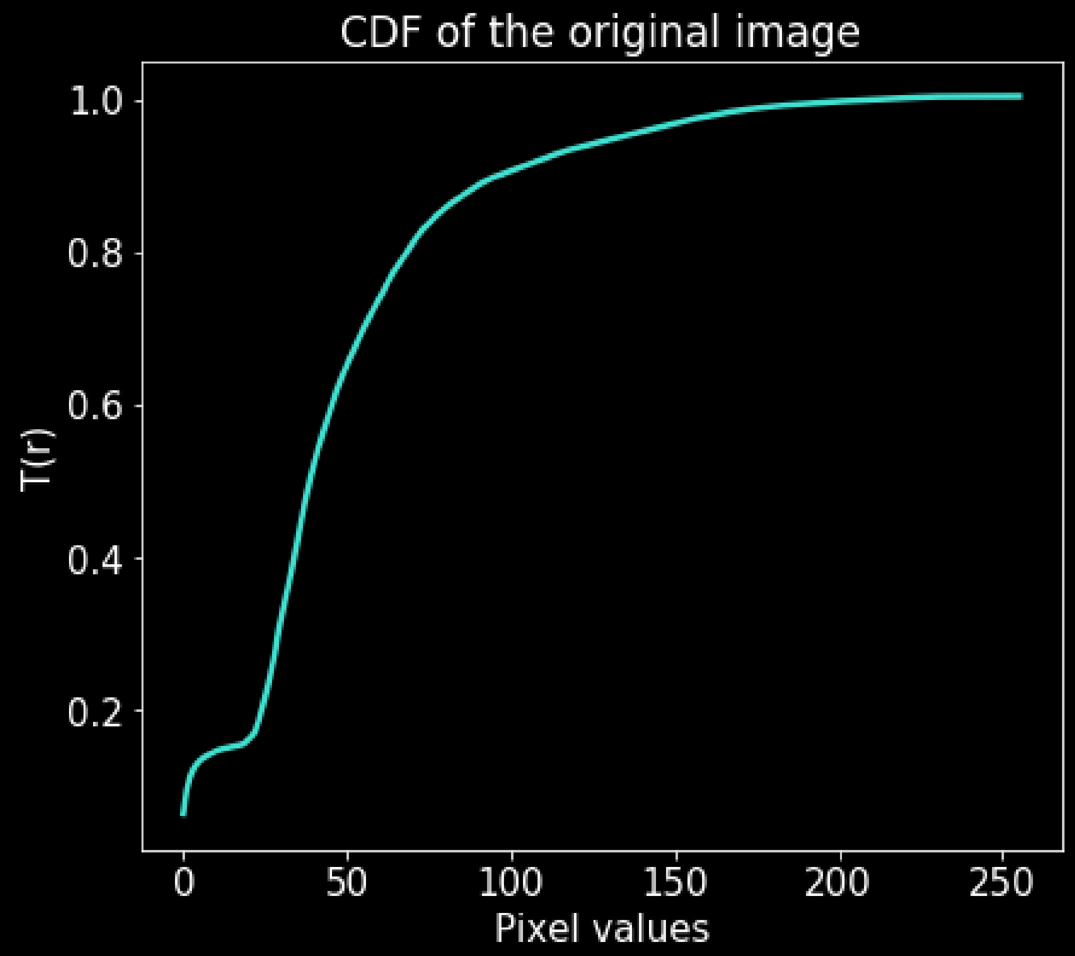
Step 7 Results

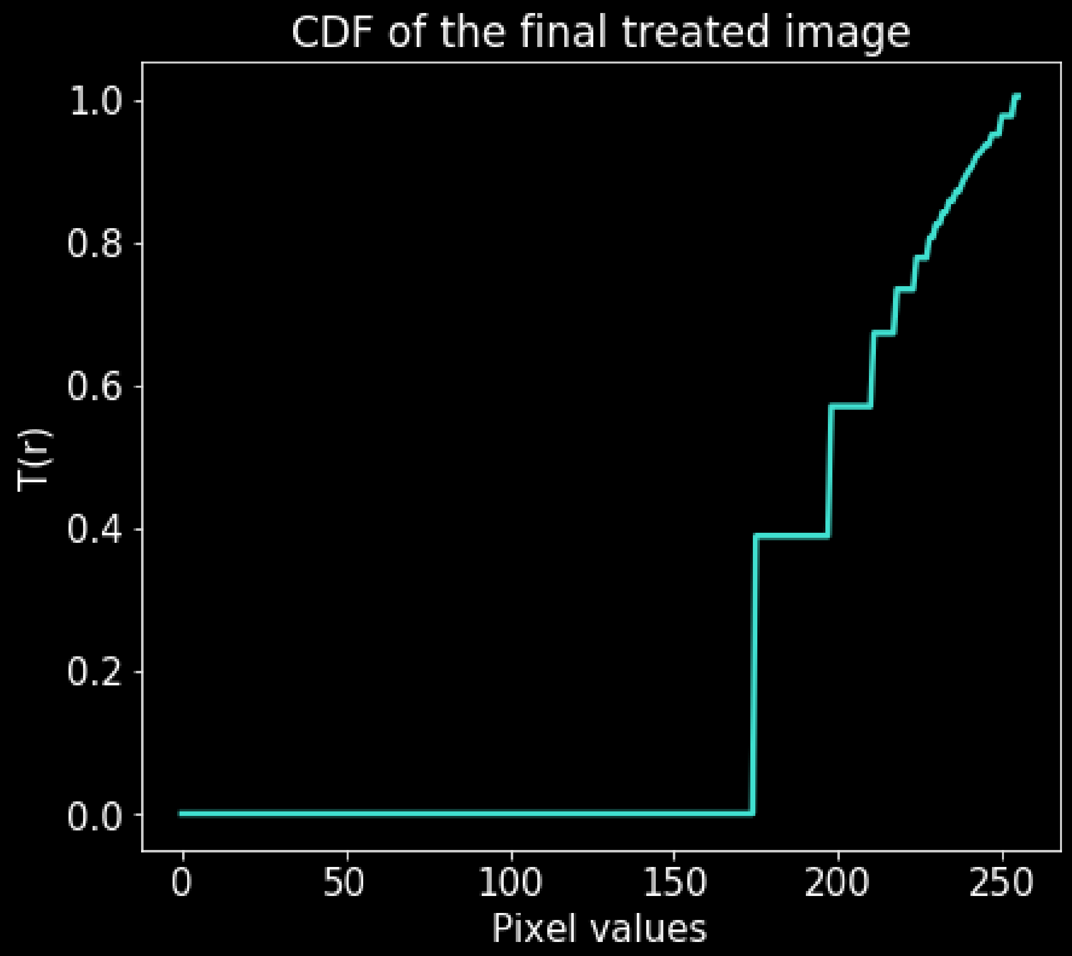

Histogram of the original image


Histogram of the final treated image

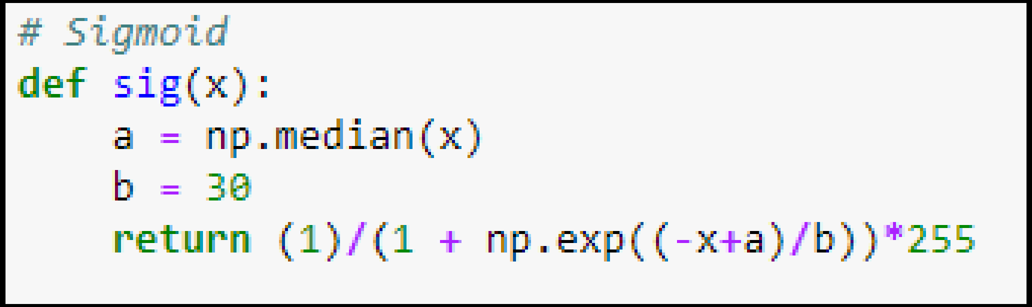
- More discrete pixel values
- More varied

Step 7 Results

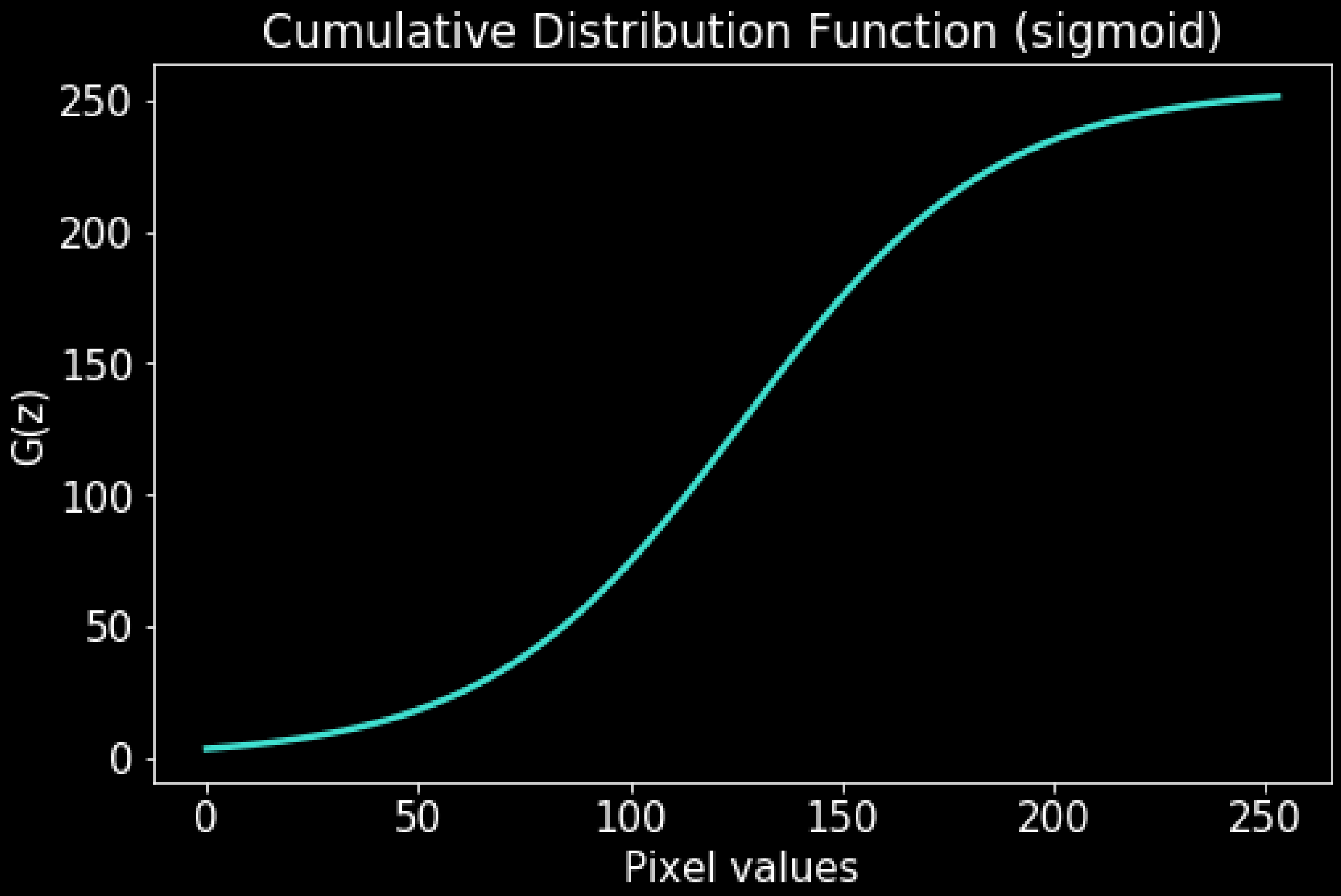




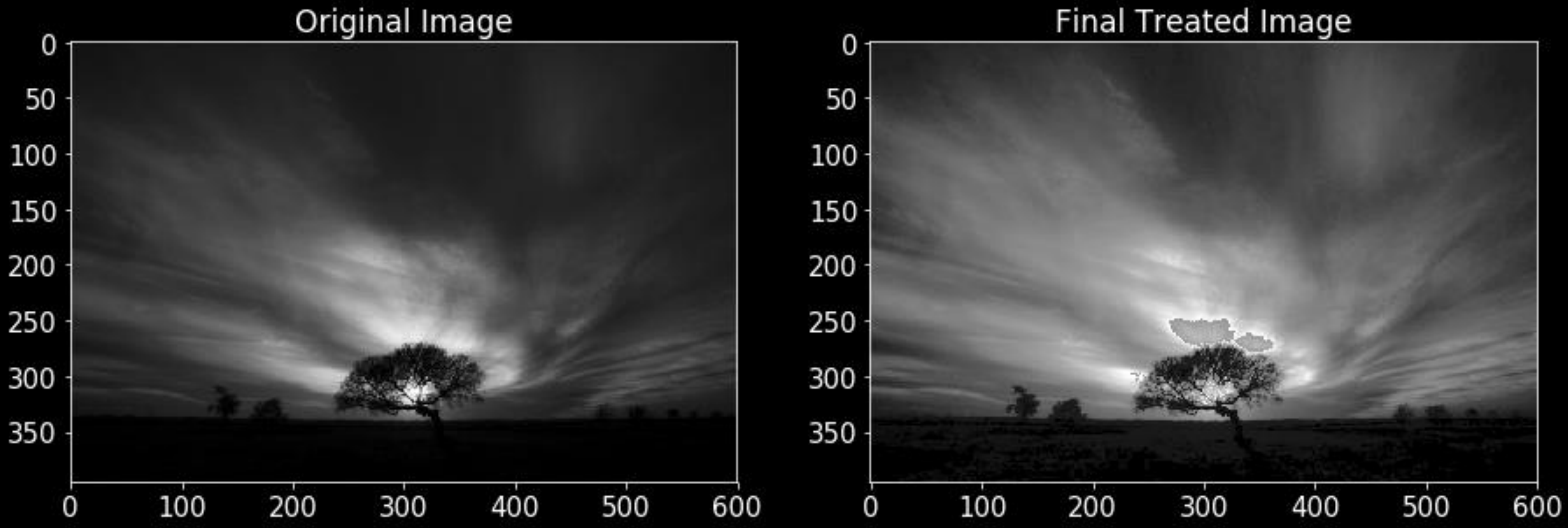
Step * Nonlinear CDF (Sigmoid)



```
# Sigmoid
def sig(x):
    a = np.median(x)
    b = 30
    return (1)/(1 + np.exp((-x+a)/b))*255
```



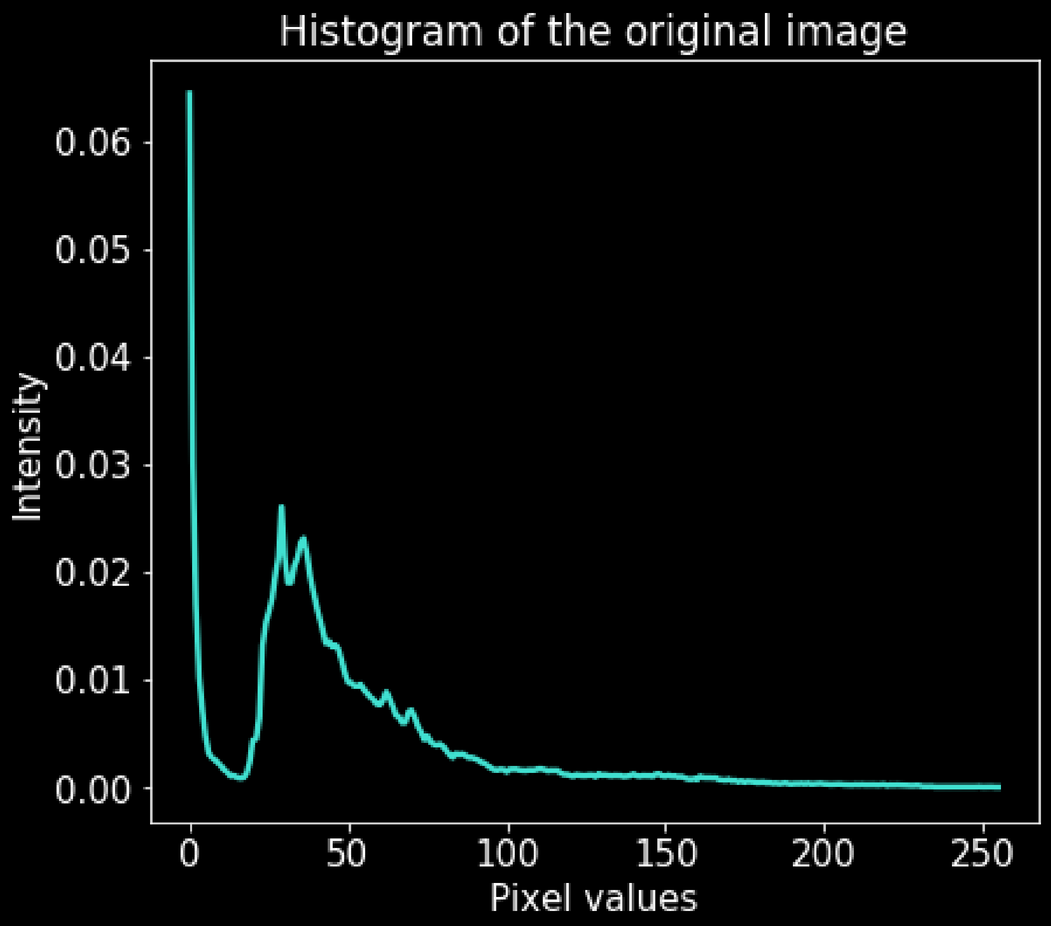
Step * Nonlinear CDF (Sigmoid) Results

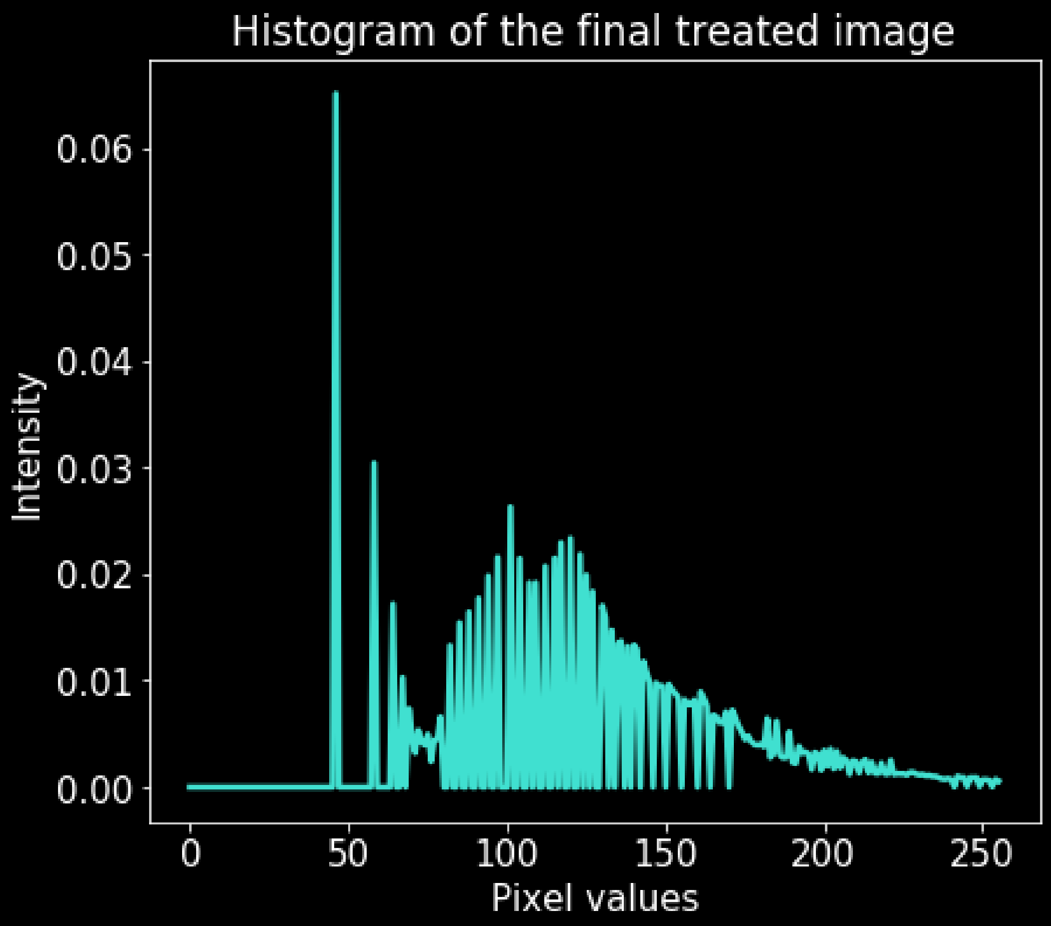

Original Image

Final Treated Image

- Brighter : sky, terrain, horizon
- Clearer : tree outlines
- Limited : tree branches
- *Splotches near the tree

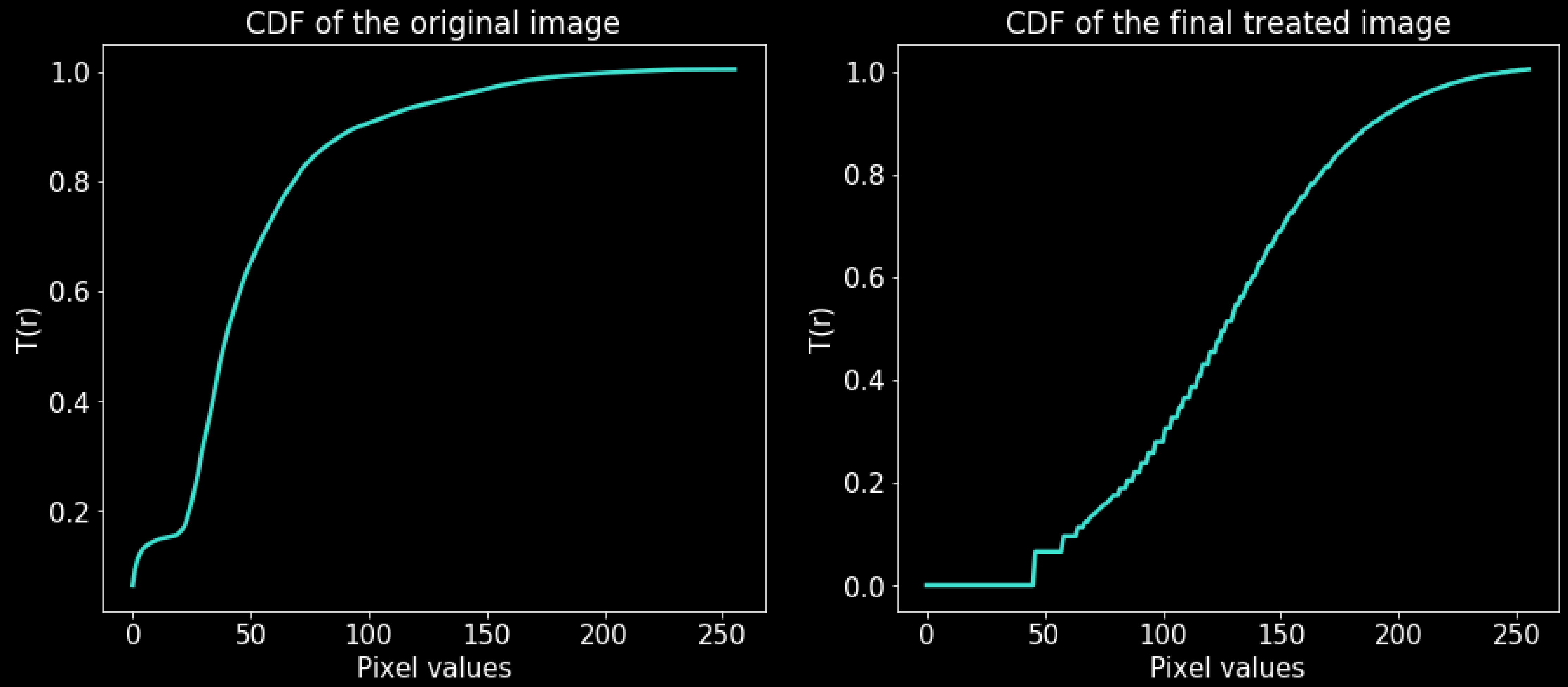
Step * Nonlinear CDF (Sigmoid)) Results


Histogram of the original image

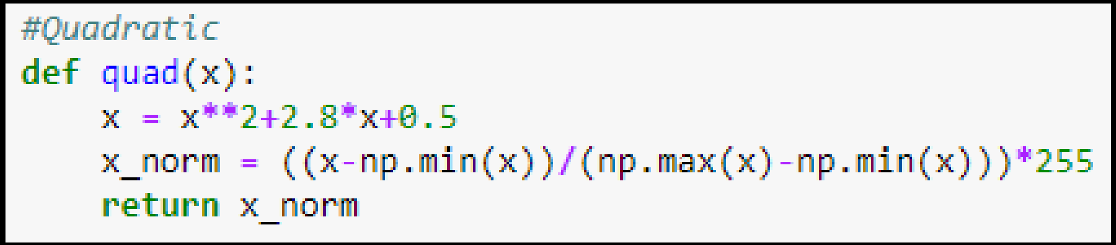

Histogram of the final treated image

- More discrete pixel values
- More varied

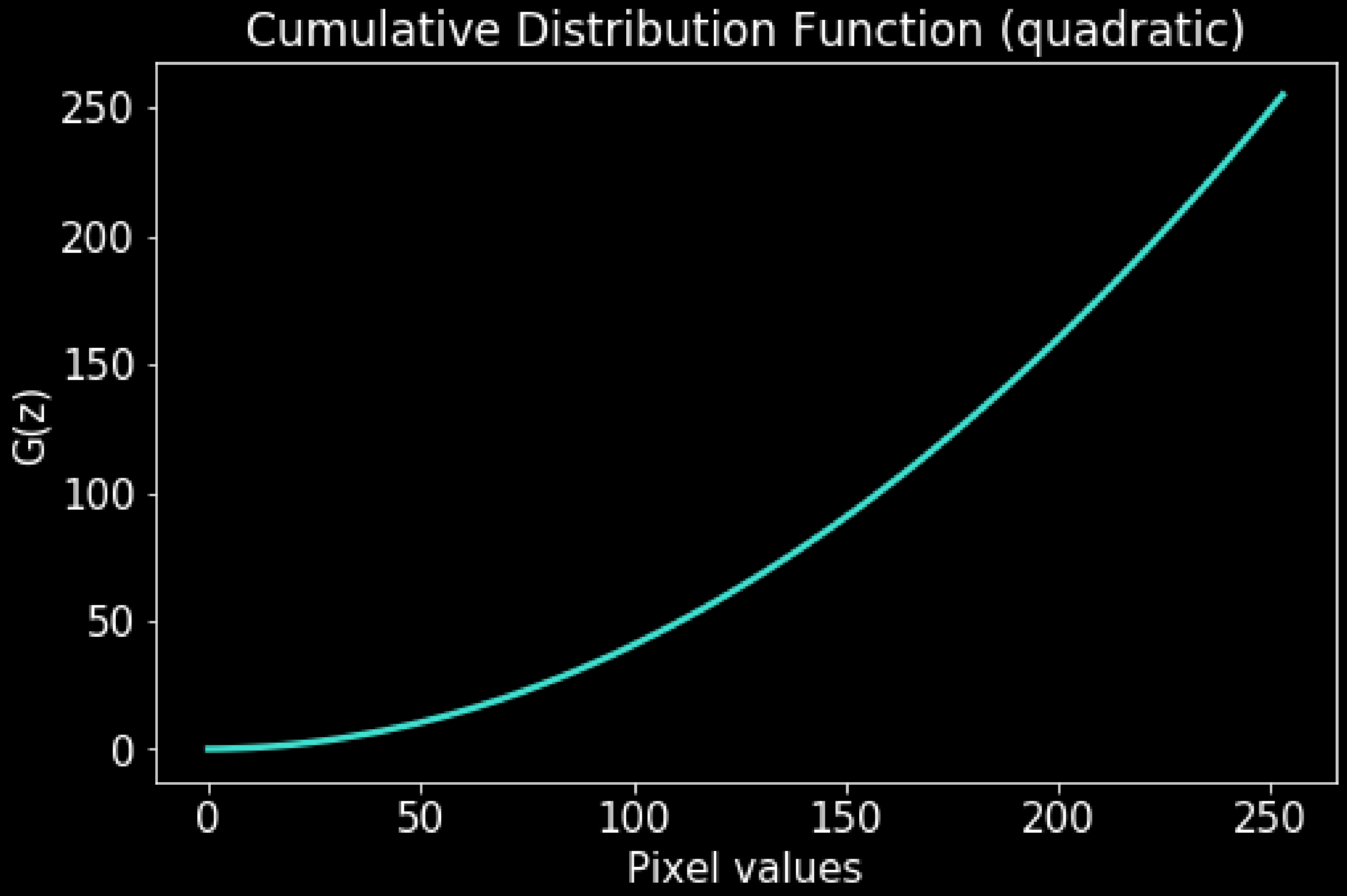
Step * Nonlinear CDF (Sigmoid)) Results



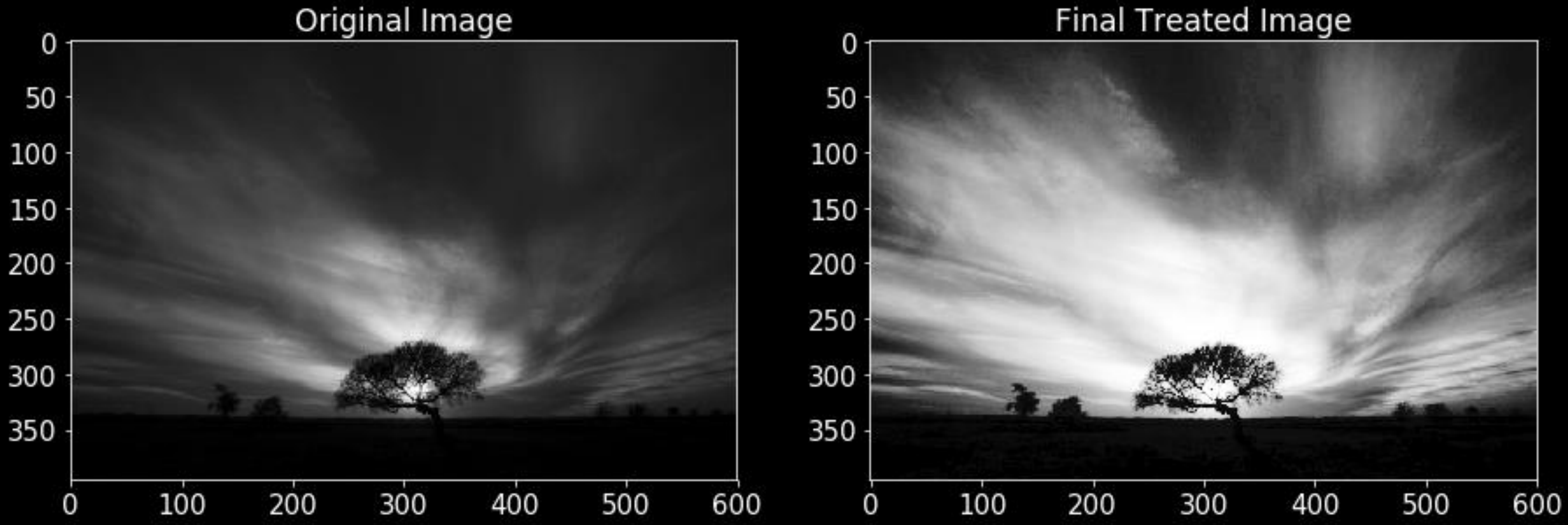
Step * Nonlinear CDF (Quadratic)



```
#Quadratic
def quad(x):
    x = x**2+2.8*x+0.5
    x_norm = ((x-np.min(x))/(np.max(x)-np.min(x)))*255
    return x_norm
```

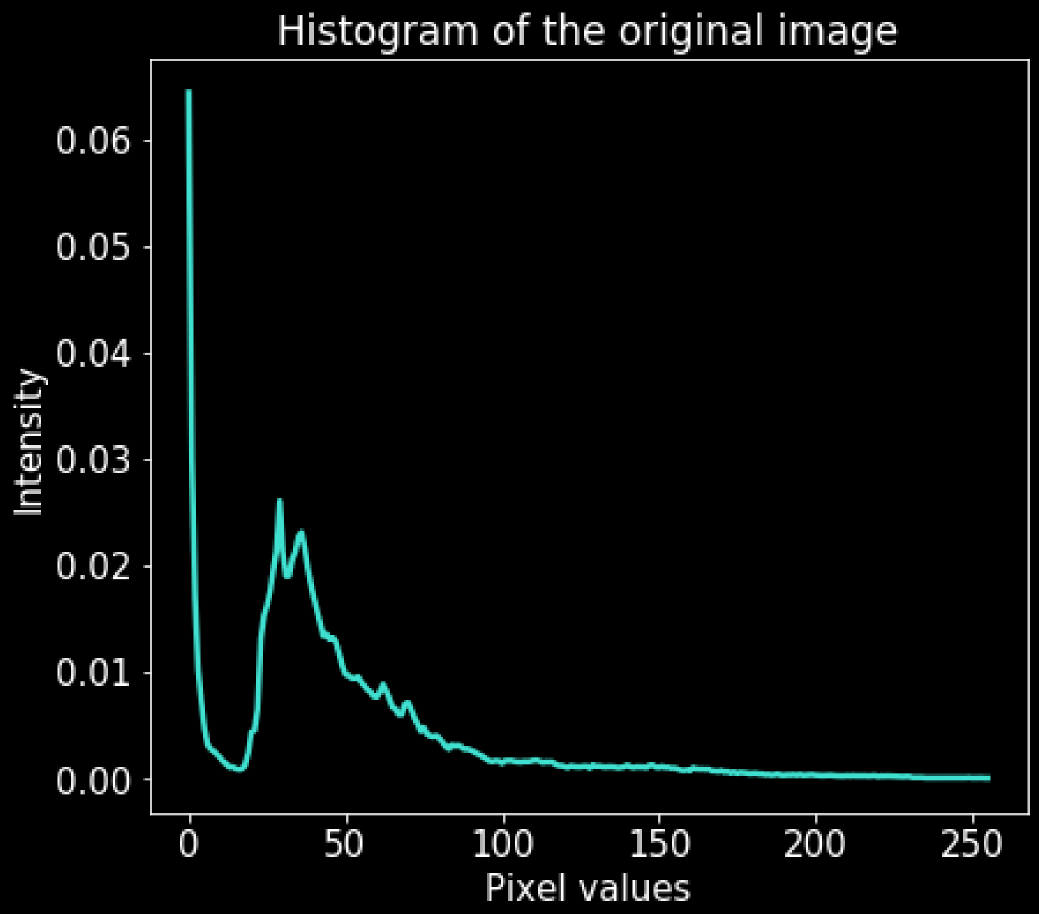


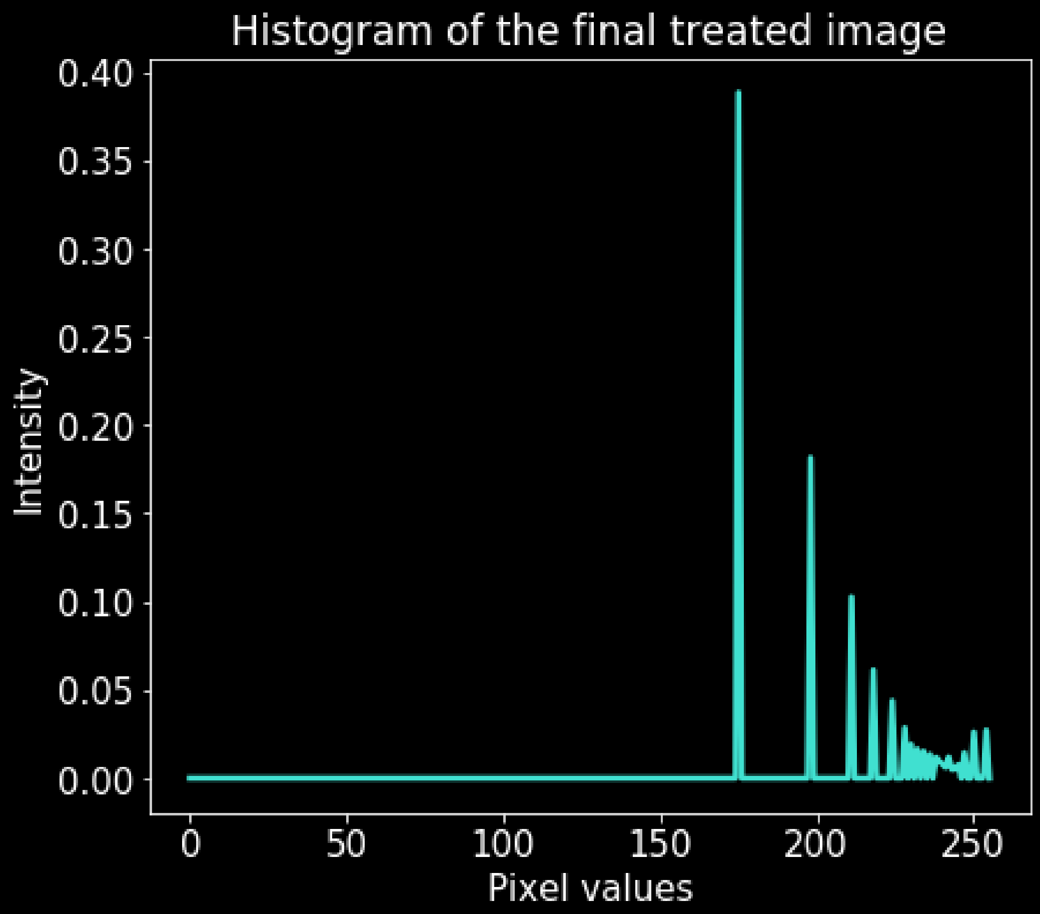
*Step * Nonlinear CDF (Quadratic) Results*



- Brighter : sky, terrain, horizon
- Clearer : tree outlines
- Limited : tree branches

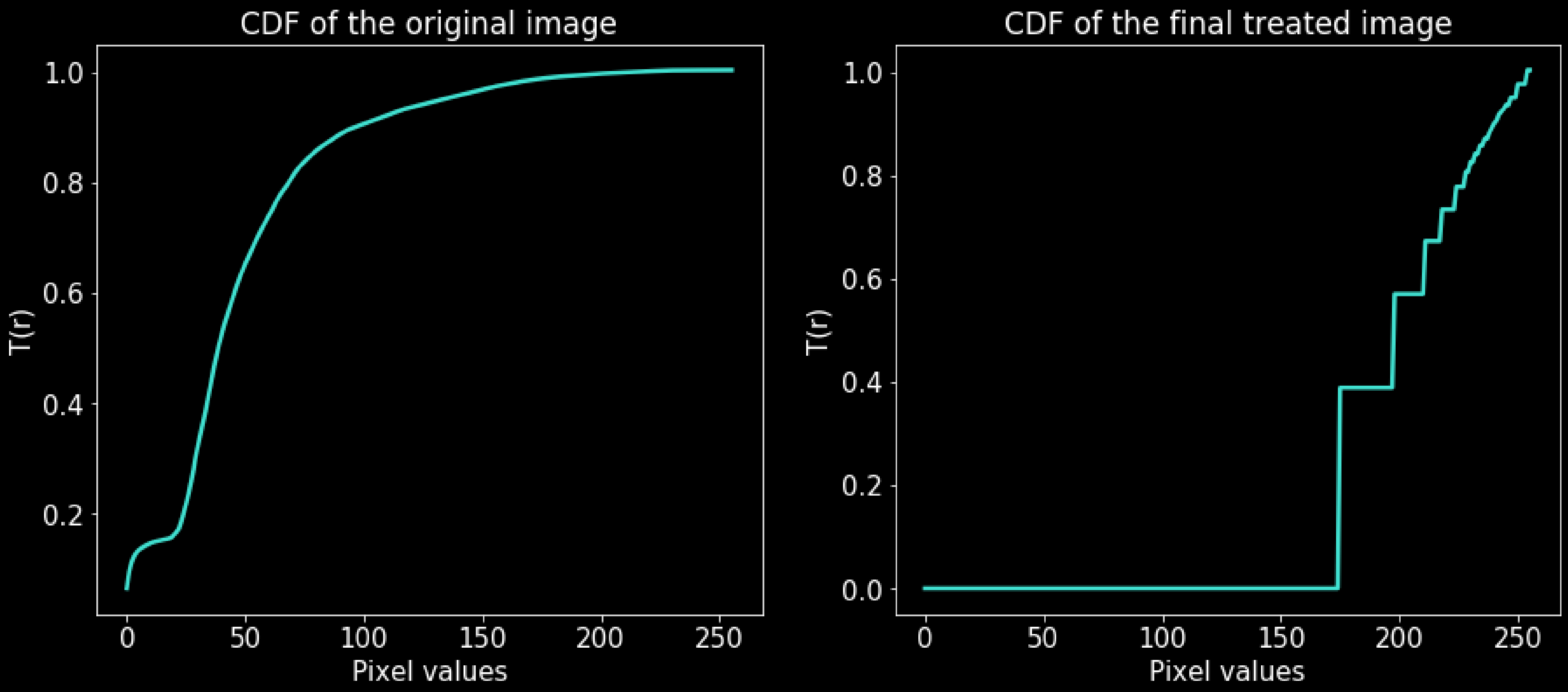
Step * Nonlinear CDF (Quadratic) Results





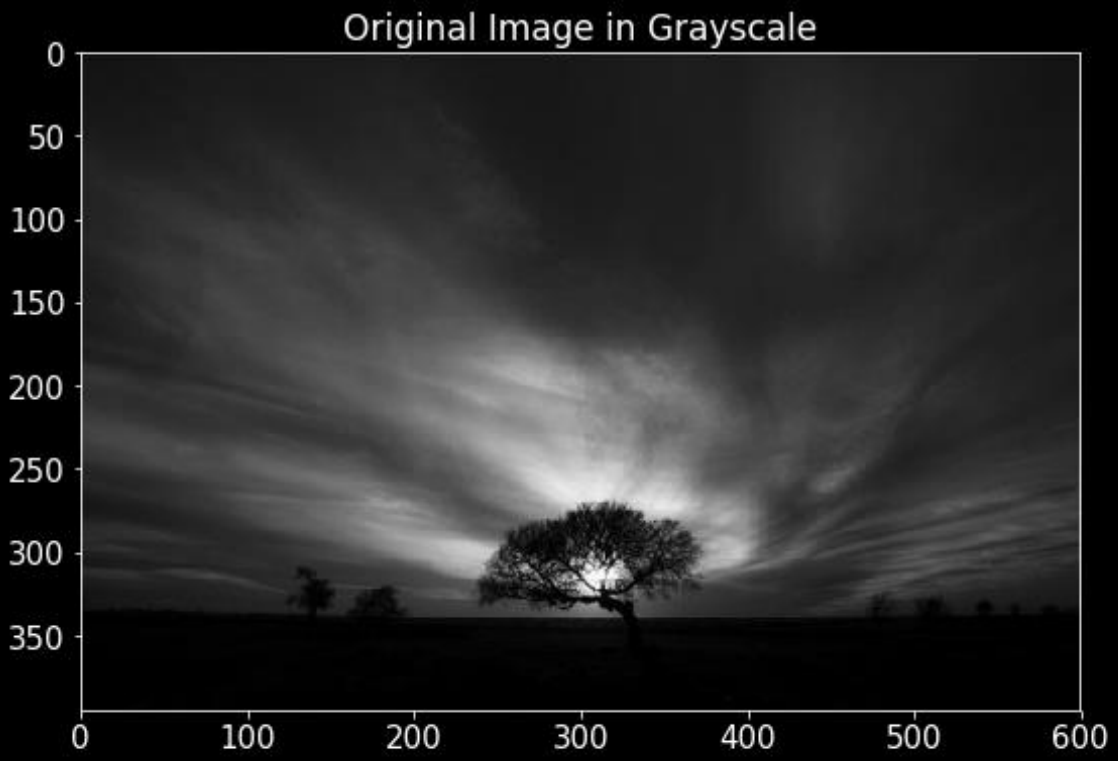
- More discrete pixel values
- More varied

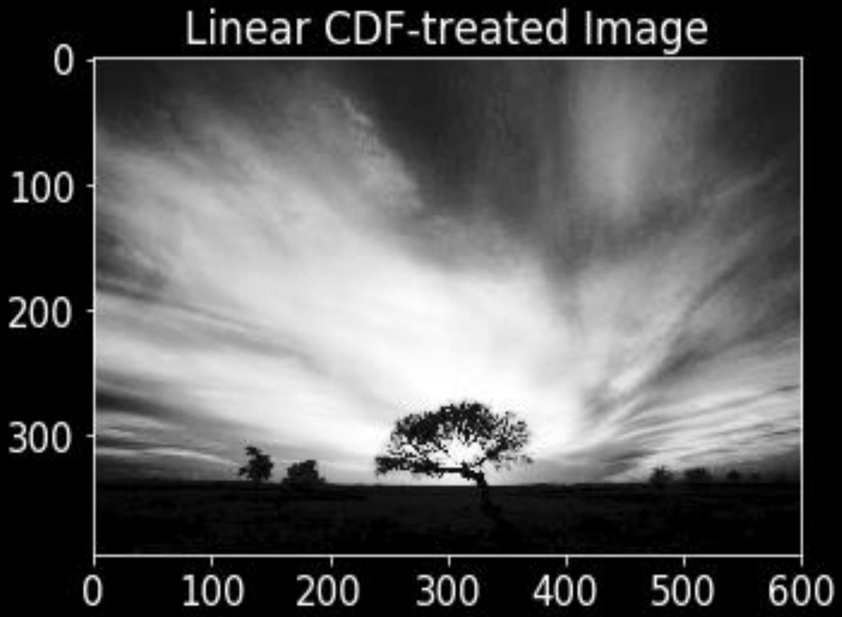
Step * Nonlinear CDF (Quadratic) Results

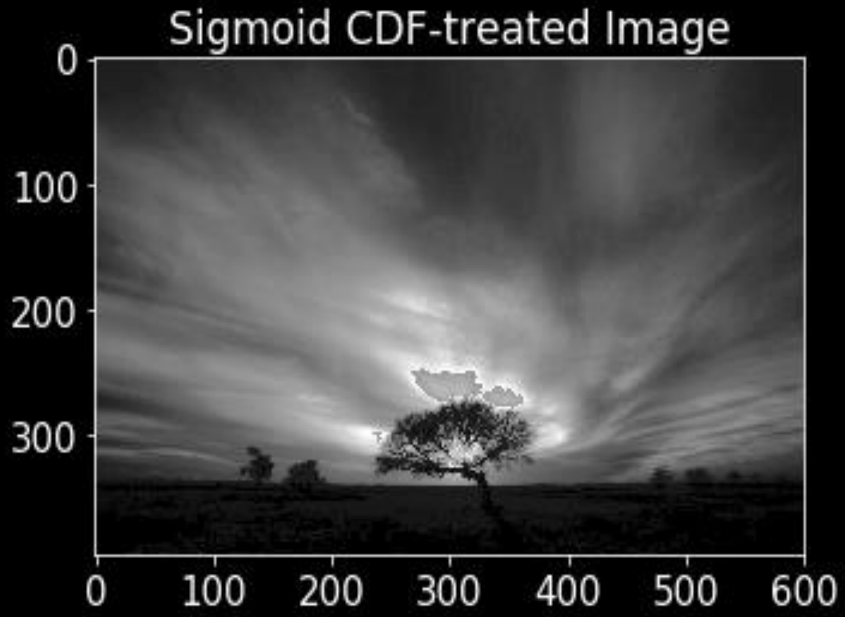


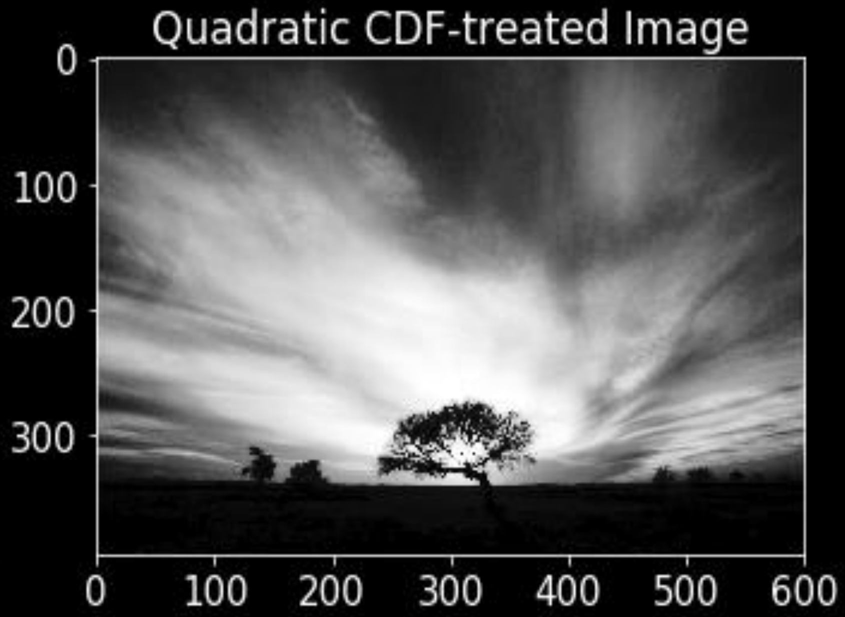
Comparison

- Brightest terrain : Sigmoid
- Brightest horizon : Sigmoid
- Brightest sky : Linear and Quadratic
- Clearest tree outlines : Sigmoid



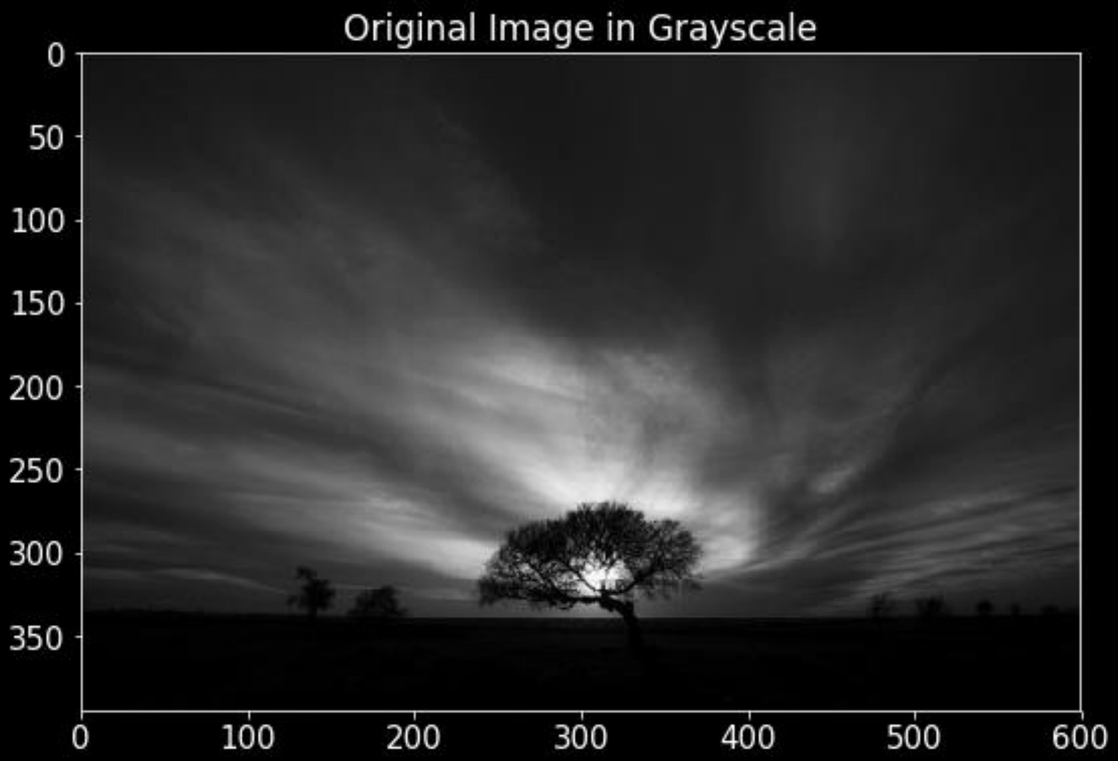


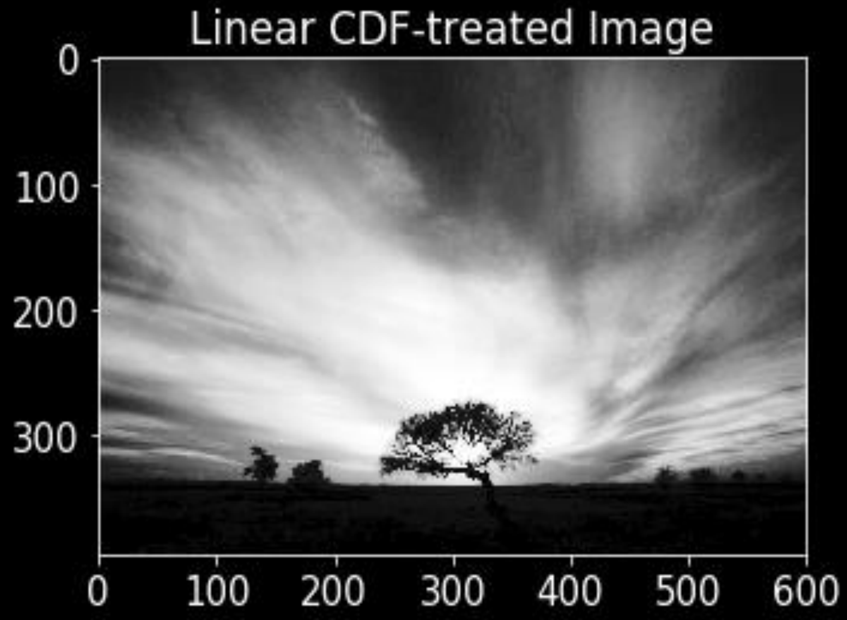


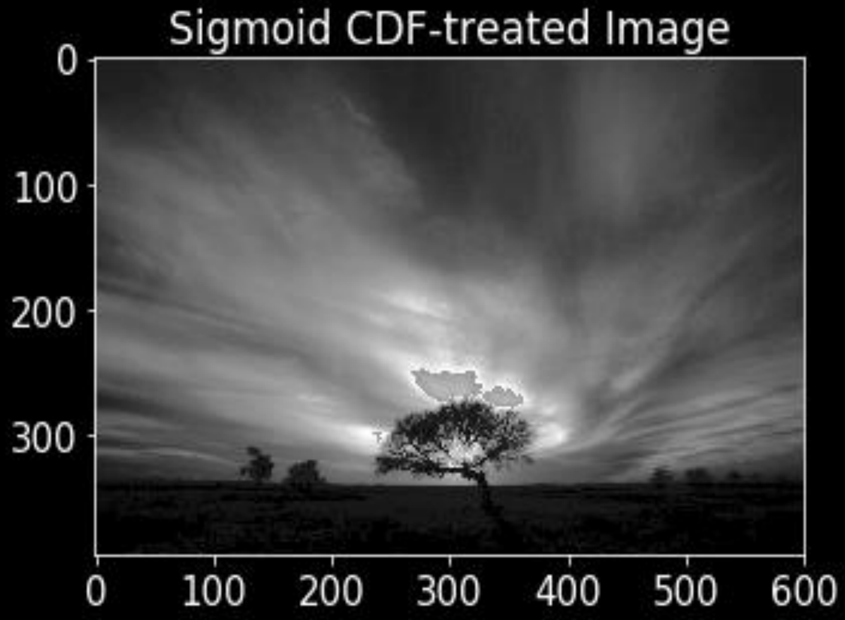


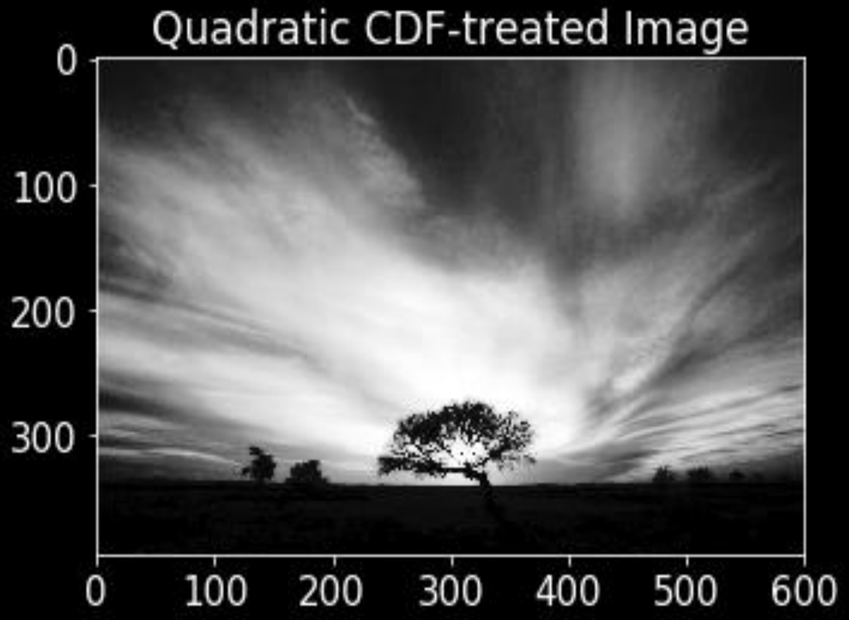
* Comparison

- Sigmoid CDF-treated image showed more details that are perceptible to my human eyes





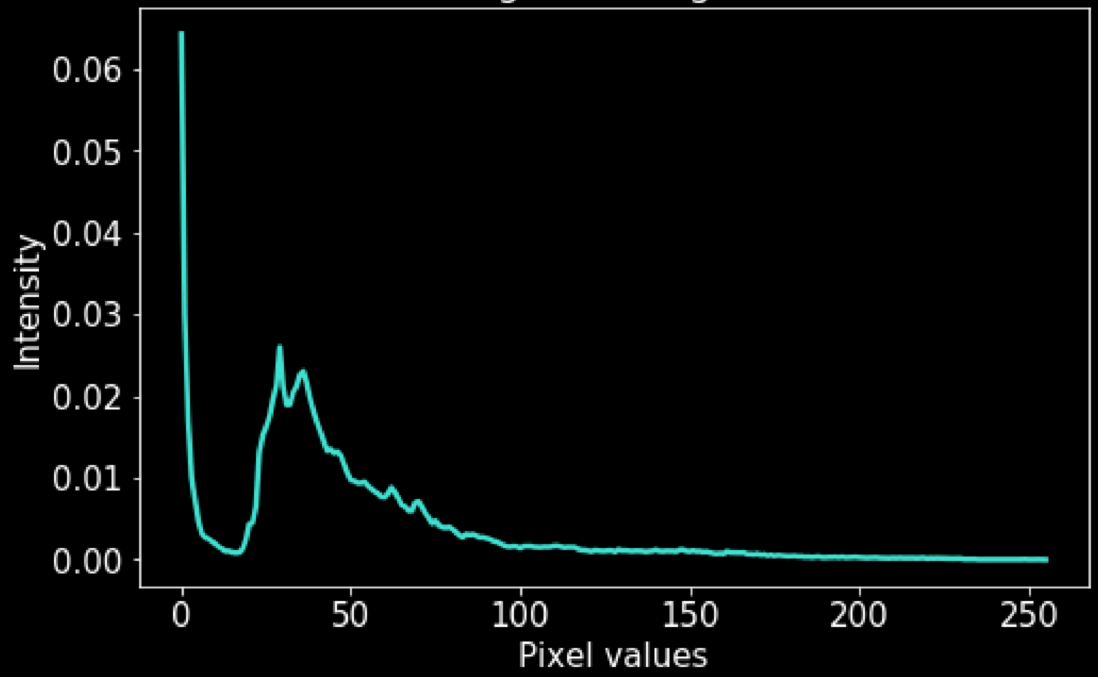




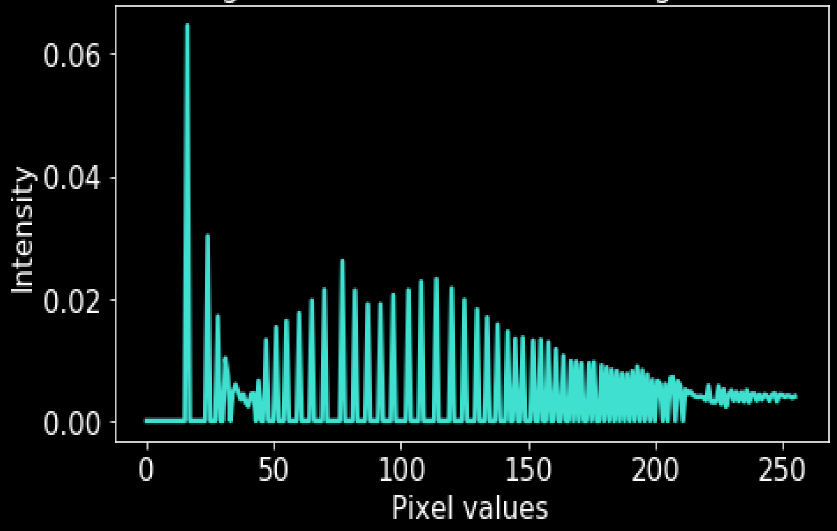
* Comparison

- Highest contrast (biggest max-min value): Linear
- Lowest contrast: Quadratic
- Most varied: Linear
- Most amount of midtones: Sigmoid

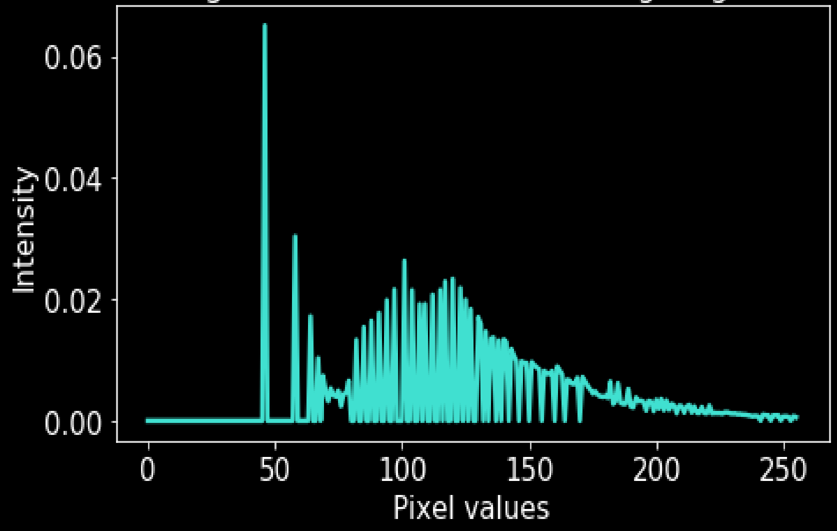
Original Histogram



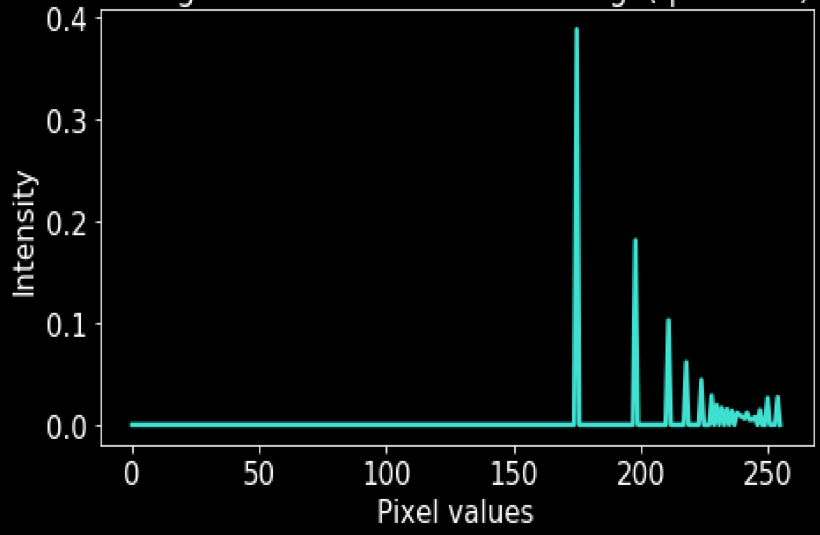
Histogram of the final treated image(linear)



Histogram of the final treated image(sigmoid)



Histogram of the final treated image(quadratic)



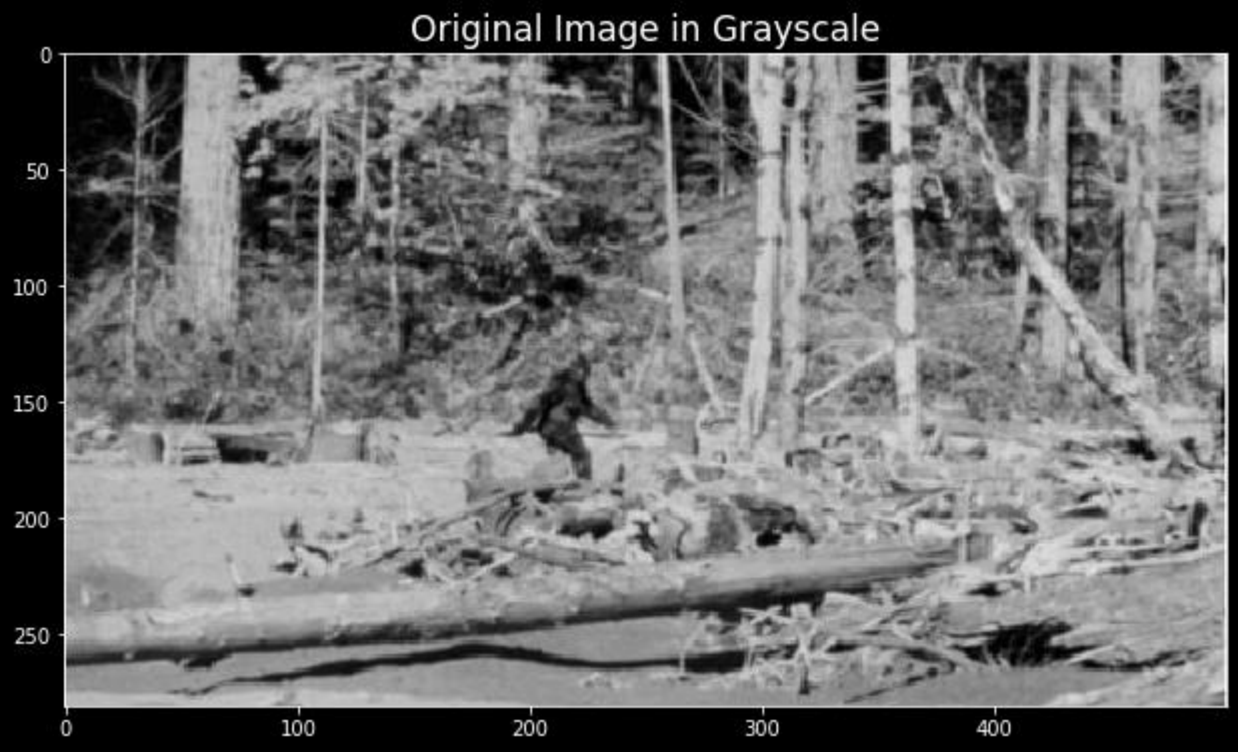
Bonus Step * *BIGFOOT*

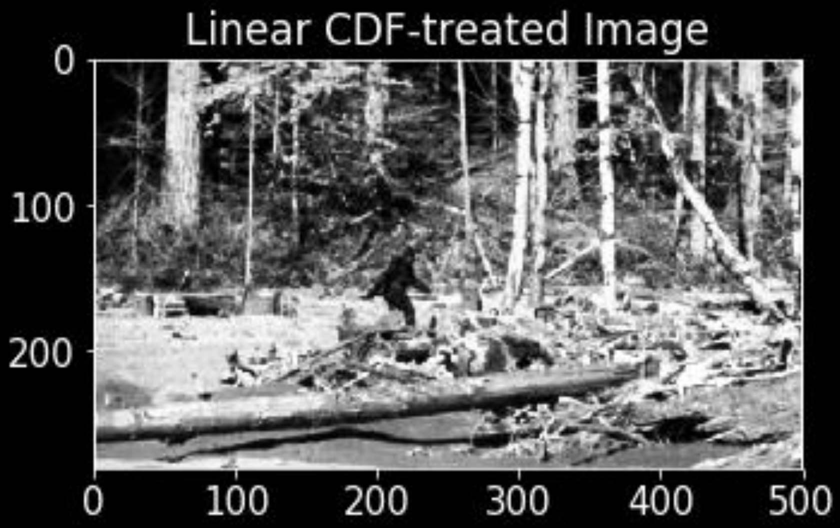
Does **BIGFOOT** actually exist? Let’s see….

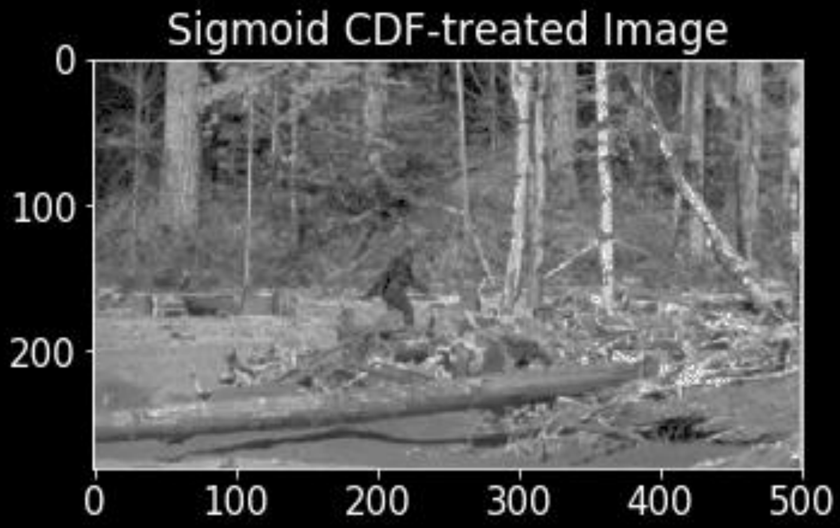
Comparison

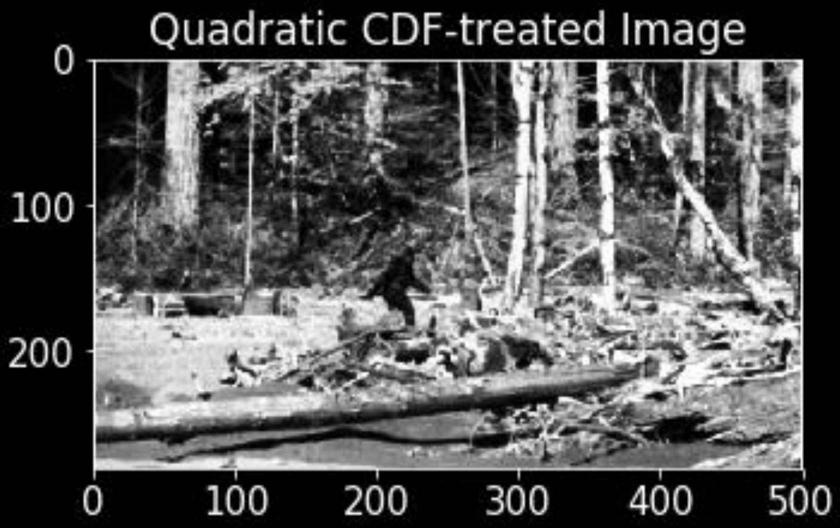
- Quadratic and Linear CDF-treated images showed more details of the so-called **BIGFOOT**.
- It looks more like a guy in a gorilla suit!!

FAKE……


Original Image in Grayscale


Linear CDF-treated Image


Sigmoid CDF-treated Image


Quadratic CDF-treated Image

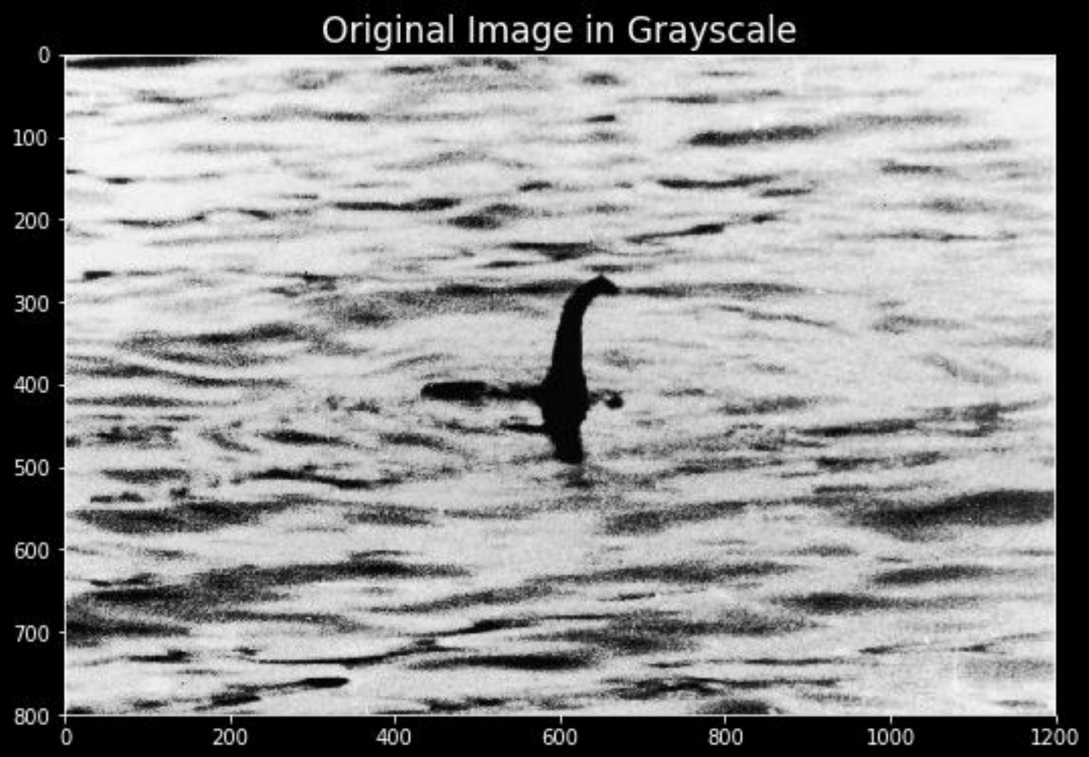
Bonus Step * BIGFOOT

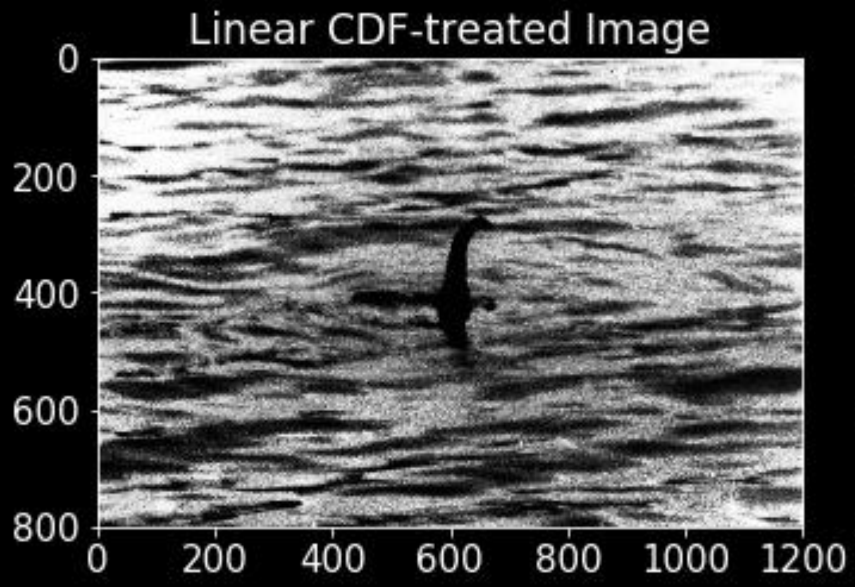
How about the Loch Ness Monster?

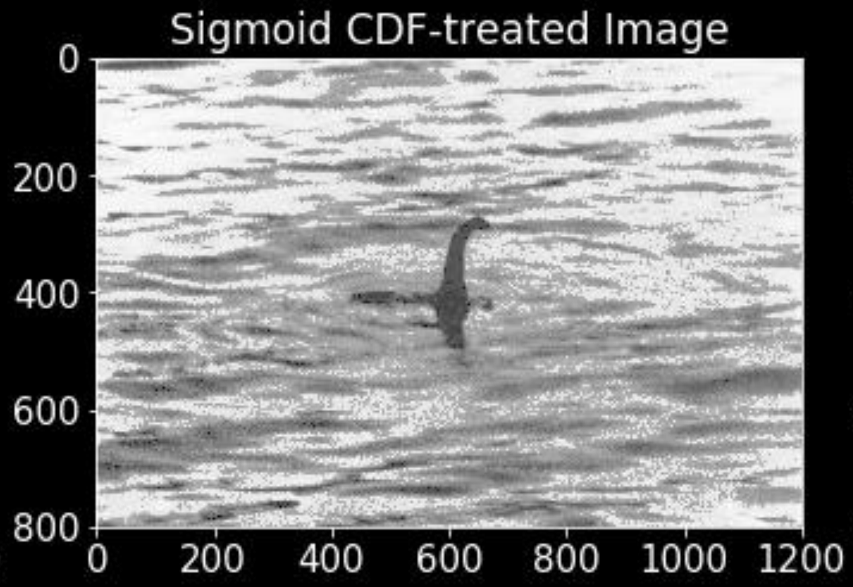
* *Comparison*

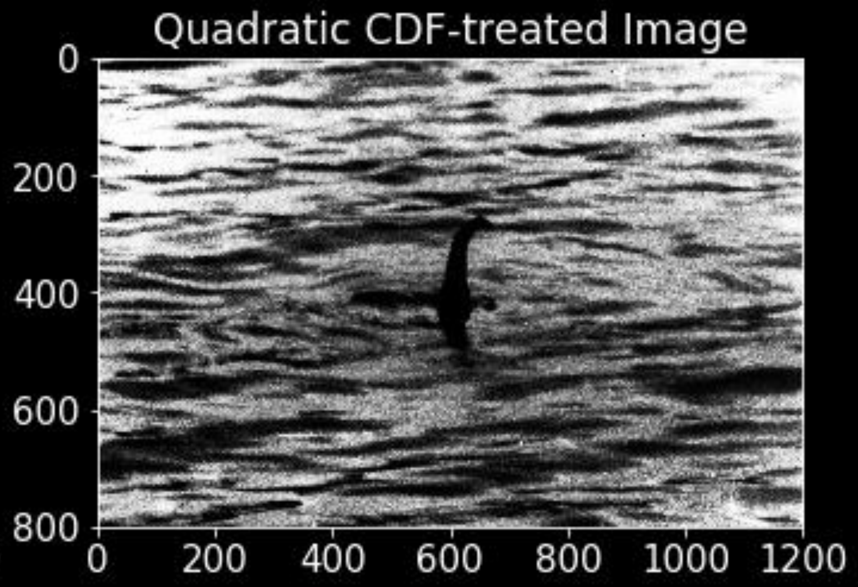
- Not a lot of details came about.
- Beautiful murky waters but what’s in it is really just a silhouette of an inflated balloon..

FAKE......


Original Image in Grayscale


Linear CDF-treated Image


Sigmoid CDF-treated Image


Quadratic CDF-treated Image

*

Pointssss

- TC : 10
- QP : 10
- IN : 2…..??

- This was so much fun ☺! Thank you!