

A histogram is a representation of frequency distribution. Histogram manipulation can be used for image enhancement. Histogram equalization is the process of stretching the histogram of an image so that its bars reach either end of the spectrum. It is a widely used contrast-enhancement technique in image processing because of its high efficiency and simplicity.

Let's see !

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

# Load the image
image_path = 'ursc_food.jpeg'
original_image = cv2.imread(image_path)

# Convert the image to grayscale
grayscale_image = cv2.cvtColor(original_image, cv2.COLOR_BGR2GRAY)

# Calculate the histogram of the grayscale image
hist = cv2.calcHist([grayscale_image], [0], None, [256], [0, 256])

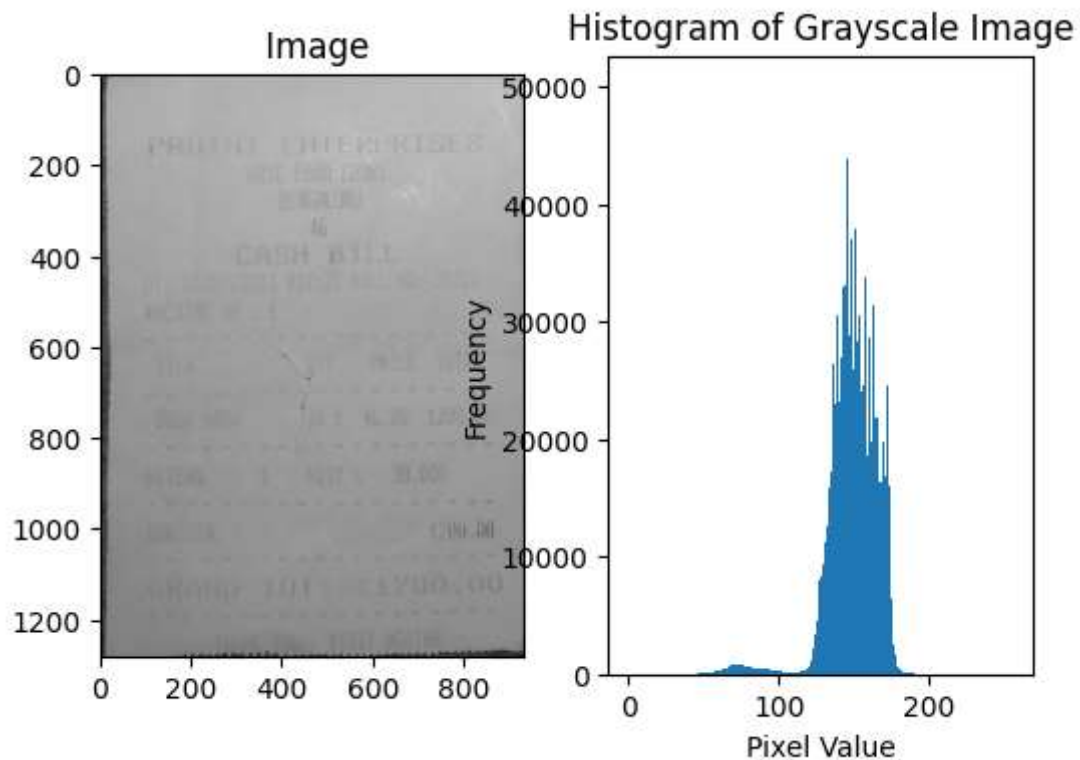
# Apply histogram equalization
equalized_image = cv2.equalizeHist(grayscale_image)
```

```
In [11]: plt.figure(figsize=(6, 4))

plt.subplot(1, 2, 1)
plt.imshow(grayscale_image, cmap='gray')
plt.title(' Image')

plt.subplot(1, 2, 2)
plt.hist(grayscale_image.ravel(), 256, [0, 256])
plt.title('Histogram of Grayscale Image')
plt.xlabel('Pixel Value')
plt.ylabel('Frequency')
plt.figure(figsize=(6, 4))

plt.tight_layout() # Ensures proper spacing between subplots
plt.show()
```



<Figure size 600x400 with 0 Axes>

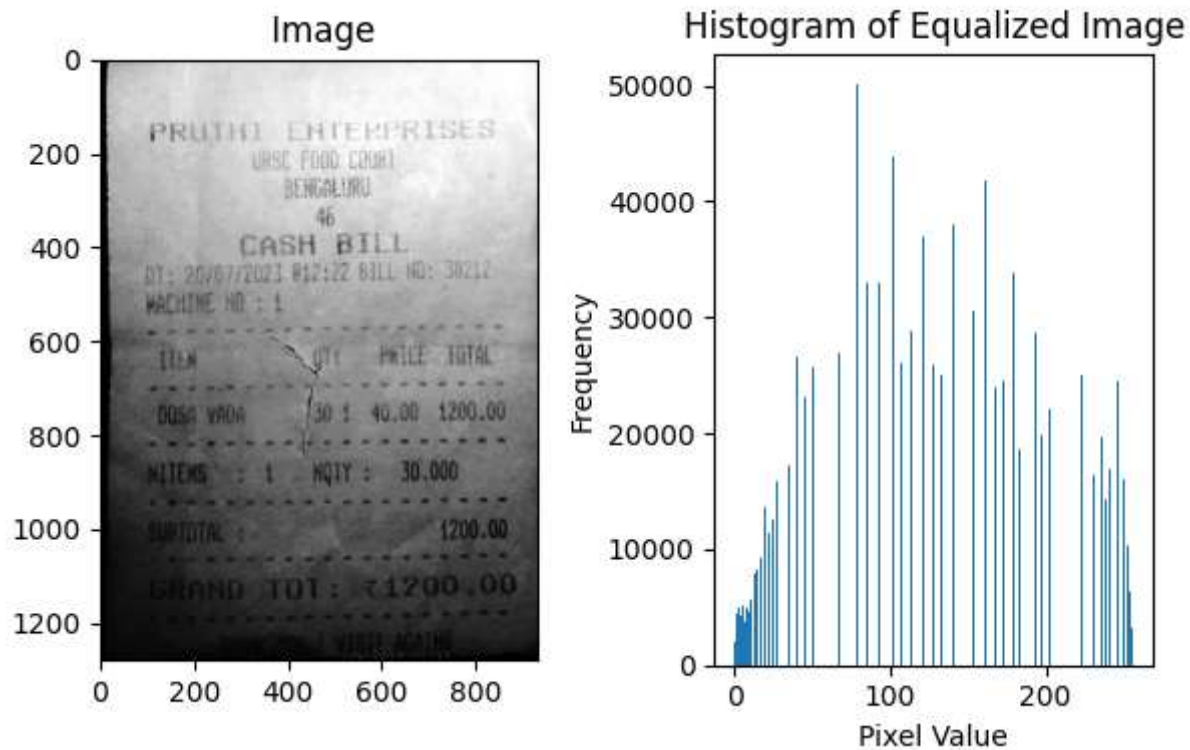
In []: The above histogram of grayscale image looks a bit concentrated towards the middle of the figure,

```
In [8]: plt.figure(figsize=(6, 4))

plt.subplot(1, 2, 1)
plt.imshow(equalized_image, cmap='gray')
plt.title(' Image')

plt.subplot(1, 2, 2)
plt.hist(equalized_image.ravel(), 256, [0, 256])
plt.title('Histogram of Equalized Image')
plt.xlabel('Pixel Value')
plt.ylabel('Frequency')

plt.tight_layout() # Ensures proper spacing between subplots
plt.show()
```



Histogram equalization method distributed the pixel intensity values further to get a more flattened histogram of equalized image.

Result comparison

```
In [13]: plt.figure(figsize=(6, 4))

plt.subplot(1, 2, 1)
plt.imshow( grayscale_image, cmap='gray')
plt.title(' Image')
plt.axis('off')

plt.subplot(1, 2, 2)
plt.imshow( equalized_image, cmap='gray')
plt.title(' Equalized Image')
plt.axis('off')

plt.tight_layout() # Ensures proper spacing between subplots
plt.show()
```

Image



Equalized Image



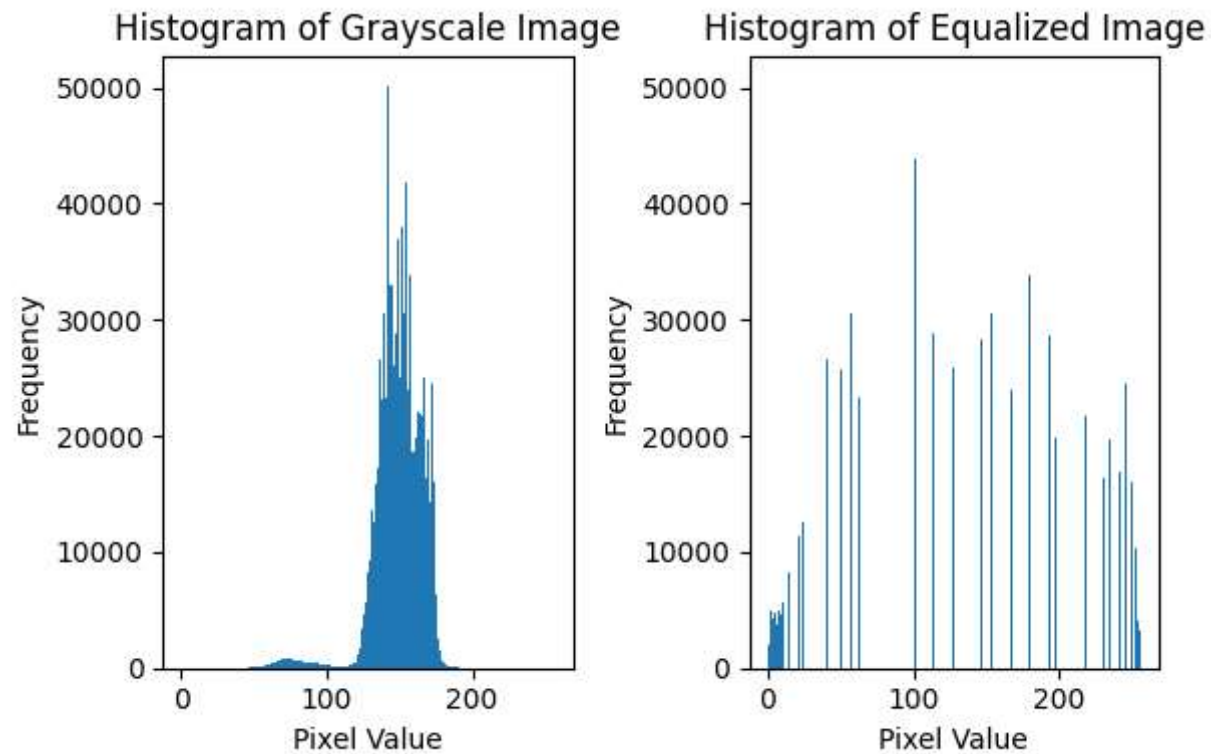
The result is very interesting as it is much clearer than the original image. The right side image is our equalized image.

```
In [16]: plt.figure(figsize=(6, 4))

plt.subplot(1, 2, 1)
plt.hist( grayscale_image.ravel(), 256, [0, 256])
plt.title('Histogram of Grayscale Image')
plt.xlabel('Pixel Value')
plt.ylabel('Frequency')

plt.subplot(1, 2, 2)
plt.hist( equalized_image.ravel(), 256, [0, 256])
plt.title('Histogram of Equalized Image')
plt.xlabel('Pixel Value')
plt.ylabel('Frequency')

plt.tight_layout() # Ensures proper spacing between subplots
plt.show()
```



The histogram of the result was flatter than the histogram of the original image, showing a better distribution of pixel intensity values across the image.

Conclusion : Histogram equalization method transformed the intensity values in a way that will make the histogram look flatter in the resulting equalized image. In other words, histogram equalization is a method that adjusts image intensities in order to enhance the contrast of the image.