Histogram Equalization

Introduction:

Histogram Equalization is a technique in image processing that improves the contrast of an image by spreading out the most frequent intensity values. It modifies the image so that its histogram (distribution of pixel intensities) becomes more uniform, making details in both dark and bright areas more visible.

Histogram Equalization Uses:

- To enhance the visibility of details in an image, especially in images with poor lighting or low contrast.
- To improve the appearance of medical images, satellite images, and other images where detail is important.
- To balance out light and dark regions of an image for better visual interpretation.

Working of Histogram Equalization Work:

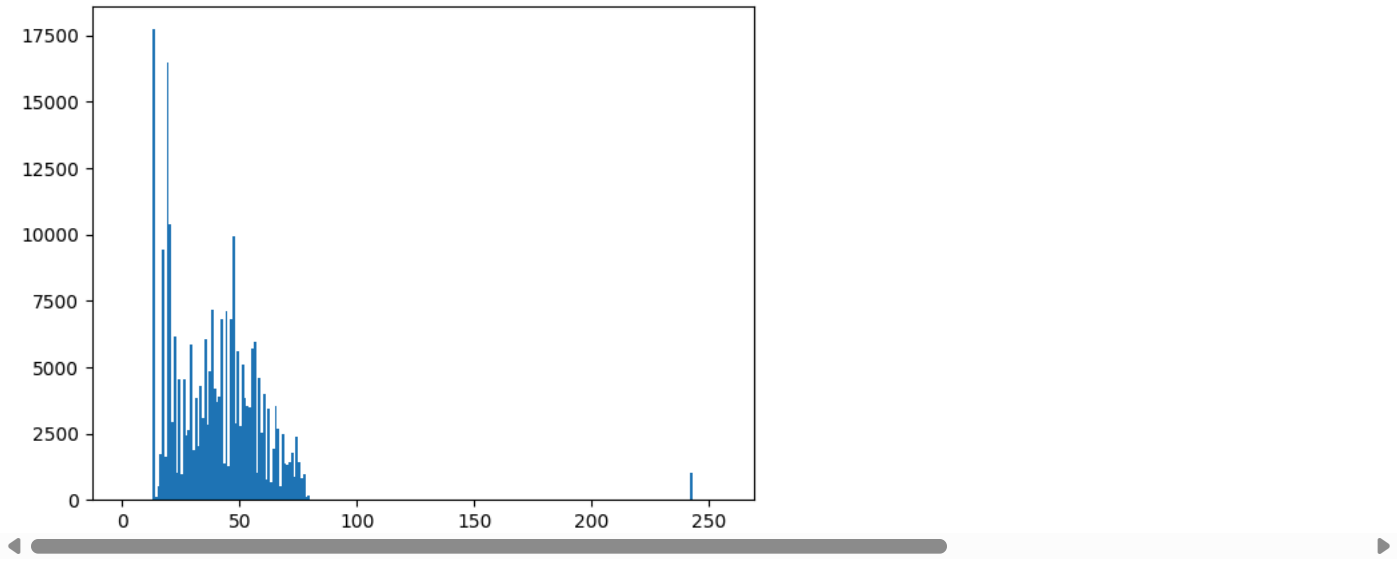
1. **Calculate the Histogram**: Count how many times each intensity value (0-255) appears in the image.
2. **Find the Cumulative Distribution Function (CDF)**: This gives the cumulative sum of histogram values.
3. **Normalize the CDF**: Scale the CDF so that it ranges between 0 and 255.
4. **Map the Old Pixel Values**: Replace each pixel's intensity with its corresponding value from the normalized CDF.
5. **Result**: The new image will have a more balanced distribution of pixel intensities, improving its contrast.

Histogram Equalization makes images clearer by adjusting brightness levels, spreading intensity values more evenly across the image.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
```

```
img = cv2.imread('/content/hist_equal_inputimg.PNG',0)
plt.hist(img.ravel(),256,[0,256])
plt.show()
```

```
<ipython-input-2-4ba807474e96>:2: MatplotlibDeprecationWarning: Passing the range parameter of hist() positionally is deprecated since M
  plt.hist(img.ravel(),256,[0,256])
```

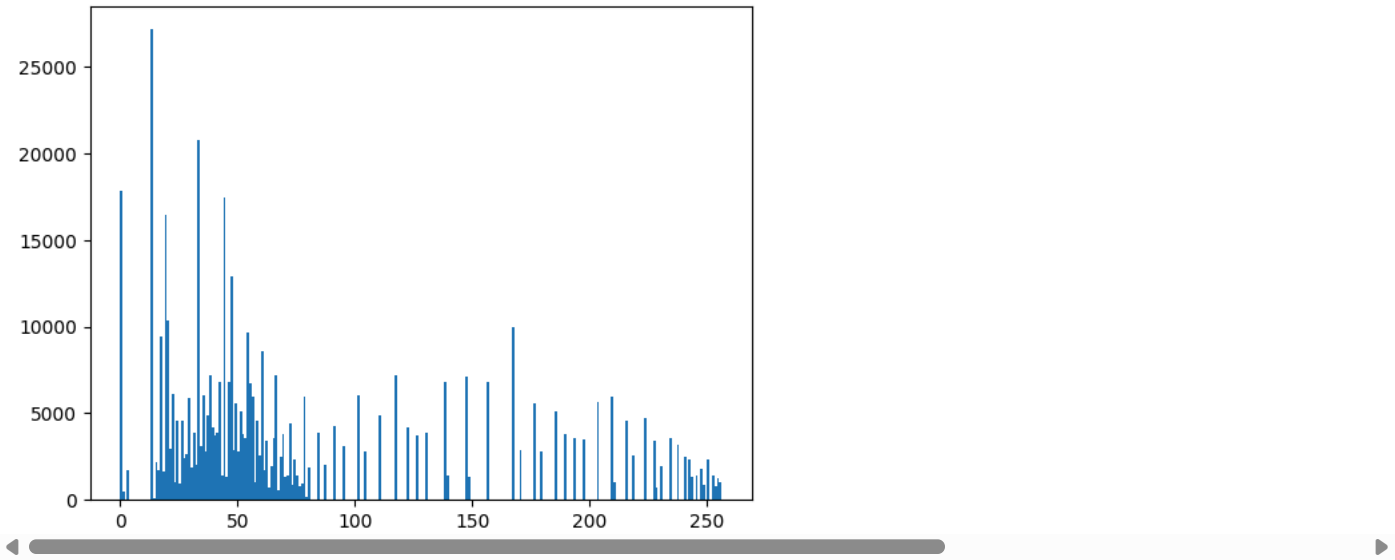



```
equ = cv2.equalizeHist(img)  # enhances the contrast of the image by redistributing the pixel intensity
                             #values more evenly across the available range (0 to 255 for 8-bit images).


output_hist = np.hstack((img,equ))  ## display images side by side – super useful when comparing the original image (img)
                             ## and the histogram-equalized image (equ).
```

```
plt.hist(output_hist.ravel(),256,[0,256])
plt.show()
```

```
<ipython-input-4-dd70c516841d>:1: MatplotlibDeprecationWarning: Passing the range parameter of hist() positionally is deprecated since M
  plt.hist(output_hist.ravel(),256,[0,256])
```




```
cv2.imwrite('outputhist.png',output_hist)
# save an image to computer using OpenCV.
```

```
True
```