Experiment 7

Shashwat Sha
60004220126
BE comps C22

Aim : Histogram equalization for an image.

Theory : Histogram equalization is an image proces techniques used to enhance the contrast of a image by redistributing its pixels intensity valu

→ Principle

The basic idea behind histogram equalization is the mapping of the pixel intensities of an image so that they follow a uniform distribution.

→ Equalization process.

The steps for performing histogram equalization can be summarized as:

1) Calculate the histogram of the given image.
2) Compute the cumulative distribution function (CDF based on the histogram.
3) Normalize CDF, to span the full range of pixel intensities (0 to 255)
4) Use of normalized CDF, to map the original pixel intensities to new values creating the equalised image.

→ Benefits and Applications - It enhances the visibility of details in an image by making the intensity distribu more uniform, which can significantly improve image clarity in situations where there are shows

FOR EDUCATIONAL USE

Scanned with CamScanner

Conclusion - In this experiment, By redistributing the intensity values, we transformed a low contrast image into one with improved detail and visibility

FOR EDUCATIONAL USE

Scanned with CamScanner

NAME: Shashwat Shah SAP ID: 60004220126 DIV/BATCH:C22

DIGITAL SIGNAL PROCESSING (DSP)
EXPERIMENT 07

AIM: To implement Histogram Equalization for an Image.

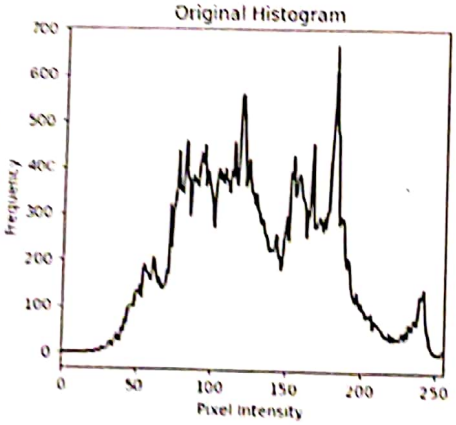
CODE:

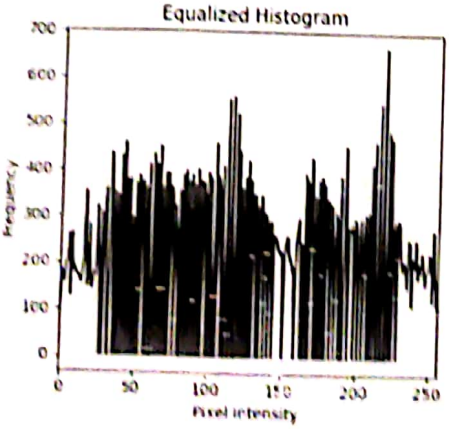
```
import cv2
import numpy as np
import matplotlib.pyplot as plt

def histogram_equalization(image_path):
    image = cv2.imread(image_path, cv2.IMREAD_GRAYSCALE)

    # Calculate the histogram of the original image original_histogram, bins =
    np.histogram(image.flatten(), bins=256,
range=[0, 256])

    # histogram equalization flat_image =
    image.flatten()
    histogram = original_histogram / original_histogram.sum() cdf = histogram.cumsum()
    cdf_normalized = np.round(cdf * 255).astype(np.uint8) equalized_image =
    cdf_normalized[flat_image].reshape(image.shape)

    # Calculate the histogram of the equalized image equalized_histogram, bins_eq
    =
np.histogram(equalized_image.flatten(), bins=256, range=[0, 256])

    # Display the original image, its histogram, the equalized image, and its histogram
    plt.figure(figsize=(8, 8))

    # Original Image and Histogram plt.subplot(2, 2,
    1) plt.title('Original Image') plt.imshow(image,
    cmap='gray') plt.axis('off')

    plt.subplot(2, 2, 2) plt.title('Original Histogram')
    plt.plot(original_histogram, color='black') plt.xlim([0, 256])
    plt.xlabel('Pixel Intensity')
```

Scanned with CamScanner

```
    plt.ylabel('Frequency')

    # Equalized Image and Histogram
    plt.subplot(2, 2, 3)
    plt.title('Equalized Image')
    plt.imshow(equalized_image, cmap='gray')
    plt.axis('off')

    plt.subplot(2, 2, 4)
    plt.title('Equalized Histogram')
    plt.plot(equalized_histogram, color='black')
    plt.xlim([0, 256])
    plt.xlabel('Pixel Intensity')
    plt.ylabel('Frequency')

    plt.tight_layout()
    plt.show()


image_path = 'image1.jpg'
histogram_equalization(image_path)
```

OUTPUT:





Scanned with CamScanner