Computer Vision

Home Work 3

Name – ROHIT DAS Student ID - 61047086s

Project – Histogram Equalization.

Language and library used: Python, OpenCV, Numpy, Matplotlib

Description: This program will perform the following functions while executing lena.bmp image file:

1. Original image and it's histogram(All in Grayscale)
2. Image with intensity divided by 3 and its histogram
3. Image after applying histogram equalization to (2) and its histogram

Parameters: None. Please Copy-paste the image path inside the program.

Algorithms Used –

Part 1: Original image and its histogram –

This is a simple code showing the image and show it's histogram

Principal Code:

```
def show_original_image():
    image_file = r"C:\Users\Rohit\Desktop\Fall 2021 NTNU\Computer Vision
NTU\Chapter-2\Homework\lena.bmp"
    image = cv2.imread(image_file, 0)  # Read Image in Grayscale
    cv2.imshow("Original Image",image)
    cv2.waitKey(0)


def show_histogram(image):
    image_histogram = np.zeros([256], np.int32)
    for h_row in range(0, image.shape[0]):
        for w_col in range(0, image.shape[1]):
            image_histogram[image[h_row, w_col]] += 1
    # Creating histogram
    plt.plot(image_histogram)
    plt.title("Histogram of Image")
    plt.xlabel("Intensity")
    plt.ylabel("Pixels")
    plt.show()
```

Part 2: Image with intensity divided by 3 and its histogram – Divide each pixel value with 3 and show the image and histogram

Principal Code:

```
def dark_image_converter(image):
    dark_image = np.copy(image)
    for h_pixel in range(image.shape[0]):
        for w_pixel in range(image.shape[1]):
            dark_image[h_pixel,w_pixel] = image[h_pixel,w_pixel]//3
    cv2.imshow("Dark Image",dark_image)
    cv2.waitKey(0)
def show_histogram(image):
    image_histogram = np.zeros([256], np.int32)
    for h_row in range(0, image.shape[0]):
        for w_col in range(0, image.shape[1]):
            image_histogram[image[h_row, w_col]] += 1
    # Creating histogram
    plt.plot(image_histogram)
    plt.title("Histogram of Image")
    plt.xlabel("Intensity")
    plt.ylabel("Pixels")
    plt.show()
```

Part 3: Image after applying histogram equalization to (2) and its histogram –

Here I used the basic algorithm of equalizing the histogram.

- Get the Dark image histogram
- Then using the formula mentioned in the ppt.

$$S_k = 255 \sum_{j=0}^{k} \frac{n_j}{n}$$

Where k is the pixel with intensity j.
N is the total number of pixels.

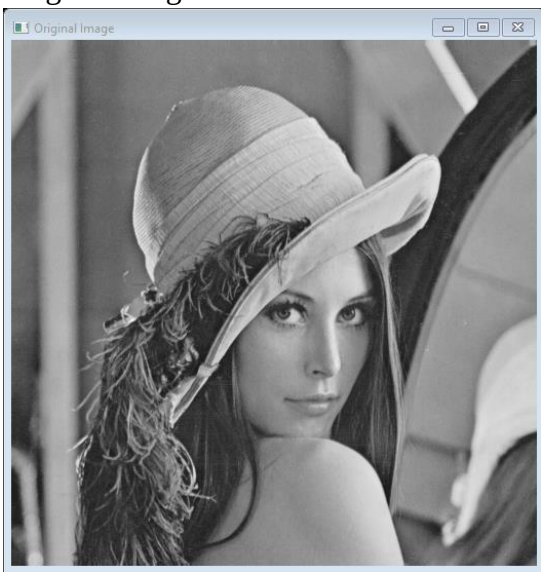
Principal Code –

```
def histogram_equalization_image(D_hist,D_image):
    height, width = D_image.shape
    histogram_equalized_image = np.copy(D_image)
    transformationTable = np.zeros(256)
# Following Histogram Equalization method shown in class s = T(r)
    for items in range(len(transformationTable)):
        intensity_sum = np.sum(D_hist[0:items + 1])
        transformationTable[items] = 255 * intensity_sum / height / width
    for h_row in range(0, D_image.shape[0]):
        for w_col in range(0, D_image.shape[1]):
            histogram_equalized_image[h_row, w_col] =
transformationTable[D_image[h_row,w_col]]
    cv2.imshow ("Equalized Image", histogram_equalized_image)
    cv2.waitKey(0)
```

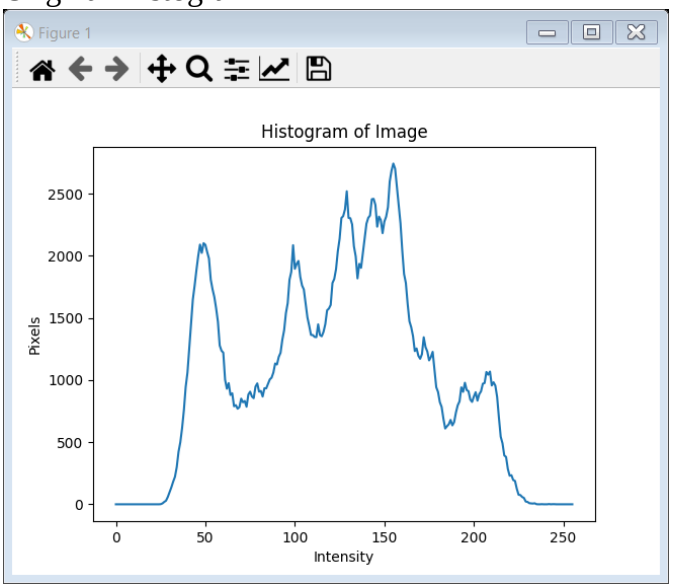
```python
def show_histogram(image):
    image_histogram = np.zeros([256], np.int32)
    for h_row in range(0, image.shape[0]):
        for w_col in range(0, image.shape[1]):
            image_histogram[image[h_row, w_col]] += 1
    # Creating histogram
    plt.plot(image_histogram)
    plt.title("Histogram of Image")
    plt.xlabel("Intensity")
    plt.ylabel("Pixels")
    plt.show()
```

Example:

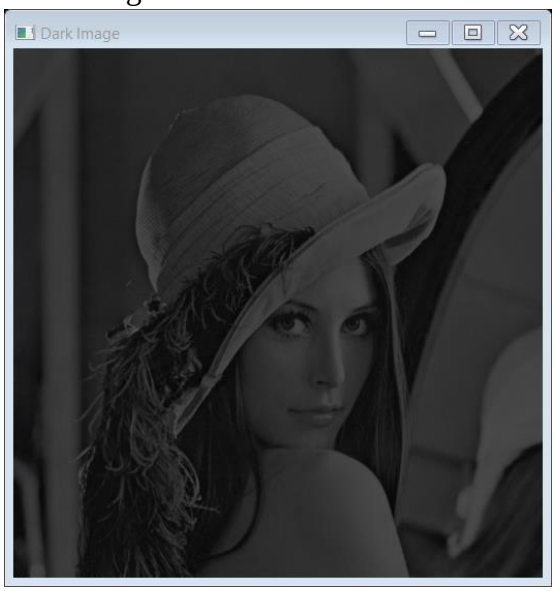
- Original image



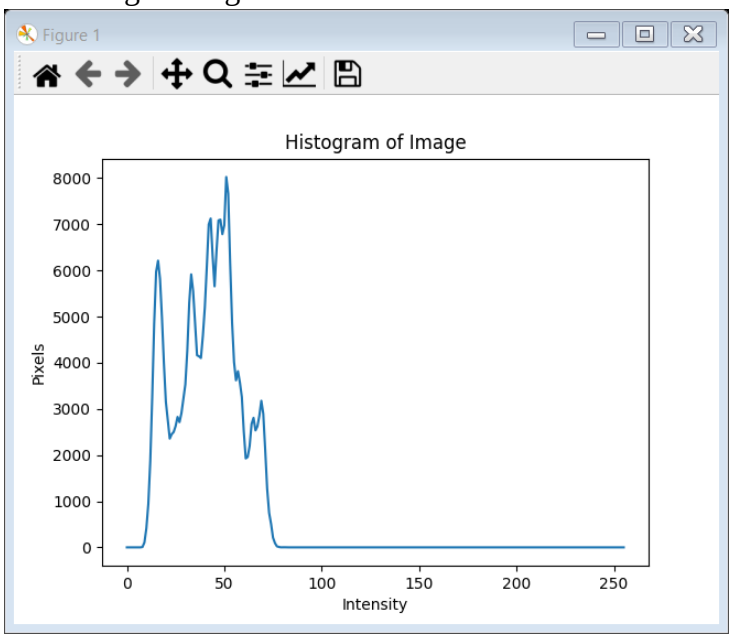
- Original Histogram



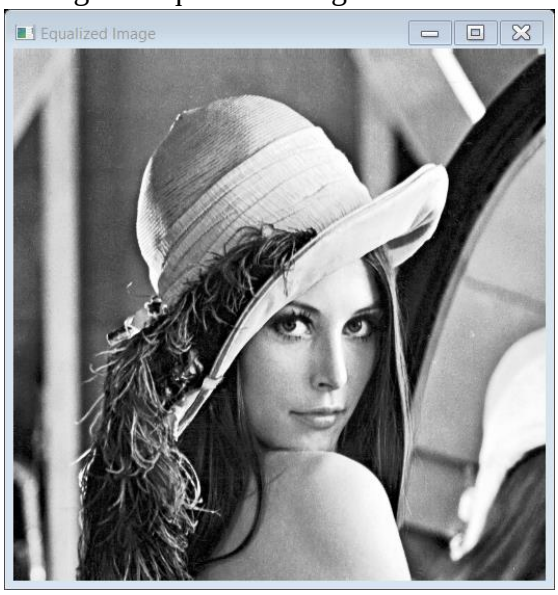
- Dark Image



- Dark image histogram



- Histogram Equalized image



- Histogram Equalized