

HW 3: Histogram Equalization

Source Code

All questions are written in Python code, please refer to the file “main.py”.

All images will be stored in the folder “res” (automatically create a new folder).

In accordance with the **FAQ** of course website:

- All parts of the question are written from scratch, except for plotting image and bar chart

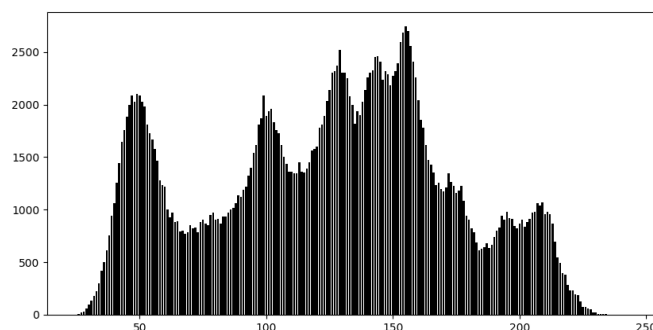
Answer

1. Original image and its histogram

Algorithm:

- 1) Count pixels and store in dictionary
- 2) Unpack keys and values of dictionary and display histogram

```
data = {}
for i in range(height):
    for j in range(width):
        if img[i][j] not in data:
            data[img[i][j]] = 0
        data[img[i][j]] += 1
pixel = list(data.keys())
count = list(data.values())
fig = plt.figure(figsize=(10,5))
plt.bar(pixel, count, color='black')
plt.savefig('res/original_histogram.png')
cv2.imwrite('res/original_lena.png', img)
```

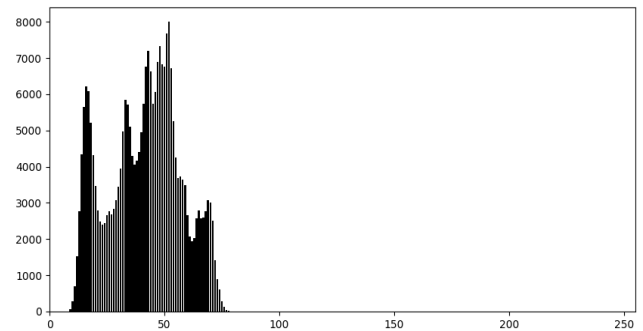
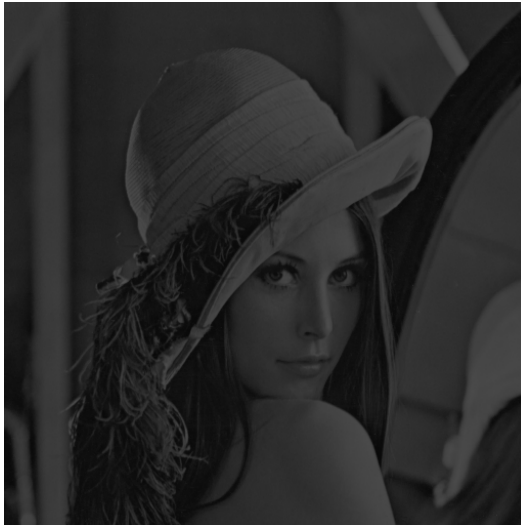


2. Image with intensity divided by 3 and its histogram

Algorithm:

- 1) For each pixel, divide by 3 and do roundings
- 2) Plot histogram using algorithm number 1

```
imgdiv3 = []
for i in range(height):
    for j in range(width):
        imgdiv3.append(round(img[i][j] / 3))
imgdiv3 = np.asarray(imgdiv3).reshape(height, width)
data = {}
for i in range(height):
    for j in range(width):
        if imgdiv3[i][j] not in data:
            data[imgdiv3[i][j]] = 0
        data[imgdiv3[i][j]] += 1
pixel = list(data.keys())
count = list(data.values())
fig = plt.figure(figsize=(10,5))
plt.xlim([0, 255])
plt.bar(pixel, count, color='black')
plt.savefig('res/div3_histogram.png')
cv2.imwrite('res/div3_lena.png', imgdiv3)
```



3. Image after applying histogram equalization to (b) and its histogram

Algorithm:

- 1) Count sum of pixels by pixel value
- 2) Calculate cumulative sum multiplied by 255 for each pixel
- 3) Replace pixels of original image with the corresponding pixels of the above calculation
- 4) Display histogram using the algorithm number 1

```
equalized_array = [0] * 256
hist = np.bincount(imgdiv3.flatten(), minlength=256)
totalpixel = height * width
temp = 0
for i in range(256):
    temp += (hist[i] / totalpixel)
    equalized_array[i] = round(255 * temp)
for i in range(height):
    for j in range(width):
        imgdiv3[i][j] = equalized_array[imgdiv3[i][j]]
cv2.imwrite("res/equalized_image.bmp", imgdiv3)
data = {}
for i in range(height):
    for j in range(width):
        if imgdiv3[i][j] not in data:
            data[imgdiv3[i][j]] = 0
        data[imgdiv3[i][j]] += 1
pixel = list(data.keys())
count = list(data.values())
fig = plt.figure(figsize=(10,5))
plt.bar(pixel, count, color='black')
```

