d11922022 資工所 楊寓鈞

Environment

- python3.9
- Pillow 10.0.0, numpy 1.25.2, pandas 2.0.3

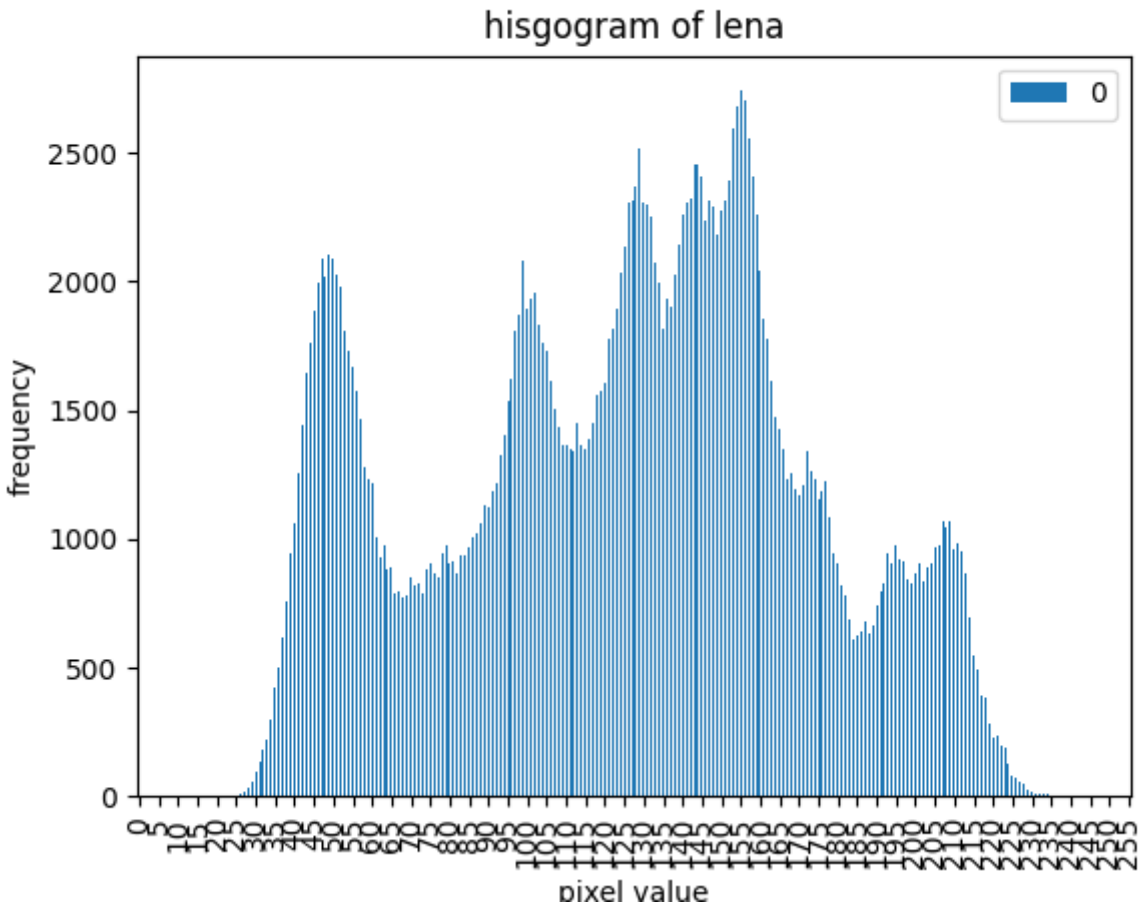
I/O

```
from PIL import Image
import numpy as np
import copy

img = Image.open('./lena.bmp') # load lena.bmp
img_array = np.array(img) # pixel content saved in np.array
width, height = img_array.shape # get `width` and `height`
img_list = img_array.tolist() # transform pixel content into list
```

a. histogram

```
import matplotlib.ticker as ticker
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline

result = img_array.copy()
histogram = dict()



for y in range(height):
        for x in range(width):
                bin = histogram.get(result[y][x], 0)
                histogram[result[y][x]] = bin + 1

histogram = dict(sorted(histogram.items(), key=lambda x: x[0]))
df = pd.DataFrame({k:[v] for k,v in histogram.items()}).T

ax = df.plot.bar(title='hisgogram of lena', xlabel='pixel value',
ylabel='frequency')
ax.xaxis.set_major_locator(ticker.MultipleLocator(base=5))
plt.savefig('histogram.png')
```

1. traverse through all pixels of the binarized image
2. create an dict to record gray level and it's corresponding pixel amounts
3. take pixel's gray level as key, accumulate the counts of the pixels at that gray level as key's value
4. plot histogram via pandas api

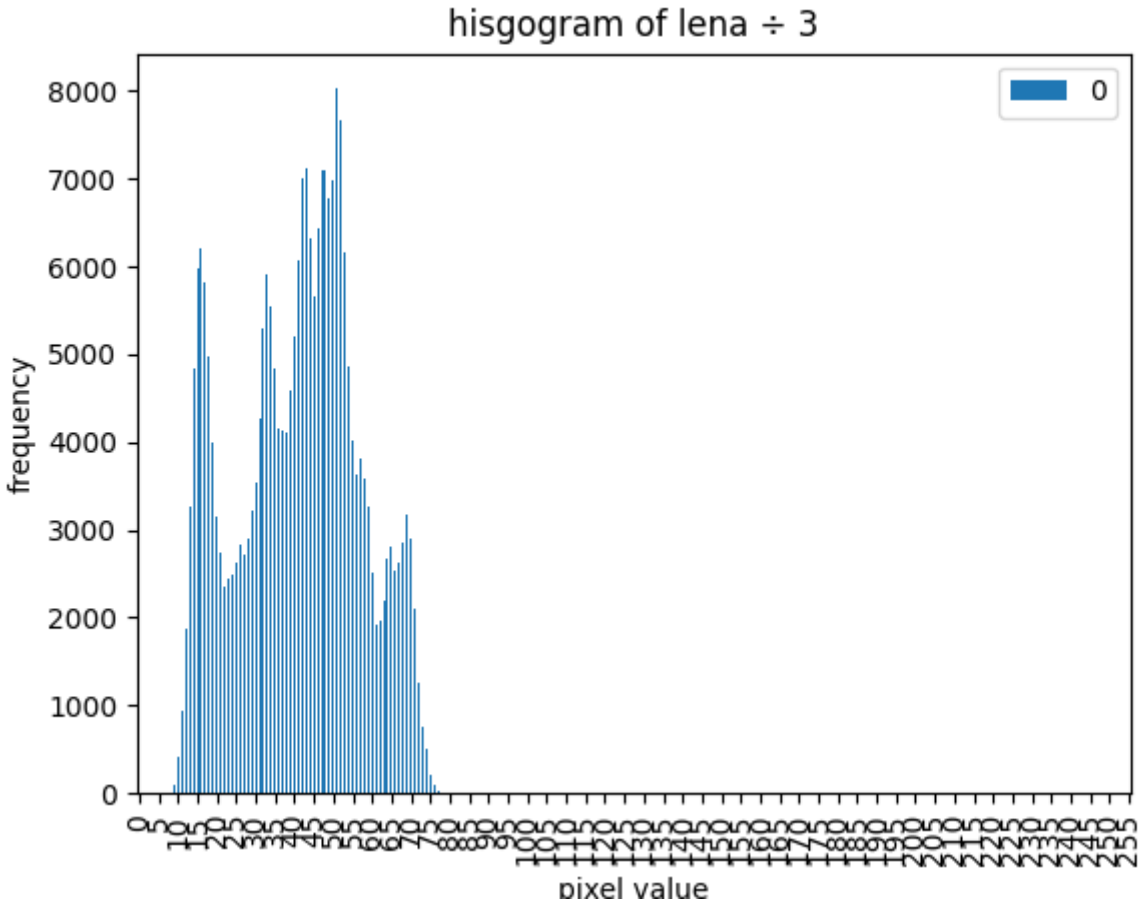


b. image intensity div by 3, then histogram

```
result = img_array.copy()

histogram = {i:0 for i in range(0, 256)}
for y in range(height):
        for x in range(width):
                result[y][x] = result[y][x]//3
                bin = histogram.get(result[y][x], 0)
                histogram[result[y][x]] = bin + 1



histogram = dict(sorted(histogram.items(), key=lambda x: x[0]))
df = pd.DataFrame({k:[v] for k,v in histogram.items()}).T
```

```
ax = df.plot.bar(title='hisgogram of lena ÷ 3', xlabel='pixel value',
ylabel='frequency')

ax.xaxis.set_major_locator(ticker.MultipleLocator(base=5))
plt.savefig('histogram_div3.png')
```

1. simply divide the pixel value by 3, and then perform the histogram in the same way as in part a.
2. you can now observe that all the bins have shifted to the left of the spectrum.




```
img_ = Image.fromarray(np.array(result, dtype='uint8'), mode='L')
img_.save('./lena_div3.bmp')
```

2. the image obtained by dividing the pixel values by 3 appears darker than the original.

c. histogram equalization on b.

```
result = img_array.copy()

# remove value = 0's key-value pair
for k, v in list(histogram.items()):
        if v == 0:
                del histogram[k]

# sort by key
histogram = sorted(histogram.items(), key = lambda x: x[0])
min_bin, min_bin_count = histogram[0]
histogram = dict(histogram)

# direct map min_bin to 0
equalized = {i:0 for i in range(0, 256)}
equalized = {0:min_bin_count}
cdf = 0
for i, (k, v) in enumerate(histogram.items()):
```

```
# min_bin has already mapped to 0
        if i == 0:
                continue
        cdf += v
        hv = round(
                ((cdf - min_bin_count)/(height*width - min_bin_count))*255
        )
        equalized[hv] = v

equalized = dict(sorted(equalized.items(), key=lambda x: x[0]))
df = pd.DataFrame({k:[v] for k,v in equalized.items()}).T
ax = df.plot.bar(title='equalized hisgogram of lena ÷ 3', xlabel='pixel
value', ylabel='frequency')

ax.xaxis.set_major_locator(ticker.MultipleLocator(base=5))
plt.savefig('equalized_histogram_div3.png')
```

1. following to histogram equalization formula:

$$h(v) = \text{round}\left(\frac{\text{cdf}(v) - \text{cdf}_{\text{min}}}{\text{height} \times \text{width}} \times 255\right)$$

equalize the histogram to make the bin spread across the spectrum uniformly