Homework 3

Histogram Equalization

Programming language: Python 3.7.3

Library used for this homework:

- Numpy
- OpenCv: to read and write the image file

Image info: lena.bmp [512(width),512(height),3(RGB)]

Code explanation:

(a)

Draw a histogram with the brightness value of all pixel.

Use matplotlib.pyplot to draw the histogram.

```
def drawHistogram():
    histData = []
    for r in range(rows):
        for c in range(columns):
            histData.append(lena[r,c,2])

    n, bins, patches = plt.hist(x=histData, bins=256)
    plt.title(r'Histogram')
    plt.savefig("answer1.png")
    plt.show()
```

(b)

Divided all the pixels' brightness value by 3. Save the result into each RGB channel.

Use a list "histDivided" to save all the value, and draw the histogram using this list.

```
def dividedHistogram():
    for r in range(rows):
        for c in range(columns):
            lenaCopy[r,c,0] = int(math.floor(lena[r,c,0]/3))
            lenaCopy[r,c,1] = int(math.floor(lena[r,c,0]/3))
            lenaCopy[r,c,2] = int(math.floor(lena[r,c,0]/3))
            histDivided.append(lenaCopy[r,c,0])
```

(C)

Calculate the PMF and CDF (cumulative distribution function) value of the darken image from part b. PMF was calculated from the hisDivided list from part B.

Histogram Equalization: https://en.wikipedia.org/wiki/Histogram_equalization

```
def equalization():
    histData = []
    pmf = np.zeros(256, dtype = int)
    cdf = np.zeros(256, dtype = int)
    cdfMin = 1000
    for count in range(len(histDivided)):
        pmf[histDivided[count]] += 1
```

```
previous = pmf[0]
for count in range(256):
    if pmf[count] != 0:
        if cdfMin > pmf[count]:
            cdfMin = pmf[count]
        cdf[count] = previous + pmf[count]
        previous = cdf[count]
```

Calculate the normalized value of each pixel using the CDF values.

The general histogram equalization formula:

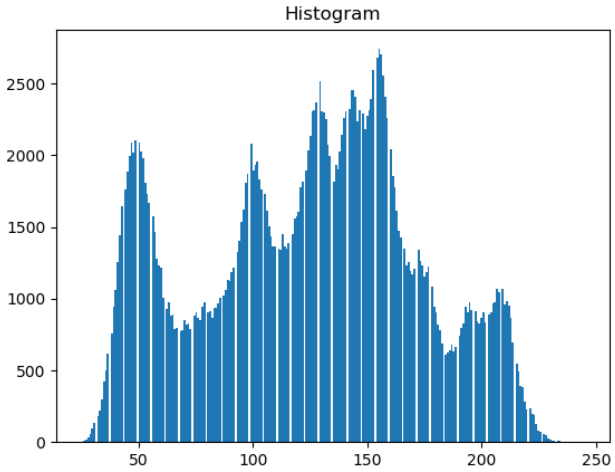
$$h(v) = \text{round}\left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L-1)\right)$$

```
for r in range(rows):
    for c in range(columns):
        temp = lenaCopy[r,c,0]
        for i in range(3):
            lenaCopy[r,c,i]= math.floor((cdf[temp] - cdfMin)/(rows*columns-cdfMin)*255)
        histData.append(lenaCopy[r,c,0])
```

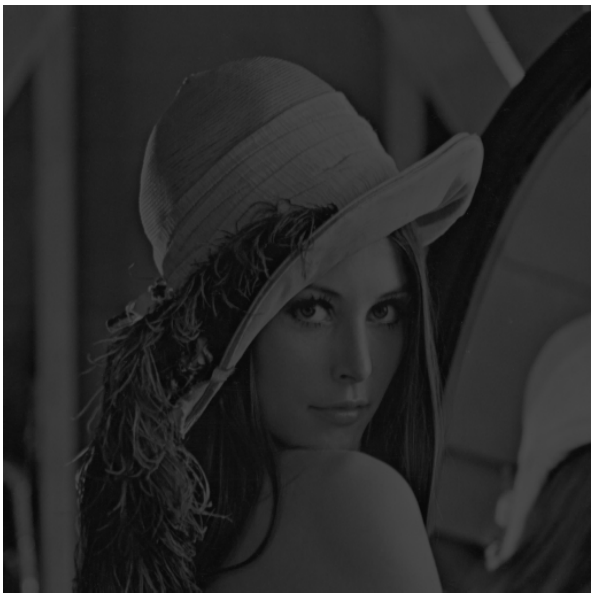
Result:

(a-1)

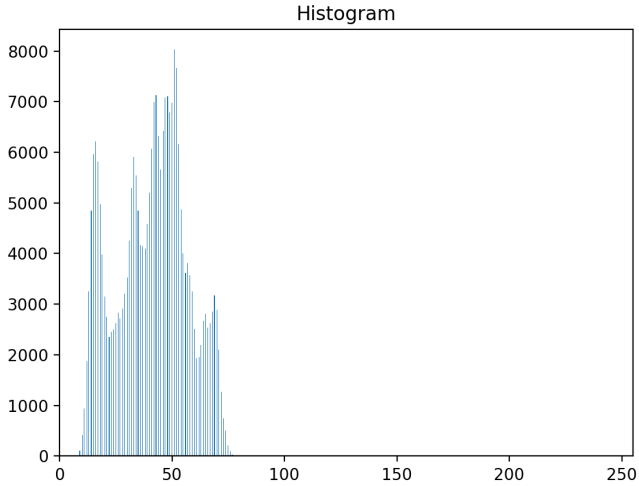
(a-2)


Histogram
2500
2000
1500
1000
500
0
50
100
150
200
250

(b-1)

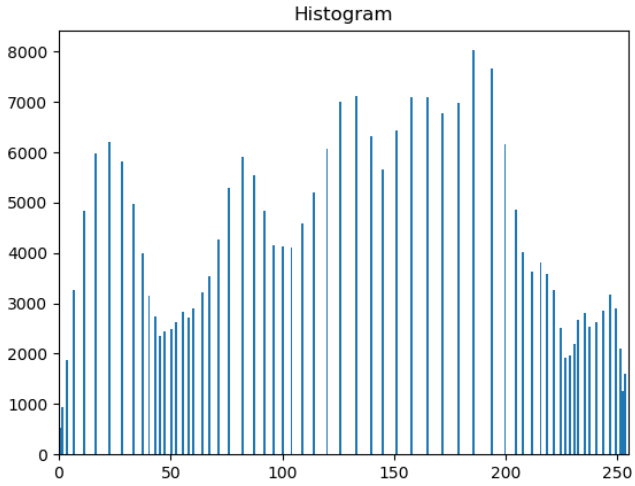


(b-2)


Histogram
8000
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250

(c-1)

(c-2)


Histogram
8000
7000
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250