

# 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.

```
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.

```
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. [https://en.wikipedia.org/wiki/Histogram\\_equalization](https://en.wikipedia.org/wiki/Histogram_equalization)

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

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

Calculate the normalized value of each pixel using the previous.

```

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] - pmf[temp]) / (rows*columns-1)*255)
        histData.append(lenaCopy[r,c,0])

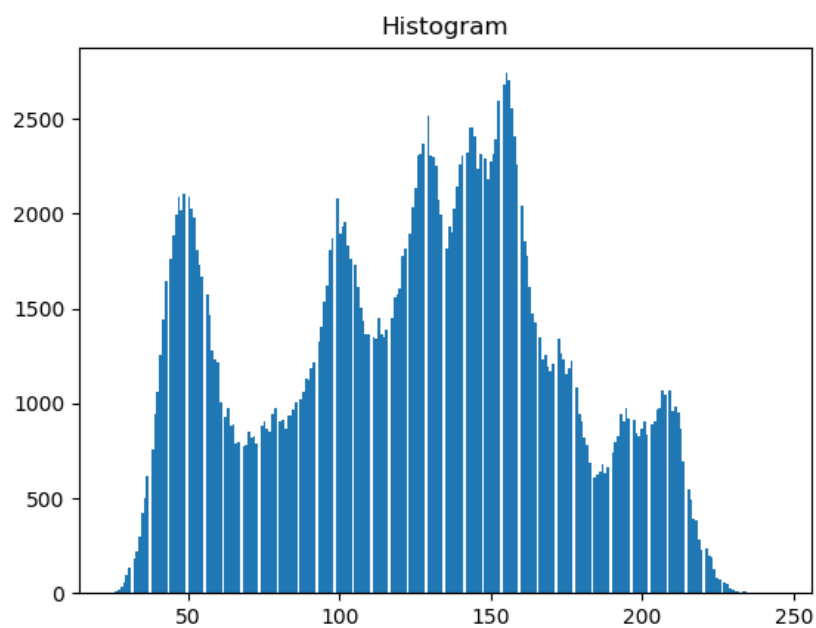
```

## Result:

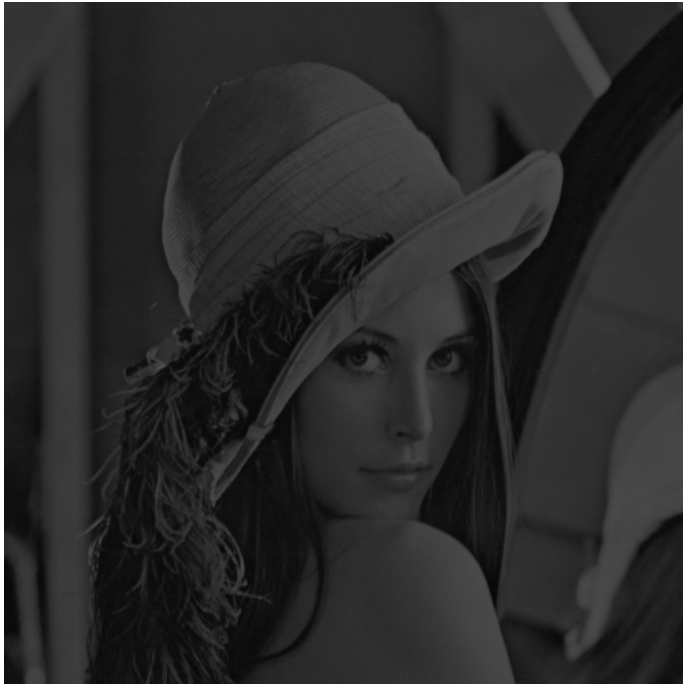
(a-1)



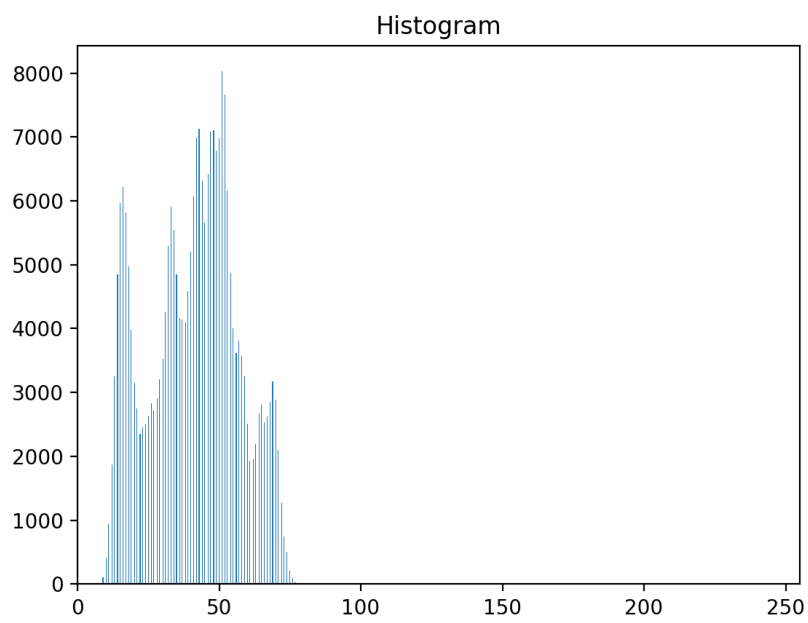
(a-2)



(b-1)



(b-2)



(c-1)



(c-2)

