Digital Image Processing – Homework 1

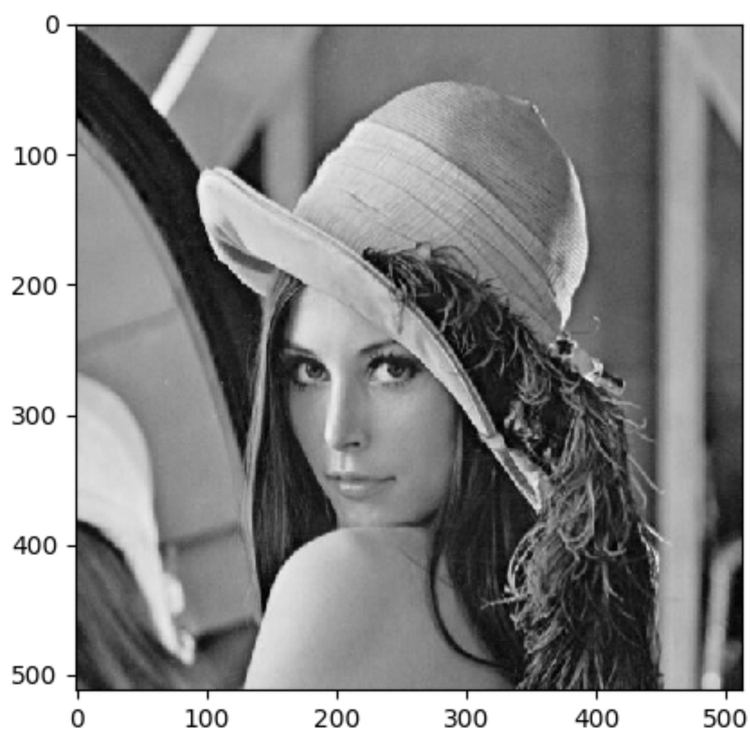
Jacob Taylor Cassady
September 3, 2019

Reduce the representation of the image lenna.jpg from 8 bits per pixel using the following substitution table:

F	G
0-15	7
16-31	23
32-47	39
…	…
240-256	247

```
Initial image:
[[126 152 171 ... 162 163 164]
 [124 155 176 ... 163 164 164]
 [121 154 175 ... 163 164 165]
 ...
 [ 94  96  96 ...  30  26  23]
 [103  99  96 ...  31  26  22]
 [104  96  93 ...  32  26  22]] (512, 512)
Quantized image:
[[119 151 167 ... 167 167 167]
 [119 151 183 ... 167 167 167]
 [119 151 167 ... 167 167 167]
 ...
 [ 87 103 103 ...  23  23  23]
 [103 103 103 ...  23  23  23]
 [103 103  87 ...  39  23  23]] (512, 512)
```

Apply imshow to image g



Compute the average error between f and g.

Mean Squared Error between intital image and quantized image = 21.998451232910156

Source Code

```
import numpy as np
import matplotlib.pyplot as plt
import imageio
import os
import sys

def mse(array1, array2, axis=None):
    return ((array1 - array2)**2).mean(axis=axis)

class Quantizer():
    @staticmethod
    def quantize_image(image_array):
        quantized_image = np.copy(image_array)

        for row_index, row in enumerate(image_array):
            for column_index, pixel in enumerate(row):
                if pixel >= 0 and pixel < 16:
                    quantized_image[row_index, column_index] = 7
                elif pixel >= 16 and pixel < 32:
                    quantized_image[row_index, column_index] = 23
                elif pixel >= 32 and pixel < 48:
                    quantized_image[row_index, column_index] = 39
                elif pixel >= 48 and pixel < 64:
                    quantized_image[row_index, column_index] = 55
                elif pixel >= 64 and pixel < 80:
                    quantized_image[row_index, column_index] = 71
                elif pixel >= 80 and pixel < 96:
                    quantized_image[row_index, column_index] = 87
                elif pixel >= 96 and pixel < 112:
                    quantized_image[row_index, column_index] = 103
                elif pixel >= 112 and pixel < 128:
                    quantized_image[row_index, column_index] = 119
                elif pixel >= 128 and pixel < 144:
                    quantized_image[row_index, column_index] = 135
                elif pixel >= 144 and pixel < 160:
                    quantized_image[row_index, column_index] = 151
                elif pixel >= 160 and pixel < 176:
                    quantized_image[row_index, column_index] = 167
                elif pixel >= 176 and pixel < 192:
                    quantized_image[row_index, column_index] = 183
                elif pixel >= 192 and pixel < 208:
                    quantized_image[row_index, column_index] = 199
                elif pixel >= 208 and pixel < 224:
                    quantized_image[row_index, column_index] = 215
```

```
            elif pixel >= 224 and pixel < 240:
                quantized_image[row_index, column_index] = 231
            elif pixel >= 240 and pixel < 256:
                quantized_image[row_index, column_index] = 247

        return quantized_image

if __name__ == "__main__":
    args = sys.argv

    if(len(args) != 2):
        print("Command Line Arguments should follow the format:")
        print("python Quantizer.py [relative_image_path]")
    else:
        image_path = sys.argv[1]

        # Read image:
        image = imageio.imread(image_path)
        print("Initial image: ")
        print(image, image.shape)
        # Display image:
        plt.imshow(image, cmap='gray')
        plt.show()

        # Quantize image:
        quantized_image = Quantizer.quantize_image(image)
        print("Quantized image: ")
        print(quantized_image, quantized_image.shape)
        # Display quantized image:
        plt.imshow(image, cmap='gray')
        plt.show()

        # Calculate Mean Squared Error
        print("Mean Squared Error between intital image and quantized image = 
{}".format(mse(image, quantized_image)))
```

2.11

2.11 Start with $s = x + m(y + Nz)$ and

(a) Derive Eq. (2-15).

$x = (S \bmod N) \bmod m$

when $N = 1 \Rightarrow x = S \bmod m$ = Eq. 2-15

(b) Derive equation (2-16)

$$y = \frac{(S \bmod N) - x}{M}$$

when $N = 1 \Rightarrow y = \frac{S - x}{M}$ = Eq. 2-16

2.14

a. Subsets S1 and S2 are not 4-adjacent.

b. Subsets S1 and S2 are 8-adjacent.

c. Subsets S1 and S2 are not m-adjacent

2.15

2.15 $V = \{1\}$

For pixel p_i in set $N_8(p)$ where p and p_i are 8-adjacent.

- Set p_i to 0 if $p_i \in N_8$ and $p_i \notin N_4$
- Set p_i to 1 if $p_i \in N_4$

2.24

2.24 $f(i) = \{2, 3, 8, 20, 21, 25, 31\}$

$\delta\{a f(i) + b f_2(i)\} \neq a\,\delta\{f_1(i)\} + b\{f_2(i)\}$

$\Rightarrow g(a f_1(i) + b f_2(i)) \neq a\{20\} + b\{20\}$

$\Rightarrow \delta(a f_1(i) + b f_2(i)) \neq 20(a+b)$

The median operator is non-linear as $H[a f_1(x,y) + b f_2(x,y)] \neq a H[f_1(x,y)] + b H[f_2(x,y)]$

2.26

2.26 a) $E\{\bar{g}(x,y)\} = f(x,y)$

→ since $\bar{g}(x,y)$ is the average of a set of images $g_i(x,y) = f(x,y) + n_i(x,y)$ and each $n_i(x,y)$ has average 0. The average of $\bar{g}(x,y)$ is $f(x,y)$.

b) $\bar{g}(x,y) = \frac{1}{K} \sum_{i=1}^{K} g_j(x,y)$

$$\sigma^2_{\bar{g}(x,y)} = \left(\frac{1}{K} \sum_{i=1}^{K} \left(f_j(x,y) + n_j(x,y) \right)\right) \sigma^2$$

$$\Rightarrow \sigma^2_{\bar{g}(x,y)} = \frac{1}{K} \sigma^2_{n(x,y)}$$

- The variance of a constant times random variable is equal to the constant squared times the variance
- The variance of the sum of uncorreleted random variables is equal to the sum of the variances of the individual random variables