922 U0610 電腦視覺
Computer Vision

Homework 3

授課教師： 傅楸善 教授

學生系級： 資工所一年級

學生姓名： 姚嘉昇

學生學號： R06922002

I. INTRODUCTION

1.1. Descriptions of Problem

This homework is to do histogram equalization with following rules:

A. Adjust the brightness of lena.bmp to one-third.
B. Do histogram equalization on dark image.
C. Show the histogram of the final image.

1.2. Programming Tools

1.2.1. Programming Language: Python3

1.2.2. Programming IDE: Visual Studio Code

II. METHOD

2.1. Algorithms

2.1.1. Histogram equalization

Step 1. Load image from file (lena.bmp).

Step 2. Get width and height of image.

Step 3. New image with the same size and ‘binary’ format.

Step 4. Put pixels with one-third brightness.

Step 5. Save dark image (dark.bmp).

Step 6. Save histogram of dark image to csv file (dark histogram.csv) and plot it on image (dark histogram.png).

Step 7. Estimate pixel transformation function (s = T(r)) which is referenced from CH3.ppt.

Step 8. Apply pixel transformation function to dark image.

Step 9. Save histogram of equalization image to csv file (histEqu histogram.csv) and plot it on image (histEqu histogram.png).

2.2. Code Fragments

2.2.1. Code fragments of this homework

```python
from PIL import Image
import matplotlib.pyplot as plt
import numpy as np
import csv

# Load image from file.
originalImage = Image.open('lena.bmp')

# Get width and height of image.
width, height = originalImage.size
# print ('width = %d, height = %d' %(width, height))

# New image with the same size and 'grayscale' format.
darkImage = Image.new('L', originalImage.size)

# Process image pixel by pixel.
for c in range(width):
    for r in range(height):
        # Get pixel from original image.
        pixelValue = originalImage.getpixel((c, r))
        # Assign 1/3 pixel value to dark image.
        darkImage.putpixel((c, r), pixelValue // 3)

# Save image to file.
darkImage.save('dark.bmp')

# Create histogram array with zeros.
darkHistogram = np.zeros(256)
```

Figure 2.2.1.1. Code of creating dark image.

```
    # Create histogram array with zeros.
    darkHistogram = np.zeros(256)

    # Process image pixel by pixel.
    for c in range(width):
        for r in range(height):
            # Get pixel from dark image.
            pixelValue = darkImage.getpixel((c, r))
            # Record count in histogram array.
            darkHistogram[pixelValue] += 1

    # Save histogram to csv file.
    csvFile = open('dark histogram.csv', 'w')
    writer = csv.writer(csvFile)
    writer.writerow(darkHistogram)

    # Clear plot.
    plt.gcf().clear()
    # Plot histogram.
    plt.bar(range(len(darkHistogram)), darkHistogram)
    # Save histogram to image file.
    plt.savefig('dark histogram.png')
```

Figure 2.2.1.2. Code of dark image histogram.

```
# Histogram Equalization
# Look up table for transformation.
transformationTable = np.zeros(256)

# Deal with each value (0 ~ 255).
for i in range(len(transformationTable)):
    transformationTable[i] = 255 * np.sum(darkHistogram[0:i + 1]) / width / height
```

Figure 2.2.1.3. Code of pixel transformation function.

```
# New image with the same size and 'grayscale' format.
histEquImage = Image.new('L', originalImage.size)

# Process image pixel by pixel.
for c in range(width):
    for r in range(height):
        # Get pixel from dark image.
        pixelValue = darkImage.getpixel((c, r))
        # Put pixel to histogram equalization image.
        histEquImage.putpixel((c, r), int(transformationTable[pixelValue]))

# Save image to file.
histEquImage.save('histogram equalization.bmp')
```

Figure 2.2.1.4. Code of creating equalization image.

```
74  # Create histogram array with zeros.
75  histEquHistogram = np.zeros(256)
76
77  # Process image pixel by pixel.
78  for c in range(width):
79      for r in range(height):
80          # Get pixel from dark image.
81          pixelValue = histEquImage.getpixel((c, r))
82          # Record count in histogram array.
83          histEquHistogram[pixelValue] += 1
84
85  # Save histogram to csv file.
86  csvFile = open('histEqu histogram.csv', 'w')
87  writer = csv.writer(csvFile)
88  writer.writerow(histEquHistogram)
89
90  # Clear plot.
91  plt.gcf().clear()
92  # Plot histogram.
93  plt.bar(range(len(histEquHistogram)), histEquHistogram)
94  # Save histogram to image file.
95  plt.savefig('histEqu histogram.png')
```

Figure 2.2.1.5. Code of equalization image histogram.

III. RESULTS

3.1. Original Image

Figure 3.1. Original lena.bmp.

3.2. Results of this homework

Figure 3.2.1. Original lena.bmp.

Figure 3.2.2. dark.bmp.

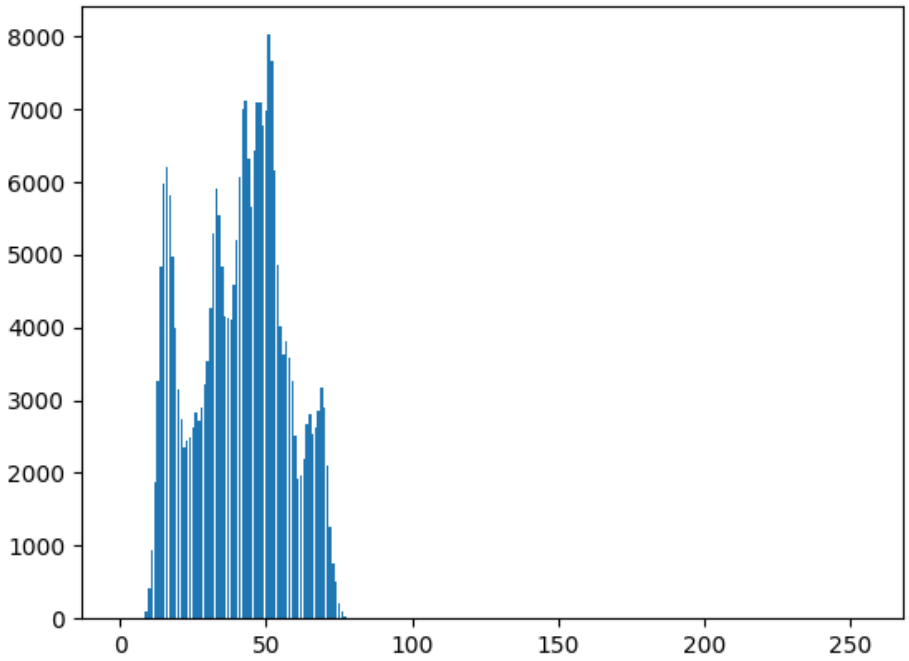

Figure 3.2.3. Histogram of dark.bmp.

Figure 3.2.4. dark.bmp.

Figure 3.2.5. histogram equalization.bmp.

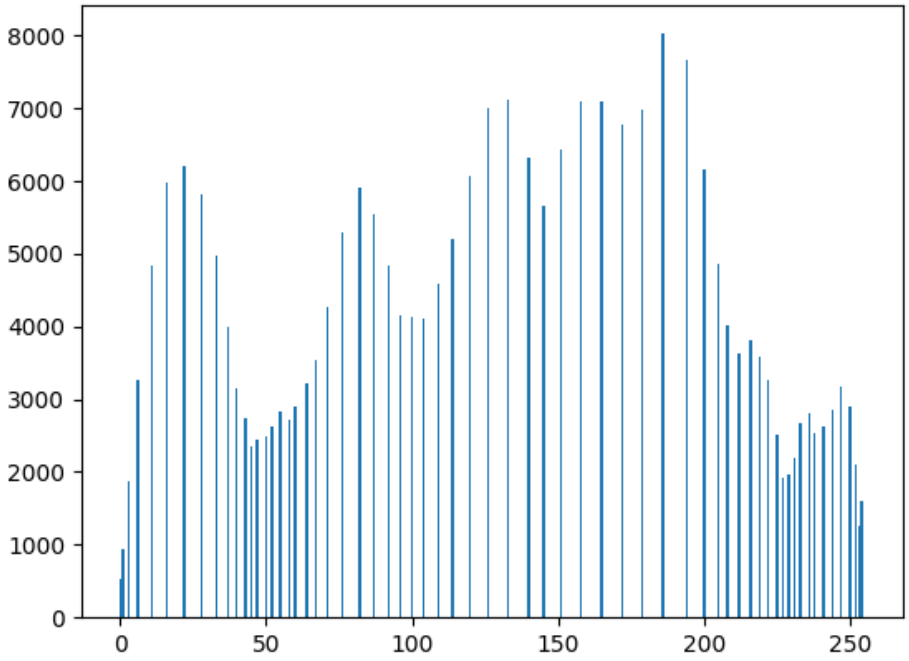

Figure 3.2.6. Histogram of histogram equalization.bmp.