Image and Video Processing Lab
Lab 2: Image Enhancement

Aruna Shaju Kollannur (SC21M111)

Sub: Image and Video Processing Lab **Department**: Avionics(DSP)
Date of Lab sheet: September 7, 2021 **Date of Submission**: September 13, 2021

Question 1: Display the negative of an image (Use mammogram.tif image)

Aim

To create the negative of an image

Discussion

The negative of an image is obtained by subtracting each image pixel from the maximum pixel value of that image. The output image obtained is inverted in pixel intensities, i.e. the darker areas become lighter and lighter areas become darker.

Algorithm

```
Step 1: Start
Step 2: Read the image
Step 3: Obtain the maximum pixel intensity of image
Step 4: Create a copy of read image to be inverted
Step 5: Initialize 2 variables to store the width and height of image
Step 6: Run a nested loop
        Step 6.1: Subtract pixel intensity at position from the maximum pixel intensity value
        obtained.
Step 7: Display the original and inverted image
```

Program Code

```
from PIL import Image

Mammogram_Image = Image.open("mammogram.tif")

# Pixel with maximum intensity in image

Max_pixel = max(Image.Image.getextrema(Mammogram_Image))

Negative_Image = Mammogram_Image.copy()
height, width = Mammogram_Image.size

# Subtracting pixel values from maximum pixel intensity to invert image
```

```
for i in range(width):
    for j in range(height):
        temp = Max_pixel- Mammogram_Image.getpixel((j,i))

        Negative_Image.putpixel((j,i),(temp))

Negative_Image.save("Mammogram_Inverted_Image.tif")

Mammogram_Image.show()
Negative_Image.show()
```

Result

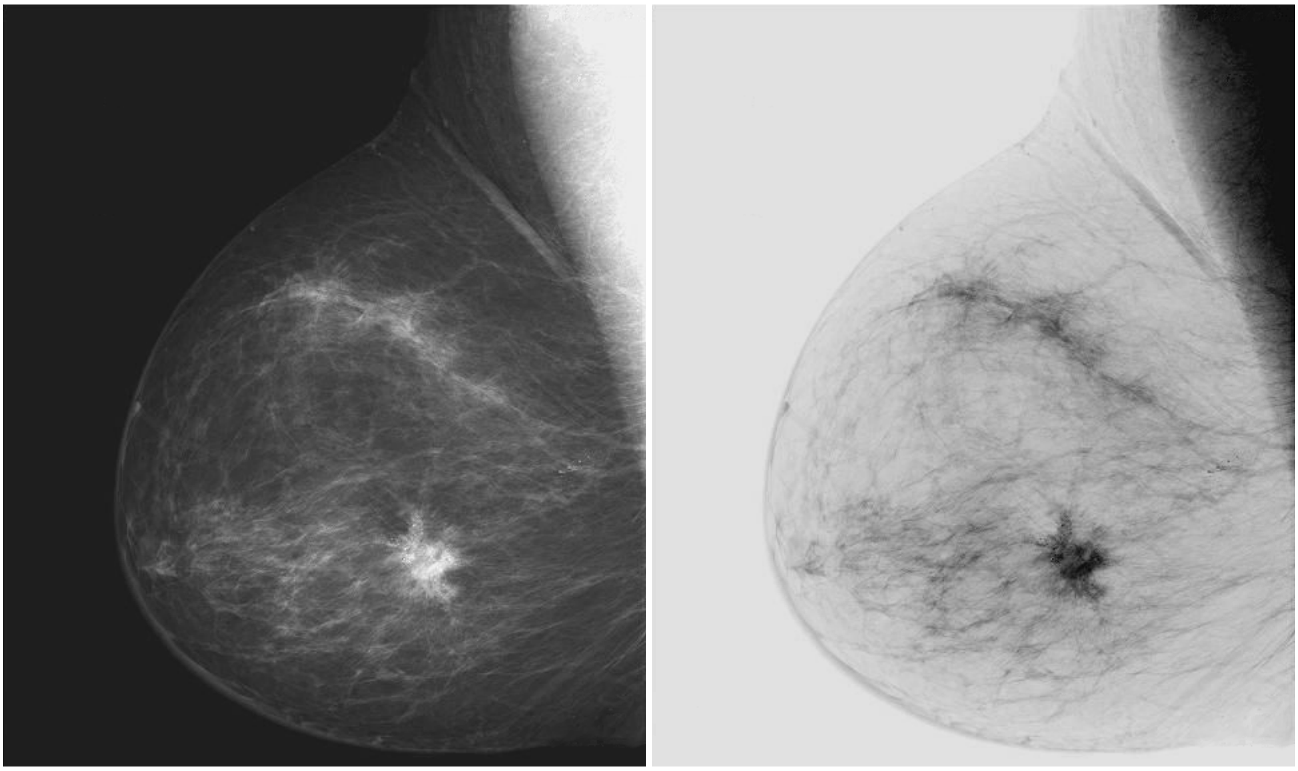

Figure 1: Mammogram : (a)Original (b)Inverted

Inference

Image inversion makes the lighter pixels dark and the darker pictures light. This can be used for highlighting details of an image with darker intensities.

Question 2: Plot the histogram of an image. (Use cameraman.tif image.)

Aim

Plotting the histogram of an image

Discussion

Histogram conveys the frequency distribution of pixel intensities of an image. It is the plot of pixel intensities against their respective count in the image.

Algorithm

```
Step 1: Start
Step 2: Read the image and convert it into gray scale
Step 3: Initialize 2 variables to store the width and height of image
Step 4: Declare an empty list to hold the pixel intensity frequencies.
Step 5: Run a nested loop
    Step 5.1: Run outermost loop to run through pixel intensities (0-255).
    Step 5.2: Initialise a temporary to hold count of pixel intensity as zero.
    Step 5.3: Increment the temporary variable for every match of image pixel
    intensity with outer loop pixel intensity.
    Step 5.4 : Append the  final pixel intensity count value to list.
    Step 5.5 : Reset temporary variable to zero
Step 6: Plot the histogram
```

Program Code

```
from PIL import Image, ImageOps

import matplotlib.pyplot as plt

Cameraman_Image = Image.open("cameraman.png")
Cameraman_gray =  ImageOps.grayscale(Cameraman_Image)
Cameraman_gray.show()
height, width = Cameraman_gray.size

# empty list to hold the frequency of each pixel intensity
Hist_density = []

# for each pixel intensity in range (0-255) , loop is run throughout th image to calculate
the count of that intensity
for pix_intensity in range(0,256):
    density = 0
    for i in range(width):
         for j in range(height):
           if (Cameraman_gray.getpixel((j,i))== pix_intensity):
               density = density + 1

    # total count of each pixel intensity appended
    Hist_density.append(density)

#print(Hist_density)
range_p = range(0,256)

plt.plot(range_p, Hist_density)
```

Result

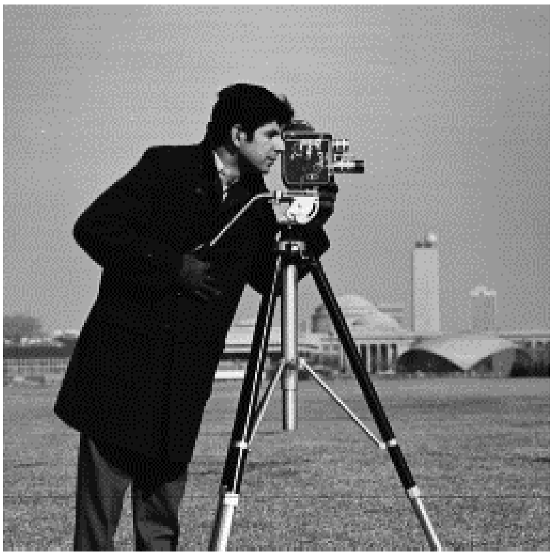

Figure 2: Cameraman : (a)Image

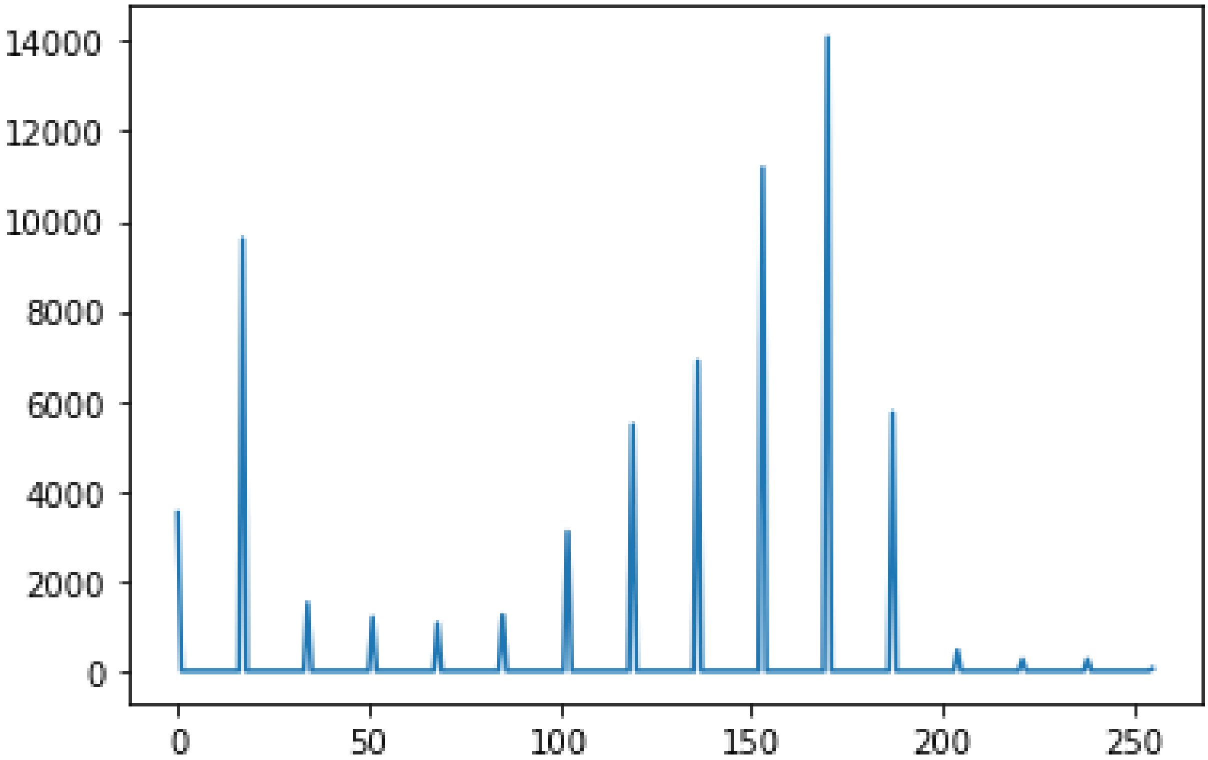

Figure 3: Cameraman : (b)Histogram

Inference

`Histogram conveys a visual interpretation of the pixel intensity frequency distribution`

```
of an image.
```

Question 3(a): Develop programs for the following image enhancement operations. (a) Brightness enhancement

Aim

Enhancing the brightness (each pixel intensity) of an image.

Discussion

Increase the intensity value at each location of the image, to obtain the brightness enhanced image. A loop is run to increase the pixel intensity of each pixel by a certain factor .

Algorithm

```
Step 1: Start
Step 2: Read the image and convert it into gray scale
Step 3: Create a copy of read image to be enhanced
Step 4: Input the Brightness Enhancement factor.
Step 5: Initialize 2 variables to store the width and height of image
Step 6: Run a nested loop
        Step 6.1: Increment the pixel intensity at each pixel position by the enhancement factor
Step 7: Display the original and brightness enhanced images
```

Program Code

```
from PIL import Image

Pollen_Image = Image.open("pollen_grains.tif")
Pollen_Image_Enhanced = Pollen_Image.copy()
Enhancement_factor = int(input('Input the Enhancement intensity factor\n'))

Pollen_Image_Enhanced.show()

height, width  = Pollen_Image.size

for i in range(width):
    for j in range(height):
        temp = Pollen_Image.getpixel((j,i)) + Enhancement_factor

        Pollen_Image_Enhanced.putpixel((j,i),(temp))

Pollen_Image_Enhanced.show()

Pollen_Image_Enhanced.save("Pollen_Image_Enhanced.tif")
```

Result

```
Input the Enhancement intensity factor
80
```

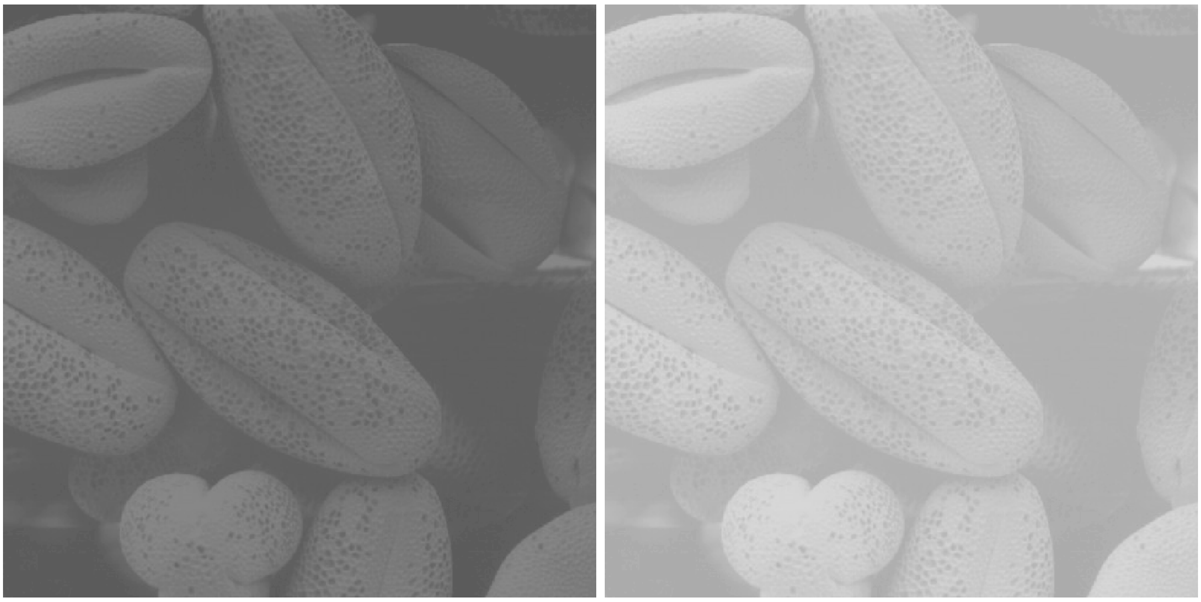

Figure 4: Pollen Grain : (a)Original (b) Brightness Enhanced

Inference

Brightness enhancement increases the pixel intensity making the image overall lighter.

Question 3(b): Develop programs for the following image enhancement operations (b) Brightness slicing- With and without background subtraction. (Use kidney.tif image)

Aim

Brightness slicing of an image with and without background subtraction

Discussion

Brightness slicing is brightness enhancement only for pixels having an intensity value within a certain range. In brightness slicing with background subtraction, pixel intensities not falling under the certain range is made zero whereas in without background subtraction, those pixels are not changed.

Algorithm

```
Step 1: Start
Step 2: Read the image and convert it into gray scale
Step 3: Create two copies of read image for brightness slicing with and without background slicing
Step 4: Input the pixel intensity range and brightness Enhancement factor.
Step 5: Initialize 2 variables to store the width and height of image
Step 6: Run a nested loop
        Step 6.1: Increment the pixel intensity at each pixel position by the enhancement
        factor for pixel lying in intensity range for both with and without background
        subtraction copies.
        Step 6.1: Make the pixel intensity  for pixel lying outside intensity range zero
        for with background subtraction copies.
Step 7: Display the brightness sliced image with and without background slicing.
```

Program Code

```
from PIL import Image

Kidney_Image = Image.open("kidney.tif")

Kidney_without_Back = Kidney_Image.copy()
Kidney_with_Back = Kidney_Image.copy()

# With and Without Background
l= int(input('Enter the range of pixel intensities for Brightness Slicing\n'))

h = int(input(''))
Enhancement_factor = int(input('Input the Enhancement intensity\n'))
height, width = Kidney_Image.size

for i in range(width):
    for j in range(height):

        temp = Kidney_Image.getpixel((j,i))
        if temp in range(l,h+1):
            Kidney_without_Back.putpixel((j,i),(temp + Enhancement_factor))

            Kidney_with_Back.putpixel((j,i),(temp + Enhancement_factor))
        else:
             Kidney_with_Back.putpixel((j,i),(0))


Kidney_without_Back.show()
Kidney_with_Back.show()

Kidney_without_Back.save("Kidney_without_Back.tif")
Kidney_with_Back.save("Kidney_with_Back.tif")
```

Result

```
Enter the range of pixel intensities for Brightness Slicing
150

220

Input the Enhancement intensity
50
```

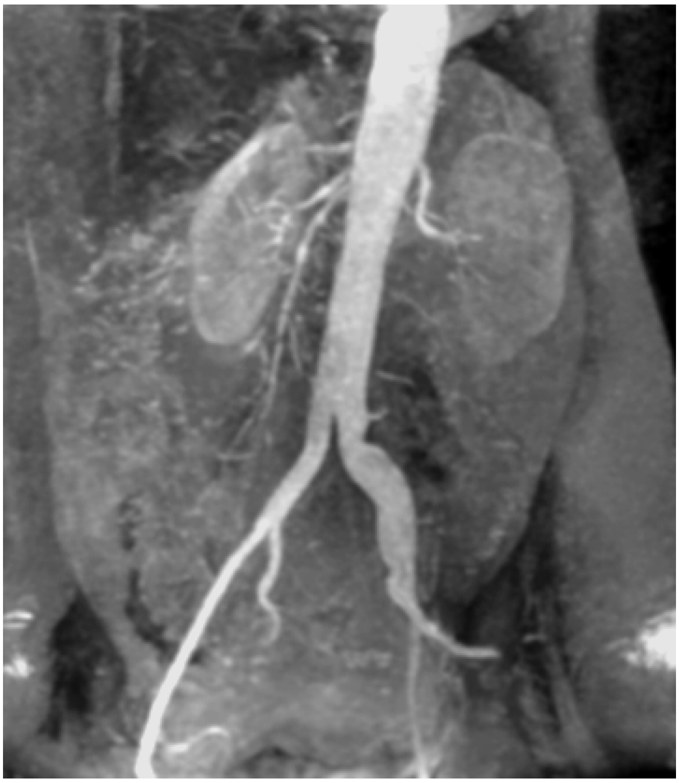

Figure 5: Kidney : (a)Original

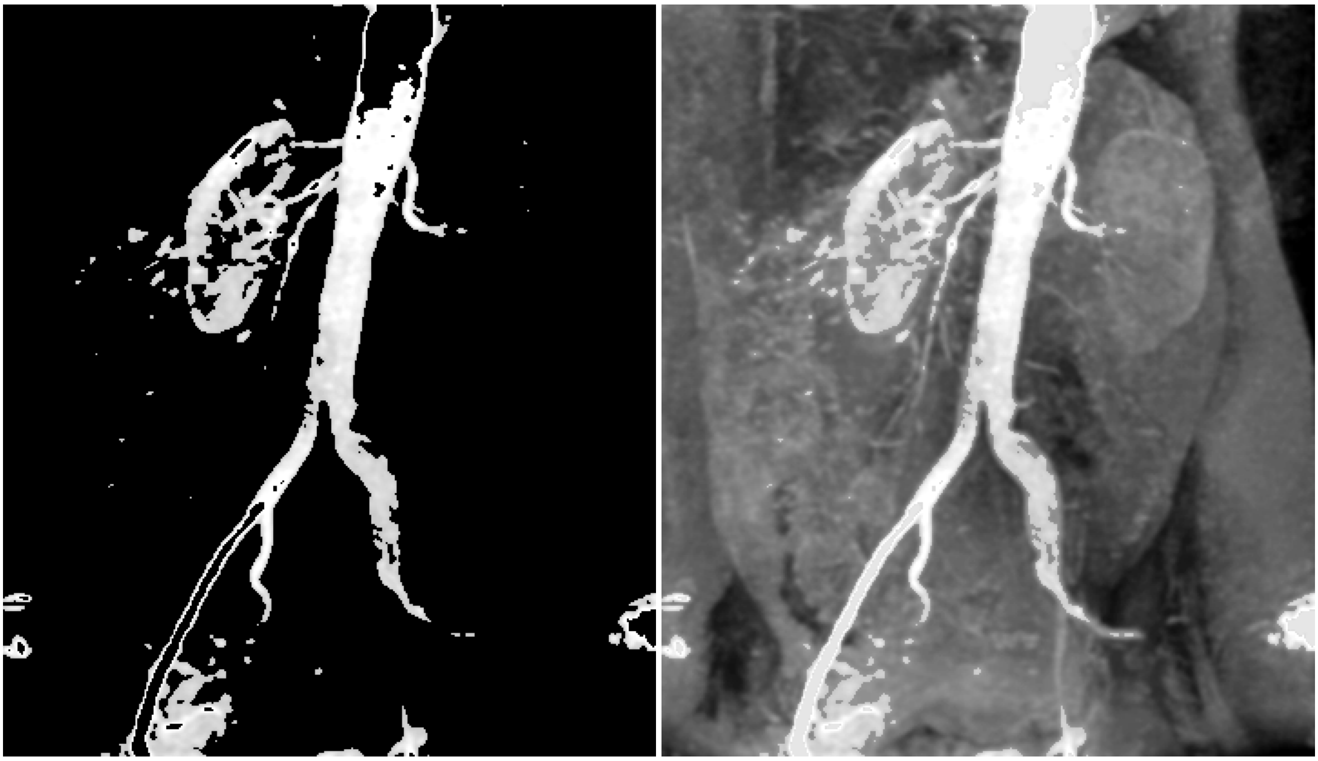

Figure 6: Kidney : (b)Brightness Slicing with Background slicing (c) Brightness Slicing without Background slicing

Inference

```
Brightness slicing can be used to isolate and enhance the pixel intensities of importance.
```