Image processing
Exercise 2

Marc Bisgaard Petersen
Marpet18@student.aau.dk

Questions:

1. A linear gray-level mapping is performed on image f(x,y) in the figure 1 where a = 1 and b = 15. What is the output value of f(2,2)?

f(x,y)

44	57	200
40	42	19
71	189	100

Figure 1

Answer:
The value for $f(2,2) = 100$
The output value is then $a \cdot f(x,y) + b = 1 \cdot f(2,2) + 15 = 100 + 15 \ = 115$

2. The image in figure 2 is correlated with the kernel. No kernel normalization is performed. At the position (1,1) the output value is 911. What is the value of x?

Image

205	204	204	206	255
206	47	208	206	206
201	199	205	206	209
61	128	213	0	205
59	65	255	206	255

Kernel

0	1	0
1	X	1
0	1	0

Figure 2

Answer:
The value for $f(1,1) = 47$ which is x in the kernel
The formel to find the output value is :$image(0,0) \cdot kernel(0,0) + \ image(0,1) \cdot kernel(0,1) \ldots + image(3,3) \cdot kernel(3,3)$
The output for $f(1,1)$ is :

$$(0 \cdot 205) + (1 \cdot 204) + (0 \cdot 204) + (1 \cdot 206) + (x \cdot 47) + (1 \cdot 208) + (0 \cdot 201) + (1 \cdot 199) + (0 \cdot 205) = 911$$

$$204 + 206 + 47x + 208 + 199 = 911$$
$$817 + 47x = 911$$
$$47x = 911 - 817$$
$$x = \frac{94}{47}$$
$$\underline{x = 2}$$

Check if x is 2

$$(0 \cdot 205) + (1 \cdot 204) + (0 \cdot 204) + (1 \cdot 206) + (2 \cdot 47) + (1 \cdot 208) + (0 \cdot 201) + (1 \cdot 199) + (0 \cdot 205) = 911$$

3. The image in figure 2 is filtered by a 3x3 Median filter (the border pixels are ignored, meaning that the output image is 3x3). A threshold is applied to the output image such that a pixel above the threshold value is set to white and otherwise black. The threshold value is set to 203. How many white pixels are present after the thresholding?

   Answer:
   The border pixels are ignored which leaves a matrix of 3x3 like this:

$$\begin{bmatrix} 47 & 208 & 206 \\ 199 & 205 & 206 \\ 128 & 213 & 0 \end{bmatrix}$$

   And the threshold is 203, which mean any value above the threshold is going to be white and below the threshold will be black

$$\begin{bmatrix} 47 & 208 & 206 \\ 199 & 205 & 206 \\ 128 & 213 & 0 \end{bmatrix} => \begin{bmatrix} 0 & 255 & 255 \\ 0 & 255 & 255 \\ 0 & 255 & 0 \end{bmatrix}$$

   So, there will be 5 pixels that are white

4. Given the image shown, and the structuring element: {{0,1,0},{1,1,1},{0,1,0}} describe which morphological operator (erode, dilate, open, close) has been used to create the 4 images below.

original image

A B C D

   Answer:
   Image A is affected by Open due to the operator is removing the corners of the foreground.
   Image B is affected by Dilate due to the operator is expanding the edges of the foreground while shrinking the background.
   Image C is affected by Erode due to the operator is shrinking the foreground.
   Image D is affected by Close due to the operator is reversing what Open operator does.

5. The image 1 represents an original image of low contrast, having the grey level histogram given in the image 2. If on this image, a grey level mapping is applied, using the transfer function from image 3, which of the following images A, B or C will be the resulting image?

Image 1

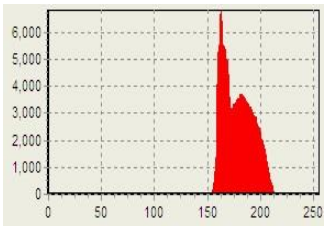

Image 2

Image 3

A

B

C

Answer:

Image 3 is matching with image B due the threshold. The threshold is a check on brightness which is why the graph is spiking at 160.

6. Consider the following 8-bit image:

The image has dimensions 20x20 pixels. Draw the histogram of this image. Please note that the borders of the images shown in black are just to highlight the boundaries. The border is not a part of the image.

Answer:

Programming:

7. Write a sketch in Processing where you are trying to produce a binary image based on the image “coin.png” where the foreground (coins) is clearly segmented from the background (use of morphological operators)

Answer:

Image:

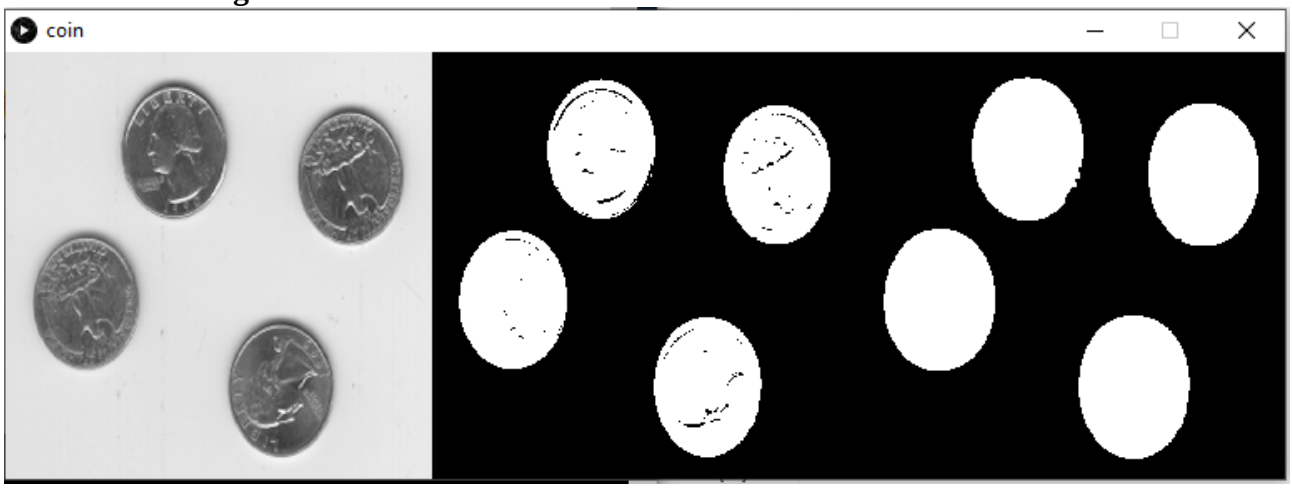

Figure 1: coin image to threshold to dilation

Code snippet:

```
PImage source, thresholded,dilated;
float colorThresh = 180f;
void setup()
{
  size(768,256);

source = loadImage("coin.png");
thresholded = createImage(source.width,source.height,RGB);
dilated = createImage(source.width,source.height,RGB);
loadImages();
}

void draw()
{
  image(source,0,0);
  image(thresholded,256,0);
  image(dilated,512,0);
}

public void loadImages()
{
    loadPixels();
  thresholded.loadPixels();
  source.loadPixels();
  dilated.loadPixels();
```

```
 for (int x = 0; x < source.width; x++ ) {
  for (int y = 0; y < source.height; y++ ) {
   int loc = x + y*source.width;
   // Test the brightness against the threshold
   if (brightness(source.pixels[loc]) > colorThresh){
    thresholded.pixels[loc] = color(0); // White
   } else {
    thresholded.pixels[loc] = color(255);   // Black
   }
  }
 }

//-------------------------DILATION 3x3
  for (int y = 1; y < source.height-1; y++) { // Skip top and bottom edges
   for (int x = 1; x < source.width-1; x++) { // Skip left and right edges
    float sum = 0; // Kernel sum for this pixel
    for (int ky = -1; ky <= 1; ky++) {
     for (int kx = -1; kx <= 1; kx++) {
      // Calculate the adjacent pixel for this kernel point
      int pos = (y + ky)*source.width + (x + kx);
      // Multiply adjacent pixels based on the kernel values
      sum += brightness(thresholded.pixels[pos])/255;
     }
    }
  if(sum >= 1)
     dilated.pixels[y*source.width + x] = color(255,255,255);

  else
     dilated.pixels[y*source.width + x] = color(0,0,0);

   }
  }
 thresholded.updatePixels();
 dilated.updatePixels();
}
```