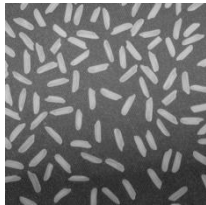


Image processing – exercise 3

Question 1

On this question we have image of rice, the purpose is to find the edges and then to fill them.

The original image:



To do so we apply multi process on the image:

1. We did brightness equalization by creating the image's gray scale using open with big kernel and then we substrate the gray scale we found from the original image (function `brightness_equalization`)
2. Use threshold to convert the image to binary (function `use_threshold`)
3. Clean noises using open with kernel of ones in size 10X10 (function `clean_noise`)
4. Then the image is ready. Now to find the outlines we erode the image and substrate the results of the erosion from the original image. (Function `find_outlines`)
5. Now we have the outlines, and we need to fill it. We do so on the inverted image so we can use open. First, we create the inverted image all black and turn on the first pixel from which we start the dilation. Then we do dilate on the inverted image and do logical and with the original image so we keep the outer pixel on and the inner pixels off. In the end we revert the image to receive the correct filled image

NOTE: if you want to see the filling process, pass True in the "show process" variable in the region filling function on line 68.

The output:

original image

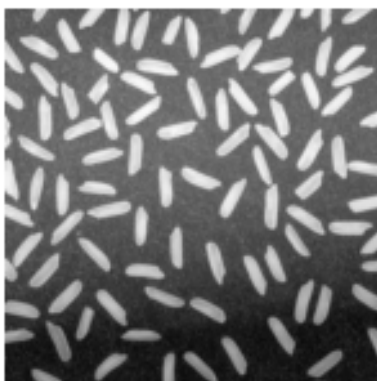


image outlines

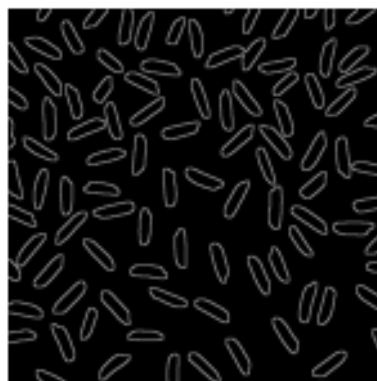


image filled



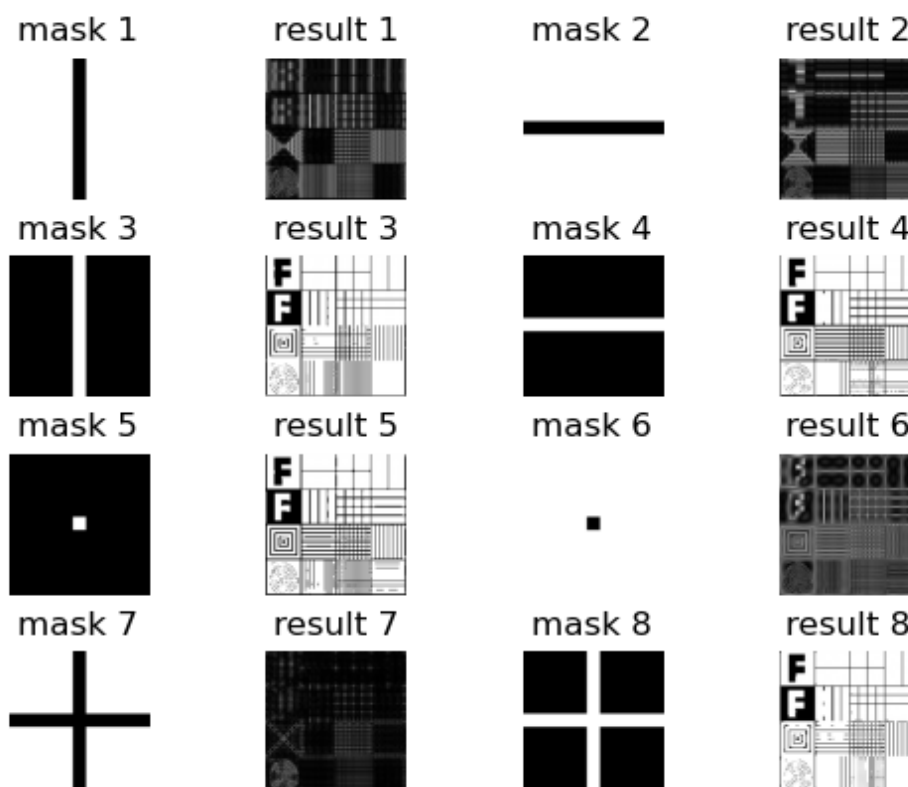
Question 2

in this question, we create 8 different masks(filters), and then apply the filters on the image in the frequency plane (using transform Fourier).

The original image:



The results:



Question 2 explanations:

First, we create the masks, then we apply them on the image:

1. Use FFT on the original image
2. Use FFT shift on the original image
3. For each filter: multiply it with the FFT image
4. Reverse FFT and FFT shift on the image with the mask.

Results:

Masks 3, 4, 5 and 8 are low pass filters: all the low frequency parts = the details, will remain

Mask 5: general low pass filter, most small details will remain

Mask 3: low pass with vertical line, so especially small details with vertical lines will remain

Mask 4: low pass with horizontal line, so especially small details with horizontal lines will remain

Mask 8: low pass with both vertical and horizontal lines, so especially small details with vertical or horizontal lines will remain

Masks 1, 2, 6 and 7 are high pass filters: all the high frequency parts = big details (for example- background color) will be deleted

Mask 7: general high pass filter. Delete the background color

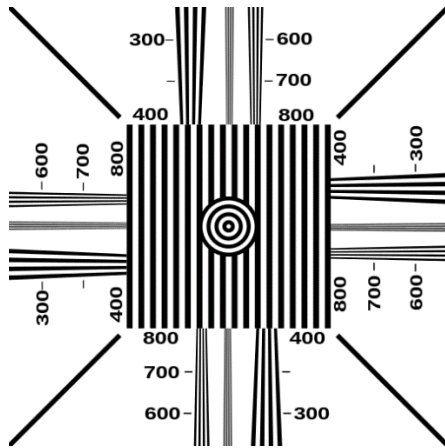
Mask 1: high pass with vertical line, so most small details with horizontal lines will be deleted

Mask 2: high pass with horizontal line, so most small details with vertical lines will be deleted

Mask 8: high pass with both vertical and horizontal lines, so small details with vertical or horizontal lines will be deleted with other big details

Question 3:

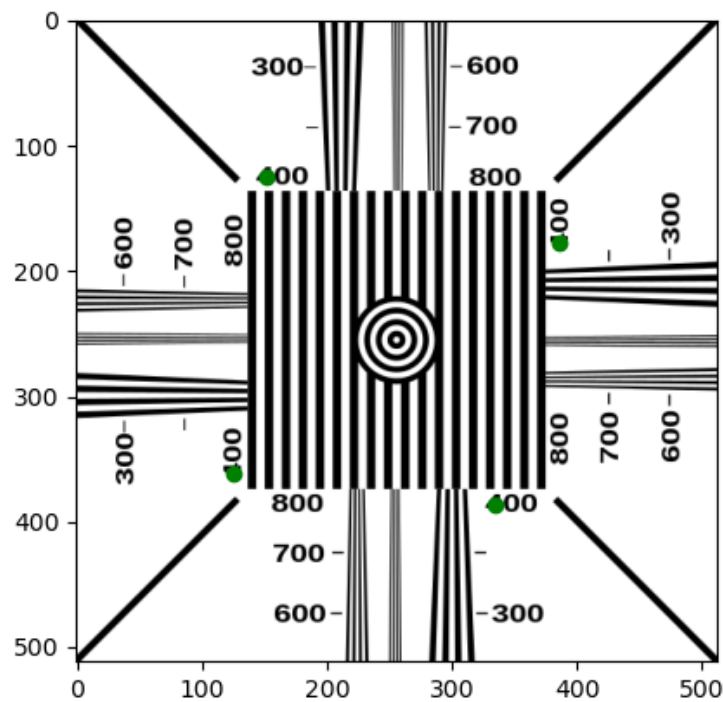
In this question, we try to find the digit 4 in the given image:



First, we create 2 filters for each appearance of the digit, regular digit, and rotated digit. Then we apply FFT and FFT shift on the original image and on the filters. Next we multiply the FFT images with each filter and do reverse FFT on the result of the multiplication.

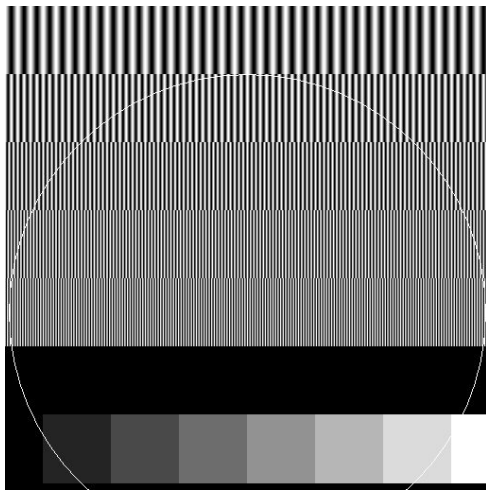
Now in the result we should get slightly brighter pixels where there is an appearance of 4. So using threshold (0.05) we find those pixels and plot them on the original image.

The results:



Question 4:

In this question, we try to find the circle in the given image:



We use the Hough transform algorithm to find circles. I chose min and max radius in the size of half the image shape (with small offset = 5) to find the big circle. After receiving the correct radius and the center pixels we plot the circle on the original image.

The results:

