UNIVERDIDAD VERACRUZANA

INSTITUTO DE INVESTIGACIONES EN INTELIGENCIA ARTIFICIAL

VISIÓN ARTIFICIAL

HOMEWORK 5

LIBRO: DIGITAL IMAGE PROCESSING

AUTOR: RAFAEL C. GONZALEZ

CONVOLUTION IMAGE

ALUMNO: JOSÉ DAVID VELAZCO MUÑOZ

PROFESOR: DR. HECTOR ACOSTA MESA

XALAPA, VERACRUZ

21 DE MARZO 2023

Objetive

- Write program to perform spatial filtering of an image (see Section 3.5 regarding implementation). You can fix the size of the spatial mask at 3×3, but the coefficients need to be variables that can be input into your program. This project is generic, in the sense that it will be used in other projects to follow. Use figure 3.35.

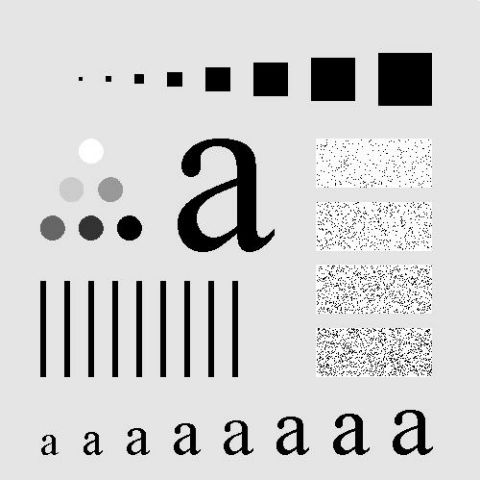


Figura 3.38(a)

Methodology

For realize the convolution is necessary following the formula:

$$y = (f * g) = \sum_{u=-m}^{m} \sum_{v=-n}^{n} f(u,v)g(x-u, y-v)$$

Where:

- f Is the matix of the image.
- g Is the matix of the filter.
- y Is the output matix.
- x Is the position real the image.
- m and n are the width and height of the filter, respectively.

The result is an image convolution of the same dimension of the original image. Each element in the output convolution image is the result of applying the filter to the image at that position.

To generate convolution image:

- Define kernel size in this case is 3x3.
- Do zeros matrix with the same image size.
- For each pixel (i, j) in the input image, we place the kernel center at that position and perform the convolution operation.
- We normalize the values.

Pseudocode.

- Determinate dimension image
- Determinate dimension kernel
- For each row i in the input image X
- For each column j in the input image X
- For each row k in the kernel H
- For each column l in the kernel H

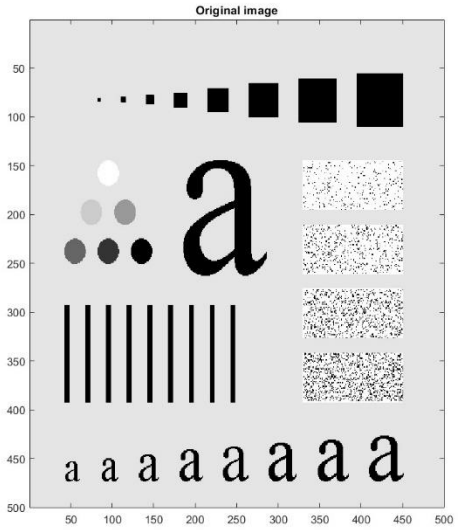
$$y(i,j) = y(i,j) + X(i+k, j+l) * H(k,l)$$

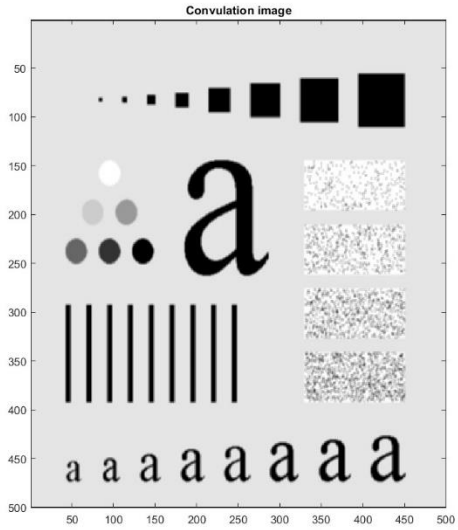
Results

Trying different kernel 3x3:

This kernel is used to smooth or blur the image, thus removing noise.

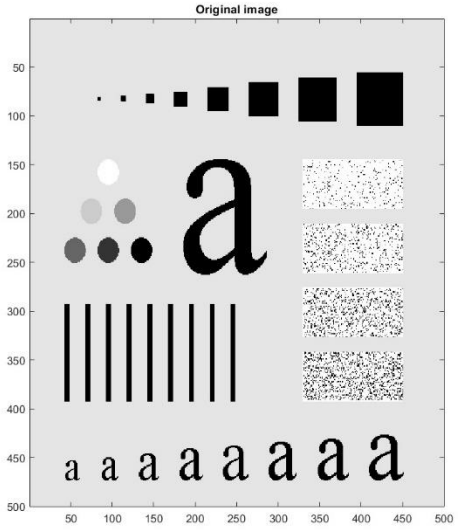
$$Blur = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

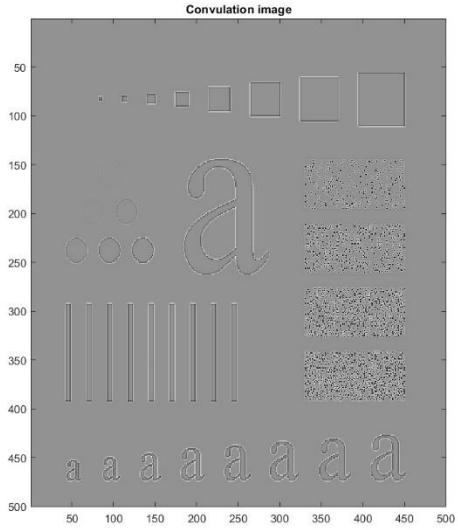




This kernel is used to edge detection the object in the image.

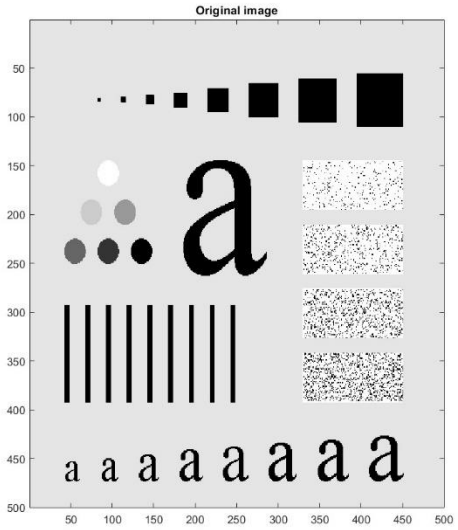
$$Edge\ detection = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

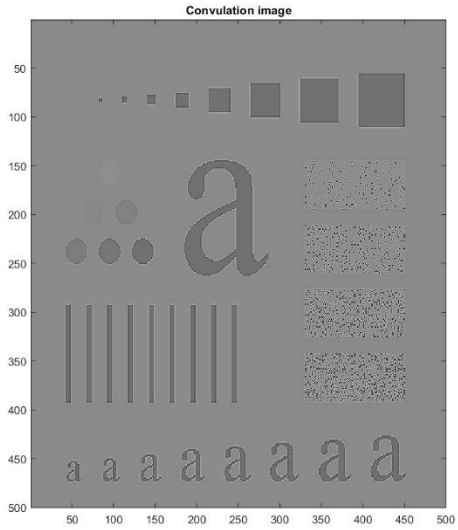




This kernel is used to edge enhancement image.

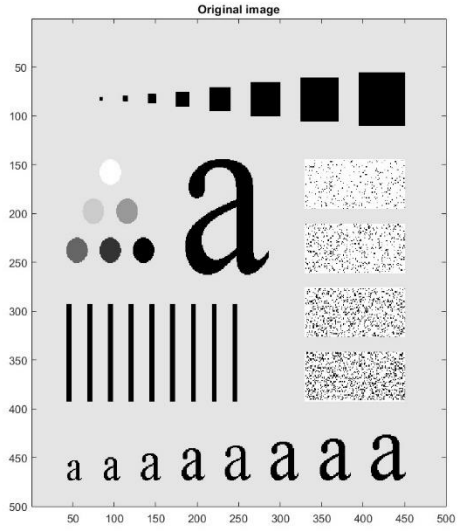
$$Edge\ enhancement = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 5 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

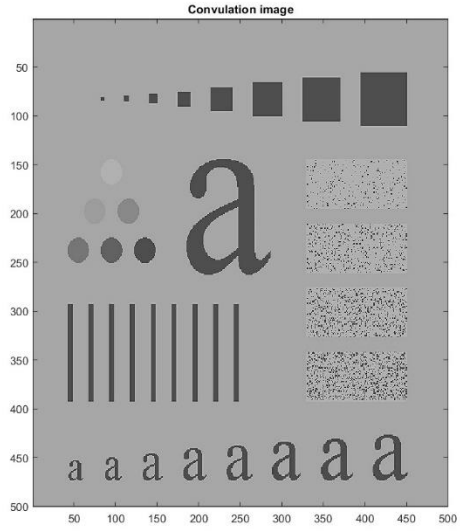




This kernel is used to sharpening image.

$$Sharpening = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$





Conclusion

We can conclude that convolution is a very useful process for image filtering, being one of the most important in artificial intelligence. Through great applications in neural networks and different connection strategies.
Convolution can have different types of filters for image processing, such as edge detection, recognition of some specific areas of the image, and many more applications. Through various tests we can determine that there is a big difference in the application of different kernels, in addition to being one of the most concurrent. We have had different results with different kernels used. The dimensions of the convolution can be set to a specific size, in this case using a 3x3 kernel there is a big difference in the results.
We can say that convolution is a mathematical operation that combines two signals and generates a third. Convolution can be used in various applications of artificial intelligence, in image processing and in different generated alternatives.

Annex: Code

```
clear all;
%Read the image
I=double(imread('Fig3.35(a).jpg'));

%Determinate the dimension
[rows,cols]=size(I);

% You can modificate the coefficients in the program

%Examples

% Edge enhancement
% [[0,-1,0],[-1,5,-1],[0,-1,0]]
% Edge detection
% [[-1, -1, -1], [-1, 8,-1], [-1, -1, -1]]
% Blur
% [[1/9, 1/9, 1/9], [1/9, 1/9,1/9], [1/9, 1/9, 1/9]]
% Sharpening
% [[0,-1,0],[-1,9,-1],[0,-1,0]]

%Determinate the kernel
mask = input('Introduce los coeficientes de la mascara espacial de 3x3 separados por
coma: ');
mask = reshape(mask, [3,3]);

% Variable where we save convolution matrix
g=zeros(rows,cols);
for i=2:rows-1
    for j=2:cols-1
        i2=I(i-1:i+1,j-1:j+1);
        % Se suman los extremos hasta i+1 y j+1
        g(i,j)=sum(sum(mask.*i2));
    end
end

% Show convolution imagen
colormap("gray");
figure(1)
subplot(1,2,1),imagesc(I),title('Original image');
subplot(1,2,2),imagesc(g),title('Convulation image');
```

References

- Haris Papasaika-Hanusch. *Digital image processing using matlab*. Institute ofGeodesy and Photogrammetry, ETH Zurich, 63, 1967.
- Gonzales, R. C., & Wintz, P. (1987). *Digital image processing*. Addison-Wesley Longman Publishing Co., Inc.