Table of Contents

Sesion 4 : Angel Prat, Haopeng Lin

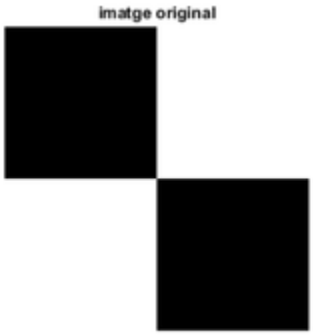
```
im1 = imread('toycars1.png');
im2 = imread('toycars2.png');
im3 = imread('toycars3.png');
figure,
subplot(1,3,1),imshow(im1),
subplot(1,3,2),imshow(im2),
subplot(1,3,3),imshow(im3),
```

Diferencia de imagenes (Operaciones de puntos)

```
res1 = imabsdiff(im1,im2);
res2 = imabsdiff(im1,im3);

figure,
subplot(1,2,1),imshow(res1),
subplot(1,2,2),imshow(res2)
```

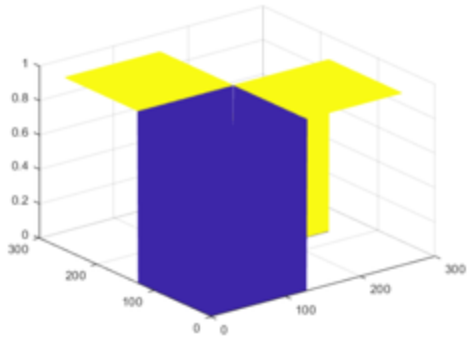
Producte de convolució (Operaciones de vecinos)

```
im = ones(256);
im(1:128,1:128) = 0;
im(129:256,129:256) = 0;
figure,imshow(im),title('imatge original')
```



Grafica 3d

```
figure,mesh(im)
% Kernel de convulució, recomendar no poner numeros negativos

h =[0,1,0;1,2,1;0,1,0]/6;
```



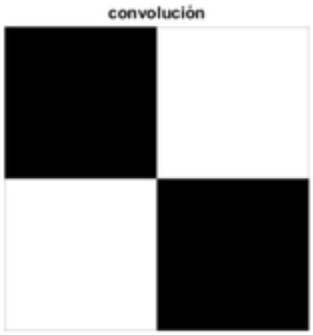
Exercici 1: Convolucionar

```
% Tamaños de la imagen y el kernel
[rows, cols] = size(im);
[kRows, kCols] = size(h);

resultado = zeros(rows, cols);

% Realizar la convolución, extendiendo los bordes como si fuera el valor
% anterior:
%    Si (1,1) = 0 => (0,1) && (0,0) && (1,0) seria 0 también, asi mismo o
%    similar con todos los bordes
for i = 1:resultRows
    for j = 1:resultCols
        % Calcular el valor de píxel convolucionado en (i, j)
        suma = 0;
        for m = 1:kRows
            for n = 1:kCols
                x = i - 1 + m;
                y = j - 1 + n;

                % Extender a 0 en los bordes desconocidos
                if x < 1
                    x = 1;
                elseif x > rows
                    x = rows;
                end

                if y < 1
                    y = 1;
                elseif y > cols
                    y = cols;
                end

                suma = suma + im(x, y) * h(m, n);
            end
        end
        resultado(i, j) = suma;
    end
end
```

```
% Mostrar la imagen resultante
imshow(resultado, []);
```

Convolucion

```
res = imfilter(im,h);
imshow(res),title('convolución')

% Convolucion 2
h2 = ones(31)/31/31;
res2 = imfilter(im,h2);
figure,imshow(res2),title('convolución 2')
```

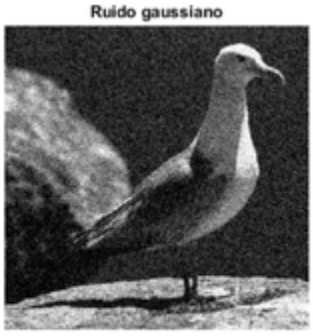


Padding replicat

```
res3 = imfilter(im,h2,'replicate');
figure,imshow(res3),title('padding replicat')
```

Ruido gaussiano

```
img = imread('gull.tif');
figure,imshow(img),title('imatge original')
% Añadir sonido al imagen (gaussiano)
img2 = imnoise(img,"gaussian");
figure,imshow(img2),title('Ruido gaussiano')
```



Filtro gaussiano (Limpiar imagen)

```
h3 = fspecial('gaussian',7,2);
result = imfilter(double(img2),h3);
figure,imshow(result,[]),title('filtro gaussiano')
```

Sal y pimienta

```
imsp = imnoise(img,'salt & pepper',0.2);
figure,imshow(imsp),title('imagen salt pimienta')

resultsp = imfilter(double(imsp),h3);
figure,imshow(resultsp,[]),title('filtro gaussiano de salt y pimienta')
```

Published with MATLAB® R2023a