Taller - Filtrado Espacial de Imágenes Matlab

William Gómez Roa
wa.gomez@javeriana.edu.co
Bioingeniería y Ciencia de Datos

Carolina Santos Baquero
carolinasantosb@javeriana.edu.co
Bioingeniería

1 Generación de imágenes con ruido

El siguiente código (adaptado de https://www.geeksforgeeks.org/what-are-different-types-of-denoisingfilters-in-matlab/) presenta diferentes opciones para agregar ruido a una imagen en escala de grises en MATLAB:

```
% cargar la imagen
img=imread("BrainMR.png");
% convertir a escala de grises
grayimg=im2gray(img);
% agregar ruido Gaussiano a la imagen
gauss_img=imnoise(grayimg,'gaussian',0,0.02);
% agregar ruido de Poisson a la imagen
poiss_img=imnoise(grayimg,'poisson');
% agregar ruido de sal y pimienta a la imagen
salpe_img=imnoise(grayimg,'salt & pepper', 0.05);
% visualizar la imagen original y sus versiones con ruido
figure(1)
subplot(2,2,1)
imshow(grayimg);
subplot(2,2,2)
imshow(gauss_img);
subplot(2,2,3)
imshow(poiss_img);
subplot(2,2,4)
imshow(salpe_img);
% guardar las imagenes con ruido
imwrite(gauss_img,"BrainMR_gauss.png");
```

```
imwrite(poiss_img,"BrainMR_poiss.png");
```

Ejercicio 1: Ejecute el código para obtener las diferentes versiones con ruido de la imagen BrainMR.png (adjunta a este enunciado), e incorpore los resultados obtenidos a su reporte.

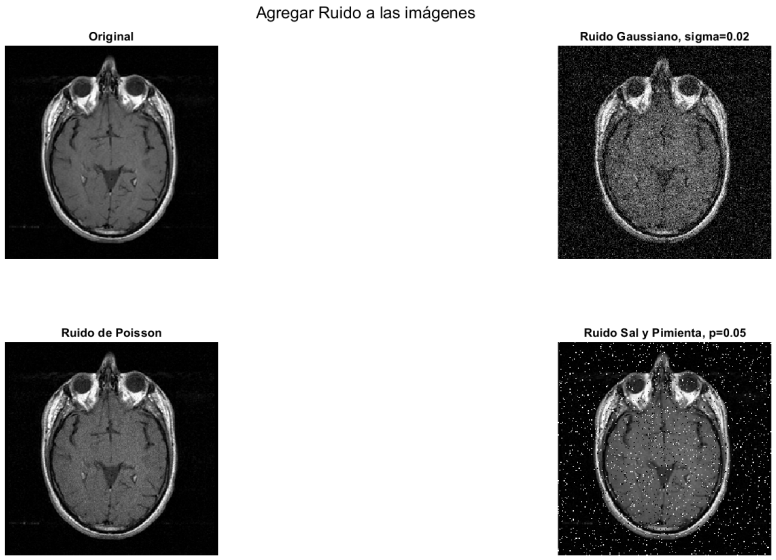

Figure 1: Efectos de agregar diferentes tipos de ruido.

Ejercicio 2: Ejecute el código anterior con otra imagen (médica) diferente, para obtener sus versiones con ruido diferente, e incorpore la imagen utilizada y los resultados obtenidos a su reporte.

```
% cargar la imagen
img=imread("ImagenMedica.jpg");
% convertir a escala de grises
grayimg=im2gray(img);
% agregar ruido Gaussiano a la imagen
gauss_img=imnoise(grayimg,'gaussian',0,0.02);
% agregar ruido de Poisson a la imagen
poiss_img=imnoise(grayimg,'poisson');
% agregar ruido de sal y pimienta a la imagen
salpe_img=imnoise(grayimg,'salt & pepper', 0.05);
% visualizar la imagen original y sus versiones con ruido
figure(1)
```

```
subplot(2,2,1)
imshow(grayimg);
title('Original');
subplot(2,2,2)
imshow(gauss_img);
title('Ruido Gaussiano');
subplot(2,2,3)
imshow(poiss_img);
title('Ruido Poisson');
subplot(2,2,4)
imshow(salpe_img);
title('Ruido sal y pimienta');
% guardar las imagenes con ruido
imwrite(gauss_img,"ImagenMedica_gauss.png");
imwrite(poiss_img,"ImagenMedica_poiss.png");
imwrite(salpe_img,"ImagenMedica_salpe.png");
sgtitle ('Diferentes ruidos a la imagen medica');
```

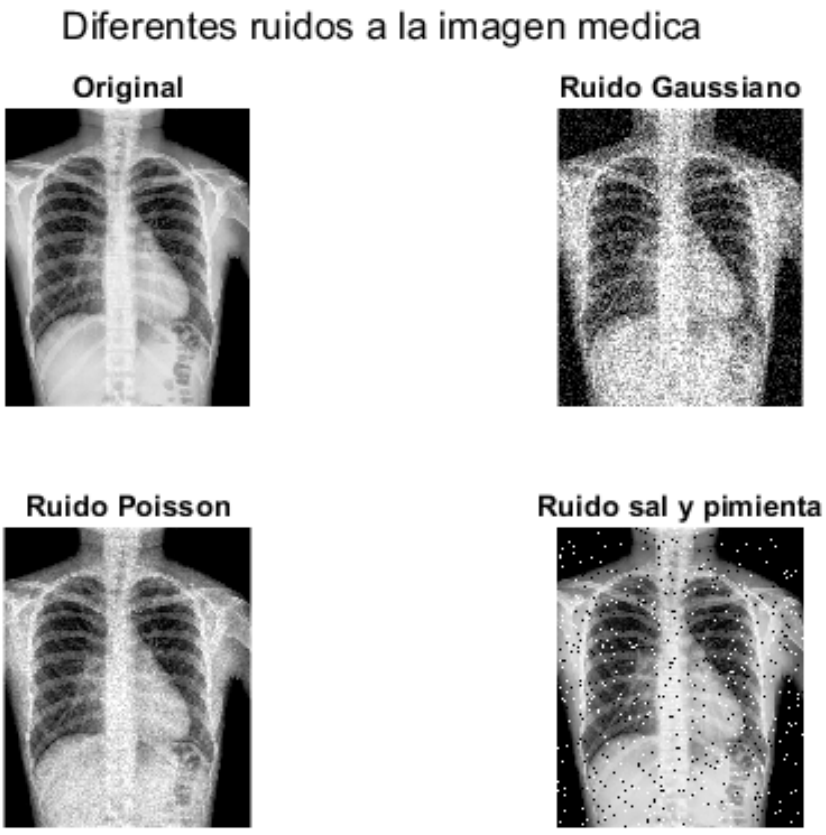

Figure 2: Diferentes ruidos en la imagen medica.

En las dos imágenes se puede observar que el ruido gaussiano se asemeja a una niebla suave que se extiende por toda la imagen, causando una disminución gradual en la calidad de la misma. Por otro lado, el ruido de Poisson es más granular y errático, provocando fluctuaciones aleatorias en los niveles de intensidad de los píxeles. En contraste, el ruido de sal y pimienta introduce puntos brillantes (sal) y oscuros (pimienta) en la imagen, creando un efecto de granos aleatorios. Llegando a que estos tres ruidos son defectos o imperfecciones que pueden estar presentes en la imagen, degradando la calidad visual de la misma.

2 Remoción del ruido en las imágenes

El siguiente código (adaptado de https://www.geeksforgeeks.org/what-are-different-types-of-denoisingfilters-in-matlab/) presenta diferentes opciones para suavizar una imagen en escala de grises con ruido en MATLAB, comparando los resultados obtenidos con respecto a la imagen original (sin ruido):

```
% cargar la imagen con ruido
img=imread("BrainMR_gauss.png");
% convertir a escala de grises
grayimg=im2gray(img);
% crear un kernel para filtro promedio 3x3
h = [1/9 1/9 1/9; 1/9 1/9 1/9; 1/9 1/9 1/9];
% filtrar la imagen con filtro promedio
prom_filt = uint8(conv2(grayimg,h,'same'));
% filtrar la imagen con filtro mediana
med_filt = medfilt2(grayimg);
% filtrar la imagen con filtro Gaussiano, sigma = 1
gauss_filt_1 = imgaussfilt(grayimg,1);
% visualizar la imagen con ruido y sus versiones filtradas
figure(1)
subplot(2,2,1)
imshow(grayimg);
subplot(2,2,2)
imshow(prom_filt);
subplot(2,2,3)
imshow(med_filt);
subplot(2,2,4)
imshow(gauss_filt_1);
% guardar las imagenes filtradas
imwrite(prom_filt,"BrainMR_gauss_prom.png");
imwrite(med_filt,"BrainMR_gauss_med.png");
imwrite(gauss_filt_1,"BrainMR_gauss_filt_1.png");

% cargar la imagen original
img2=imread("BrainMR.png");
```

```
% convertir a escala de grises
grayimg2=im2gray(img2);
% calcular las diferencias con respecto a la imagen original

diff1 = grayimg2-grayimg;
num_diff1 = sum(diff1(:)==0) / size(diff1(:),1)
diff2 = grayimg2-prom_filt;
num_diff2 = sum(diff2(:)==0) / size(diff2(:),1)
diff3 = grayimg2-med_filt;
num_diff3 = sum(diff3(:)==0) / size(diff3(:),1)
diff4 = grayimg2-gauss_filt_1;
num_diff4 = sum(diff4(:)==0) / size(diff4(:),1)
% visualizar las diferencias con respecto a la imagen original
figure(2)
subplot(2,2,1)
imshow(diff1);
subplot(2,2,2)
imshow(diff2);
subplot(2,2,3)
imshow(diff3);
subplot(2,2,4)
imshow(diff4);
```

Ejercicio 3: Ejecute el código para obtener las versiones filtradas de las imágenes $BrainMR_gauss.png, BrainMR_poiss.pngyBrainMR_salpe.png$, e incorpore los resultados obtenidos a su reporte. Utilice los valores calculados de las diferencias (valores en variables num_diff*) para identificar cuál es el filtro que mejor reduce cada uno de los tipos de ruido agregados en la primera parte.

Las imágenes con ruido Gaussiano, de poisson y sal y pimienta fueron filtradas utilizando 3 diferentes filtros de suavisado: Filtro promedio, filtro mediana y filtro Gaussiano.

BrainMRGauss:

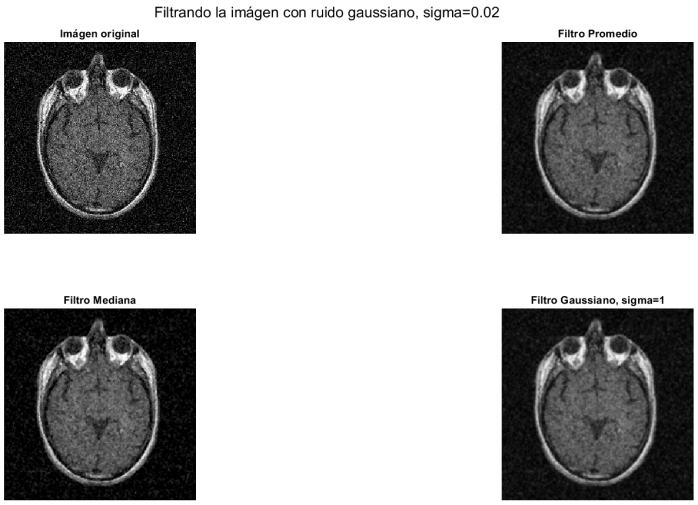

Figure 3: Filtrando la imágen con ruido Gaussiano , sigma=0.02

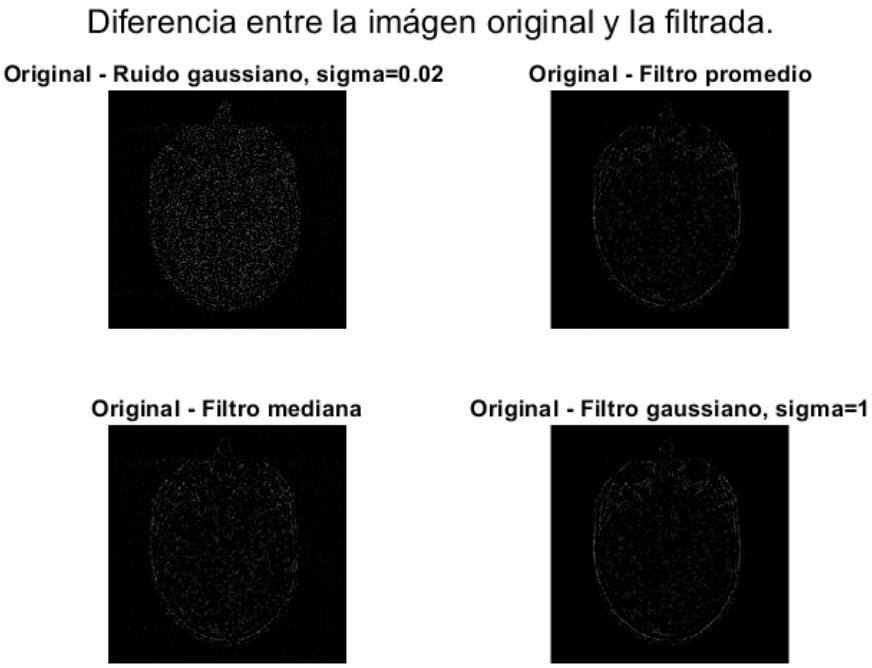

Figure 4: Diferencia entre la imagen con ruido y la filtrada.

```
num_diff1 =

    0.5645


num_diff2 =

    0.7780


num_diff3 =

    0.5621


num_diff4 =

    0.7852
```

Figure 5: Error calculado entre la imagen con ruido y la filtrada.

En esta parte se utilizó la imágen con ruido gaussiano cuyo sigma fue de 0.02. Se puede observar que el la imágen original comparada con la imágen con este ruido presnetabá un error de 0.5645; después de agregar el filtrado a la imágen esta diferencia llego a reducirse solo hasta 0.5621 utilizando el filtro mediana, en los otros casos el ruido aumenta.

BrainMRPoisson:

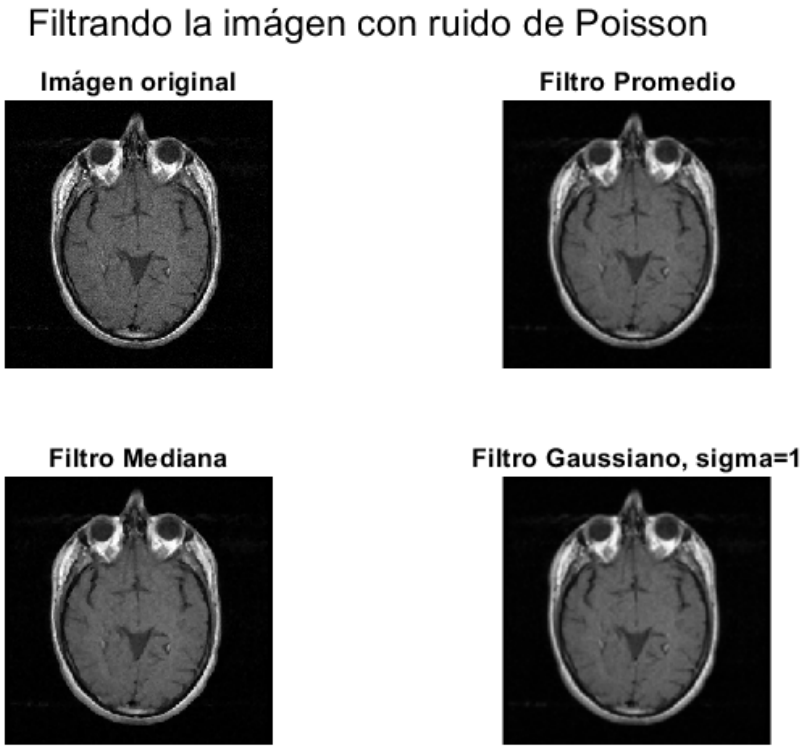

Figure 6: Filtrando la imágen con ruido de Poisson

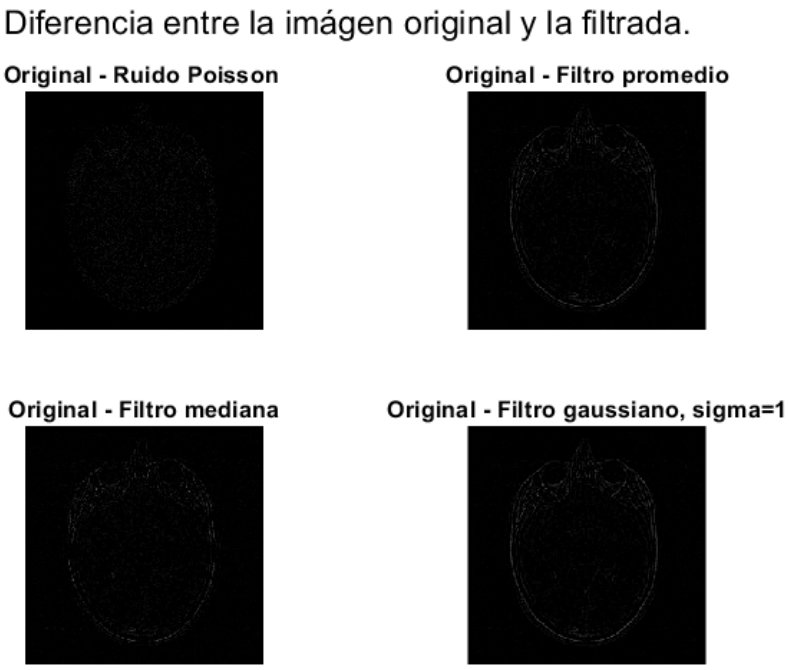

Figure 7: Diferencia entre la imagen con ruido y la filtrada.

```
num_diff1 =

    0.6076

num_diff2 =

    0.6075

num_diff3 =

    0.5471

num_diff4 =

    0.6150
```

Figure 8: Error calculado entre la imagen con ruido y la filtrada.

En esta parte se utilizó la imágen con ruido de Poisson. Se puede observar que el la imágen original comparada con la imágen con este ruido presnetabá un error de 0.6076; después de agregar el filtrado a la imágen esta diferencia llego a reducirse hasta 0.5471 utilizando el filtro mediana.

BrainMRSaltPeper:

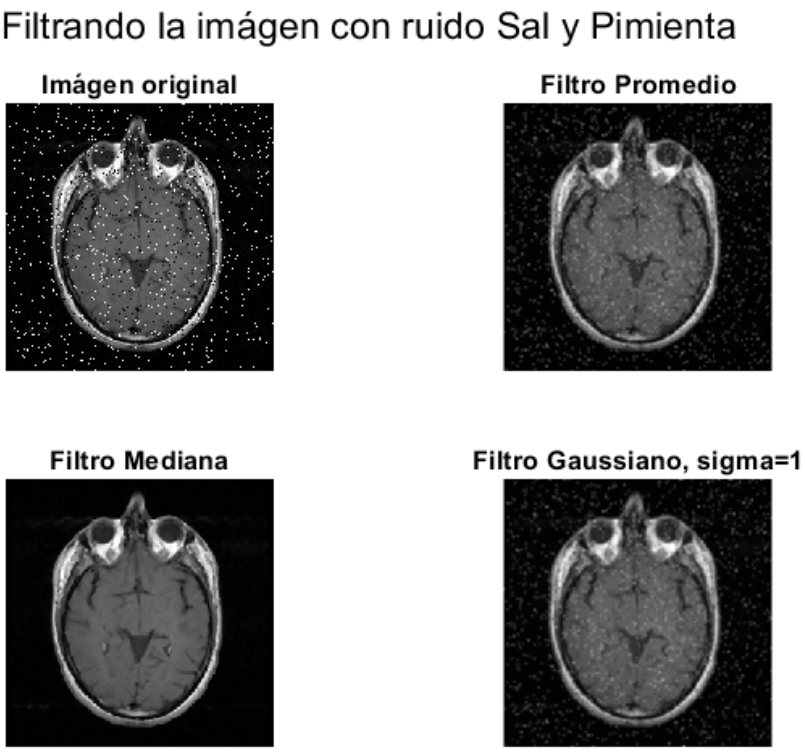

Figure 9: Filtrando la imágen con ruido de Saly y Pimienta

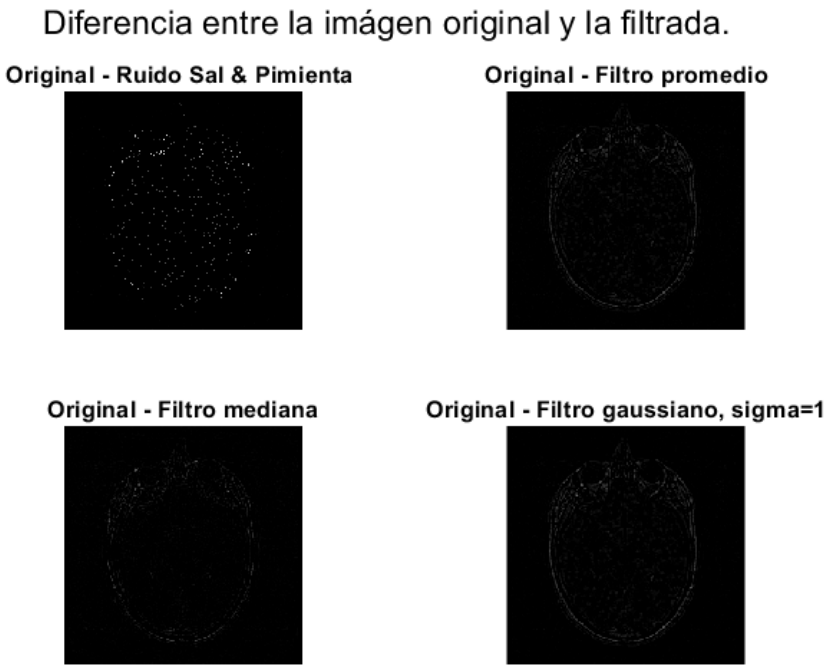

Figure 10: Diferencia entre la imagen con ruido y la filtrada.

```
num_diff1 =

    0.9782

num_diff2 =

    0.6488

num_diff3 =

    0.6573

num_diff4 =

    0.6969
```

Figure 11: Error calculado entre la imagen con ruido y la filtrada.

En esta parte se utilizó la imágen con ruido de Sal y & Pimienta. Se puede observar que el la imágen original comparada con la imágen con este ruido presnetabá un error de 0.9782; después de agregar el filtrado a la imágen esta diferencia llego a reducirse hasta 0.6488 utilizando el filtro promedio, aunque el filtro mediana es muy similar.

Ejercicio 4: Replique el ejercicio anterior (3), ahora con las imágenes con ruido generadas en el Ejercicio 2, e incorpore los resultados obtenidos en su reporte, así como los valores calculados de las diferencias para identificar de nuevo el filtro que mejor reduce el ruido en cada caso.

Las imágen médica seleccionada con ruido Gaussiano, de poisson y sal y pimienta fueron filtradas utilizando 3 diferentes filtros de suavisado: Filtro promedio, filtro mediana y filtro Gaussiano.

IMGauss:

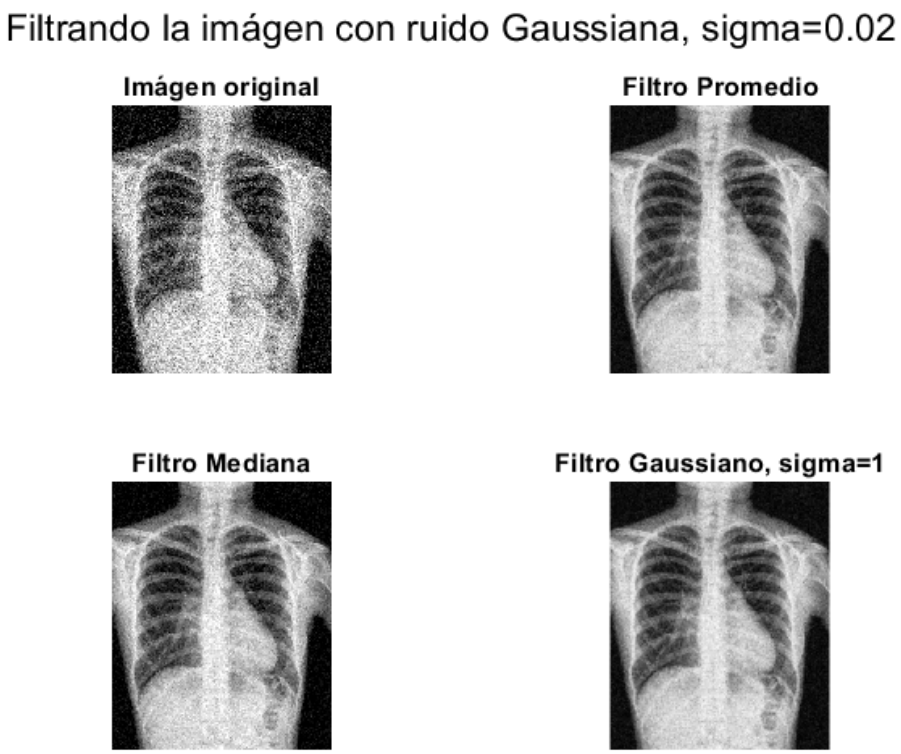

Figure 12: Filtrando la imágen con ruido Gaussiano , sigma=0.02

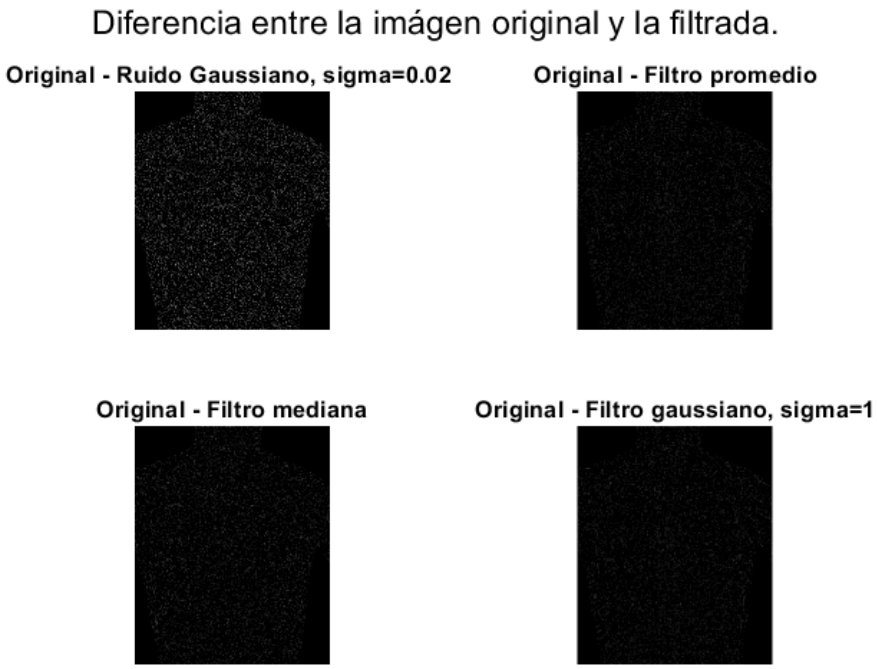

Figure 13: Diferencia entre la imagen con ruido y la filtrada.

```
num_diff1 =

    0.5943

num_diff2 =

    0.5853

num_diff3 =

    0.6000

num_diff4 =

    0.5845
```

Figure 14: Error calculado entre la imagen con ruido y la filtrada.

En esta parte se utilizó la imágen con ruido gaussiano cuyo sigma fue de 0.02. Se puede observar que el la imágen original comparada con la imágen con este ruido presnetabá un error de 0.5943; después de agregar el filtrado a la imágen esta diferencia llego a reducirse solo hasta 0.58 utilizando el filtro promedio o el gaussiano con sigma=1.

IMPoisson:

Figure 15: Filtrando la imágen con ruido de Poisson

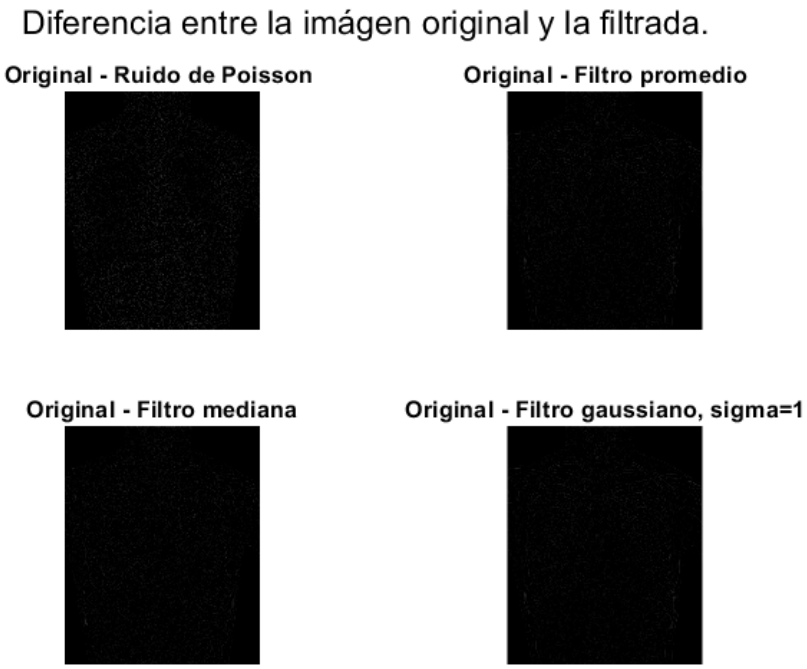

Figure 16: Diferencia entre la imagen con ruido y la filtrada.

```
num_diff1 =

    0.6031


num_diff2 =

    0.6090


num_diff3 =

    0.6013


num_diff4 =

    0.6153
```

Figure 17: Error calculado entre la imagen con ruido y la filtrada.

En esta parte se utilizó la imágen con ruido de Poisson. Se puede observar que el la imágen original comparada con la imágen con este ruido presnetabá un error de 0.6031; después de agregar el filtrado no se redujo el error existente.

IMSaltPeper:

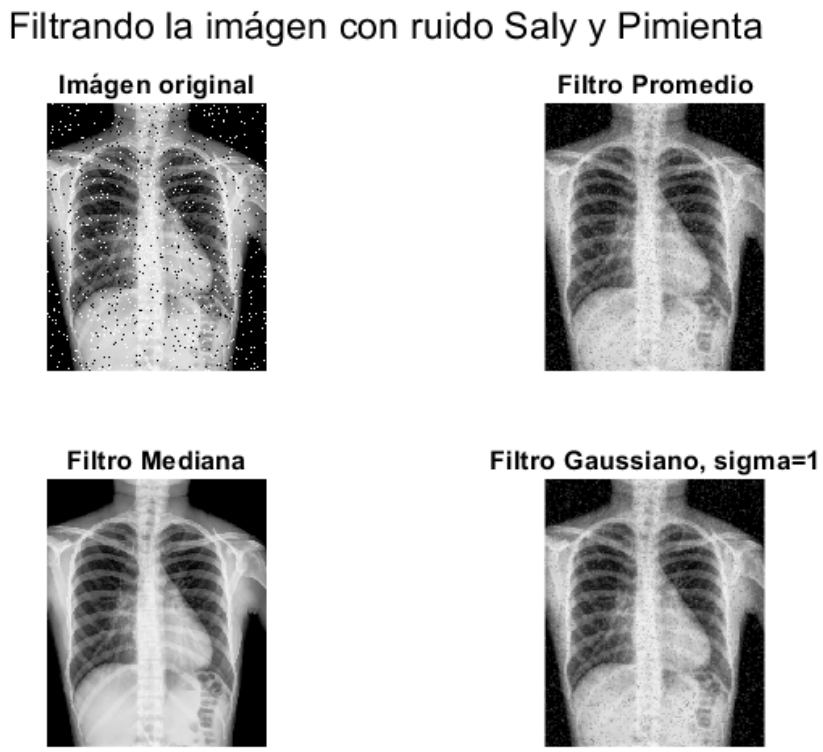

Figure 18: Filtrando la imágen con ruido de Saly y Pimienta

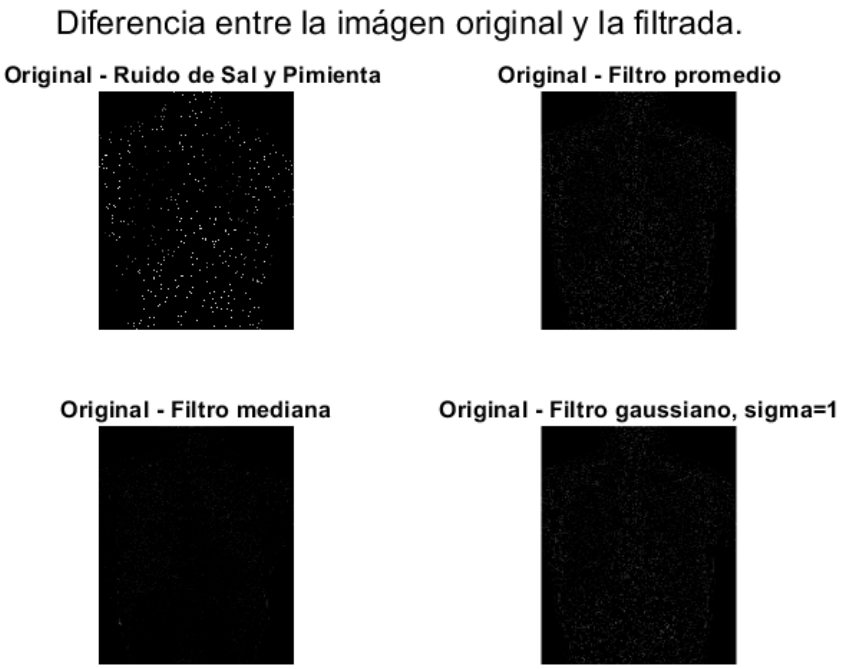

Figure 19: Diferencia entre la imagen con ruido y la filtrada.

```
num_diff1 =

    0.9793

num_diff2 =

    0.6171

num_diff3 =

    0.7233

num_diff4 =

    0.5949
```

Figure 20: Error calculado entre la imagen con ruido y la filtrada.

En esta parte se utilizó la imágen con ruido de Sal y & Pimienta. Se puede observar que el la imágen original comparada con la imágen con este ruido presnetabá un error de 0.9793; después de agregar el filtrado a la imágen esta diferencia llego a reducirse hasta 0.5949 utilizando el filtro gaussiano con sigma=1, aunque el filtro promedio es bunea opción también.

3 Realce de bordes en las imágenes

El siguiente código presenta un ejemplo simple de opciones para realzar los bordes y texturas de una imagen en escala de grises en MATLAB:

```
% cargar la imagen
img=imread("BrainMR.png");
% convertir a escala de grises
grayimg=im2gray(img);
% crear un kernel para filtro de realce 3x3
h1 = [0 -1 0; -1 5 -1; 0 -1 0];
% filtrar la imagen con filtro de realce
real_filt_1 = uint8(conv2(grayimg,h1,'same'));
% crear otro kernel para filtro de realce 3x3
h2 = [-1 -1 -1; -1 9 -1; -1 -1 -1];
% filtrar la imagen con nuevo filtro de realce
real_filt_2 = uint8(conv2(grayimg,h2,'same'));
% visualizar la imagen y sus versiones filtradas
figure(1)
subplot(2,2,1)
imshow(grayimg);
subplot(2,2,3)
imshow(real_filt_1);
subplot(2,2,4)
imshow(real_filt_2);
% guardar las imagenes filtradas
imwrite(real_filt_1,"BrainMR_real_1.png");
imwrite(real_filt_2,"BrainMR_real_2.png");
```

Ejercicio 5: Ejecute el código para obtener las versiones filtradas de la imagen BrainMR.png (adjunta a este enunciado), e incorpore los resultados obtenidos a su reporte.

```
% cargar la imagen
img=imread("BrainMR.png");
% convertir a escala de grises
grayimg=im2gray(img);
% crear un kernel para filtro de realce 3x3
h1 = [0 -1 0; -1 5 -1; 0 -1 0];
% filtrar la imagen con filtro de realce
real_filt_1 = uint8(conv2(grayimg,h1,'same'));
% crear otro kernel para filtro de realce 3x3
```

```
h2 = [-1 -1 -1; -1 9 -1; -1 -1 -1];
% filtrar la imagen con nuevo filtro de realce
real_filt_2 = uint8(conv2(grayimg,h2,'same'));
% visualizar la imagen y sus versiones filtradas
figure(1)
subplot(2,2,1)
imshow(grayimg);
title('Original')
subplot(2,2,3)
imshow(real_filt_1);
title('Filtro de realce')
subplot(2,2,4)
imshow(real_filt_2);
title(' Nuevo filtro de realce');
% guardar las imagenes filtradas
imwrite(real_filt_1,"BrainMR_real_1.png");
imwrite(real_filt_2,"BrainMR_real_2.png");
sgtitle('Realce de bordes y texturas del cerebro');
```

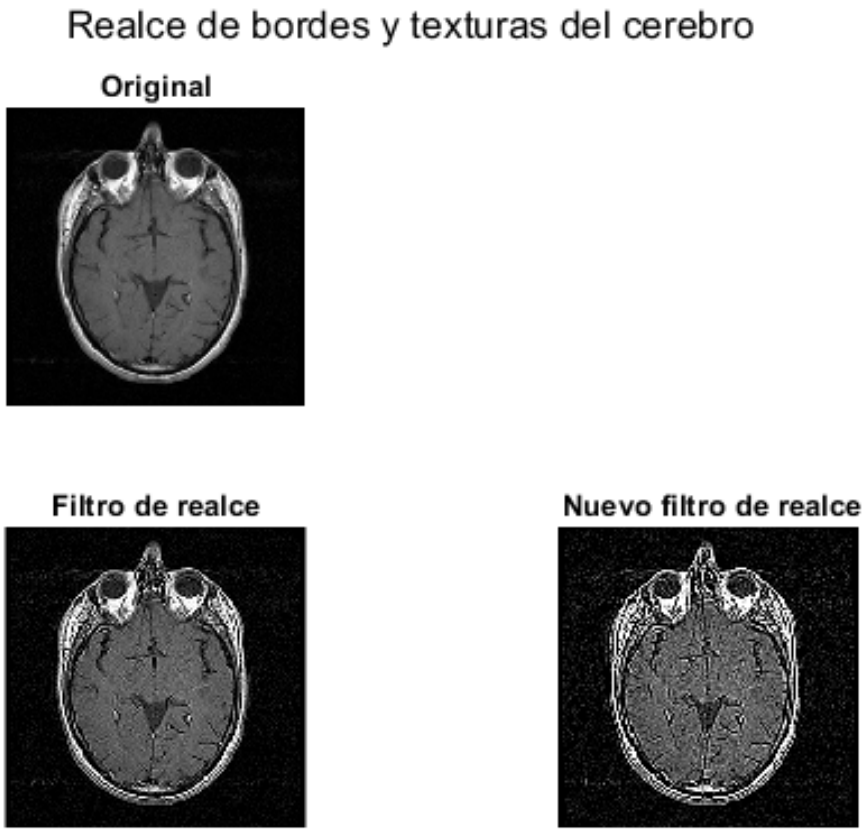

Figure 21: Realce de bordes y texturas del cerebro

Ejercicio 6: Ejecute el código anterior con otra imagen (médica) diferente, para obtener sus versiones con realce, e incorpore la imagen utilizada y los resultados obtenidos a su reporte.

```
% cargar la imagen
img=imread("ImagenMedica.jpg");
% convertir a escala de grises
grayimg=im2gray(img);
% crear un kernel para filtro de realce 3x3
h1 = [0 -1 0; -1 5 -1; 0 -1 0];
% filtrar la imagen con filtro de realce
real_filt_1 = uint8(conv2(grayimg,h1,'same'));
% crear otro kernel para filtro de realce 3x3
h2 = [-1 -1 -1; -1 9 -1; -1 -1 -1];
% filtrar la imagen con nuevo filtro de realce
real_filt_2 = uint8(conv2(grayimg,h2,'same'));
% visualizar la imagen y sus versiones filtradas
```

```
figure(1)
subplot(2,2,1)
imshow(grayimg);
title('Original')
subplot(2,2,3)
imshow(real_filt_1);
title('Filtro de realce')
subplot(2,2,4)
imshow(real_filt_2);
title(' Nuevo filtro de realce');
% guardar las imagenes filtradas
imwrite(real_filt_1,"ImagenMedica_real_1.png");
imwrite(real_filt_2,"ImagenMedica_real_2.png");
sgtitle('Realce de bordes y texturas de la imagen medica elegida');
```

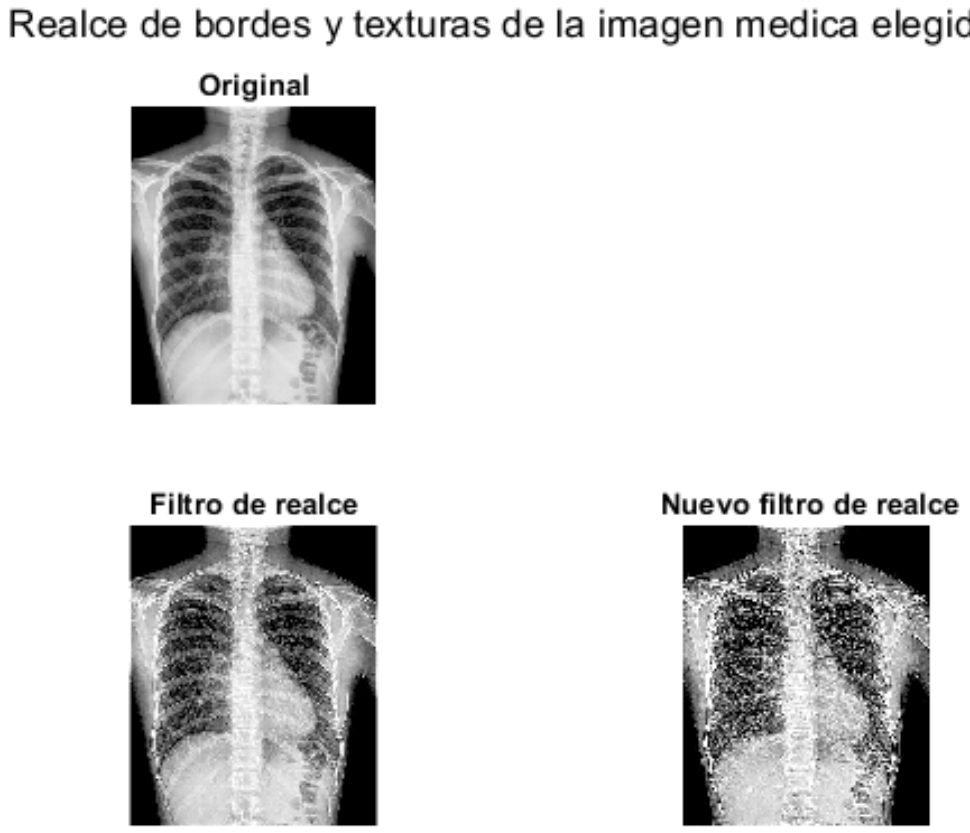

Figure 22: Realce de bordes y texturas de la imagen medica elegida

Se puede observar que el primer filtro de realce, resalta los bordes en la imagen. Al aplicarlo, se enfatizan los cambios bruscos en la intensidad de los píxeles, lo que generalmente realza los bordes. Sin embargo,

introduce un cierto nivel de ruido y amplifica las pequeñas fluctuaciones de intensidad. Por otra parte, el nuevo filtro de realce es más agresivo en su realce y da lugar a un efecto de nitidez más pronunciado en la imagen. Esto hace que los bordes sean más visibles, pero también aumenta el ruido y las imperfecciones como se puede ver claramente en la Imagen Medica que elegimos.