

Práctica 1:

Métodos Avanzados de Pre-procesado de Imagen Médica

Análisis de Imagen Médica

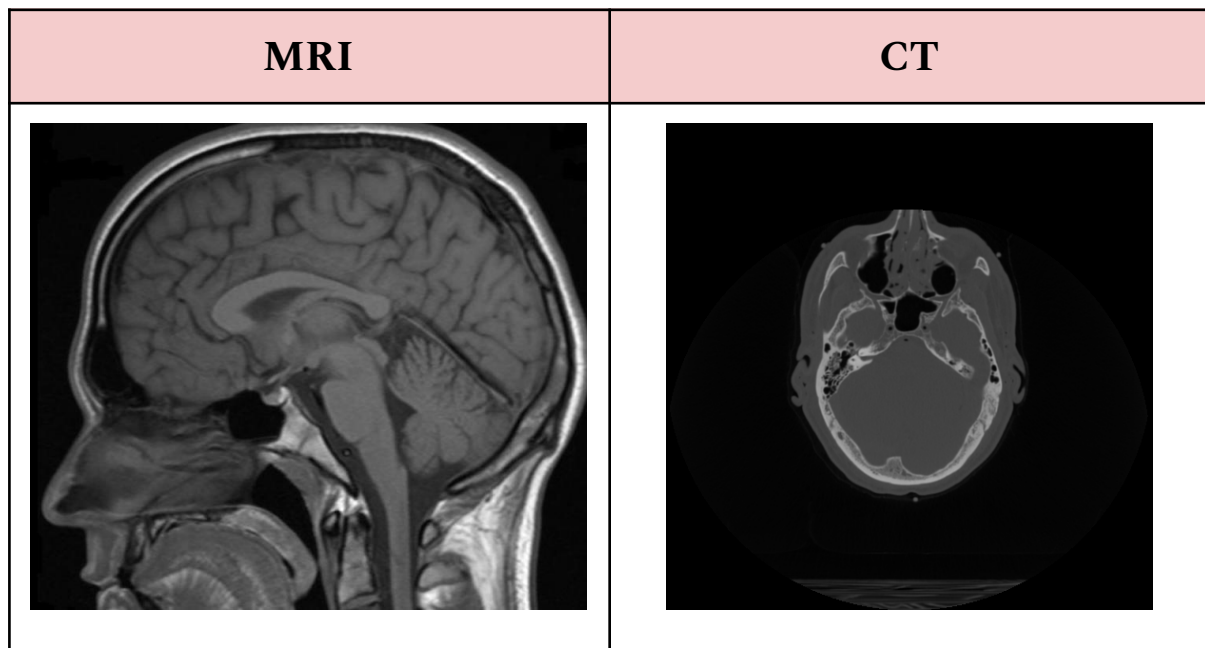


**Universidad
Rey Juan Carlos**

Sofía Gutiérrez Díaz
Teresa Vargas Rodríguez
Daniel Mancheño Castaño

0. Importar imágenes y librerías. [1], [2], [11]:

Comenzamos importando las librerías que se van a utilizar para construir el código en *python3*, cómo serían *numpy* y *scikit-image*. Las imágenes a filtrar, una imagen de resonancia magnética (MRI) y una de tomografía computarizada (CT), se han importado con la librería de *scikit-image*.



1. Adición de ruido:

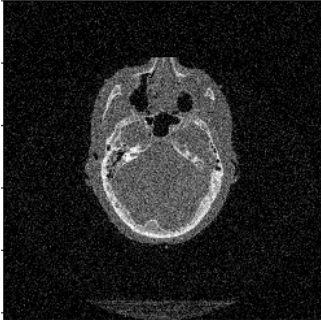
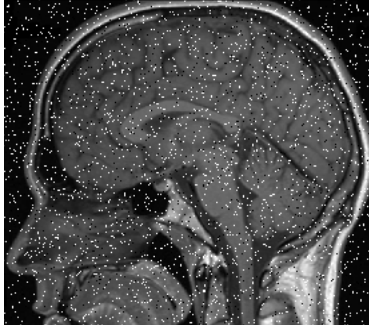
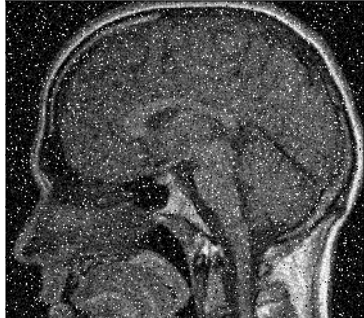
Previamente a la adición de ruido gaussiano e impulsivo se ha normalizado la imagen de manera que la intensidad de los píxeles se distribuya en el intervalo $[0,1]$. Para la implementación de ambos tipos de ruido se ha recurrido de nuevo a diferentes módulos de la librería de *scikit-image*. Además, se han modificado diferentes parámetros del módulo *skimage.util.random_noise* [2] para visualizar su impacto en la visualización de las imágenes:

1.1 Ruido Gaussiano [2]:

Modificando el parámetro *var* podemos controlar la intensidad del ruido gaussiano. Al aumentar la varianza, la probabilidad de que un valor de gris se vea afectado aumenta, y viceversa.

1.2 Ruido Impulsivo (salt & pepper) [2]:

Modificando el parámetro *amount* podemos cambiar la proporción de píxeles afectados por este tipo de ruido. El parámetro *salt_vs_pepper*, en cambio, afecta a la proporción de *salt* (píxeles blancos) y *pepper* (píxeles negros), de manera que valores más cercanos al 1 generan más píxeles blancos y valores más cercanos al 0 generan más píxeles negros.

GAUSSIANO	IMPULSIVO	MIXTO
		

2. Filtrado Non-Local Means:

2.1 Funcionamiento del filtro:

Para reducir el ruido se ha recurrido a la función *denoise_nl_means* [3]. Esta técnica de filtrado ayuda a eliminar el ruido de tipo aleatorio o gaussiano de una imagen, e implica realizar el promedio ponderado de todos los píxeles de la imagen en función de su similitud de intensidad al píxel a filtrar. La expresión matemática sería la siguiente:

$$NL[v](i) = \sum_{j \in I} w(i, j) v(j)$$
$$w(i, j) = \frac{1}{Z(i)} e^{-\frac{d(i, j)}{h^2}}$$

Aquí, la h sería un parámetro de similitud asociado al grado de filtrado que deseamos aplicar y $Z(i)$ una constante de normalización relacionada con los pesos $w(i,j)$.

2.2 Parámetros [4]:

La función *denoise_nl_means* implementada toma por argumento tanto el tamaño del parche (*patch_size*) como el parámetro h .

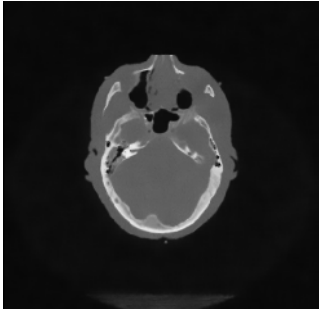
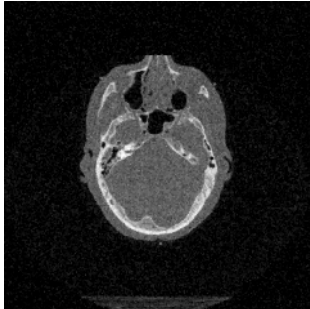
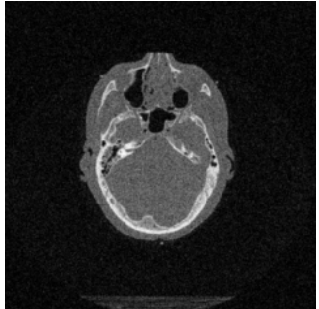
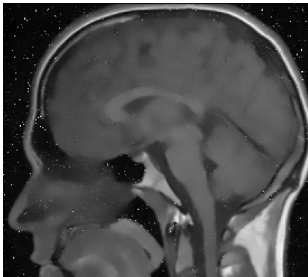
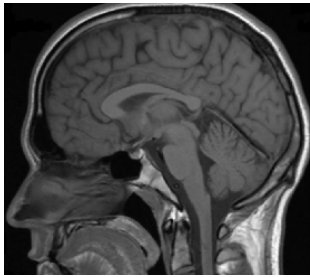

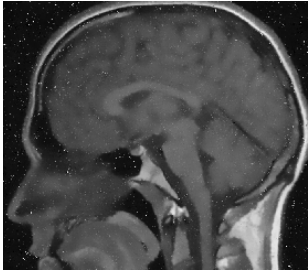


Modificando ambos valores se puede regular el nivel de filtrado de la imagen: a mayor valor de h y de tamaño de ventana, mayor es el difuminado de la imagen. Esto se debe a que cuanto más píxeles se tengan en cuenta a la hora de realizar el promedio, menor peso tendrá la intensidad del píxel filtrado. De igual manera, el parámetro h contribuye al peso de cada término en el promediado: al aumentar h se aumenta el grado de filtrado, lo que conlleva una mayor homogeneización del nivel de grises.

Aquí vemos un compromiso entre la reducción de ruido y la conservación de bordes. Finalmente hemos optado por perder resolución espacial, de manera que se dé preferencia a la eliminación de ruido frente a preservación de detalles finos. Para ello hemos elegido unos valores altos tanto de tamaño del parche como del parámetro h .

2.3 Comparativa con filtros convencionales [3]:

El filtrado NLM ofrece generalmente mejor resolución de bordes y mayor eliminación de ruido en imágenes afectadas por ruido gaussiano, tal y como se observa para la imagen de CT. No obstante, esta preservación de detalles finos se pierde a la hora de enfrentarse al efecto *salt & pepper*. Este defecto se ve acentuado debido a la alta intensidad del ruido que contamina nuestras imágenes de resonancia. Para neutralizar dicho ruido impulsivo es necesario establecer un valor demasiado alto de parámetro h , de manera que para eliminar al completo este ruido las imágenes contaminadas pierden resolución en exceso. Así, y teniendo en mente que el filtro de la **mediana** y de la **media** es más efectivo a la hora de neutralizar el ruido de sal y pimienta, determinamos que el filtro de la mediana funciona mejor en este caso. La eficiencia de estos filtros convencionales frente al ruido impulsivo se debe a que la mediana es un filtro resistente frente a valores atípicos y la media realiza un

promediado de todos los píxeles por igual, de manera que los píxeles afectados por el efecto de sal y pimienta apenas contribuyen a la imagen filtrada.

Tipo de Ruido	Filtrado NLM	Filtro de Mediana	Filtro de Media
Gaussiano			
Impulsivo			
Mixto			

3. Filtrado Anisotrópico: Perona y Malik:

3.1 Funcionamiento del filtro [5], [6], [8]:

Los filtros de difusión isotrópica para la eliminación de ruido son invariantes en el espacio, de manera que toda la imagen se suaviza de manera uniforme, resultando en una pérdida de resolución espacial importante. Para solucionar este problema se puede recurrir a los filtros de difusión anisotrópica. Mediante la implementación de derivadas parciales, estos filtros mejoran la preservación de bordes y detalles finos,

pues el suavizado ahora depende del grado de variación de la intensidad en la imagen. De esta manera, aquellas zonas con un gradiente alto, tal y como serían los bordes, sufrirán menos difusión que aquellas áreas más homogéneas, donde se encuentran gradientes más bajos.

3.2 Parámetros [5], [6], [8]:

El filtrado de Perona y Malik es uno de estos filtros de difusión anisotrópica iterativa. Permite eliminar el ruido de la imagen sin comprometer la resolución espacial, siempre que los parámetros del modelo se elijan correctamente. El modelo se rige por la siguiente expresión matemática:

$$\frac{\partial I}{\partial t} = \text{div} (c(x, y, t) \nabla I) = \nabla c \cdot \nabla I + c(x, y, t) \Delta I$$

Donde ∇ se refiere al Laplaciano, Δ al Gradiente, $\text{div}()$ al operador divergencia y $c(x, y, t)$ al coeficiente de difusión. Las dos funciones propuestas inicialmente para este coeficiente son:

$$c(\|\nabla I\|) = e^{-(\|\nabla I\|/K)^2}$$

and

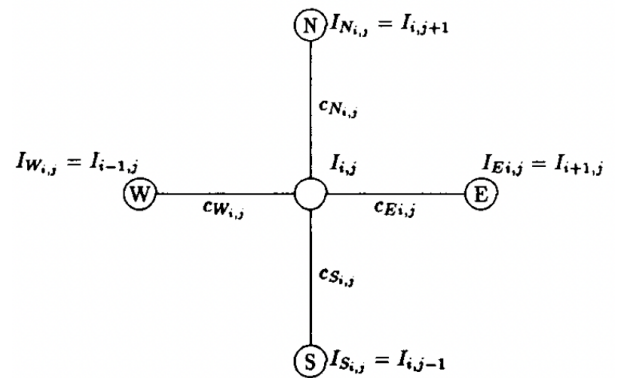
$$c(\|\nabla I\|) = \frac{1}{1 + \left(\frac{\|\nabla I\|}{K}\right)^2}$$

En estas expresiones reside el principal cambio en el paradigma del filtrado: mientras que en la difusión isotrópica este coeficiente se asume constante, en la difusión anisotrópica es una función que impide la difusión de bordes (toma valores cercanos a 0) y permite el emborronamiento de zonas homogéneas (toma valores cercanos a 1)

El parámetro K se puede interpretar como un umbral, que define cuando una región es reconocida como un borde: a mayor K , menor es la diferencia de intensidad requerida entre píxeles para que se identifique como un borde.

Para definir la función exacta de estos coeficientes se debe estimar la localización de los bordes, de manera que en cada una de las iteraciones dicha estimación se actualiza. Cada iteración se aplica a los 4 píxeles adyacentes al que se quiere filtrar:




$$I_{i,j}^{t+1} = I_{i,j}^t + \lambda [c_N \cdot \nabla_N I + c_S \cdot \nabla_S I + c_E \cdot \nabla_E I + c_W \cdot \nabla_W I]_{i,j}^t$$



Consecuentemente, modificando el parámetro *lambda* se consigue regular la intensidad del filtrado.

3.3 Comparativa con filtros convencionales [7]:

Tipo de Ruido	Filtrado de Perona y Malik	Filtro de Mediana	Filtro de Media
Gaussiano			
Impulsivo			

<p>Mixto</p>			
---------------------	---	--	---

El filtro de la **mediana** demuestra nuevamente que es el método más efectivo a la hora de lidiar con ruido impulsivo. No obstante, el filtrado anisotrópico logra tratar mejor el ruido gaussiano, aunque la diferencia es mínima. En cuanto a la imagen mixta, debido a que la intensidad del ruido impulsivo domina frente al ruido gaussiano, es el filtro de la mediana aquel que devuelve una imagen ligeramente más limpia y con bordes mejor definidos.

En comparación con el filtro de la **media**, el filtrado anisotrópico parece ser más eficiente neutralizando tanto ruido gaussiano como ruido impulsivo. Ambos filtros parecen tener el mismo efecto sobre los bordes, de manera que ninguno causa una distorsión significativa.

4. Otros métodos de filtrado avanzado: Filtro Bilateral

4.1 Funcionamiento del filtro [9]:

El filtro bilateral es una técnica de filtrado no lineal que, de manera similar al filtro de Perona y Malik, busca reducir el ruido sin perder resolución espacial. La idea principal reside en combinar dos filtros Gaussianos, uno espacial y otro de intensidad. De esta forma, para que un píxel influya en el valor de otro no solo debe ser un píxel vecino, sino que también debe tener un nivel de intensidad similar. El principal problema del filtrado bilateral viene dado por su alto coste computacional:

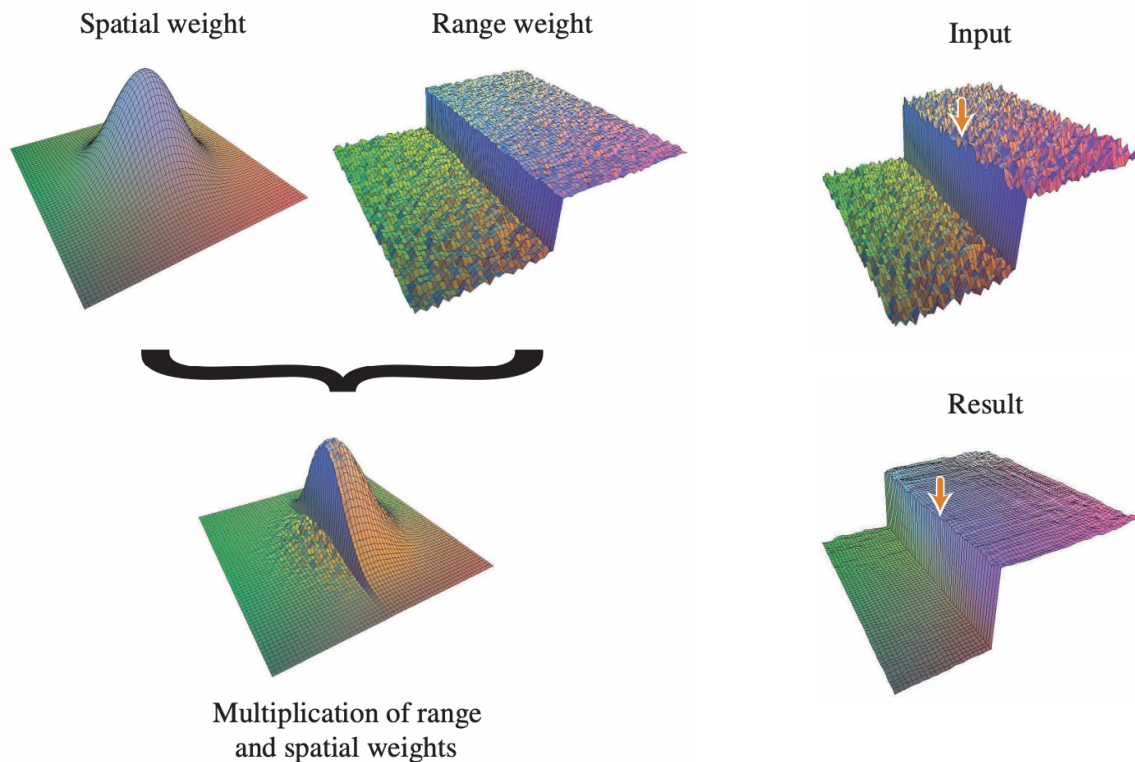
Se realiza una suma ponderada de las intensidades de los píxeles adyacentes al píxel a filtrar, donde el peso de cada término es inversamente proporcional a la distancia entre el píxel vecino y el píxel central. A medida que la distancia entre ambos píxeles

aumenta, la contribución de la intensidad del píxel vecino a la imagen filtrada disminuye. En el filtrado bilateral, la distancia representa tanto localización espacial como diferencia en niveles de gris. La expresión matemática que combina ambos filtros Gaussianos es la siguiente:

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in \mathcal{S}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|) I_{\mathbf{q}},$$

Donde G_{σ_s} representa el filtro gaussiano espacial, G_{σ_r} el filtro gaussiano de intensidades, p y q representan el píxel central y píxeles vecinos respectivamente, I_p y I_q sus correspondientes intensidades, y W_p es una constante de normalización.

Una representación visual del efecto conjunto de ambos filtros Gaussianos sería:

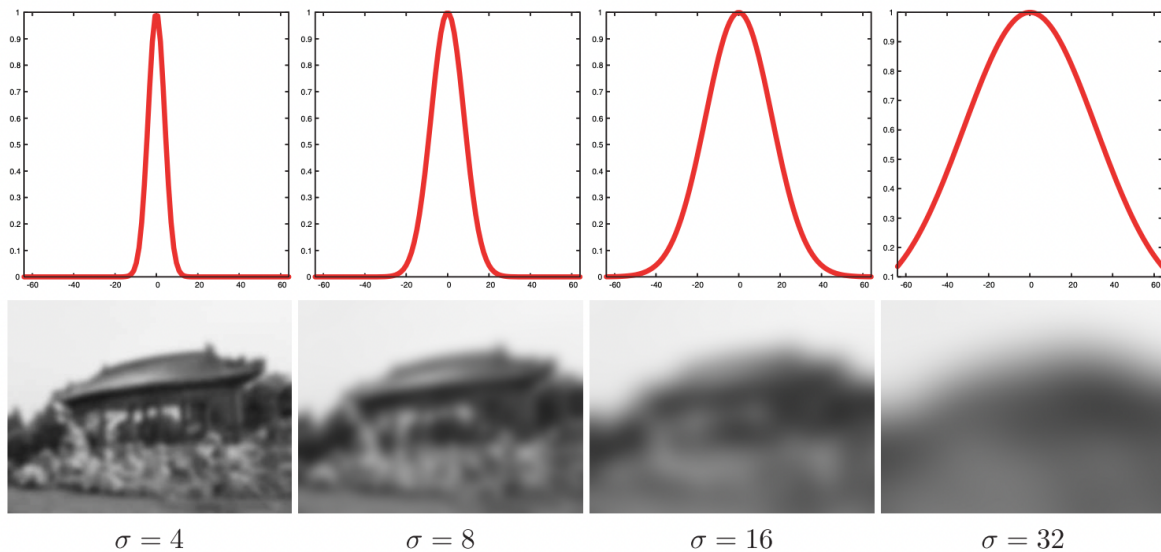


Estos pesos permiten al filtro preservar los bordes, puesto que al ser áreas con diferencias significativas en intensidades, el peso del filtro Gaussiano de intensidad reduce la influencia de los píxeles al otro lado del borde. En el resto de la imagen,

donde la intensidad fluctúa mínimamente, el filtro suaviza el ruido, actuando como un filtro Gaussiano corriente.

4.2 Parámetros [9]:

El parámetro fundamental que rige este modelo es la desviación estándar (σ) de los filtros Gaussianos, puesto que define el peso que se le asignará a los píxeles vecinos en función de su cercanía al píxel central:



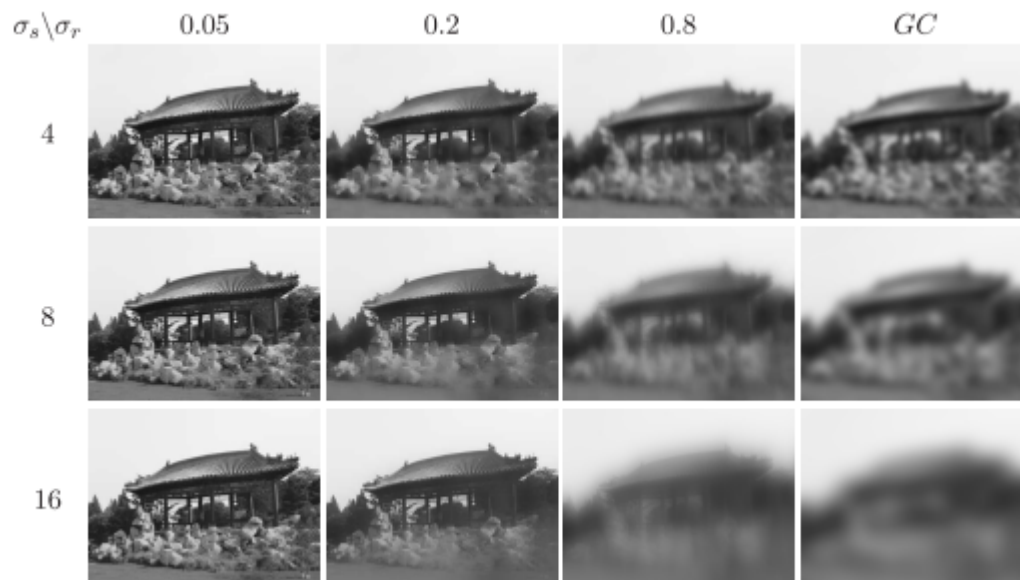
A medida que aumenta σ , la influencia de píxeles más lejanos también aumenta, de manera que se incrementa el suavizado de la imagen.

Una característica importante del filtro bilateral es que los pesos (ambos filtros Gaussianos) se multiplican, por lo que si alguno de ellos es cercano a 0, no se produce suavizado.

De esta manera, a medida que aumenta el parámetro de intensidad (σ_r), el filtro se va aproximando a la convolución Gaussiana. Este parámetro determina la influencia de los píxeles vecinos en el píxel a filtrar basándose en su diferencia de intensidad: si un píxel vecino tiene una diferencia muy alta en intensidad respecto al píxel a filtrar, este tendrá un menor peso, y en consecuencia una menor influencia.

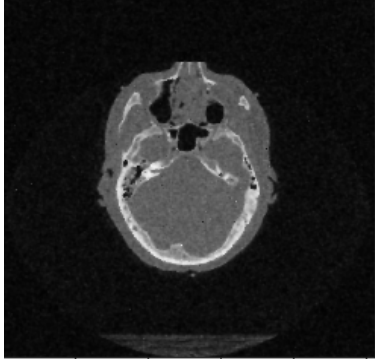
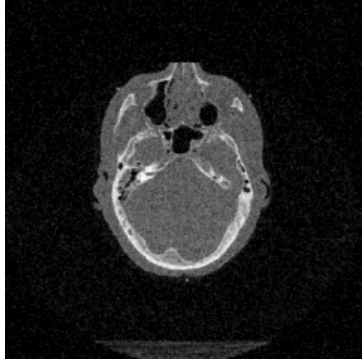


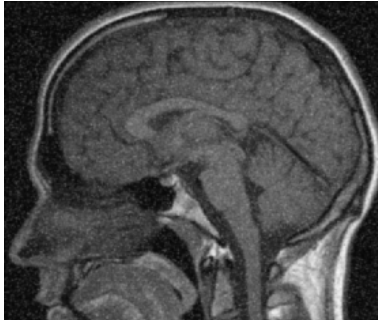

Lo mismo sucede con el parámetro espacial σ_s ; al aumentarlo se suaviza más la imagen, ya que asigna mayor peso a píxeles lejanos (incluso si sus intensidades son bastante diferentes), provocando que la imagen se emborrone más fácilmente.

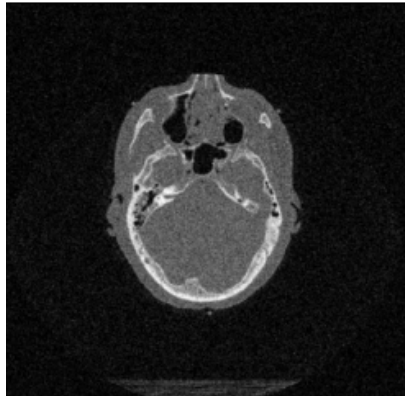
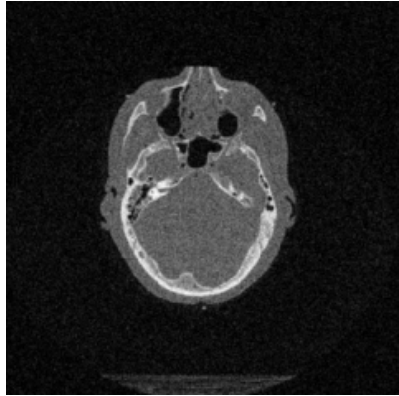




En consecuencia, el aumento del parámetro espacial no desdibujará un borde mientras que el parámetro de intensidad sea menor que la amplitud del borde:



Estos dos parámetros proporcionan un control mayor en el filtro Gaussiano convencional, puesto que podemos decidir el peso que tienen los píxeles cercanos al píxel a filtrar, determinando así su influencia en el mismo. En el momento en el que cualquiera de los pesos del filtro bilateral alcance valores cercanos a cero, no se produce suavizado.

4.3 Comparativa con otros filtros [10]:

Tipo de Ruido	Filtrado de Perona y Malik	Filtro de Mediana
Gaussiano		
Impulsivo		
Mixto		

Tipo de Ruido	Filtrado Bilateral	Filtro de Media
Gaussiano		
Impulsivo		
Mixto		

Una vez más, el filtro de la mediana, es el que mejores resultados aporta la hora de filtrar el **ruido impulsivo**. Además, también genera una imagen ligeramente más limpia que con el resto de métodos a la hora de eliminar el **ruido mixto** debido a que este tiene una mayor componente de ruido *s&p* (impulsivo).

En cuanto al **ruido gaussiano**, podemos observar que a simple vista, todos los métodos proporcionan un filtrado similar, aunque el filtrado de Perona y Malik reduce levemente más el grano de la imagen.

5. Comparación final

Una vez vistos todos los filtros, podemos hacer una evaluación sobre cuál es mejor para cada tipo de ruido y cuál preserva mejor los bordes.

Como hemos podido observar, el **filtro NLM**, permite una buena eliminación del ruido gaussiano aunque sacrificando los bordes de la imagen -si la imagen tiene mucho detalle, como es el caso de la imagen de la cabeza en plano transversal, posiblemente se pierda bastante información debido al suavizado-, mientras que el resto de filtros no consiguen el mismo nivel de filtrado, pero si uno aproximado. Siempre y cuando la imagen a filtrar no tenga mucho detalle, este filtro sería una buena opción.

El **filtrado de media**, al igual que el **filtrado bilateral**, son capaces de filtrar imágenes contaminadas con ruido gaussiano e impulsivo por igual, aunque dejando un ligero componente de grano. Concluimos por tanto que se conservan bien los bordes, siempre y cuando se ajusten adecuadamente los parámetros. No obstante, se deberá llegar a un equilibrio entre preservación de resolución y eliminación de ruido.

El **filtrado de Perona y Malik**, proporciona imágenes filtradas con un menor nivel de grano en comparación con las imágenes de los filtros de media y bilateral, sin dejar de preservar los bordes.

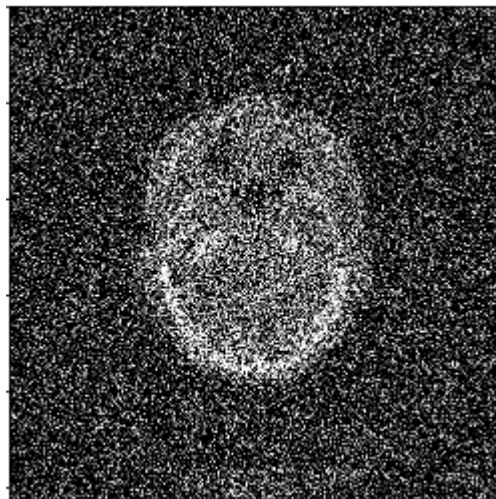
Por último, el resultado de utilizar el **filtrado de mediana** en las imágenes con ruido gaussiano, es ligeramente peor que el que ofrecen el resto de filtros, aunque su efectividad a la hora de lidiar con ruido impulsivo es indiscutible. También se puede decir que la preservación de bordes con este filtro es muy buena.

6. Extra: utilización de varios filtros

El empleo de un único filtro para eliminar el ruido puede ser efectivo en situaciones con niveles moderados de ruido. No obstante, cuando tenemos niveles elevados de ruido, esto puede no ser suficiente. Por ello, hemos considerado la aplicación combinada de varios filtros para mejorar la calidad de la imagen.

Primero hemos utilizado distintos filtros de manera individual, y posteriormente hemos analizado las diferencias existentes entre estas imágenes y las filtradas con dos filtros de manera consecutiva.

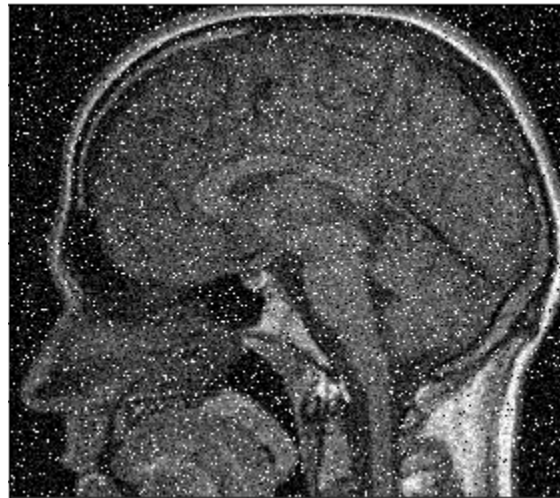
Imagen con ruido:


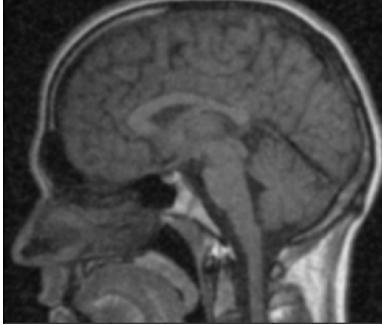



Tipo de Ruido	Filtro de Mediana	Filtro NLM	Filtro de Mediana + Filtro NLM
Gaussiano			

En otro ejemplo:

Imagen con ruido:



Tipo de Ruido	Filtro de Mediana	Filtro Perona y Malik	Filtro de Mediana + Filtro PyM
Mixto			

7. Bibliografía

Librerías:

[1] Numpy: (No date) *NumPy*. Available at: <https://numpy.org/> (Accessed: 16 October 2023).

[2] Scikit-Image: *News* (no date) *scikit*. Available at: <https://scikit-image.org/> (Accessed: 16 October 2023).

Filtro de NLM:

[3] *Skimage.restoration#* (no date) *skimage.restoration - skimage 0.22.0 documentation*.

Available at:

https://scikit-image.org/docs/stable/api/skimage.restoration.html#skimage.restoration.denoise_nl_means (Accessed: 16 October 2023).

[4] Buades, A.; Coll, B.; Morel, J.M. Non-Local Means Denoising. *Image Processing On Line*, 1 (2011), pp. 208–212.

Filtro de Perona-Malik:

[5] Kamalaveni, V., Rajalakshmi, R.A. and Narayanankutty, K.A. (2015) ‘Image denoising using variations of Perona-Malik model with different edge stopping functions’, *Procedia Computer Science*, 58, pp. 673–682. doi:10.1016/j.procs.2015.08.087.

[6] (No date a) *Perona-Malik/RE4017_Project1.pdf at master · Pastapleton ... - github*.

Available at:

https://github.com/pastapleton/Perona-Malik/blob/master/RE4017_Project1.pdf

(Accessed: 16 October 2023).

[7] (No date a) *GitHub*. Available at:

<https://github.com/fubel/PeronaMalikDiffusion/blob/master/main.py> (Accessed: 16 October 2023).

[8] Perona, P. and Malik, J. (1990) ‘Scale-space and edge detection using anisotropic diffusion’, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7), pp. 629–639. doi:10.1109/34.56205.

Filtro Bilateral:

[9] Bilateral Filtering: Paris, S. *et al.* (2008) ‘Bilateral filtering: Theory and applications’, *Foundations and Trends® in Computer Graphics and Vision*, 4(1), pp. 1–75. doi:10.1561/06000000020.

[10] *Skimage.restoration#* (no date a) *skimage.restoration - skimage 0.22.0 documentation*. Available at: https://scikit-image.org/docs/stable/api/skimage.restoration.html#skimage.restoration.denoise_bilateral (Accessed: 16 October 2023).

Imágenes:

[11] Alberto Olivares González, Profesor de la Universidad Rey Juan Carlos, Área de Estadística e Investigación Operativa del Departamento de Teoría de la Señal y Comunicaciones y Sistemas Telemáticos y Computación