

GRADO EN FÍSICA

Reducción de ruido en imágenes cancerosas mediante CNN tipo U-NET

Presentado por:

ALEJANDRO VILAR ALFONSO

Dirigido por:

ALEJANDRO CALABUIG BARROSO

CURSO ACADÉMICO 2024-2025

Resumen

En este estudio se desarrolla y se evalúa una red neuronal tipo U-Net para reducir el ruido gaussiano en imágenes de tomografía computarizada (TC), la motivación de este trabajo reside en la necesidad de mejorar el diagnóstico. Se revisarán los fundamentos de la TC y los fundamentos físicos que ocasionan ruido, donde en la metodología se detalla la generación controlada de ruido y el diseño y entrenamiento de la red neuronal, evaluando sus resultados mediante métricas de error, así como el desarrollo de una taxonomía de técnicas de eliminación de ruido.

Abstract

In this study, a U-Net neural network is developed and evaluated to reduce Gaussian noise in computed tomography (CT) images. The motivation for this work lies in the need to improve diagnosis. The fundamentals of CT and the physical foundations that cause noise will be reviewed, where the methodology details the controlled generation of noise and the design and training of the neural network. Its results are evaluated using error metrics, as well as the development of a taxonomy of noise removal techniques.

Índice general

1	Introducción	4
1.1	Contexto y motivación del proyecto	4
2	Objetivos	5
2.1	Objetivo general	5
2.2	Objetivos específicos	5
3	Marco teórico	7
3.1	Tomografía computarizada	7
3.1.1	Reconstrucción de la imagen	15
3.1.2	Diferentes tipos avanzados de tomografía computarizada	19
3.2	Fundamentos de la generación del ruido gaussiano	21
3.2.1	Conversión parcial de energía mediante la interacción de los fotones con la materia	21
3.2.2	Eficiencia cuántica del detector	31
4	Metodología	35
4.1	Explicación fundamental de una red neuronal	35
4.2	Redes neuronales convolucionales	39
4.3	Metodología utilizada	40
4.3.1	Preparación de datos y ruido gaussiano	41
4.3.2	Modelo de red neuronal U-NET	45
5	Resultados y discusión	48
5.1	Tipo de arquitectura utilizada	48
5.2	Resultados del modelo	53
5.3	ODSs empleadas	58

6 Conclusiones	60
Anexos	64
A Taxonomía de técnicas de eliminación de ruido	64
A.1 Técnicas básicas	64
A.1.1 Filtro promedio	64
A.1.2 Filtro mediano	66
A.1.3 Filtro bilateral	67
A.1.4 Filtro gaussiano	69
A.1.5 Transformada de Radón	70
A.1.6 Transformadas de Fourier y Wavelet	71
A.2 Técnicas avanzadas	73
A.2.1 Non-Local Means (NLM)	73
A.2.2 PCA (Probabilistic Component Analysis o Principal Component Analysis)	75
B Código Python	77

1. Introducción

1.1. Contexto y motivación del proyecto

La detección temprana del cáncer es considerada como un factor que aumenta la tasa de supervivencia de los pacientes, por tanto, es necesario que se realice un diagnóstico preciso, esto a menudo es complicado debido a la presencia de ruido en las imágenes médicas.

Las imágenes medicas pueden ser obtenidas por tomografía computarizada (CT), resonancias magnéticas (RM), etc, donde el ruido dificulta la identificación de tumores u otras características relevantes, reduciendo así la eficiencia de los sistemas de diagnóstico asistidos por computadora.

Las redes neuronales profundas, en concreto las que poseen arquitectura convolucional tipo U-NET, está demostrado que son altamente efectivas para tareas de segmentación y restauración de imágenes médicas de diagnóstico, ya que son capaces de aprender patrones complejos pudiendo así eliminar el ruido y representar una imagen de alta calidad, este trabajo de fin de grado se enfocará en fundamento físico de la tomografía computarizada, así como de su generación del ruido, posteriormente se realizará una taxonomía de técnicas de eliminación de ruido, y por último el uso de una red neuronal tipo U-ET con imágenes tomadas por CT pertenecientes a cáncer de pelvis.

La motivación detrás de este trabajo radica en la necesidad urgente de mejorar la precisión del diagnóstico médico mediante el uso de tecnologías avanzadas de procesamiento de imágenes. Con la creciente disponibilidad de grandes bases de datos de imágenes médicas y el avance de las técnicas de inteligencia artificial, las redes neuronales profundas como U-NET pueden ofrecer soluciones efectivas a problemas persistentes en la medicina moderna, mejorando la calidad de vida de los pacientes al permitir diagnósticos más rápidos y certeros.

2. Objetivos

Este apartado se definirán de forma clara los objetivos que se pretenden alcanzar mediante este trabajo.

2.1. Objetivo general

Implementar y evaluar una metodología de eliminación de ruido mediante redes neuronales convolucionales tipo U-Net para conseguir reducir el ruido gaussiano en imágenes CT, con el fin de mejorar el diagnóstico.

2.2. Objetivos específicos

Para alcanzar el objetivo general, el estudio se organiza en los siguientes objetivos específicos:

Revisión bibliográfica y fundamentación teórica.

- Investigar los fundamentos físicos de la tomografía computarizada como modelo de reconstrucción de imagen, sus diferentes modalidades y las limitaciones inherentes.
- Analizar los mecanismos de generación de ruido gaussiano en sistemas de TC, incluyendo la interacción fotón–materia y la eficiencia cuántica del detector.
- Examinar métodos clásicos y avanzados de eliminación de ruido para identificar fortalezas y carencias.

Construcción y preparación del conjunto de datos.

- Seleccionar fuentes de imágenes TC con licencia apropiada o de dominio público.
- Diseñar un protocolo de generación controlada de ruido gaussiano de distintas intensidades para simular escenarios clínicos reales.

- Dividir los datos en conjuntos de entrenamiento, validación y prueba, garantizando una distribución equilibrada de casos y niveles de ruido.

Diseño e implementación de la arquitectura U-Net.

- Definir la topología de la red (profundidad, número de filtros, tamaños de kernel, conexiones de salto).
- Incorporar técnicas de regularización (dropout, batch normalization) y optimización (función de pérdida, tasa de aprendizaje, optimizador).

Entrenamiento y ajuste de hiperparámetros.

- Realizar experimentos de entrenamiento variando parámetros clave.
- Emplear técnicas para prevenir sobreajuste y determinar el mejor conjunto de hiperparámetros.
- Documentar curvas de aprendizaje y métricas intermedias durante el entrenamiento.

Evaluación comparativa y análisis de resultados.

- Medir cuantitativamente la calidad de las imágenes.

Integración con Objetivos de Desarrollo Sostenible (ODS).

- Identificar cómo la mejora en la calidad de imagen contribuye a la ODS 3 (“Salud y bienestar”) y ODS 9 (“Industria, innovación e infraestructura”).
- Proponer recomendaciones para la adopción de esta tecnología en entornos hospitalarios y su impacto en la eficiencia de diagnóstico.

3. Marco teórico

En esta sección se explicarán los fundamentos físicos de la evolución histórica de la tomografía computarizada, así como el proceso de reconstrucción de imagen y los diferentes tipos de CT que existen, por otro lado también se desarrollarán cuales son los fenómenos físicos que contribuyen a la generación del ruido gaussiano.

3.1. Tomografía computarizada

Las imágenes obtenidas por tomografía computarizada son el tipo de imagen que utilizaremos para el uso de la red neuronal tipo U-NET, esta técnica es considerada como uno de los mayores avances científicos en lo referente al diagnóstico médico originada por el descubrimiento de los rayos-X hecho por Röntgen, esta técnica fue tan innovadora debido a su uso de procedimientos digitales en el campo de la radiología médica, más allá de este método, también se expone el problema de la obtención de información sobre estructuras tridimensionales a través de la transmisión de rayos X, sin embargo esto nos resuelve varios problemas existentes previos a la invención de esta técnica.

En primer lugar mejoramos la eficiencia de absorción de las combinaciones pantalla-película, donde una película es un material que debido a sus propiedades fluorescencia cambia los rayos X en un espectro visible, y una película fotosensible funciona de tal manera que captura la imagen formada por la luz emitida por la pantalla intensificadora y su sensibilidad térmica afecta la eficiencia global del sistema, y la eficiencia de absorción, se define como la fracción de fotones de rayos X incidentes que son absorbidos y convertidos en luz en la pantalla intensificadora.

Para darle un enfoque más matemático, la probabilidad de que un fotón de rayos X sea absorbido en la pantalla viene dada por la ley de atenuación exponencial (1), ya que cuando un haz de rayos X atraviesa un material, su intensidad disminuye debido a la absorción y dispersión.

$$I(x) = I_0 e^{-\mu x} \quad (1)$$

donde:

- I_0 es la intensidad inicial que tienen los rayos X antes de atravesar la pantalla.
- $I(x)$ es la intensidad que tienen los rayos X después de atravesar un espesor x de la pantalla.
- μ es el coeficiente de atenuación lineal del material, el cual varía dependiendo de su composición y de la energía de los rayos X (cm^{-1}).
- x es el espesor de la pantalla (cm).

La eficiencia de absorción de la pantalla se define como la fracción de fotones de rayos X absorbidos en el material, dada por (2).

$$A = 1 - e^{-\mu x} \quad (2)$$

donde:

- A es la eficiencia de absorción, es decir, la fracción de fotones de rayos X que son absorbidos y no atraviesan la pantalla.

Si la pantalla es muy gruesa ($x \gg 1/\mu$), la absorción es casi total ($A \approx 1$), mientras que si la pantalla es delgada, parte de los rayos X puede atravesarla sin ser absorbidos, ahora bien existen unas combinaciones específicas de transporte de energía que transportan de un 25% aproximadamente que son las que se solían utilizar.

Cada fotón de rayos X absorbido excita electrones del material fluorescente, los cuales emiten múltiples fotones de luz visible al desexcitarse. La eficiencia cuántica de conversión se define como (3).

$$\eta_q = \frac{E_X}{E_L} = \frac{h\nu_X}{h\nu_L} = \frac{\nu_X}{\nu_L} \quad (3)$$

donde:

- η_q es la eficiencia cuántica de conversión, es decir, el número de fotones de luz visible emitidos por cada fotón de rayos X absorbido.
- $E_X = h\nu_X$ es la energía de un fotón de rayos X.
- $E_L = h\nu_L$ es la energía de un fotón de luz visible.
- h es la constante de Planck.
- ν_X es la frecuencia de los rayos X.
- ν_L es la frecuencia de la luz visible emitida.

Dado que la frecuencia de los rayos X (ν_X) es mucho mayor que la de la luz visible (ν_L), un solo fotón de rayos X puede generar cientos de fotones de luz visible.

Entonces como hemos visto no toda la energía del fotón de rayos X se convierte en luz visible; una parte se pierde en forma de calor u otras interacciones. La eficiencia de conversión de energía se define como (4).

$$\eta_E = \frac{E_{\text{luz emitida}}}{E_X} \quad (4)$$

donde:

- η_E es la eficiencia de conversión de energía, que representa la fracción de la energía de los rayos X que se convierte en luz visible.
- $E_{\text{luz emitida}}$ es la energía total de los fotones de luz visible generados.

- E_X es la energía del fotón de rayos X absorbido.

En pantallas convencionales, η_E suele estar entre el 5% y el 20%, lo que significa que gran parte de la energía de los rayos X se disipa en otras formas.

También existía una elevada contribución de la radiación dispersa a la hora de reconstruir la imagen, si utilizamos un haz de rayos X demasiado ancho, produce que la dispersión de fotones contribuya hasta un 50% del total que llega al sistema de reconstrucción e imagen, lo que genera ruido.

A día de hoy se sigue teniendo en cuenta el tamaño de los haces que administramos al paciente, ya que el motivo principal por el que se debe de minimizar esta dosis siempre que sea posible, es debido a que los rayos X presentan una interacción con los tejidos del cuerpo humano de la forma que generan radiación ionizante, lo que ocasiona que los átomos del cuerpo puedan perder o ganar electrones, esto es peligroso ya que entre otras cosas se puede generar cáncer, así como afectar a la estructura molecular del ADN.

Sin embargo cuando un rayo de este tipo atraviesa algún tejido, se puede transmitir sin causar ninguna interacción, su energía puede ser absorbida por el material debido a efectos como el fotoeléctrico o el Compton donde los fotones pueden cambiar su dirección.

La intensidad total que llega al detector es la suma de los fotones primarios (que no han interactuado con nada), es decir, no han cambiado su dirección desde su salida del haz hasta su entrada en el detector y dispersos (que si que han interactuado) (5).

$$I_{\text{total}} = I_{\text{primario}} + I_{\text{disperso}} \quad (5)$$

La fracción de fotones dispersos respecto al total puede expresarse como (6).

$$f_{\text{disperso}} = \frac{I_{\text{disperso}}}{I_{\text{total}}} \quad (6)$$

Donde I_{disperso} es la intensidad de los fotones dispersados, I_{total} es la intensidad total, y la dispersión depende del ancho angular $\Delta\theta$.

Para tratar como afecta la anchura del haz, la dispersión de Compton sigue la distribución de Klein-Nishina, lo que implica que los fotones dispersos pueden describir muchas trayectorias diferentes. El número de fotones dispersos que alcanzan al detector está relacionado con el tamaño del campo del haz (A).

La distribución de Klein-Nishina describe la distribución angular de los fotones dispersos por electrones en el proceso de dispersión Compton, que es una interacción en la que un fotón (por ejemplo un rayo X) se dispersa por un electrón, cambiando su dirección y energía. Fue propuesta por Oskar Klein y Yoshio Nishina en 1929, como una mejora sobre la fórmula clásica de Rayleigh, que solo describe la dispersión elástica. La distribución de Klein-Nishina es relevante cuando los fotones contienen alta energía, es decir, cuando los fotones involucrados tienen energías comparables o mayores que la energía de reposo del electrón.

La fórmula para la distribución de Klein-Nishina para la dispersión de fotones por electrones en función del ángulo de dispersión es (7).

$$\frac{d\sigma}{d\Omega} = \frac{\alpha r_e^2}{2} \left(\frac{\omega'}{\omega}\right)^2 \left(\frac{\omega'}{\omega} + \frac{\omega}{\omega'} - \sin^2 \theta\right) \quad (7)$$

Donde:

- $\frac{d\sigma}{d\Omega}$ es el diferencial de sección eficaz por unidad de ángulo sólido.
- α es la constante de estructura fina ($\alpha \approx \frac{1}{137}$).

- r_e es el radio clásico del electrón, $r_e \approx 2.817 \times 10^{-15}$ m.
- ω es la energía del fotón incidente.
- ω' es la energía del fotón dispersado.
- θ es el ángulo de dispersión entre el fotón incidente y el fotón dispersado.
- $d\Omega$ es el diferencial de ángulo sólido.

Relacionando esta distribución con el tamaño del haz, obtenemos que (8).

$$S \propto A = w \cdot h \quad (8)$$

Donde:

- w : Ancho del haz.
- h : Altura del haz.

Por tanto como vemos si w aumenta el área A y aumenta la cantidad de fotones dispersos que llegan a el detector.

El contraste de la imagen disminuye cuando la fracción de dispersión es alta. Se puede definir el factor de deterioro del contraste (C) debido a la dispersión como (9).

$$C = \frac{P}{S + P} \quad (9)$$

Donde:

- C : Contraste de la imagen, que indica la calidad visual en cuanto a la diferencia de intensidad entre las áreas de la imagen.

- P : Intensidad de los fotones primarios, es decir, los fotones que no han sufrido dispersión y forman la imagen útil.
- S : Intensidad de los fotones dispersos, los cuales han interactuado con el medio (típicamente debido al efecto Compton) y contribuyen al ruido en la imagen.

Si S (los fotones dispersos) aumenta, la relación de contraste C disminuye, lo que reduce la calidad de la imagen generando más ruido y menos diferencia de intensidad entre las áreas de la imagen.

La superposición y enmascaramiento de las imágenes, provoca que las estructuras tridimensionales no se solapen entre sí, generando así una pérdida de detalle en áreas de interés como puede ser la región de cáncer, también las áreas con elementos de tamaño pequeño y bajo contraste se ven muy afectadas.

El problema que poseían las técnicas de diagnóstico previas a la tomografía computarizada, es que la tomografía convencional, ocurría de tal manera que se hacía mover un tubo y el sistema de la imagen en direcciones opuestas en torno a un punto de referencia llamado fulcrum, mediante sistemas digamos complejos, desde el punto de vista mecánico.

Con un plano de la región tridimensional que sea de interés radiográfico, queda bastante enfocada durante todo el proyecto, y debido al movimiento, se producían regiones difuminadas, por tanto, se conseguía que la región de interés estuviese lo suficientemente enfocada para la época.

La solución que propone la tomografía computarizada a este tema, producido por la radiación dispersa, como hemos mencionado previamente, es la utilización de rayos X lo más finos posibles y para evitar la superposición de estructuras, un buen enfoque fue abordar la región tridimensional desde diferentes perspectivas, fue lo que dio lugar a la TC.

En 1963, Cormack, el premio nobel de fisiología y medicina en 1979, propuso la idea de poder reconstruir los coeficientes de atenuación en una región del tejido, y con la ayuda de Doyle formularon el problema en términos de integrales de línea de los coeficientes de atenuación (10).

$$I = I_0 \exp \left[- \int_L f(s) ds \right] \quad (10)$$

Donde:

- I : Es la intensidad resultante después de atravesar el medio.
- I_0 : Es la intensidad inicial antes de atravesar el medio.
- $f(s)$: Es una función que representa el coeficiente de atenuación del medio en función de la posición s .
- \int_L : Es una integral a lo largo del camino L que recorre la luz o radiación en el medio.

Estas ecuaciones bastante complejas para la época se resolvieron mediante transformadas de Fourier, limitando el número de términos de expansión, y también para mayor facilidad se utilizó una escala de grises.

Por otra parte Godfrey Hounsfield, ingeniero quien compartió el nobel con Cormack fue quien diseñó con éxito el primer equipo de tomografía computarizada, basándose en la idea de que todas las medidas de transmisión de rayos X que salen en todas las direcciones posibles del cuerpo humano, contienen toda la información necesaria para realizar una reconstrucción de la imagen.

Para analizar el paso del rayo X desde su salida hasta su detección se analizó dividiendo la trayectoria del rayo en secciones tridimensionales con forma de cubo, las cuales se denominan voxels. Sin embargo, el problema que surge a la hora de la reconstrucción de la imagen, es el número elevado de ecuaciones lineales debido a la cantidad de coeficientes de atenuación que existen, por tanto se empleó el uso de un programa para la resolución de estas ecuaciones.

3.1.1 Reconstrucción de la imagen

Normalmente el fundamento físico de la reconstrucción de la imagen, suele competir al fabricante del equipo debido al software que tiene asociado, sin embargo el fundamento de la reconstrucción se basa en reconstruir una función a partir de sus proyecciones, trabajo realizado por Johann Radon en 1917 (11).

$$p_L = \ln \left(\frac{I_0}{I} \right) = \int_L \mu(x, y) ds \quad (11)$$

Donde:

- I_0 : Es la intensidad inicial antes de atravesar la línea L .
- I : Es la intensidad medida después de atravesar la línea L .
- $\mu(x, y)$: Es la distribución espacial de la propiedad de atenuación lineal en la superficie.
- $\int_L \mu(x, y) ds$: Representa la integral (proyección) de $\mu(x, y)$ a lo largo de la línea L .
- p_L : Es el resultado de dicha proyección a lo largo de L .

Radon consiguió demostrar que $\mu(x, y)$ podía ser calculado a partir del conjunto de las proyecciones p_L , expresandolas como $p(r, \theta)$, donde r y θ son las coordenadas polares de cada línea L mediante (12).

$$\mu(x, y) = -\frac{1}{2\pi^2} \int d\theta \int dr' \left[\frac{1}{(r' - r)} \right] \frac{\partial p(r', \theta)}{\partial r'} \quad (12)$$

Donde:

- $\mu(x, y)$: Representa la distribución de la propiedad.
- r y r' : Son variables radiales.

- θ : Es el ángulo utilizado en la integración.
- $p(r', \theta)$: Es una función dependiente de las variables r' y θ .

Esto es una solución matemática para el problema de la reconstrucción de imágenes CT, sin embargo Cormack no tenía constancia de esta solución y en sus primeros pasos, tanto Cormack como Hounsfield, emplearon diferentes aproximaciones para la obtención de imagen. Aunque a lo largo del tiempo se han propuesto multitud de técnicas, variando sus resultados en ruido, tiempo de cálculo, complejidad computacional, viabilidad de los protocolos etc.

Siendo realistas llegaron a la conclusión de que estas soluciones no se ajustaban a la realidad desde un punto de vista matemático, los problemas que surgieron fueron los siguientes:

Como hemos visto previamente la integral de la línea medible poseía forma de logaritmo natural, y la idea de los detectores se basa en que el coeficiente de atenuación lineal es un valor fijo el cual va asociado a cada voxel, pero este coeficiente, como hemos visto es una función energética de la radiación, la cual atraviesa el cuerpo, y este rayo no es monoenergético, es decir, cuando tenemos un haz de rayos X funcionando a 120kV, emite fotones en todo el intervalo de energía desde 10 KeV hasta 120 keV, por tanto para la mayoría de materiales que atraviesa el haz se trata de tejidos musculares, órganos huesos, etc, lo que supone cambios relevantes en el valor del coeficiente de atenuación n .

A medida que el haz atraviesa el cuerpo, su energía promedio aumenta debido a un fenómeno conocido como endurecimiento del haz, este proceso ocurre porque los fotones de baja energía dentro del espectro del haz de rayos X son absorbidos con mayor facilidad al interactuar con la materia, especialmente en zonas con un elevado coeficiente de atenuación, como los huesos o materiales densos, los fotones de baja energía poseen una mayor probabilidad de ser absorbidos a través de interacciones como el efecto fotoeléctrico, mientras que los fotones de alta energía es más probable de que atraviesen el material sin ser absorbidos.

Conforme los fotones de baja energía son eliminados del haz (es decir son absorbidos), el espectro energético del haz se modifica, sin embargo, esto no significa que el haz gane energía en términos absolutos, sino que la energía promedio de los fotones que lo componen aumenta. Este cambio en el espectro es lo que se denomina endurecimiento del haz, ya que los fotones remanentes son más "duros" o penetrantes, es decir, tienen una longitud de onda más corta y una mayor energía.

Esta alteración en el espectro del haz genera efectos importantes en la reconstrucción de imágenes, la presencia de zonas con un elevado coeficiente de atenuación, como el tejido óseo, amplifica los efectos del endurecimiento del haz, ya que estas regiones absorben de forma desproporcionada los fotones de baja energía. Como resultado, el haz que emerge de estas áreas densas es más energético y genera sombras y patrones característicos en la imagen, conocidos como artefactos de endurecimiento del haz.

Los artefactos causados por este fenómeno incluyen:

- **Bandas oscuras o brillantes:** Son bandas que aparecen debido a un coeficiente de atenuación elevado, que como hemos mencionado puede ser el tejido óseo.
- **Oscurecimiento del centro del objeto:** El centro de un artefacto es muy probable que aparezca más oscuro en comparación con sus bordes debido al endurecimiento.
- **Distorsión de la información cuantitativa:** El valor de la imagen en cada pixel está asociado a una densidad, pues debido a esto puede ocasionar errores en la medición de este parámetro.

Por estas razones, se emplean estrategias para mitigar el endurecimiento del haz, como la filtración física del haz antes de que atraviese el cuerpo (mediante filtros de materiales como aluminio o cobre) y algoritmos de corrección computacional que ajustan las mediciones para compensar las distorsiones.

Por otro lado surge otro problema que es causante de la aparición de ruido gaussiano y se trata de la dispersión de la radiación debido a que los fotones son dispersados, los detectores captan estos fotones los cuales no corresponden con la imagen real, esto se traduce en un incremento de valores de baja frecuencia como ya hemos visto, así como la producción de sombras y artefactos.

Otro problema, es la existencia de un carácter no lineal en los sistemas de detección, produciendo ruido gaussiano ocasionado por los sesgos dependientes del canal y de la temperatura que varía a medida que el rayo se propaga. Considerando que residen inexactitudes en los objetos radiografiados, debido a movimientos que hace el paciente en el momento de la toma de imagen, es por esto que en la red neuronal se realizará data augmentation utilizando diferentes técnicas, sin embargo, cabe mencionar que existen diferentes metodologías para que el paciente permanezca estático siendo así el tratamiento reproducible, aunque siempre existan factores como el humano que provoque un movimiento que por muy despreciable que pueda llegar a ser, pueda afectar perjudicialmente a los órganos de riesgo.

Existen dos métodos para la reconstrucción de la imagen los cuales se dividen en métodos algebraicos y métodos analíticos, los algebraicos se basan en plantear un sistema de ecuaciones, con un número suficiente de ecuaciones para poder dar una solución única al elevado número de variables existentes, las cuales son independientes, ya que existe al menos una por cada voxel, se resuelven mediante procesos de iteraciones que consiste en repetir una secuencia de pasos o cálculos varias veces, con el objetivo de aproximarse progresivamente a una solución deseada o a un resultado final, en estos métodos desde el primer instante, las medidas que obtienen los detectores, introducen una descripción numérica discreta del caso, es decir, los valores se encuentran separados, como podría ser una secuencia de números que solo son enteros. Debido a la complejidad computacional que tiene esto, no se ha empleado mucho este método.

Los métodos analíticos, son más eficientes ya que realizan las operaciones sobre las funciones

en un espacio continuo y en la etapa final traducen estos valores continuos en valores discretos de las variables, en la mayor parte de estas aplicaciones analíticas se basan en el teorema de Fourier que prueba que a partir de cada proyección del objeto es posible obtener la transformada bidimensional de Fourier, y calculando el número suficiente de proyecciones en el intervalo de 0 a π , dispondremos del espacio de Fourier completo pudiendo así reconstruir el objeto.

Con esta técnica podemos completar el proceso de reconstrucción TC pudiendo reducir a un conjunto de transformadas monodimensionales de Fourier, siguiendo con una transformada bidimensional de Fourier inversa.

Las soluciones analíticas, debido a su menor coste computacional, son las que han sido utilizadas de manera genérica en los equipos comerciales, aunque existen equipos avanzados con procesadores de alta velocidad que facilitan los métodos algebraicos lo cual permite una reconstrucción más exacta con menor ruido, pudiendo así reducir la dosis en el paciente, como vemos el método de reconstrucción utilizado suele estar determinado por el fabricante.

3.1.2 Diferentes tipos avanzados de tomografía computarizada

Comenzaremos explicando la tomografía axial convencional, también conocida como TAC axial o secuencial, funciona obteniendo las imágenes del cuerpo mediante cortes individuales de manera secuencial gracias a un tubo de rayos X emitiendo una radiación la cuál atraviesa el cuerpo del paciente desde diferentes ángulos, posteriormente los detectores captan la información saliente en forma de rayos, seguidamente el tubo de rayos X realiza una rotación completa, es decir, 360 grados alrededor del paciente obteniendo así la imagen de un solo corte, después la mesa en la que se encuentra el paciente se mueve unos milímetros para completar cada rotación permitiendo la adquisición del siguiente corte, teniendo en cuenta que una imagen en solo corte quiere decir que la imagen es tomada en un solo plano, como el nombre nos indica el plano axial, por último

mediante procesamiento de imagen por ordenador se genera una representación del cuerpo.

Como vemos, debido al método secuencial que proporciona este tipo de CT solo se produce una imagen a la vez, también tiene bastantes limitaciones ya que tenemos el movimiento de la mesa y una exploración más lenta, es decir, tarda más tiempo en tomar todas las mediciones para acabar con la imagen fina, el riesgo de artefactos es mayor ya que tarda más tiempo, lo que provoca que el paciente se pueda mover más causando así más dispersión en la imagen, otra desventaja que provoca la cantidad de tiempo es que el paciente debe recibir mayor dosis de radiación ya que por cada corte, el paciente requiere otra dosis de rayos X. Debido a todas estas desventajas, este tipo de CT se ha quedado obsoleto.

Otro tipo de interés es la tomografía axial computarizada helicoidal o comúnmente conocida como TAC helicoidal o espiral, la cual introdujo bastantes mejoras respecto al tipo anterior ya que la mesa en la que se encuentra el paciente tiene un movimiento continuo mientras que el tubo emite rayos X girando y formando una trayectoria en espiral, simultáneamente se mueve la mesa de forma continua y se toman las imágenes produciendo así una trayectoria helicoidal, posteriormente los detectores captan la información generada por los múltiples cortes superpuestos, permitiendo así una mejor reconstrucción de la imagen, por último necesita un software capaz de reconstruir las imágenes tridimensionalmente, permitiendo la visualización de los órganos desde diferentes ángulos.

Esto nos facilita que la exploración sea más rápida gracias al movimiento continuo de la mesa, obtenemos menos ruido y artefactos ya que no existen pausas entre los cortes generando una mayor resolución, nos permite tener las reconstrucciones en 3D respecto a la cantidad de radiación suministrada al paciente, la cual es menor gracias a la reducción de tiempo en la exploración. Esta técnica se emplea para diagnosticar enfermedades pulmonares y abdominales, estudios vasculares así como la detección temprana de tumores.

Por último la tomografía axial computarizada multicorte, conocida como TAC multicorte o multidetector, esta evoluciona en comparación a la modalidad helicoidal ya que incorpora filas de detectores pudiendo así capturar varios cortes simultáneamente por cada rotación que realiza el tubo de rayos X, el cual emite radiación constante mientras gira alrededor del paciente, similarmente se mueve la mesa de forma continua, generando así patrones helicoidales en la adquisición de datos, y gracias a los múltiples detectores introducidos, se capturan simultáneamente varios cortes en vez de uno solo como en el TAC helicoidal, por último necesitaríamos un software que se encargue de reconstruir la imagen en múltiples planos, esto contribuye a una mayor precisión en detalles finos lo que nos facilita el estudio de imágenes cancerosas, también nos permite suministrar menor dosis de radiación así como exploraciones grandes. Las aplicaciones de esta técnica residen en estudios cardíacos y el diagnóstico de enfermedades oncológicas con buena precisión.

3.2. Fundamentos de la generación del ruido gaussiano

En esta sección se explicarán los fenómenos físicos que contribuyen a la formación de ruido en la toma de una imagen médica, especialmente para la tomografía computarizada.

3.2.1 Conversión parcial de energía mediante la interacción de los fotones con la materia

Si consideramos un fotón, este carece de masa en reposo así como de carga, y consideramos la interacción de los fotones no polarizados con energía E , y átomos cuyo número atómico es Z , el intervalo energético que nos concierne es aproximadamente 0,1 keV hasta 1 GeV, donde los procesos de interés son el efecto fotoeléctrico, la dispersión Rayleigh, el efecto Compton y la creación de pares electrón-positrón, es importante considerar que existen otros modos de interacción que son las reacciones fotonucleares, sin embargo la probabilidad de esta interacción es despreciable por tanto no se tendrán en cuenta, finalmente la energía en reposo del electrón será de 511 keV.

Respecto al efecto fotoeléctrico, un fotón es absorbido por un átomo blanco y un electrón de la capa i -ésima, este queda absorbido hacia un ángulo sólido, que queda caracterizado por el ángulo polar. Este proceso solo es posible si $E > U_i$, por tanto cada vez que ocurre esto, se abre un nuevo canal de absorción. La energía que describe esta interacción queda descrita por (13).

$$\varepsilon_e = E - U_i \quad (13)$$

donde:

- ε_e es la energía cinética del electrón emitido.
- E es la energía del fotón incidente.
- U_i es la función trabajo o energía de ionización del material.

Considerando la aproximación de Born, la cual es no relativista, y simplificando, podemos reemplazar la función de onda de un fotoelectrón por una onda plana. Sin embargo esta aproximación solo dará resultados acercados a la realidad si la energía cinética del electrón emitido es mucho mayor a la función trabajo o energía de ionización del materia, en este caso tejidos del cuerpo. La expresión que se obtiene de la sección eficaz en esta aproximación es (14).

$$\frac{d\sigma}{d\Omega_e} = 2^{3/2} \alpha^8 a_0^2 Z^5 \kappa^{-7/2} \frac{\sin^2 \theta_e}{[1 - (v_e/c) \cos \theta_e]^4} \approx 2^{3/2} \alpha^8 a_0^2 Z^5 \kappa^{-7/2} \sin^2 \theta_e, \quad (14)$$

donde:

- $\frac{d\sigma}{d\Omega_e}$ es la sección eficaz diferencial.
- σ es la sección eficaz.
- Ω_e es el ángulo sólido asociado al electrón.
- α es la constante de estructura fina.
- a_0 es el radio de Bohr.

- Z es el número atómico del material.
- κ es un parámetro adimensional.
- θ_e es el ángulo de dispersión del electrón.
- v_e es la velocidad del electrón.
- c es la velocidad de la luz.

Como vemos debido a la constante de seno cuadrado que aparece en (14), nos indica que el fotoelectrón tiende a irse perpendicularmente a la dirección con la que ha incidido el fotón, posteriormente integrado llegamos a (15).

$$\sigma(E) = \int_{-1}^1 \frac{d\sigma}{d\Omega_e} 2\pi d(\cos \theta_e) = \frac{2^{9/2}}{3} \alpha^8 a_0^2 Z^5 \kappa^{-7/2}. \quad (15)$$

La sección eficaz es proporcional a Z y E , crece rápidamente al aumentar el número atómico y disminuyendo la energía.

Respecto a la aproximación dipolar eléctrica no relativista, el Hamiltoniano de interacción entre la radiación electromagnética y el átomo es (16).

$$\mathcal{H}'_{\text{ph}} = \sum_{j=1}^N \frac{e}{m_e c} \mathbf{A}(\mathbf{r}_j) \cdot \mathbf{p}_j + \sum_{j=1}^N \frac{e^2}{2m_e c^2} A^2(\mathbf{r}_j) \equiv \mathcal{H}'_{\text{ph},1} + \mathcal{H}'_{\text{ph},2}. \quad (16)$$

donde:

- \mathcal{H}'_{ph} es el Hamiltoniano de interacción del sistema.
- e es la carga del electrón.
- m_e es la masa del electrón.
- c es la velocidad de la luz en el vacío.

- $\mathbf{A}(\mathbf{r}_j)$ es el potencial vectorial en la posición \mathbf{r}_j .
- \mathbf{p}_j es el operador de momento del electrón j .
- $\mathcal{H}'_{\text{ph},1}$ es el término de interacción lineal con el campo electromagnético.
- $\mathcal{H}'_{\text{ph},2}$ es el término cuadrático con el campo electromagnético.

Ahora empleando teoría de perturbaciones a primer orden obtenemos la sección eficaz del efecto fotoeléctrico como (17).

$$\sigma(E) = 4\pi^2\alpha E \left| \left\langle \Psi_f \left| \sum_{j=1}^Z \mathbf{r}_j \right| \Psi_i \right\rangle \right|^2, \quad (17)$$

donde:

- E es la energía incidente del sistema.
- Ψ_i y Ψ_f son las funciones de onda inicial y final, respectivamente.
- \mathbf{r}_j es la posición del electrón j .

Las funciones de onda inicial y final son aproximadas mediante determinantes Slater y el operador es a un cuerpo, por tanto solo están permitidas las interacciones que involucren a un solo electrón, el sumatorio de la matriz no es necesario ponerlo en este caso, las funciones de onda iniciales y finales corresponden a los orbitales de espín monoparticulares inicial y final respectivamente de electrón que efectúa la transición, las funciones radiales reducidas inicial, $P_{n\ell}(r)$, y final, $P_{\varepsilon,\ell}(r)$, del electrón activo se obtienen resolviendo numéricamente la ecuación de Schrödinger radial, y llegamos a la ecuación (18).

$$\sigma(E) = \frac{4\pi^2\alpha}{3} N_{n\ell}(\varepsilon_e - \varepsilon_{n\ell}) \left[\frac{\ell}{2\ell+1} \mathcal{R}_{\ell-1}^2 + \frac{\ell+1}{2\ell+1} \mathcal{R}_{\ell+1}^2 \right], \quad (18)$$

donde:

- $N_{n\ell}$ es un coeficiente de normalización.

- ε_e y ε_{nl} son las energías del electrón incidente y del estado inicial, respectivamente.
- ℓ es el número cuántico orbital.
- $\mathcal{R}_{\ell-1}$ y $\mathcal{R}_{\ell+1}$ son funciones radiales reducidas asociadas a los estados de momento angular $\ell - 1$ y $\ell + 1$.

Recordando que $E = \varepsilon_e - \varepsilon_{nl}$, con las integrales radiales dadas por (19).

$$\mathcal{R}_{\ell\pm 1}(\varepsilon_e) \equiv \int_0^\infty P_{\varepsilon_e, \ell\pm 1}(r) r P_{nl}(r) dr. \quad (19)$$

Las funciones de onda radiales reducidas ligadas y libres de los átomos de un electrón admiten expresiones analíticas. Cuando el estado inicial es el $1s$, la integral radial es (20).

$$\left(\int_0^\infty P_{\varepsilon_e, 1}(r) r P_{10}(r) dr \right)^2 = 2^8 Z^{-4} \frac{a_0^3}{e^2} f(Ze^2/\hbar v_e). \quad (20)$$

con

$$f(x) = \left(\frac{x^2}{1+x^2} \right)^5 \frac{\exp[-4x \arctan(1/x)]}{1 - \exp(-2\pi x)}. \quad (21)$$

Entonces debido a las interacciones que tiene el fotoelectrón con los demás materiales, se disipa energía en forma de calor así como este también puede caer en un hueco y originar un fotón de rayos X que no corresponde de la imagen generando así un ruido gaussiano.

Respecto a la dispersión Rayleigh, este proceso involucra a un fotón el cual es dispersado elásticamente por los electrones atómicos ligados, sin que el átomo se encuentre excitado, por tanto la onda inicial es igual a la final, sus energías son iguales. Desarrollando las ecuaciones a partir de el Hamiltoniano de interacción tenemos que (22).

$$\frac{d\sigma}{d\Omega} = \frac{d\sigma_T}{d\Omega} [F(q, Z)]^2, \quad (22)$$

donde

$$\frac{d\sigma_T}{d\Omega} = r_e^2 \frac{1 + \cos^2 \theta}{2}. \quad (23)$$

Donde (23) es la sección eficaz diferencial de Thompson para la dispersión por un electrón libre y en reposo.

$$F(q, Z) = \left\langle \Psi_0 \left| \sum_{j=1}^Z \exp(i\mathbf{q} \cdot \mathbf{r}_j / \hbar) \right| \Psi_0 \right\rangle. \quad (24)$$

donde:

- Ψ_0 es la función de onda del estado fundamental.
- Z es el número total de partículas cargadas (por ejemplo, electrones en un átomo).
- \mathbf{q} es el vector de transferencia de momento.
- \mathbf{r}_j es la posición del j -ésimo electrón.
- \hbar es la constante de Planck reducida.
- $\exp(i\mathbf{q} \cdot \mathbf{r}_j / \hbar)$ representa la fase de la función de onda del electrón debido a la dispersión.

En el factor de forma atómico, Ψ_0 denota la función de onda atómica del estado fundamental y la suma recorre los Z electrones atómicos, donde \mathbf{q} es el vector de transferencia de momento, cuyo módulo es (25).

$$q = 2 \left(\frac{E}{c} \right) \sin(\theta/2). \quad (25)$$

El factor de forma atómico puede ser expresado como la transformada de Fourier de la densidad electrónica $\rho(\mathbf{r})$ del átomo, la cual, teniendo en cuenta la simetría esférica, se simplifica a (26).

$$F(q, Z) = \int_0^\infty \rho(r) \frac{\sin(qr/\hbar)}{qr/\hbar} 4\pi r^2 dr. \quad (26)$$

Considerando estas ecuaciones vemos que la dispersión Rayleigh, favorece los ángulos pequeños, y en el caso del átomo de hidrógeno no relativista, podemos evaluar analíticamente el factor de forma atómico (27).

$$F_H(q, Z) = \left[1 + \left(\frac{qa_0/\hbar}{2Z} \right)^2 \right]^{-2}. \quad (27)$$

donde:

- $F_H(q, Z)$ es el factor de forma atómico del hidrógeno.
- a_0 es el radio de Bohr.

La expresión (27) se conoce como la fórmula de Prienne, los átomos con más de un electrón serán calculados numéricamente y podemos mirar en tablas de literatura científica como las de Hubell, estas han sido obtenidas a partir de densidades atómicas.

Debido a que esta interacción es elástica, no existe absorción de energía en el material, por lo que no se convierte en calor, sin embargo, como existe luz dispersado el material no emite radiación nueva pero redistribuye la luz incidente en diferentes direcciones, aunque como hemos visto anteriormente, los ángulos de dispersión son bastante pequeños, y la frecuencia de la luz dispersada es igual que la de la luz incidente por lo que no hay emisión de nuevos rayos X, puede variar su dirección, lo que puede causar anomalías en la toma de imagen.

Respecto a la interacción denominada efecto Compton, donde un fotón con energía $E = \kappa m_e c^2$, interactúa inelásticamente con un electrón atómico que electrón atómico y lo absorbe remitiendo un fotón con energía $E' \equiv \tau E < E$, posterior a la colisión, el electrón retrocede con una energía cinética ε_e en la dirección θ'_e . Teniendo en cuenta la conservación de energía y momento lineal tenemos las relaciones (28) y (29).

$$\cos \theta = \frac{1}{\kappa} \left(\kappa + 1 - \frac{1}{\tau} \right) \quad (28)$$

$$\cos \theta = (\kappa + 1) \left(\frac{1 - \tau}{\kappa [2 + \kappa(1 - \tau)]} \right)^{1/2}. \quad (29)$$

De la ecuación (28) se sigue que (30).

$$\tau = \frac{1}{1 + \kappa(1 - \cos \theta)}, \quad (30)$$

es decir (31),

$$E' = \frac{E}{1 + \frac{E}{m_e c^2} (1 - \cos \theta)} \equiv E_c. \quad (31)$$

Los valores mínimo y máximo de τ son (32).

$$\tau_{\min} = \frac{1}{1 + 2\kappa} \quad \text{y} \quad \tau_{\max} = 1. \quad (32)$$

donde:

- θ es el ángulo de dispersión del fotón después de la colisión.
- κ es un parámetro adimensional relacionado con la energía del fotón incidente.
- τ es la fracción de energía retenida por el fotón dispersado.
- E es la energía inicial del fotón incidente.
- E' es la energía del fotón después de la dispersión.
- m_e es la masa del electrón en reposo.
- c es la velocidad de la luz en el vacío.

El electrón a medida que gana energía cinética, se mueve a través del material y debido a que tenemos colisiones inelásticas, el electrón es probable que choque con otros electrones en los

tejidos, transfiriendo parte de la energía en calor, si el electrón causa excitación, los átomos del material podrían liberar energía en forma calórica o fluorescente.

Esta interacción genera radiación secundaria ya que si el electrón deja un hueco en la capa atómica, el electrón que se encuentre en una capa superior, puede caer para llenar ese hueco, esto involucra una transición en la que se emite un fotón de rayos X característico para cada transición, además debido a la radiación de frenado conocida como radiación Bremsstrahlung, si el electrón es liberado con una energía elevada, es probable que dentro del material se frene, emitiendo este tipo de radiación en forma de rayos X continuos.

Respecto a la creación de pares, el fotón queda consumido y su energía es usada para crear un electrón y su antipartícula que es el positrón, las energías cinéticas de estas partículas, las cuales se encuentran cargadas, con E_{\pm} , la conservación de dicha energía queda descrita por (33).

$$E = E_- + E_+ + 2m_e c^2 \quad (33)$$

Como se puede ver, la creación de esta interacción sólo es posible si $E > 2m_e c^2 \approx 1.022 \text{ MeV}$, aunque la conservación del momento hace que el fenómeno tenga lugar en una región próxima al material que pueda absorber el exceso de momento, para que una partícula masiva absorba una pequeña cantidad de movimiento y una pequeña cantidad de energía la cual se suele considerar como despreciable, esta partícula es considerada como un núcleo.

Para el cálculo de las secciones eficaces, en este proceso, debemos recurrir a la electrodinámica cuántica, y utilizando este formalismo, suponemos que las partículas cargadas son lo suficientemente rápidas para que puedan ser descritas como ondas planas véase la aproximación de Born (14), por otro lado Bethe y Heitler obtuvieron la expresión para la sección eficaz diferencial en la energía del electrón creado (34).

$$\frac{d\sigma_{\text{BH}}}{dE_-} = Z(Z + 1) f(Z, E; E_-). \quad (34)$$

Como vemos $f(Z, E; E_-)$ depende del factor de forma atómico y como sabíamos aparecen dos contribuciones la primera siendo proporcional a Z^2 , que está asociada a la creación de pares en el campo del núcleo y la segunda que es proporcional a Z se debe a la contribución de los Z electrones atómicos lo cual se basa en la creación de tripletes.

El ángulo polar promedio de emisión del electrón y del positrón está dado por (35).

$$\langle \theta_{\pm} \rangle \approx \left(1 + \frac{E}{m_e c^2} \right)^{-1}. \quad (35)$$

La sección eficaz total de creación de pares se obtiene al integrar $d\sigma_{\text{BH}}/dE_-$ respecto de E_- .

Por tanto, debido a la absorción de energía en el material, así como cuando el electrón y el positrón son creados tienen mucha energía cinética significativa, a medida en la que estas partículas se van desplazando ocasionan colisiones con átomos y electrones del material, y parte de su energía se transforma en calor, también produciendo excitación así como ionización en el material pudiendo aumentar su temperatura. Se origina radiación secundaria debido a la aniquilación del positrón a rayos gamma, ya que después de su formación, el positrón viaja una corta distancia antes de encontrar un electrón del material, y cuando este fenómeno ocurre, el positrón y el electrón se aniquilan generando dos fotones de rayos gamma de 511 keV cada uno (36).



También como las partículas poseen altas energías cinéticas, se genera radiación de frenado, emitiendo rayos X de frenado.

Como ya tenemos bastante conocimiento de lo que ocurre dentro del cuerpo cuando atraviesa

un rayo X, para cada modo de interacción podemos definir el coeficiente de atenuación lineal como (37).

$$\mu_i = \mathcal{N} \sigma_i, \quad (37)$$

Donde \mathcal{N} es el número de átomos o moléculas por unidad de volumen. Como \mathcal{N} es proporcional a la densidad ρ , resulta más conveniente definir los coeficientes de atenuación másicos parciales μ_i/ρ , y el coeficiente de atenuación másico total (38).

$$\frac{\mu}{\rho} = \frac{N_A}{A_w} \sum_i \sigma_i. \quad (38)$$

3.2.2 Eficiencia cuántica del detector

La eficiencia cuántica que posea el detector es otro de los factores fundamentales que ocasiona la generación de ruido en imágenes médicas.

La descripción de la eficiencia cuántica en un detector de imágenes, en este caso de tipo TC, se refiere a la eficiencia con la que el detector captura y utiliza los rayos X que inciden sobre él, se describe como la fracción de los fotones absorbidos por un material que se encuentra en el detector respecto al número de fotones que llegan al detector, esto depende de las propiedades físicas del detector, donde predomina el número atómico efectivo (Z), la densidad, y el grosor de la capa detectora del material, ya que un material de número atómico efectivo elevado y una alta densidad así como de un espesor elevado, tendrá mayor probabilidad de absorber los fotones del rayo debido a interacciones fotoeléctricas o efecto Compton, lo que aumenta la eficiencia cuántica intrínseca del detector, por tanto, cuanto menores sean estos parámetros menor eficiencia tendrá el detector.

Para poder caracterizar de mejor manera este fenómeno debemos distinguir dos conceptos

clave.

El primero se trata de la eficiencia cuántica intrínseca, que se basa en la eficiencia basada en el porcentaje de rayos que inciden e interaccionan con el detector, si consideramos un haz de fotones monoenergéticos, la probabilidad de que un fotón interactúe es de $1 - e^{-\mu x}$, donde μ es el coeficiente de atenuación lineal del material y x , se trata del espesor, por tanto, esta expresión nos describe la eficiencia intrínseca del material y como vemos aumenta con el espesor, la densidad del material y con un coeficiente de atenuación elevado, el cual crece con el número atómico del material, por tanto, con una energía elevada, la dificultad de absorber los fotones aumenta, por lo que la eficiencia cuántica tiende a disminuir, pero se puede compensar si utilizamos un material con un número atómico efectivo elevado, o aumentando el espesor del material.

El segundo concepto relevante es la eficiencia cuántica de detección (DQE), en este parámetro además de tener en cuenta la cantidad de fotones absorbidos, también se tiene en cuenta como de eficiente es ese fotón para calcular la imagen, este tipo de eficiencia se define como la relación entre la relación señal a ruido (SNR) (39) al cuadrado y la imagen de salida y la SNR al cuadrado de la señal de entrada (40).

$$\text{SNR} = \frac{\text{Señal útil}}{\text{Nivel de ruido}} \quad (39)$$

$$(\text{SNR}_{\text{out}})^2 = (\text{SNR}_{\text{in}})^2 T, \quad (40)$$

Como vemos un detector ideal aprovecharía de una manera perfecta cada cuanto de rayos X, por tanto la fracción sería de 1, es interesante considerar que se combinan los aspectos de la eficiencia de la absorción, la detección del fotón y la conversión en señales su posterior veracidad que posee esa señal con la imagen real, incluyendo la posibilidad de que se introduzca ruido, también tiene en cuenta la resolución con la que el detector transfiere la información espacial, la cual se

encuentra determinada por la función de transferencia modulada.

Respecto al impacto que tiene la eficiencia cuántica en la calidad de la imagen, como hemos visto la influencia de la eficiencia cuántica en los detectores es directa en la calidad de la imagen, si tenemos un detector DQE con un índice del 0,5 se necesitaría el doble de dosis para obtener el mismo SNR que un detector ideal, es decir, aprovecha de manera mediocre la información de cada fotón.

La resolución espacial, es un punto importante a la hora de capturar la mayor cantidad de fotones y mantener la nitidez de la imagen, ya que, por ejemplo, consideramos un cristal centelleador, el cual a mayor grosor, se aumenta la eficiencia cuántica, sin embargo si se aumenta demasiado el grosor, la luz que se emite podría ser dispersada dentro del mismo material o en ángulos oblicuos, produciendo así una dispersión lateral lo que perjudicaría a la resolución espacial en altas frecuencias, como vemos la idea de los detectores es buscar un equilibrio consiguiendo optimizar los dos apartados, ya que si la capacidad de rendir detalles finos con un buen SNR se ve disminuida si el detector no posee suficiente resolución intrínseca, los fabricantes de máquinas CT, buscan optimizar este balance empleando materiales con luminiscencia elevada y un rápido decaimiento, un detector común podría ser los llamados detectores de cerámica.

Otro apartado a tener en cuenta es el contraste y la capacidad de detección de lesiones, ya que a mayor número de fotones detectados, debido a una eficiencia alta, implica que entre los píxeles exista una menor variabilidad estadística, por tanto se aumenta la resolución en el contraste que posee la imagen, esto ayuda a diferenciar tejidos que tienen aproximadamente la misma densidad, como pueden ser lesiones hipodensas, así como la detección de artefactos pequeños de bajo contraste mejora con detectores con alto DQE, debido a que con un nivel de contraste en relación objeto-fondo, se necesita un umbral de SNR para percibirlo, un detector con un coeficiente de eficiencia bajo podría pasar por alto un artefacto pequeño o de bajo contraste, así como un índice

bajo provoca un post-procesamiento de la imagen que puede no corresponderse con la realidad, lo que podría llevar a un diagnóstico erróneo, o un cálculo incorrecto de una dosis de radiación.

Para la relación entre eficiencia cuántica y la dosis de radiación, tenemos que tener en cuenta que una prueba de imagen tipo CT necesita más radiación que por ejemplo una prueba CBCT para tomar la imagen, como hemos visto antes un detector con mayor eficiencia cuántica precisa de menor radiación para tomar la imagen, en un escáner múltiples cortes, antiguamente se utilizaba como material un gas presurizado de xenón y posteriormente se comenzaron a utilizar detectores de cristal de centelleador los cuales son capaces de absorber casi hasta el cien por ciento de los rayos X.

4. Metodología

En esta sección explicaremos la metodología empleada introduciendo la explicación del funcionamiento y fundamentos de las redes neuronales, las redes neuronales de tipo convolucional, la forma en que hemos realizado la preparación de datos y por último la explicación del modelo U-NET.

4.1. Explicación fundamental de una red neuronal

Como se puede consultar en el apartado anexado de la taxonomía de las técnicas de eliminación de ruido (Appendix A), necesitamos un tipo de método que elimine el ruido de la imagen cancerosa y que este método sea capaz de aprender de las imágenes, no necesariamente tenemos que tener definido el tipo de ruido para que el método no elimine detalles finos que como hemos visto hasta ahora son tan importantes.

El método que hemos escogido como más óptimo es la red neuronal tipo U-NET. Explicando el concepto de red neuronal, podemos considerar que es un conjunto de algoritmos de aprendizaje automático que intentan modelar determinados patrones, en esta caso patrones de datos de alto nivel asociados a la intensidad de cada píxel, y mediante arquitecturas computacionales, las cuales admiten transformaciones no lineales e iterativas de nuestros datos preparados de forma matricial o tensorial. Explicando la estructura de una red comenzamos con el concepto de neurona que se refiere a la estructura mínima de la cual esta compuesta la red, en este sistema podemos encontrarlas total, parcialmente conectadas o simplemente que las neuronas no se encuentren conectadas entre sí, cada neurona recibe un conjunto de entradas y en base a estas entradas se calcula el peso que tiene cada una, donde este valor está asociado al grado de importancia que tiene cada neurona, en el caso que nos ocupa una neurona podría ser un solo píxel de la imagen, posteriormente se realiza una suma ponderada ajustando el sesgo de los datos, donde el sesgo es un término constante y se agrega a la suma ponderada de las entradas, esto nos permite tener flexibilidad y desplazamiento

en la función de activación, pudiendo así mediante este hiperparámetro controlar la adaptación del modelo sin necesidad de modificar los datos, además este parámetro nos ayuda a que el modelo sea capaz de aprender patrones más complejos. Por último se aplica una función de activación a la suma para obtener una salida de la neurona, el principal objetivo de la función de activación es introducir no linealidad en los datos, debido a cómo se encuentran construidas las arquitecturas de una red neuronal, si aplicamos una función de activación lineal a las neuronas, la salida de estas, se convertirá en una combinación lineal de las entradas, por tanto, sólo aprendería patrones lineales, y no es el objetivo de estas ya que necesitamos que aprenda patrones a complejos, posteriormente se explicará con más detalle matemático el funcionamiento de estas y porque la más adecuada para la U-NET consideramos que es la función ReLU.

Previo a la explicación del funcionamiento de una red neuronal, las funciones de pérdida consisten en cuantificar la discrepancia entre las predicciones del modelo y los valores reales esperados, por tanto nuestro principal objetivo es minimizar esta función, tomando las predicciones del modelo podemos etiquetarlas como consideremos, y representa el costo o error asociado con las predicciones actuales, proporcionándonos así una medida cuantitativa que nuestra red neuronal utiliza para ajustar los pesos y sesgos, esto nos influye en la convergencia de la red y evita problemas como podrían ser el sobreajuste o el subentrenamiento.

Una vez explicados estos conceptos fundamentales, el funcionamiento de una red neuronal se basa en una estructura desglosada en capas, cada capa es compuesta de una serie de neuronas, también conocidas como nodos, los cuales se encuentran conectados desde la entrada hasta la salida de la red, la estructura común se desglosa en tres capas.

La primera de estas capas denominada como la capa de entrada es el inicio de la red, cada nodo representa una característica de nuestro conjunto de datos inicial, como es en este caso, la intensidad de un píxel, esta capa sirve simplemente para reconocer los datos, no se realiza ningún

tipo de operación.

La siguiente capa, denominada capa oculta, puede estar compuesta de varias capas ya que es donde se realiza la mayor parte del procesamiento, las neuronas de las capas ocultas reciben las salidas de la capa anterior, pudiendo ser de la capa de entrada o de otras capas ocultas, teniendo en cuenta que cada neurona tiene un peso y un sesgo asociado. Respecto a la profundidad de la red, indica el número de capas ocultas y el ancho de la red se refiere al número de neuronas en cada capa.

Por último la capa de salida de la red, su función radica en generar los resultados finales que pueden tratarse de la producción de valores de regresión, problemas de clasificación etc.

El descenso del gradiente es una técnica de optimización de la red que se usa para ajustar los parámetros como los pesos y los sesgos, el objetivo de esta técnica es como hemos mencionado previamente minimizar la función de pérdida, ya que realizamos pequeñas variaciones en los pesos moviéndolos en la dirección opuesta al gradiente de la función de pérdida respecto a cada peso, permitiendo así reducir el error en cada época, donde una época es el cómputo de toda la red en cada muestra del conjunto de datos, donde como veremos se ajustan los pesos y los sesgos.

A la hora de controlar como aprende la red neuronal, tenemos un hiperparámetro llamado tasa de aprendizaje, el cual nos mide la velocidad con la que la red examina los pesos y sesgos en el proceso de aprendizaje, es crucial elegir adecuadamente este parámetro ya que una tasa de aprendizaje muy alta puede llevar a que el modelo no sea capaz de aprender correctamente los patrones complejos, y si esta es muy pequeña, puede llevarnos a que la red quede oscilando en un mínimo local de la función de pérdida.

El método utilizado para que la red tenga una rápida convergencia, se eviten mínimos locales

y que el modelo posea adaptabilidad, es la implementación de optimizadores, los cuales son algoritmos que consiguen minimizar la función de pérdida ajustando los pesos y los sesgos.

Las estrategias de entrenamiento son esenciales para que nuestro modelo logre llegar al mejor resultado posible, en primer lugar es importante tener en cuenta como van a inicializar los pesos, ya que la red en la capa de entrada no conoce el grado de importancia de los nodos, es decir, el valor de los pesos, y es crucial iniciarlos de manera correcta porque si no el modelo puede ocasionar la desaparición o explosión del gradiente, ralentizando el entrenamiento o provocando que cada vez que compilemos el modelo los resultados sean muy dispares, es decir, que los resultados no converjan, por ejemplo si inicializamos los pesos en cero, es decir, le damos el mismo grado de importancia, en este caso nula, a cada neurona, todas estas recibirán el mismo gradiente durante la retropropagación, por tanto, todas las neuronas aprenderán las mismas características, por otro lado si inicializamos los pesos con valores muy pequeños, los valores del gradiente pueden volverse muy pequeños, y si los pesos son demasiado grandes, el valor del gradiente puede ser aumentado exponencialmente, generando una explosión del gradiente. Por estas razones existen diferentes técnicas de iniciación de los pesos, y posteriormente explicaremos la que hemos elegido.

Para mejorar la capacidad de generalización de datos externos a la red y que la red no se acostumbre a trabajar con los datos del entrenamiento existen técnicas de regularización, en primer lugar la técnica denominada dropout consiste en apagar aleatoriamente un porcentaje de neurona durante el entrenamiento en cada época, por otro lado las técnicas de regularización L2 y L1, consisten en añadir un término de penalización a la función de pérdida el cual es proporcional a la magnitud de los pesos.

Por último el early stopping consiste en imponer una condición para cuando el modelo no tenga una bajada lo suficientemente relevante en la función de pérdida, el modelo se deje de entrenar evitando así el sobreajuste en el entrenamiento.

4.2. Redes neuronales convolucionales

Las redes neuronales convolucionales son el tipo de red que hemos empleado para la reducción de ruido en imágenes cancerosas, estas se basan en arquitecturas espaciales aprovechando esta relación espacial de los datos pudiendo así reducir significativamente la cantidad de datos entrenables, se basa fundamentalmente en la operación convolución que consiste en aplicar un filtro denominado kernel a la imagen para extraer sus características relevantes, matemáticamente la operación se define como 41.

$$O(i, j) = \sum_{m=0}^{f-1} \sum_{n=0}^{f-1} K(m, n) \cdot I(i + m, j + n) \quad (41)$$

Donde:

- $O(i, j)$: El elemento de salida en la posición (i, j) del resultado de la convolución.
- $K(m, n)$: El elemento del kernel o matriz de convolución en la posición (m, n) .
- $I(i + m, j + n)$: El elemento de la imagen de entrada que se encuentra desplazado por m filas y n columnas desde la posición (i, j) .
- $\sum_{m=0}^{f-1}$ y $\sum_{n=0}^{f-1}$: Las sumatorias sobre las dimensiones del kernel, que iteran desde 0 hasta $f - 1$, donde f es el tamaño del kernel.

Posteriormente tendremos que definir un parámetro llamado stride el cual define el paso con el que el kernel se mueve por la imagen, por ejemplo si el stride es de 1, el filtro se mueve píxel por píxel por la imagen, si el stride es 2, el filtro se salta un pixel en cada movimiento, véase en la fórmula 42.

$$O_{\text{dim}} = \left\lfloor \frac{n - f}{s} + 1 \right\rfloor \quad (42)$$

Los componentes de la fórmula son:

- O_{dim} : La dimensión de la salida después de aplicar la convolución.
- n : La dimensión de la entrada (por ejemplo, número de filas o columnas).
- f : El tamaño del filtro (kernel) utilizado en la convolución.
- s : El paso (stride) con el que se desplaza el filtro sobre la entrada.
- $\lfloor \cdot \rfloor$: Denota la función piso, que redondea hacia abajo al entero más cercano.

Esto afecta directamente a las dimensiones de la imagen de salida por eso debemos de aplicar una técnica llamada padding, la cual nos permite determinar las dimensiones de la imagen de salida, en nuestro caso, como es evidente elegimos un padding de tal manera que las dimensiones de la imagen de entrada coincidan con las de la imagen de salida, se calcula mediante 43.

$$O_{\text{dim}} = n + 2p - f + 1 \quad (43)$$

Los componentes de la fórmula son:

- n : La dimensión de la entrada (por ejemplo, número de filas o columnas).
- p : El padding, es decir, el número de filas/columnas adicionales agregadas a cada lado de la entrada para mantener o ajustar la dimensión de salida.

Por último la técnica del pooling consigue reducir la dimensional de la imagen cosa necesaria en este tipo de arquitecturas, para que el modelo sea mas robusto frente a las variaciones, funcionando por ejemplo que en una región pongamos 2x2 escoja solo el valor más alto, o el valor promediando entre los 4 valores, convirtiendo una región 2x2 en una 1x1.

4.3. Metodología utilizada

Una vez explicados los conceptos básicos de las redes neuronales, así como de las redes neuronales de tipo convolucional, procedemos a explicar toda la metodología utilizada en el código para el

tratamiento de datos y el uso de una red convolucional tipo U-NET.

4.3.1 Preparación de datos y ruido gaussiano

En primer lugar las librerías utilizadas corresponden al código 1, entre estas se encuentra matplotlib para la correcta visualización de los datos, numpy para tener mayor facilidad con las operaciones, el uso de TensorFlow y Keras que son las librerías principales en lo que aprendizaje automático se refiere, ya que TensorFlow nos ofrece una API de bajo nivel permitiéndonos crear operaciones y capas personalizables lo cual es necesario debido a la compleja arquitectura de la red neuronal tipo U-NET, así como Keras es una API de alto nivel para redes neuronales, y debido a su interfaz simple nos facilita mucho la programación, además nos facilita poner los datos espaciales en forma tensorial, lo cual como veremos, es necesario para la red.

Posteriormente, debido a la limitación de datos que tenemos, he obtenido 5 imágenes cancerosas de pelvis de The Cancer Imaging Archive, 2025, debido a que estas imágenes se encuentran procesadas por métodos que desconozco de eliminación de ruido, las he tomado como mi resultado ideal, posteriormente, hemos añadido a cada imagen un ruido gaussiano para que el resultado obtenido sea extrapolable a la realidad.

El ruido gaussiano sigue una distribución normal presentando una interferencia aleatoria, por tanto este ruido se le agrega de forma aleatoria a cada píxel de la imagen (128x128), con la condición de que la media sea cero, es decir, el ruido no se encuentra orientado en ninguna dirección y el ruido de cada píxel es independiente.

El ruido, matemáticamente se define como una variable aleatoria (44).

$$X \sim N(\mu, \sigma^2) \quad (44)$$

Donde:

- μ es la media (valor esperado) de la distribución. En el contexto de ruido, comúnmente se asume $\mu = 0$.
- σ^2 representa la varianza, que controla la dispersión de los valores de la variable aleatoria.
- σ es la desviación estándar, y es la raíz cuadrada de la varianza σ^2 . Esta mide la magnitud del ruido.

La función de densidad de probabilidad (PDF) de la distribución normal es (45).

$$f(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (45)$$

Donde:

- x representa las variables de la muestra o los valores posibles que pueden ser tomados por la variable aleatoria X .
- μ es la media de la distribución, que en esta fórmula se ubica en el centro del campana de Gauss.
- σ^2 es la varianza que determina la amplitud de la curva. Cuanto mayor es la varianza, más dispersos están los datos alrededor de la media.
- σ es la desviación estándar y aparece en el denominador de la fracción que normaliza la función, asegurando que el área bajo la curva sea 1.
- $\sqrt{2\pi}$ es un factor de normalización para que la integral total de la PDF a lo largo de todos los posibles valores de x sea igual a 1, cumpliendo así con la propiedad de una distribución de probabilidad.
- El término $e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ es la función exponencial que forma la "campana" característica de la distribución normal, centrándose en la media μ y extendiéndose a ambos lados según la desviación estándar σ .

Utilizando funciones de la librería numpy generamos un valor aleatorio con distribución normal y asegurándome que el ruido tenga la misma forma que la imagen, además me aseguro de que los valores después del ruido se encuentren en el rango $(0,1)$, para que se corresponda con la escala de grises que es con la que trabajaremos, también, hemos tomado que la varianza sea del 0.1 para que los valores se puedan corresponder con la realidad.

Tenemos 5 imágenes con ruido y sin ruido, las cuales se encuentran agrupadas por pares, es decir, cada par se encuentra compuesto por una imagen ruidosa, y su imagen sin ruido correspondiente.

Por otro lado, debido a la poca cantidad de datos, utilizo una técnica llamada data augmentation, que consiste en que en base a las imágenes que tenemos, realizar pequeñas variaciones en estas pudiendo así generar más imágenes para que el modelo pueda entrenar con más diversidad de datos y no tienda a sobreajustarse, aprovechando que cada imagen es diferente para cada paciente debido a su peso, estatura, etc, he realizado transformaciones de zoom aleatorio y de brillo aleatorio, respecto al zoom aleatorio, se aplica un zoom en el rango de $(0.8, 1.2)$ a cada imagen y en el brillo, actúa como si existiesen diferentes escalas de grises lo cual puede ser diferente dependiendo del tipo de máquina de toma de la imagen que tengamos, el rango elegido es de $(-0.2, 0.2)$. Gracias a estas transformaciones he aumentado mis imágenes de 5 a 50, siguen siendo escasas como para entrenar una red y que pueda predecir con éxito imágenes que no haya visto, sin embargo, el objetivo de esta investigación debido a nuestra limitación de datos y computacional, es realizar una predicción de los datos ya observados.

En este punto del desarrollo tenemos 50 pares de imágenes, donde en cada par tenemos una imagen ruidosa, y su imagen asociada post-procesada, que sería el resultado ideal del modelo, como se muestra en las figuras 1 y 2.



Figure 1: Imagen ruidosa después de aplicar ruido gaussiano.

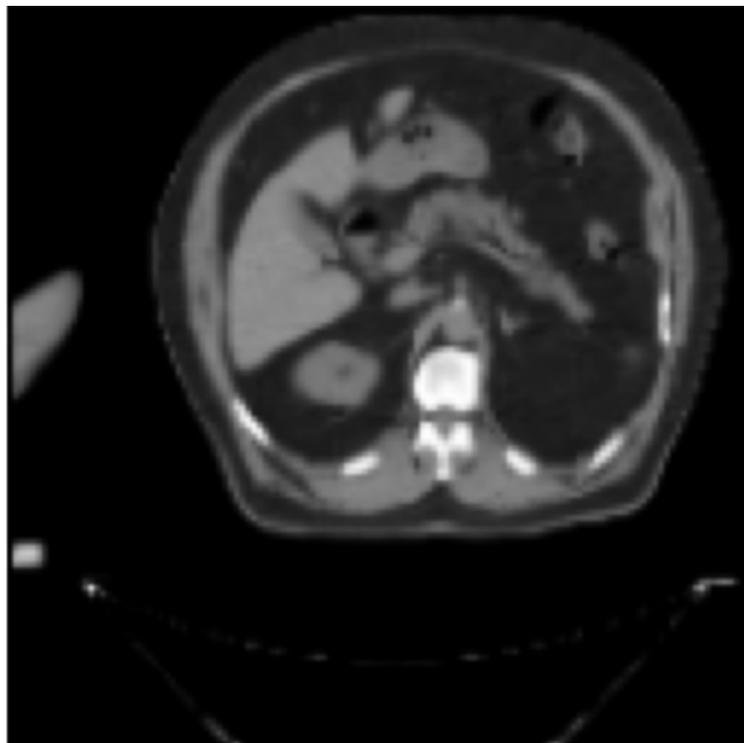


Figure 2: Imagen sin ruido en el caso ideal.

Posteriormente, para tener mayor facilidad a la hora de trabajar con los datos, generamos 2 datasets, uno con las 50 imágenes ruidosas, y otro con las 50 imágenes limpias, para seguidamente generar otro dataset conjunto donde tendremos un batch size de 50, sin embargo hemos elegido este parámetro ya que queremos que los 50 pares de imágenes se incluyan en cada lote de procesado. Una vez tenemos el dataset con cada par de imágenes, realizo una operación para visualizar cuales son las dimensiones de estas imágenes, ya que necesito que cumplan la forma tensorial del tipo (n, x, y, w) , donde "n" es el número de lote que tiene asociado cada par de imágenes, siendo este menor o igual a 50, "x" e "y", son las dimensiones espaciales de entrada que requiere el modelo, donde he redimensionado las imágenes para que tengan dimensiones de (128×128) , para poder tener mayor eficiencia computacional debido a la reducida cantidad de píxeles, y por último, "w" es la escala de colores de la imagen, que como se encuentran en escala de grises, su valor mínimo es 0 y su valor máximo es 1, que corresponde a escala de grises.

4.3.2 Modelo de red neuronal U-NET

El uso de las redes neuronales convolucionales era bastante limitado anteriormente, pero debido a los avances de Krizhevsky, quien entreno una red con 8 capas y un conjunto de un millón de imágenes, gracias al notable éxito en su trabajo, fue que surgió el interés por redes aun más profundas, sin embargo, el uso habitual de este tipo de redes se basaba en tareas de clasificación.

Posteriormente en el ámbito de la física médica se comenzó a estudiar el procesamiento de imágenes médicas, concretamente tareas visuales, donde la salida que se estudiaba debía incluir una localización, en otras palabras se le asignaba una etiqueta de clase a cada píxel. Debido a la limitación de archivos que tenían en ese momento se decidió entrenar una red de configuración de ventana deslizante para predecir la etiqueta de clase de cada píxel proporcionando una región local que se trataba de un parche alrededor de ese píxel, donde los parches eran mucho más grandes que el número de imágenes de entrenamiento, esta red resultó ganadora en varios premios en 2012, sin embargo, esta estrategia presentaba bastantes inconvenientes, el primero, es la redundancia de los

parques que se encontraban superpuestos, en segundo lugar, existía una compensación entre la precisión de la localización y el uso del contexto ya que los parches más grandes requerían más capas de agrupación máxima y se reducía la precisión de la localización, y los parches más pequeños provocaban que la red observase poco contexto.

Debido a estos inconvenientes, surgió la idea de una red totalmente convolucional, que se basa en una arquitectura mucho más elegante, la cual tenga éxito con pocas imágenes de entrenamiento, como es nuestro caso, y se produzcan segmentaciones precisas, lo que da lugar a la arquitectura de la red neuronal tipo U-NET como se puede ver en la figura 3.

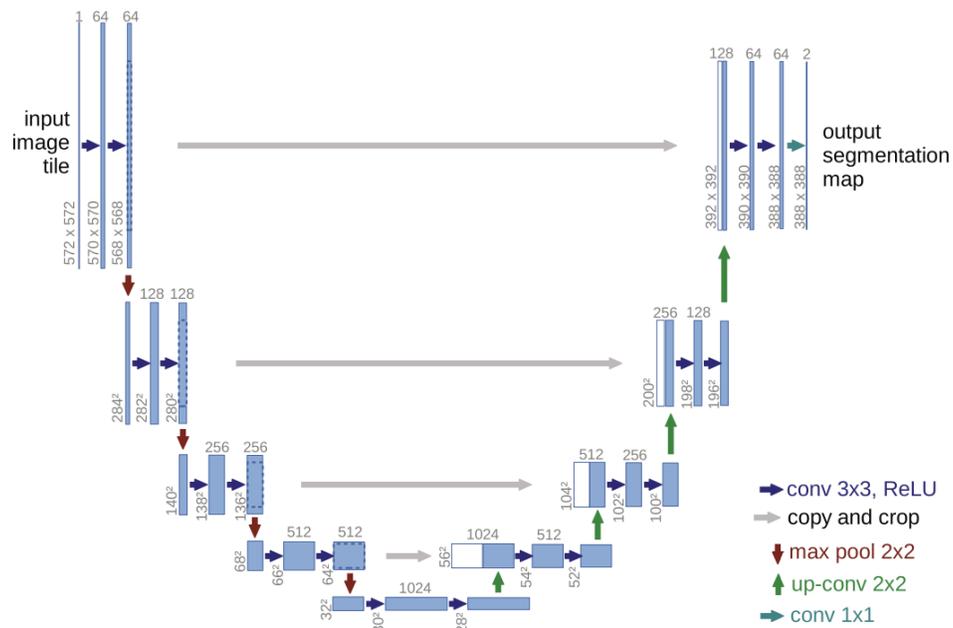


Figure 3: Arquitectura red neuronal convolucional tipo U-NET

La idea se basó en complementar la red de contracción de dimensiones, técnica común cuando tu arquitectura se fundamenta en capas sucesivas, los operadores de agrupación fueron reemplazados por operadores de muestreo ascendente, aumentando así la resolución de la salida. Como vemos se demostró que una capa covolucional consecutiva puede originar una salida más precisa.

La modificación más relevante fue respecto al apartado del muestreo, ya que como tenemos

un gran número de canales de características, esto permite propagar la información de contexto a través de capas con mayor resolución. Esto provocó que la ruta fuese bastante simétrica respecto a la arquitectura de contracción, generando una forma de "U", este tipo de red no posee capas completamente conectadas y solo se utiliza la salida de la convolución, es decir, el mapa de segmentación solo contiene los píxeles, para los cuales el contexto se encuentra completo en la imagen de entrada, esto nos permite la segmentación de las imágenes como lo podría hacer hoy en día la técnica de segmentación de tipo Canny.

Si incidimos más en la figura 3, la arquitectura consta de un camino que fluye desde el lado izquierdo hasta el derecho, donde el lado izquierdo se considera como el camino de la contracción y es el primero de la red y por otro lado, el derecho se considera como el camino expansivo. Sigue la estructura de una red neuronal convolucional convencional, y consiste en una aplicación repetida de convoluciones con kernel 3x3, cada una seguida de la función de activación ReLU, con una agrupación máxima de 2x2, en cada paso del muestreo descendente, se duplican el número de canales, en cambio en la ruta expansiva tratamos un muestreo ascendente del mapa de características seguido de una convolución 2x2, la cual reduce a la mitad el número de características, una concatenación con el mapa de características recortado correspondientemente de la ruta de contracción y dos convoluciones de 3x3, cada una seguida por una ReLU. El recorte es imprescindible debido a que poseemos una pérdida de píxeles de borde en cada convolución. En la capa final de características se usa una convolución tipo 1x1, para poder asignar cada vector de características el número deseado de clases, esta estructura posee 23 capas convolucionales.

5. Resultados y discusión

En esta sección explicaremos la arquitectura concreta utilizada para la reducción de ruido mediante una explicación detallada del código así como los resultados que proporciona el modelo.

5.1. Tipo de arquitectura utilizada

Para explicar el tipo de arquitectura utilizada tenemos en cuenta en el código 2, el cual se encuentra dividido en cinco partes bien definidas, en primer lugar se encuentra la capa de entrada, en segundo lugar tenemos el downsampling, seguido del bottleneck, posteriormente el upsampling, y por último la capa de salida.

La función de activación utilizada es la ReLU (Rectified Linear Unit, o Unidad Lineal Rectificada), esta función matemáticamente se define como 46, lo que significa que teniendo un valor de entrada mayor que 0, la salida será x , es decir, no sufre ningún cambio, sin embargo si el valor de entrada x es menor o igual a 0, la salida será 0, esto ocasiona una función lineal para valores positivos y cero para los valores negativos, parece que se trate de una función lineal que ocasionaría una salida de combinaciones lineales de los datos de entrada, sin embargo, introduce no linealidad debido a que esta recorta los valores negativos, por tanto, como nuestras imágenes poseen pequeñas variaciones que pueden ser partes del artefacto que puede ser identificado como tumor, esto nos ayuda a aprender patrones complejos sin perder información, y nos permite eliminar el ruido no deseado. Computacionalmente hablando esta función es bastante eficiente debido a que no utiliza operaciones complejas, como sí las utilizan las funciones tangente hiperbólica o la sigmoideal. Respecto al desvanecimiento del gradiente, esta función es especialmente útil para evitarlo porque como su derivada es constante, no se acerca a 0 para valores positivos facilitando así el flujo del gradiente durante la retropropagación.

$$\text{ReLU}(x) = \max(0, x) \quad (46)$$

En un contexto de imágenes médicas, ReLU es especialmente útil ya que, nos ayuda a eliminar el ruido de manera eficaz, debido a que el ruido suele manifestarse como valores de píxeles bajos, o incluso negativos, como ocurre en algunos casos, donde los valores negativos pueden ser artefactos de las imágenes o ruido aleatorio, aplicando ReLU, permitimos que la red se centre en las características más relevantes como pueden ser las áreas de alta intensidad como pueden ser artefactos identificables como tumores y sus bordes, dejando intactas estas características y descartando las pequeñas variaciones que no sean relevantes. Como vemos esto nos permite diferenciar entre las características relevantes y el fondo ruidoso, presentando robustez frente a pequeñas variaciones sin que el ruido afecte a la capacidad de aprendizaje del modelo. Como los valores negativos son eliminados, ReLU hace activaciones significativas, mejorando la precisión del modelo en detectar patrones relevantes.

La capa de entrada de la red U-NET se muestra en el código 3, donde definimos el nombre del modelo y posteriormente definimos que el tipo de datos que acepta el modelo sea imágenes 128x128, y que tengan como escala de colores tipo grises.

Por otro lado tenemos el código 4 que pertenece a la parte del downsampling.

La capa de downsampling se basa en una reducción de las dimensiones espaciales durante el entrenamiento de la red, esto se consigue mediante convoluciones y otras operaciones que reducen las dimensiones de la imagen, con la condición que se conserve la información relevante.

Este proceso nos permite que la red aprenda representaciones jerarquizadas, además de patrones a priori complejos que poseen los datos de entrada.

En primer lugar la función Conv2D, es la función que ejecuta la convolución y a la derecha tenemos la cantidad de filtros que se aplica, como es recomendable para este tipo de arquitecturas

escogemos que el número de filtros sea 16 ya que es múltiplo de 128, que son las dimensiones de la imagen original, posteriormente el tamaño del kernel que se usa es de 3x3, seguidamente aplicamos la función de activación ReLU y por último escogemos que el padding sea tipo same para que las dimensiones de las imágenes de salida después de cada capa sean las mismas que las de la capa de entrada o de la capa anterior. Como se ve aplicamos dos capas con 16 filtros cada una donde el stride no se encuentra definido, por tanto es de uno, posteriormente se aplica una capa de MaxPooling2D la que toma el valor máximo de una dimensión específica que como se ve en el código es de 2x2, por tanto de 4 valores tomamos solo uno, el más alto, reduciendo así las dimensiones de la imagen, y por último aplicamos una capa de batchnormalization para ayudar a que el entrenamiento sea más estable y rápido, esto se consigue ajustando la media y la desviación estándar de la función de activación para poder centrarlas en torno a cero con una desviación estándar de 1.

Seguimos aplicando la misma estructura de cuatro capas, bajo la condición de que aumento el número de filtros que pasa de 16 a 32 y posteriormente a 64, por tanto, vamos extrayendo patrones cada vez más complejos a medida que vamos aumentando el número de filtros, y las capas de pooling tienen como función regular y controlar el entrenamiento.

En el bottleneck, como indica su nombre, nos encontramos en el cuello de la botella, donde la idea es reducir el flujo de datos dentro de la red, procesamos la información proveniente de la capa pool3, la cual posee una alta dimensión, así como las imágenes de dimensión reducida las capas de pooling que hemos aplicado, por tanto esto permite a la red extraer características abstractas de manera eficiente, como vemos en el código 5, la arquitectura de esta parte se basa en dos capas convolucionales con una aplicación de 128 filtros con un tamaño de kernel de dimensiones 3x3, con un stride de 1 seguido de una función de activación tipo ReLU con padding same.

El upsampling, se basa en un muestreo superior con la finalidad de aumentar las dimensiones

espaciales que nos permite reconstruir la imagen de salida a partir de las representaciones comprimidas que hemos obtenido en el downsampling, se basa en hacer la inversa del downsampling, como se ven en el código 6, en la arquitectura aplicamos en primer lugar una capa de desconvolución con 64 filtros, un kernel de 2x2, así como un stride de 2x2, posteriormente duplicamos las dimensiones de la imagen mediante la función `layers.concatenate`, y nos permite combinar las características de las capas anteriores definidas como `conv1`, `conv2` y `conv3`, seguidamente aplicamos dos capas convolucionales con 64 filtros, kernel de 3x3, una función de activación ReLU y un padding de tipo `same` para que las imágenes continúen saliendo con las mismas dimensiones, esto nos ayuda a refinar y aprender más detalles de las características combinadas, esta misma estructura se repite dos veces más bajo la condición de que el número de filtros en la capa desconvolucional y las 2 capas convolucionales, se reduzcan a la mitad.

Por último tenemos la capa de salida, como vemos en el código 7, comienza con una capa convolucional con un solo filtro lo que significa que el modelo produce solamente una salida por cada píxel, lo que se traduce en una sola imagen de un solo canal, por otro lado que el tamaño del filtro sea de 1x1 significa que cada píxel de salida es una combinación ponderada de los valores de cada píxel de su entrada correspondiente, la capa actúa como una reducción del canal, tomando las activaciones de las capas anteriores, condensando así un solo valor por píxel, en este caso la función de activación elegida es la sigmoideal debido a que cada pixel se encontrará en el rango [0, 1], esta función nos permite mapear el valor real dentro de dicho rango, la fórmula de esta función se encuentra descrita por 47.

$$\varphi(z) = \frac{1}{1 + e^{-z}} \quad (47)$$

Esta función como vemos se trata de una función suave y diferenciable, lo que nos facilita el cálculo de los gradientes, también debido a que la función no se encuentra centrada en cero, la salida será siempre positiva lo que ralentiza la convergencia del entrenamiento debido a las oscilaciones en las actualizaciones de los pesos.

Como vemos la capa convolucional se aplica a la última salida de la fase del upsampling definida como up1, dando lugar a la predicción final del modelo, en este caso produciendo una máscara de segmentación en un solo canal, donde cada valor en el mapa de salida indica la probabilidad de que el píxel pertenezca a la clase de interés.

Para compilar el modelo, como vemos en el código 8, hemos utilizado el optimizador Adam (Adaptive Moment Estimation), debido a que este se basa en una mejora respecto otros métodos de optimización como podrían ser el descenso del gradiente o por ejemplo el optimizador RMSProp, ya que este combina dos conceptos clave en el ámbito de la optimización, el primero utiliza un promedio de momentos para conseguir adaptarse mejor a la topología del espacio de parámetros y las características de las funciones de pérdida, también realiza una estimación de momentos en primer y segundo orden, empleando el gradiente y la segunda derivada, permitiéndonos así adaptar la tasa de aprendizaje de una manera óptima.

El Adam emplea una actualización de parámetros, donde por ejemplo, tomando los parámetros de los pesos, Adam calcula el gradiente de la función de pérdida y posteriormente calcula la media del gradiente donde el primer momento es una media ponderada de los gradientes pasados como se ve en (48).

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_{\theta} J(\theta) \quad (48)$$

Donde:

- m_t : Es el primer momento (la media) del gradiente en el tiempo t .
- β_1 : Es el factor de decaimiento exponencial para el primer momento (típicamente alrededor de 0.9).
- m_{t-1} : Es el primer momento del paso de tiempo anterior $t - 1$.

- $\nabla_{\theta} J(\theta)$: Es el gradiente de la función objetivo J con respecto a los parámetros θ en el tiempo t .

Como vemos en la arquitectura, no se especifica la manera en la que se inicializan los pesos, por tanto, se utilizan los inicializadores por defecto de la librería keras, que se trata de la iniciación de Xavier, comúnmente conocida como Glorot initialization, está diseñada para lograr mantener la varianza de las activaciones constante a través de las capas, esto se consigue inicializando los pesos de cada neurona con una distribución uniforme o normal mediante la fórmula (49).

$$W \sim U\left(-\frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}, \frac{\sqrt{6}}{\sqrt{n_{\text{in}} + n_{\text{out}}}}\right) \quad (49)$$

Donde:

- W representa los pesos iniciales de la capa.
- $U(a, b)$ denota una distribución uniforme en el intervalo $[a, b]$.
- n_{in} es el número de neuronas de entrada en la capa.
- n_{out} es el número de neuronas de salida en la capa.

El resumen de la arquitectura y los parámetros utilizados se encuentra en la tabla 1.

5.2. Resultados del modelo

Respecto al número escogido de épocas, debido a nuestra limitación de datos, queremos que el modelo sepa predecir los datos con los que entrena, por tanto, el número escogido de épocas es de 500, debido a que la mejora que obtiene el modelo comienza a ser prácticamente nula en la época 500 como se puede observar en la figura 4 debido a que la función de pérdida comienza en aproximadamente 0,16 y desciende hasta 0.00073624, lo que nos indica que el modelo ha conseguido alcanzar un punto de convergencia donde no existirán grandes mejoras, por otro lado la métrica de error utilizada es el error absoluto medio el cual comienza en 0.3554, y alcanza también un

punto de convergencia en la época 500 de un valor de 0.0136 como se puede ver en la gráfica. La evolución de la función de pérdida indica que el modelo aprende de manera efectiva en las primeras épocas y a partir de la época 30 la mejora del modelo no es tan significativa y a medida que aumentan las épocas la mejora va disminuyendo alcanzando el punto de convergencia.

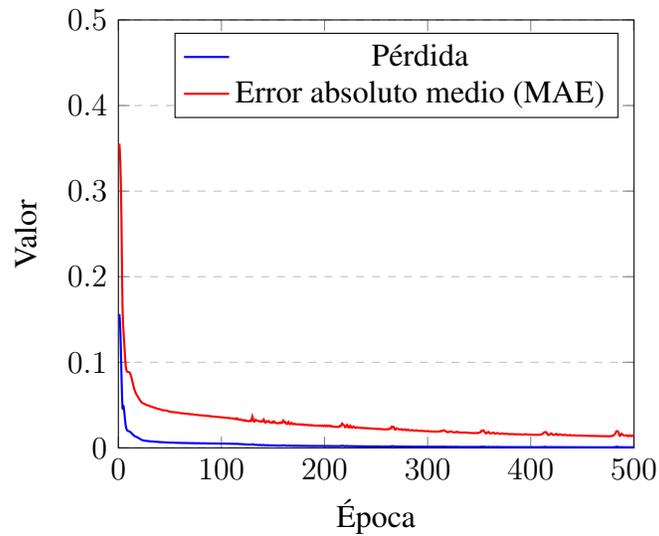


Figure 4: Evolución de la pérdida y el MAE durante el entrenamiento de la red neuronal.

Posteriormente validamos el modelo asegurándonos de que el modelo no solo haya aprendido en base a los datos de entrenamiento, sino que pueda generalizar bien datos no vistos, en este caso datos de prueba, también nos permite evaluar si el modelo se ha sobre ajustado, por tanto no se desempeña bien con datos no vistos. La función de pérdida es muy baja siendo esta de 0.0022, esto indica que el modelo ha conseguido minimizar esta función en base a los datos de prueba, sugiriéndonos que el modelo se ajusta correctamente a los datos, el error medio absoluto es relativamente bajo con un valor de 0.0258, significando que el error promedio entre las predicciones del modelo y los valores teóricamente perfectos es pequeño, por tanto es probable que sea capaz de realizar predicciones con bastante precisión.

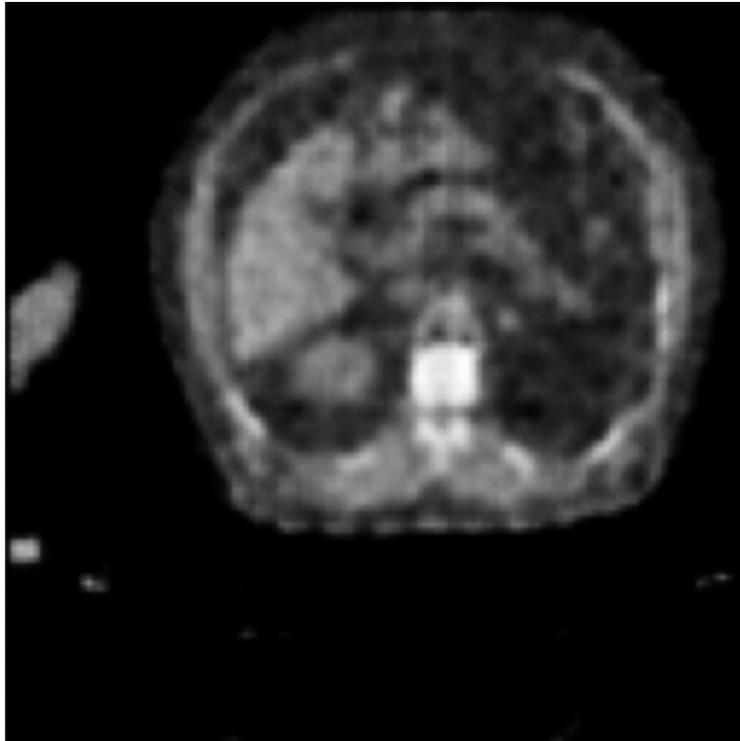


Figure 5: Imagen procesada por el modelo

Como vemos en la figura 5, la imagen procesada por el modelo se corresponde a la predicción de las figuras 2 y 1, evaluando sus métricas de error, tenemos un error medio cuadrado entre la imagen sin ruido y la que contiene ruido de 0.0708, y la diferencia entre la imagen sin ruido, y la imagen procesada es de 0.0524, por tanto, como vemos, la mejora es notable. Respecto al error medio absoluto, la diferencia entre la imagen ruidosa y la imagen sin ruido es de 0.2142, y la diferencia entre la imagen procesada y sin ruido es de 0.1375, sin embargo aunque vemos que la mejora respecto a los dos tipos de errores es notable, evaluaremos el modelo realizando 50 predicciones aleatorias, para posteriormente evaluarlo mediante una gráfica de correlación.

MSE entre imagen con ruido y original vs. imagen procesada por el modelo

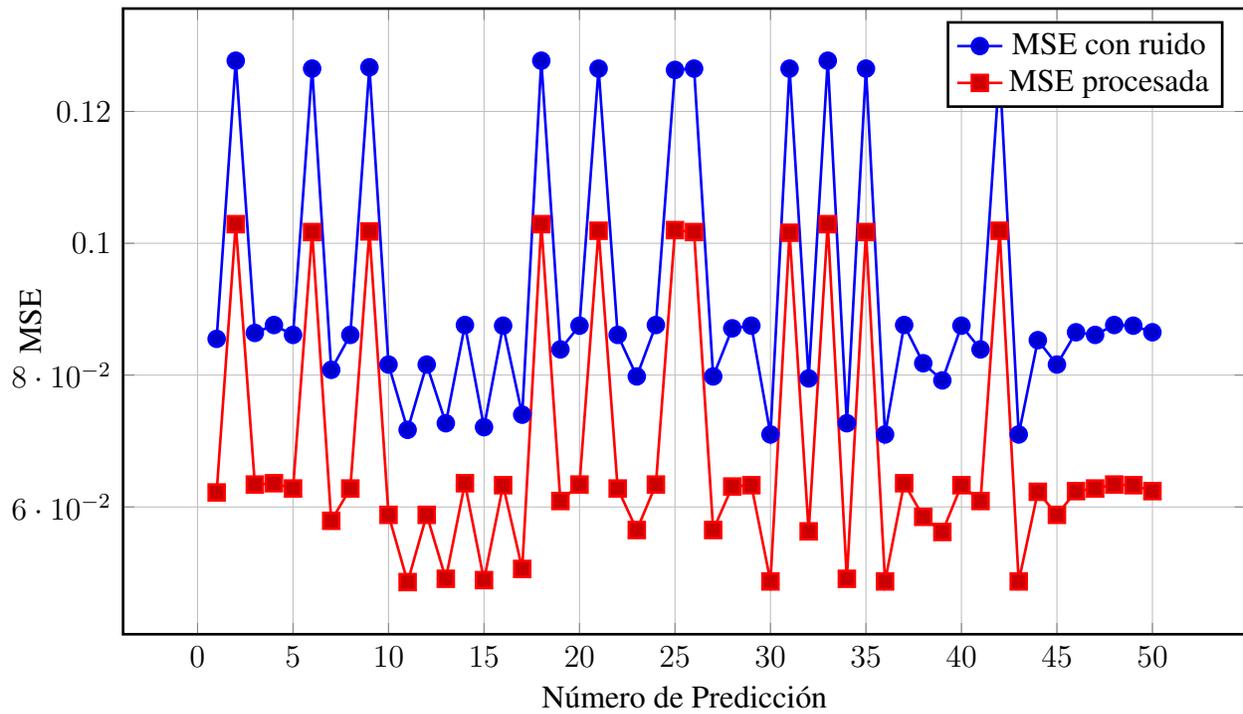


Figure 6: MSE para las imágenes ruidosas y procesadas por el modelo.

MAE entre imagen con ruido y original vs. imagen procesada por el modelo

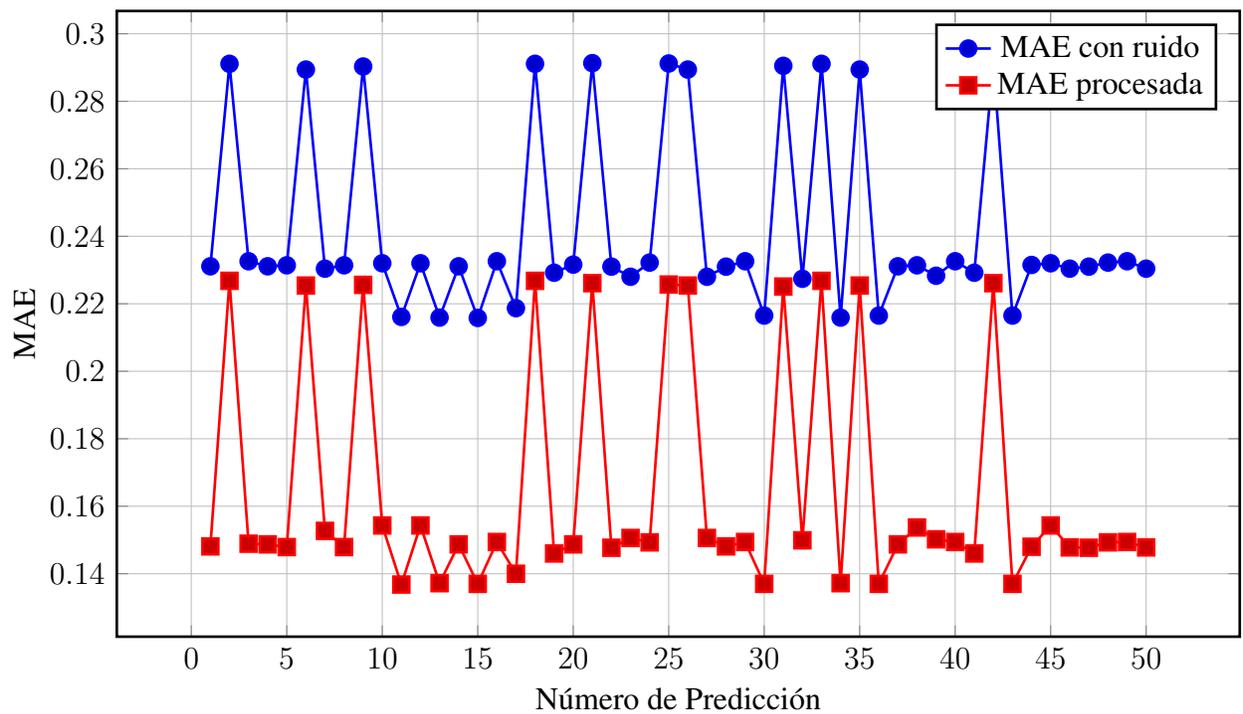


Figure 7: MAE para las imágenes ruidosas y procesadas por el modelo.

Como vemos en las figuras 6 y 7, las diferencias de error cuadrático medio, así como de error absoluto medio en 50 predicciones aleatorias de los datos ya vistos, las diferencias oscilan de manera similar, con una media de diferencias de 0.0237 en el mse y una media 0.0774 en el mae, esto se atribuye a un comportamiento homogéneo de los errores ya que el modelo tiene un error consistente en todas las predicciones, por tanto podemos afirmar que no existen valores atípicos teniendo en cuenta que el mse es un tipo de error, a menudo, sensible a este tipo de errores, siendo el mae más robusto, por otro lado el mse penaliza más los errores con valor más alto por su término cuadrático.

Si tenemos en cuenta que la diferencia de cada tipo de error es bastante similar en todas las 50 predicciones, esto indica que el modelo durante el entrenamiento ha cometido los mismos errores en algún punto del entrenamiento, como era de esperar observando la función de pérdida ya que como hemos comentado llega a un punto de convergencia.

Por otro lado el modelo ha empleado una única técnica de entrenamiento que es batchnormalization, utilizado para normalizar las activaciones durante el entrenamiento, lo que nos permite reducir el desplazamiento covariante interno, por tanto la red es capaz de aprender de manera rápida y estable, también conseguimos así que la red sea menos sensible a la manera en la que se inicializan los pesos, sin embargo, hemos intentado implementar técnicas de regularización de tipo dropout, L1 y L2, pero estas técnicas empeoraban el resultado de la red debido a la poca cantidad de datos, ya que el dropout desactiva aleatoriamente un porcentaje de neuronas en cada capa que se utilice, lo que provoca que el modelo posea dificultades de aprender patrones relevantes en base a las imágenes y se reducía su capacidad de aprendizaje, respecto a las regularizaciones L1 y L2, estas técnicas añaden términos que penalizan a la función de pérdida llevando a que los pesos del modelo se mantengan los más pequeños posibles incluso en 0 en el caso de L1, por tanto si aplicamos una penalización impide al modelo aprender relaciones complejas.

A pesar de que este tipo de resultados son muy prometedores, existen diversos inconvenientes. En primer lugar, se necesita un gran volumen de datos para entrenar de manera eficaz estos modelos, debido a la dificultad existente de encontrar imágenes médicas sin procesar, lo que nos ha llevado a añadir un ruido sintético, por tanto el modelo puede no generalizar bien en situaciones clínicas reales.

El segundo inconveniente que viene derivado por el primer inconveniente es el riesgo de que el modelo se encuentre sobreajustado, como hemos visto existe un elevado número de parámetros, donde la red memoriza las características específicas del conjunto de entrenamiento, perdiendo así su capacidad de generalización, el modelo que ha sido entrenado se encuentra sobreajustado debido a que el objetivo era lograr eliminar el ruido de las imágenes que forman parte del entrenamiento para que el resultado fuese extrapolable a la realidad. Por otro lado el coste computacional que tiene entrenar una red neuronal es bastante elevado, debido a esta limitación de recursos disponibles, hemos escogido pocas imágenes.

Existen diversas técnicas de eliminación de ruido en imágenes médicas, las cuales se encuentran en el apartado anexo de taxonomía de técnicas de eliminación de ruido, donde se explica detalladamente su fundamento matemático teórico y una breve enunciación de sus ventajas y desventajas respecto a la red neuronal convolucional tipo U-Net, así como la explicación de porque consideramos que este tipo de técnica es el más óptimo.

5.3. ODSs empleadas

La motivación de este trabajo se basa en la implementación de ODSs a el ámbito de la física, más concretamente en la física médica, donde hemos empleado el ODS 3 (salud y bienestar) ya que hemos tenido con objetivo principal mejorar la precisión y la rapidez en los diagnósticos y la optimización de la dosis de radiación que recibe el paciente para que no reciba radiación innecesaria perjudicando así más de la cuenta su salud general, esto promueve la meta de garantizar una

vida sana promoviendo el bienestar de la población. El ODS 4 (educación de calidad), también ha sido implementado ya que hemos recibido ayuda por parte del departamento de radiofísica del Consorcio Hospitalario Provincial de Castellón, así como he difundido el desarrollo de este tipo de técnicas en el mismo departamento, el ODS 9 (industria, innovación e infraestructura), nos hemos basado en el uso de tecnologías avanzadas como es en este caso el uso de redes neuronales en el ámbito sanitario, impulsando así el sector tecnológico de la salud, y por último el ODS 17 (alianzas para lograr los objetivos), como he mencionado previamente el desarrollo tecnológico precisa de alianzas entre diferentes instituciones como pueden ser empresas privadas, universidades o como en este caso hospitales, lo que también promueve la transferencia de conocimiento.

6. Conclusiones

El objetivo principal de este trabajo se desglosa en dos apartados, el primero ha sido realizar una descripción bibliográfica de los fenómenos físicos que ocurren en el transcurso de un rayo de radiación con la finalidad de calcular una imagen, así como la explicación de el funcionamiento de la tomografía computarizada y sus diferentes tipos, y una taxonomía de técnicas de eliminación de ruido en imágenes médicas.

El segundo apartado se basa en una investigación con el objetivo de la programación de una red neuronal tipo CNN U-NET, con la finalidad de la reducción de ruido en imágenes médicas de diagnóstico de tipo CT, ya que debido a los fenómenos físicos que ocurren en la toma de la imagen, se genera un ruido habitualmente de tipo gaussiano.

Hemos conseguido mejorar la calidad de la imagen, la cual había sido previamente procesada añadiendo ruido gaussiano, lo que nos indica que este algoritmo contribuye al diagnóstico de tumores, enfermedades etc, así como la precisión en la radioterapia adaptativa, ya que permite a los radiofísicos y a los médicos un mayor control sobre la enfermedad y una mayor optimización del cálculo de radiación al paciente, lo que inhibe al paciente de una absorción de dosis de radiación innecesaria.

Estos algoritmos suelen estar implementados en la propia máquina de tomografía computarizada, sin embargo sigue existiendo ruido en las imágenes calculadas, por tanto un buen uso de este tipo de redes podría mejorar el cálculo de la imagen, y sería relativamente fácil de implementar debido a la gran cantidad de datos que existen, concretamente en el contexto de imágenes médicas cancerosas y la gran capacidad de cómputo que existe actualmente.

Esta herramienta desarrollada es útil en comparación con otros métodos ya que este tipo de

técnicas no trata el ruido como si solo existiese un tipo de ruido en toda la imagen o como si el ruido fuese a seguir un patrón determinado, sino que es capaz de aprender de las propias imágenes, e identificar que valores se corresponden o no a la realidad.

Durante el desarrollo del código me ha resultado especialmente complicado la obtención de imágenes médicas de cualquier tipo que se encuentren en dominio público, lo cual es bastante sorprendente ya que desde mi punto de vista, que este tipo de datos sea de libre acceso es importante para el desarrollo y el progreso tecnológico, siempre que se respete el acuerdo de confidencialidad médico paciente y que los datos se encuentren anonimizados, por otro lado, obtuve un gran inconveniente a la hora del tiempo de entrenamiento del modelo ya que tardaba alrededor de dos horas en finalizarse, lo que me llevo bastante tiempo en optimizar los hiperparámetros utilizados, de haber tenido mayor cantidad de datos el trabajo hubiera sido inviable debido a la capacidad de cómputo que dispongo, en el caso de tener mayor capacidad, podría haber entrenado un dataset mayor y en ese caso la optimización de la red neuronal, habría sido diferente a la actual ya que sería importante añadir técnicas de regularización para evitar el sobre ajuste, y el modelo sería muy robusto ante la predicción de imágenes no vistas por el modelo.

Respecto a las nuevas líneas de investigación que puede sugerir este modelo es la implementación de tiny machine learning en las máquinas CT ya que permitiría reducir el tiempo de cómputo, mediante los microcontroladores, que es donde se introduce el tiny ML, los cuales se encontrarían conectados a los receptores de los rayos de radiación optimizando así el modelo.

Por otro lado esta clase de técnicas se implementarían a la perfección en el flujo de trabajo existente entre los radiofísicos, los médicos y los pacientes ya que sería una implementación automática y realizaría una mejora en todos los ámbitos como el cálculo de la dosis de radiación, reducción del tiempo de diagnóstico etc.

Esta investigación, demuestra que la implementación de las redes neuronales convolucionales con la finalidad de el análisis de imágenes médicas ofrece una herramienta para el diagnóstico asistido por computadora, durante esta investigación, se evidencia que esta técnica es capaz de clasificar patrones complejos superando los métodos convencionales, mejorando así la atención sanitaria.

Bibliografía

- Brosed, A. (Ed.). (2023). *Fundamentos de física médica. volumen 1: Medida de la radiación* [Editor de la colección: Antonio Brosed. Editor del volumen: Antonio Brosed.]. Sociedad Española de Física Médica (SEFM).
- Brosed Serreta, A., & Ruiz Manzano, P. (Eds.). (2023). *Fundamentos de física médica. volumen 2: Radiodiagnóstico: Bases físicas, equipos y control de calidad* [Editor de la colección: Antonio Brosed Serreta. Editor del volumen: Pedro Ruiz Manzano.]. Sociedad Española de Física Médica (SEFM).
- Ronneberger, O., Fischer, P., & Brox, T. (2015, May 18). U-net: Convolutional networks for biomedical image segmentation. <https://doi.org/10.48550/arXiv.1505.04597>
- The Cancer Imaging Archive. (2025). Nbia search - the cancer imaging archive [Accessed: 2025-04-11]. <https://nbia.cancerimagingarchive.net/nbia-search/>

Anexos

A. Taxonomía de técnicas de eliminación de ruido

En el procesamiento de imágenes médicas, la eliminación de ruido es un paso fundamental para garantizar que los datos sean precisos y útiles para el diagnóstico clínico. A lo largo de los años, se han desarrollado numerosas técnicas con el objetivo de mejorar la calidad de las imágenes al reducir el impacto del ruido introducido por dispositivos de captura o por el entorno de adquisición.

Para estructurar este estudio, se presenta una taxonomía de estas técnicas, clasificándolas según su naturaleza, principios matemáticos y aplicabilidad en el ámbito de la física médica. Esta taxonomía permitirá identificar las fortalezas y limitaciones de cada enfoque, proporcionando un marco comparativo que servirá como base para justificar el uso de redes neuronales, en particular el modelo U-NET, como solución para el problema.

La clasificación se organizará de la siguiente manera:

- Técnicas clásicas de eliminación de ruido: Métodos estadísticos y filtros.
- Transformadas matemáticas: Uso de wavelets y otras transformadas.
- Modelos basados en aprendizaje automático y profundo: PCAs y métodos avanzados.

A.1. Técnicas básicas

A.1.1 Filtro promedio

El objetivo de este filtro es realizar un suavizado de la imagen de forma lineal trabajando como un filtro pasa-bajas. Su estructura matemática consiste en dividir la imagen en regiones cuadradas y reemplazar el valor de cada pixel por la media (promedio) de esta región incluyendo el pixel en

cuestión.

Supongamos que tenemos una imagen I de tamaño $m \times n$ y un filtro promedio de tamaño $k \times k$. La matriz del filtro promedio F se define como (50).

$$F = \frac{1}{k^2} \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \quad (50)$$

Este filtro se aplica a la imagen I mediante la operación de convolución, que se puede escribir como (51).

$$I_{\text{filtrada}}(i, j) = \sum_{m=-\frac{k-1}{2}}^{\frac{k-1}{2}} \sum_{n=-\frac{k-1}{2}}^{\frac{k-1}{2}} I(i + m, j + n) \cdot F(m, n) \quad (51)$$

Donde $I_{\text{filtrada}}(i, j)$ es el valor del píxel filtrado en la posición (i, j) , y la matriz F contiene los valores $\frac{1}{k^2}$ en todos sus elementos.

La consecuencia de aplicar este filtro es el de eliminar el valor del píxel el cual no se corresponde con el valor de sus vecinos eliminando así lo que podría considerarse como un valor atípico, es decir ruido, sin embargo como vemos es poco efectivo a cambios bruscos en la imagen como podrían ser bordes muy marcados, los cuales corresponderían a las frecuencias altas, por tanto posee una sensibilidad notable a la hora de generar nuevas intensidades, generalmente grises, las cuales no existían en la imagen original.

El filtro promedio, aunque útil para reducir el ruido en imágenes, no es lo suficientemente preciso para aplicaciones críticas como el cálculo de dosis de radiación en tratamientos oncológicos. Al suavizar los bordes de los tumores, puede perder detalles importantes, lo que afecta la precisión

en la identificación de las estructuras. En la planificación de radioterapia, pequeñas variaciones en los contornos pueden influir en la dosis administrada, lo que pone en riesgo la exactitud del tratamiento, por lo tanto, aunque es simple y eficiente, el filtro de media no es adecuado cuando se requiere alta precisión, como en la planificación de tratamientos de radioterapia.

A.1.2 Filtro mediano

Este filtro funciona de forma similar al filtro promedio, sin embargo, en vez de reemplazarlo por el promedio de sus vecinos, es reemplazado por la mediana la cual es calculada extrayendo los valores de los píxeles dentro de la vecindad, posteriormente son ordenados los valores en orden ascendente y se toma el valor central del conjunto y se reemplaza el calor del píxel central con el de la mediana.

Supongamos que tenemos una imagen I y aplicamos un filtro mediano en una vecindad de tamaño $k \times k$ alrededor de un píxel (i, j) .

La fórmula general del filtro mediano aplicada a un píxel (i, j) es (52).

$$I_{\text{filtrada}}(i, j) = \text{med} \left\{ I(i + m, j + n) : m, n \in \left[-\frac{k-1}{2}, \frac{k-1}{2} \right] \right\} \quad (52)$$

Donde $\{I(i + m, j + n)\}$ son los valores de los píxeles dentro de la vecindad de $k \times k$, ordenados de menor a mayor, y la mediana es el valor central de los valores ordenados.

El filtro mediano es eficaz para reducir el ruido impulsivo en las imágenes, especialmente el ruido de tipo sal y pimienta, que se presenta como píxeles de intensidad muy alta o baja de forma aleatoria. Este tipo de ruido puede ser particularmente problemático en imágenes médicas, ya que puede enmascarar detalles cruciales de la imagen. El filtro mediano tiene la ventaja de eliminar este tipo de ruido de manera eficiente sin difuminar en exceso los bordes y las estructuras importantes de la imagen, lo cual es esencial para preservar la información visual de los tumores y otras características críticas.

Sin embargo, este filtro tiene limitaciones cuando se trata de imágenes más complejas, como las que se utilizan en el diagnóstico y tratamiento del cáncer. Aunque el filtro mediano funciona bien con el ruido sal y pimienta, no es tan efectivo frente a otros tipos de ruido, como el ruido gaussiano, que puede ser más común en imágenes médicas. Además, el filtro mediano no es adaptativo, lo que significa que no tiene en cuenta las variaciones locales en la imagen.

En un contexto médico, donde las características del tejido y los tumores pueden variar en diferentes regiones, puede suponer un problema, ya que el filtro no se ajusta de manera eficiente a las necesidades específicas de cada área de la imagen.

Otro inconveniente importante es que, aunque el filtro mediano preserva los bordes, puede suavizar detalles más finos, en imágenes de cáncer, cada pequeño detalle cuenta para la localización precisa de un tumor y la planificación de tratamientos, como la radioterapia, la pérdida de estos detalles debido al filtrado puede afectar la precisión de las decisiones clínicas.

A.1.3 Filtro bilateral

Este filtro de naturaleza no lineal debido a la forma en la que pondera los valores de los píxeles en función de dos factores, toma la suma ponderada de los píxeles en una vecindad local y los pesos, que son un parámetro que nos determina el grado de importancia de cada pixel, dependen tanto de la distancia de intensidad así como de la distancia espacial, mediante este método conservamos los bordes mientras realizamos un promedio del ruido.

La fórmula general del filtro bilateral aplicada a un píxel (i, j) es (53).

$$I_{\text{filtrada}}(i, j) = \frac{1}{W} \sum_{m,n} I(i+m, j+n) \cdot \exp\left(-\frac{(m^2 + n^2)}{2\sigma_s^2}\right) \cdot \exp\left(-\frac{(I(i+m, j+n) - I(i, j))^2}{2\sigma_r^2}\right) \quad (53)$$

Donde W es un factor de normalización, σ_s es el parámetro de suavizado espacial, σ_r es el parámetro de suavizado de intensidad y $I(i, j)$ es la intensidad del píxel en la posición (i, j) , la cual se encuentra comprendida en el intervalo $(0, 255)$, donde 0 es negro y 255 es blanco.

El parámetro de normalización W en el filtro bilateral se define como la suma de las ponderaciones aplicadas a todos los píxeles en la vecindad, de modo que la fórmula es (54).

$$W = \sum_{m,n} \exp\left(-\frac{(m^2 + n^2)}{2\sigma_s^2}\right) \cdot \exp\left(-\frac{(I(i+m, j+n) - I(i, j))^2}{2\sigma_r^2}\right) \quad (54)$$

Este parámetro asegura que el filtro bilateral se normalice adecuadamente, para que la intensidad filtrada tenga un valor adecuado dentro del rango de intensidades de la imagen.

El filtro bilateral es útil para suavizar imágenes y reducir el ruido, especialmente el ruido de tipo "sal y pimienta", sin difuminar los bordes, lo que como hemos visto es importante en imágenes médicas. Sin embargo, su principal desventaja es su alta complejidad computacional, lo que lo hace lento para procesar grandes volúmenes de imágenes médicas, como tomografías computarizadas o resonancias magnéticas. Además, aunque es efectivo en ciertos tipos de ruido, puede no ser tan útil en imágenes con ruido más complejo. También, al ser un filtro no lineal, puede suavizar detalles finos, lo que es crítico en la planificación de tratamientos médicos como la radioterapia, donde se requiere una alta precisión para calcular dosis de radiación y localizar tumores de manera exacta.

A.1.4 Filtro gaussiano

Respecto a este tipo de filtro podríamos pensar que es el ideal para el problema que nos ocupa, ya que poseemos un ruido con una distribución gaussiana como ya hemos visto previamente.

Este tipo de filtro no lineal nos suaviza la imagen reduciendo la diferencia que existe entre las frecuencias más altas y más bajas, por tanto, limita demasiado la precisión a la hora de recuperar detalles finos cruciales para la detección de bordes del cáncer.

El filtro se aplica convolucionando la imagen con un kernel gaussiano definido por la fórmula (55).

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2 + y^2}{2\sigma^2}\right) \quad (55)$$

Donde:

- $G(x, y)$: Valor del filtro en la posición (x, y) .
- σ : Desviación estándar de la distribución gaussiana, que controla el grado de suavizado.
- x, y : Coordenadas en el espacio del kernel.

Este filtro toma un píxel de la imagen y calcula su nuevo valor mediante un promedio ponderado de su propia intensidad y las intensidades de sus vecinos centrandolo el kernel sobre el píxel en cuestión y se ejecuta una multiplicación de las intensidades por los pesos, es decir, para cada píxel cubierto por el kernel, multiplico su intensidad por el valor correspondiente en el kernel, lo que se conoce como el producto Haddamard, posteriormente se hace una suma ponderada de los productos de la operación anterior y se normaliza el resultado dividiendo la suma obtenida por la suma de todos los valores del kernel, que normalmente suele ser 1.

El efecto que tiene este filtro es que la suma de todos los valores adicionales que se han sumado a cada pixel tiene que ser igual a 0, respecto a los bordes es bastante probable que queden difuminados, también se genera un ruido aleatorio el cual es reducido porque sus valores extremos se suavizan mediante el promedio dependiendo del valor de σ , ya que si σ es muy pequeño se perderán menos estructuras de interés como bordes.

Como vemos el filtro gaussiano no es adecuado para la eliminación de ruido en imágenes de diagnóstico médico porque suaviza los bordes que separan estructuras anatómicas importantes, como órganos, tejidos musculares o tumores, lo que compromete detalles esenciales para una interpretación precisa. Aunque está diseñado para reducir ruido gaussiano, típico de la dispersión de rayos X o variaciones térmicas en los detectores, no diferencia entre ruido y patrones relevantes, lo que puede llevar a la pérdida de información crítica. Dado que en física médica es fundamental preservar las transiciones entre regiones de diferentes densidades para un diagnóstico fiable, el uso de este filtro puede degradar la calidad de las imágenes y afectar la precisión clínica.

A.1.5 Transformada de Radón

La transformada de Radón es una herramienta ampliamente utilizada en el procesamiento de imágenes médicas, concretamente en la tomografía computarizada, la transformada sigue la fórmula (56).

$$R_f(s, \theta) = \int_{-\infty}^{+\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - s) dx dy \quad (56)$$

Donde:

- $f(x, y)$: Función que representa la imagen a reconstruir.
- $R_f(s, \theta)$: Valor de la proyección obtenida para un ángulo θ y una posición s a lo largo de la proyección.

- δ : Función delta de Dirac, que actúa como selector integrando únicamente sobre la línea definida por $x \cos \theta + y \sin \theta = s$.
- θ : Ángulo de la proyección respecto al eje horizontal.
- s : Distancia desde el origen a la línea de integración.

Como vemos se trata de una operación lineal la cual mediante integrales a lo largo de líneas permite analizar una función, esta transformada se expresa en términos de valores puntuales pertenecientes a $f(x,y)$, y la transformada se expresa en términos de proyecciones las cuales pueden ser la suma o en este caso definida como integral de los valores a lo largo de cada línea de la imagen, esto es útil ya que en tomografía computarizada se obtiene la imagen desde múltiples ángulos, y a partir de estas se puede reconstruir la imagen.

Se suele implementar un filtrado multiescala previo a la retroproyección, para poder separar cada proyección en diferentes umbrales de detalle y lograr atenuar únicamente las componentes denominadas como ruido, sin que se afecte a las estructuras relevante.

El proceso se basa principalmente en calcular para cada ángulo y cada desplazamiento la integral de la imagen en esa línea, y posteriormente se puede invertir recuperando la imagen original, esta técnica debido a su simplicidad matemática es sencillo trabajar con ella, sin embargo, debido a que es una técnica lineal, no es capaz de aprender patrones complejos a diferencia de las redes neuronales CNN, además que en la retroproyección tiene que estar definido el ruido.

A.1.6 Transformadas de Fourier y Wavelet

La idea de la transformada de Fourier es descomponer una señal, que en este caso corresponde a una imagen, y mediante la fórmula (57), descomponerla en sus componentes sinusoidales, es decir, sus componentes frecuenciales.

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad (57)$$

donde:

- $f(x, y)$ es la imagen en el dominio espacial.
- $F(u, v)$ es la representación en el dominio de la frecuencia.
- u y v son las frecuencias espaciales.
- j es la unidad imaginaria ($j^2 = -1$).

Posteriormente, puedo aplicar diferentes tipos de filtros, como un filtro de paso bajo, que permite el paso de las frecuencias bajas y elimina las altas, atenuando así los cambios bruscos de intensidad en la imagen. Sin embargo, este enfoque no es adecuado para mi caso, ya que el ruido presente en la imagen es de tipo gaussiano, el cual se distribuye de forma aleatoria por toda la imagen. Además, también podría emplear máscaras diseñadas para reducir ruidos periódicos específicos, pero enfrentaría el mismo problema: estas técnicas no son efectivas para eliminar de manera adecuada el ruido gaussiano.

La transformada wavelet descompone una señal o imagen en una serie de funciones base llamadas wavelets, que se encuentran localizadas tanto en frecuencia como en el espacio. Esto permite analizar simultáneamente las frecuencias y su ubicación en la imagen. Esta transformada esta regida por la fórmula (58).

$$W(a, b) = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{\infty} f(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (58)$$

donde:

- $W(a, b)$ es el coeficiente wavelet, que depende de la escala a y la traslación b .
- $f(t)$ es la señal original en el dominio temporal.

- $\psi(t)$ es la función wavelet madre.
- $\psi^*(t)$ representa el conjugado complejo de la wavelet madre.
- a controla la escala de la wavelet (estrechamiento o expansión).
- b controla la posición de la wavelet en el tiempo.
- $\frac{1}{\sqrt{|a|}}$ es un factor de normalización para conservar la energía de la señal.

Sin embargo, enfrentamos el mismo inconveniente, ya que para identificar el ruido solemos emplear una umbralización global, eliminando las frecuencias altas o bajas según lo que se desee resaltar. No obstante, en una imagen cancerosa, el ruido no puede tratarse de esta manera, ya que no se reduce a un único umbral, debido a su naturaleza compleja y distribución aleatoria.

A.2. Técnicas avanzadas

Hasta ahora hemos visto técnicas bastante simples, matemáticamente, conceptualmente y computacionalmente hablando, comencemos a ver técnicas más complejas y puedan reconocer patrones automáticamente.

A.2.1 Non-Local Means (NLM)

La técnica conocida como Non-Local Means (NLM) se basa en aprovechar los patrones presentes en una imagen, los cuales suelen mantener una estructura de valores similares en diferentes regiones. Estos patrones permiten distinguir el ruido de la información verdaderamente relevante. A diferencia de otros métodos que utilizan un promedio local, NLM emplea un promedio ponderado de los valores de intensidad de los píxeles en toda la imagen. Los pesos de este promedio están determinados por la similitud entre las distintas regiones, lo que permite un filtrado más preciso y adaptado a la estructura de la imagen. Esta técnica utiliza la fórmula (59).

$$\hat{I}(x) = \frac{\sum_{y \in \Omega} w(x, y) I(y)}{\sum_{y \in \Omega} w(x, y)} \quad (59)$$

Donde:

- $\hat{I}(x)$ es la nueva intensidad calculada para el píxel x .
- Ω es el conjunto de píxeles en la imagen (o una ventana de búsqueda alrededor de x).
- $w(x, y)$ es el peso asignado a cada píxel y , basado en la similitud entre los parches alrededor de x y y .
- $I(y)$ es la intensidad del píxel y en la imagen original.

El peso de similitud se calcula como (60).

$$w(x, y) = \exp\left(-\frac{\|P(x) - P(y)\|^2}{h^2}\right) \quad (60)$$

Donde:

- $P(x)$ es el parche centrado en el píxel x .
- $\|P(x) - P(y)\|^2$ es la norma euclidiana al cuadrado entre los parches $P(x)$ y $P(y)$, que mide su diferencia.
- h es un parámetro de suavizado que controla la sensibilidad del filtro.

La idea es tomar un parche $P(x)$ de tamaño fijo centrado en x y compararlo con el parche (región) $P(y)$ de otros píxeles y dentro de una región de búsqueda la cual puede ser toda la imagen o una ventana alrededor, posteriormente se calcula $w(x,y)$ utilizando la fórmula del peso y se calcula la nueva intensidad (x) como un promedio ponderado de todas las intensidades, donde los pesos indican similitud.

Esta técnica es ideal para eliminar ruido aleatorio preservando detalles estructurales, sin embargo en las imágenes médicas obtenidas por tomografía computarizada que tenemos puede no ser la más indicada debido a que el ruido gaussiano no solamente es aleatorio sino que está influido por factores previamente mencionados como la cantidad de fotones detectados, la sensibilidad a las variaciones térmicas que poseen los sensores, etc. Lo que genera patrones un tanto intrincados y no siempre tendremos la facilidad de modelar de manera efectiva mediante similitudes locales y no locales como se emplean en NLM, además como hemos visto, NLM basa su efectividad en una comparación de parches dentro de una región definida, pero en las imágenes tomadas por CT el ruido puede ser tan sutil que las similitudes entre regiones pueden no considerar un parámetro de interés como ruido, en especial áreas con detalles anatómicos complejos.

Otro punto en contra de esta técnica es que es bastante sensible al ajuste de los parámetros como puede ser el tamaño del parche porque podemos no identificar bien una región que puede ser de interés y es complicado de ver a simple vista.

A.2.2 PCA (Probabilistic Component Analysis o Principal Component Analysis)

Esta técnica basada en estadística nos permite reducir la dimensionalidad de nuestros datos, pudiendo así comprender mejor los datos sin perder valores relevantes, en el contexto de eliminación de ruido, podemos tener un mejor control de lo que puede considerarse como ruido y los valores que corresponden a la imagen real.

Para poder lograr la reducción de dimensionalidad se utiliza una transformación basada en la descomposición de la matriz de covarianza del conjunto de datos, la cual se define como (61).

$$\Sigma = \frac{1}{n} \sum_{i=1}^n (X_i - \mu)(X_i - \mu)^T \quad (61)$$

donde:

- X_i representa cada muestra del conjunto de datos.
- μ es el vector de medias de los datos.
- n es el número total de muestras.
- Σ es la matriz de covarianza resultante.

Es decir, estamos calculando la media de los datos, construyo la matriz de la covarianza, posteriormente obtengo los valores propios y sus respectivos vectores propios y nos permite realizar una selección de los componentes principales.

Debido a que trabajamos con imágenes médicas, nuestros datos contienen una alta dimensionalidad por tanto necesitamos reducir el número de variables para comprimir la información en sus componentes principales, las cuales son variables nuevas generadas a partir de las variables pertenecientes al conjunto de datos, las componentes capturan la máxima varianza posible, pudiendo así representar la información de manera compacta, como vemos al no estar eliminando o perdiendo ningún tipo de información de la imagen nos permite tener un mayor control de interpretación en los datos, así como mayor eficiencia computacional debido a sus variables reducidas.

Cuando identificamos las componentes principales, los vectores propios que tengan los valores propios mas grandes, corresponden a las direcciones donde existe mayor varianza, y las primeras componentes principales son las que contienen la información más relevante, y las últimas son las que tienen los valores propios más pequeños lo que corresponde a menor varianza que comúnmente se atribuye al ruido, eliminamos las últimas componentes y posteriormente reconstruimos la imagen ya limpia.

Como vemos esta técnica nos brinda muchas ventajas en comparación con la red neuronal, sin embargo, como para cada imagen el número de componentes principales es un parámetro ajustable supone mucho tiempo perdido en la reconstrucción de una sola imagen debido a que se debe

eliminar correctamente el ruido sin eliminar información anatómica relevante, por otro lado, a diferencia de la red no distingue lo que es el ruido, por tanto podría eliminar anomalías relevantes ya no solo para la eliminación de ruido, sino para el diagnóstico, aunque es cierto que este tipo de técnica trabaja muy bien con ruido gaussiano, consideramos más completa la red neuronal de tipo U-NET.

B. Código Python

El código puede consultarse en GitHub: <https://github.com/alejandroovilar/TFG-AVA-CNN-U-NET>.

git

```
1 import matplotlib.pyplot as plt
2 import matplotlib.image as mpimg
3 import numpy as np
4 import tensorflow as tf
5 from tensorflow.keras import layers, models, regularizers
6 from tensorflow.keras.preprocessing.image import ImageDataGenerator
7 from tensorflow.keras.models import Model
8 from tensorflow.keras.models import load_model
9 from tensorflow.keras.layers import Input, Conv2D, MaxPooling2D,
    Conv2DTranspose, concatenate
10 from tensorflow.keras.callbacks import EarlyStopping
11 import os
12 import re
13 import cv2
14 import shutil
15 from skimage.metrics import structural_similarity as ssim
```

Código 1: Librerías

```
1 def improved_unet(input_shape=(128, 128, 1)):
2     inputs = layers.Input(shape=input_shape)
```

```
3
4 # Downsampling
5 conv1 = layers.Conv2D(16, (3, 3), activation="relu", padding="same")(
inputs)
6 conv1 = layers.Conv2D(16, (3, 3), activation="relu", padding="same")(conv1
)
7 pool1 = layers.MaxPooling2D((2, 2))(conv1)
8 pool1 = layers.BatchNormalization()(pool1)
9
10 conv2 = layers.Conv2D(32, (3, 3), activation="relu", padding="same")(pool1
)
11 conv2 = layers.Conv2D(32, (3, 3), activation="relu", padding="same")(conv2
)
12 pool2 = layers.MaxPooling2D((2, 2))(conv2)
13 pool2 = layers.BatchNormalization()(pool2)
14
15 conv3 = layers.Conv2D(64, (3, 3), activation="relu", padding="same")(pool2
)
16 conv3 = layers.Conv2D(64, (3, 3), activation="relu", padding="same")(conv3
)
17 pool3 = layers.MaxPooling2D((2, 2))(conv3)
18 pool3 = layers.BatchNormalization()(pool3)
19
20 # Bottleneck
21 bottleneck = layers.Conv2D(128, (3, 3), activation="relu", padding="same")
(pool3)
22 bottleneck = layers.Conv2D(128, (3, 3), activation="relu", padding="same")
(bottleneck)
23
24 # Upsampling
25 up3 = layers.Conv2DTranspose(64, (2, 2), strides=(2, 2), padding="same")(
bottleneck)
26 up3 = layers.concatenate([up3, conv3])
```

```
27 up3 = layers.Conv2D(64, (3, 3), activation="relu", padding="same")(up3)
28 up3 = layers.Conv2D(64, (3, 3), activation="relu", padding="same")(up3)
29
30 up2 = layers.Conv2DTranspose(32, (2, 2), strides=(2, 2), padding="same")(
up3)
31 up2 = layers.concatenate([up2, conv2])
32 up2 = layers.Conv2D(32, (3, 3), activation="relu", padding="same")(up2)
33 up2 = layers.Conv2D(32, (3, 3), activation="relu", padding="same")(up2)
34
35 up1 = layers.Conv2DTranspose(16, (2, 2), strides=(2, 2), padding="same")(
up2)
36 up1 = layers.concatenate([up1, conv1])
37 up1 = layers.Conv2D(16, (3, 3), activation="relu", padding="same")(up1)
38 up1 = layers.Conv2D(16, (3, 3), activation="relu", padding="same")(up1)
39
40 # Salida
41 outputs = layers.Conv2D(1, (1, 1), activation="sigmoid")(up1)
42
43 model = models.Model(inputs, outputs)
44 return model
```

Código 2: U-Net

```
1 def improved_unet(input_shape=(128, 128, 1)):
2     inputs = layers.Input(shape=input_shape)
```

Código 3: Capa de entrada

```
1 # Downsampling
2 conv1 = layers.Conv2D(16, (3, 3), activation="relu", padding="same")(inputs)
3 conv1 = layers.Conv2D(16, (3, 3), activation="relu", padding="same")(conv1)
4 pool1 = layers.MaxPooling2D((2, 2))(conv1)
5 pool1 = layers.BatchNormalization()(pool1)
6
```

```
7 conv2 = layers.Conv2D(32, (3, 3), activation="relu", padding="same")(pool1)
8 conv2 = layers.Conv2D(32, (3, 3), activation="relu", padding="same")(conv2)
9 pool2 = layers.MaxPooling2D((2, 2))(conv2)
10 pool2 = layers.BatchNormalization()(pool2)
11
12 conv3 = layers.Conv2D(64, (3, 3), activation="relu", padding="same")(pool2)
13 conv3 = layers.Conv2D(64, (3, 3), activation="relu", padding="same")(conv3)
14 pool3 = layers.MaxPooling2D((2, 2))(conv3)
15 pool3 = layers.BatchNormalization()(pool3)
```

Código 4: Downsampling

```
1 #Bottleneck
2 bottleneck = layers.Conv2D(128, (3, 3), activation="relu", padding="same")(
    pool3)
3 bottleneck = layers.Conv2D(128, (3, 3), activation="relu", padding="same")(
    bottleneck)
```

Código 5: Bottleneck

```
1 #Upsampling
2 up3 = layers.Conv2DTranspose(64, (2, 2), strides=(2, 2), padding="same")(
    bottleneck)
3 up3 = layers.concatenate([up3, conv3])
4 up3 = layers.Conv2D(64, (3, 3), activation="relu", padding="same")(up3)
5 up3 = layers.Conv2D(64, (3, 3), activation="relu", padding="same")(up3)
6
7 up2 = layers.Conv2DTranspose(32, (2, 2), strides=(2, 2), padding="same")(
    up3)
8 up2 = layers.concatenate([up2, conv2])
9 up2 = layers.Conv2D(32, (3, 3), activation="relu", padding="same")(up2)
10 up2 = layers.Conv2D(32, (3, 3), activation="relu", padding="same")(up2)
11
12 up1 = layers.Conv2DTranspose(16, (2, 2), strides=(2, 2), padding="same")(
```

```

up2)
13 up1 = layers.concatenate([up1, conv1])
14 up1 = layers.Conv2D(16, (3, 3), activation="relu", padding="same")(up1)
15 up1 = layers.Conv2D(16, (3, 3), activation="relu", padding="same")(up1)
    
```

Código 6: Upsampling

```

1 #Salida
2 outputs = layers.Conv2D(1, (1, 1), activation="sigmoid")(up1)
3
4 model = models.Model(inputs, outputs)
5 return model
    
```

Código 7: Capa de salida

```

1 #Compilar el modelo
2 model.compile(optimizer="adam", loss="mse", metrics=["mae"])
    
```

Código 8: Compilación del modelo

Table 1: Resumen del modelo

Layer (type)	Output Shape	Param #	Connected to
input_layer_5 (InputLayer)	(None, 128, 128, 1)	0	-
conv2d_48 (Conv2D)	(None, 128, 128, 16)	160	input_layer_5[0][0]
conv2d_49 (Conv2D)	(None, 128, 128, 16)	2,320	conv2d_48[0][0]
max_pooling2d_3 (MaxPooling2D)	(None, 64, 64, 16)	0	conv2d_49[0][0]

Continúa en la página siguiente

Table 1 – continuación de la anterior

Layer (type)	Output Shape	Param #	Connected to
batch_normalization_13 (BatchNormalization)	(None, 64, 64, 16)	64	max_pooling2d_3[0][0]
conv2d_50 (Conv2D)	(None, 64, 64, 32)	4,640	batch_normalization_13[0][0]
conv2d_51 (Conv2D)	(None, 64, 64, 32)	9,248	conv2d_50[0][0]
max_pooling2d_4 (MaxPooling2D)	(None, 32, 32, 32)	0	conv2d_51[0][0]
batch_normalization_14 (BatchNormalization)	(None, 32, 32, 32)	128	max_pooling2d_4[0][0]
conv2d_52 (Conv2D)	(None, 32, 32, 64)	18,496	batch_normalization_14[0][0]
conv2d_53 (Conv2D)	(None, 32, 32, 64)	36,928	conv2d_52[0][0]
max_pooling2d_5 (MaxPooling2D)	(None, 16, 16, 64)	0	conv2d_53[0][0]
batch_normalization_15 (BatchNormalization)	(None, 16, 16, 64)	256	max_pooling2d_5[0][0]
conv2d_54 (Conv2D)	(None, 16, 16, 128)	73,856	batch_normalization_15[0][0]
conv2d_55 (Conv2D)	(None, 16, 16, 128)	147,584	conv2d_54[0][0]
conv2d_transpose_6 (Conv2DT Transpose)	(None, 32, 32, 64)	32,832	conv2d_55[0][0]
Continúa en la página siguiente			

Table 1 – continuación de la anterior

Layer (type)	Output Shape	Param #	Connected to
concatenate_6 (Concatenate)	(None, 32, 32, 128)	0	conv2d_transpose_6[0][0]...
conv2d_56 (Conv2D)	(None, 32, 32, 64)	73,792	concatenate_6[0][0]
conv2d_57 (Conv2D)	(None, 32, 32, 64)	36,928	conv2d_56[0][0]
conv2d_transpose_7 (Conv2DT Transpose)	(None, 64, 64, 32)	8,224	conv2d_57[0][0]
concatenate_7 (Concatenate)	(None, 64, 64, 64)	0	conv2d_transpose_7[0][0]...
conv2d_58 (Conv2D)	(None, 64, 64, 32)	18,464	concatenate_7[0][0]
conv2d_59 (Conv2D)	(None, 64, 64, 32)	9,248	conv2d_58[0][0]
conv2d_transpose_8 (Conv2DT Transpose)	(None, 128, 128, 16)	2,064	conv2d_59[0][0]
concatenate_8 (Concatenate)	(None, 128, 128, 32)	0	conv2d_transpose_8[0][0]...
conv2d_60 (Conv2D)	(None, 128, 128, 16)	4,624	concatenate_8[0][0]
conv2d_61 (Conv2D)	(None, 128, 128, 16)	2,320	conv2d_60[0][0]
conv2d_62 (Conv2D)	(None, 128, 128, 1)	17	conv2d_61[0][0]