



UNIVERSIDAD NACIONAL DEL SUR

Proyecto final de carrera

**Reducción de ruido en imágenes de
Tomografías Computadas utilizando
Redes Neuronales**

Autor:
Juan I. Pisula

Directores:
Claudio Delrieux
Felix Thomsen

Índice

| | |
|---|-----------|
| 1. Introducción | 2 |
| 2. Estudios de Tomografías Computadas | 5 |
| 2.1. Instalaciones | 5 |
| 2.2. Escáner | 6 |
| 2.3. Proyecciones y reconstrucción de imágenes | 8 |
| 2.4. Uso clínico de TCs | 11 |
| 2.5. Monitoreo de hueso | 12 |
| 2.6. Ruido y artefactos presentes en las Tomografías Computadas | 13 |
| 2.6.1. Artefacto de anillo | 13 |
| 2.6.2. Endurecimiento y dispersión del haz de rayos X | 13 |
| 2.6.3. Artefactos por movimiento | 14 |
| 2.6.4. Ruido | 14 |
| 3. Trabajos previos en reducción de ruido de TCs | 16 |
| 3.1. Procesamiento de sinogramas | 16 |
| 3.2. Reconstrucción Iterativa | 16 |
| 3.3. Procesamiento sobre imágenes reconstruidas | 18 |
| 4. Redes Neuronales de Convolución | 20 |
| 4.1. Introducción | 20 |
| 4.2. Arquitectura | 20 |
| 4.2.1. Capas de convolución | 20 |
| 4.2.2. Capas de <i>pooling</i> | 23 |
| 4.2.3. Capas de activación | 24 |
| 4.2.4. Capas lineales | 24 |
| 4.3. Entrenamiento de una Red Neuronal | 25 |
| 5. RNC para reducción de ruido en TCs | 27 |
| 5.1. Estrategia propuesta | 27 |
| 5.2. Arquitectura | 28 |
| 5.3. Conjunto de datos | 29 |
| 5.4. Entrenamiento | 31 |
| 6. Resultados | 35 |
| 7. Discusión | 39 |
| 8. Conclusión | 40 |

1. Introducción

La Tomografía Computada (TC) es un método de adquisición de imágenes basado en rayos X que permite obtener imágenes de cortes transversales de un objeto. Su principio de operación se basa en una fuente de rayos X que gira alrededor del objeto en cuestión, y que en su lado opuesto posee detectores de rayos X. Una vez adquiridos los datos del escaneo, los cortes transversales del objeto se obtienen a partir de un proceso de reconstrucción, donde los valores de los píxeles reflejan los coeficientes de absorción de rayos X. Con un amplio uso en medicina, esta modalidad de imágenes es utilizada para analizar distintas partes del cuerpo con el objetivo de encontrar y monitorizar tumores, hemorragias, malformaciones, y otros tipos de condiciones, o para realizar diagnósticos de ciertas patologías.

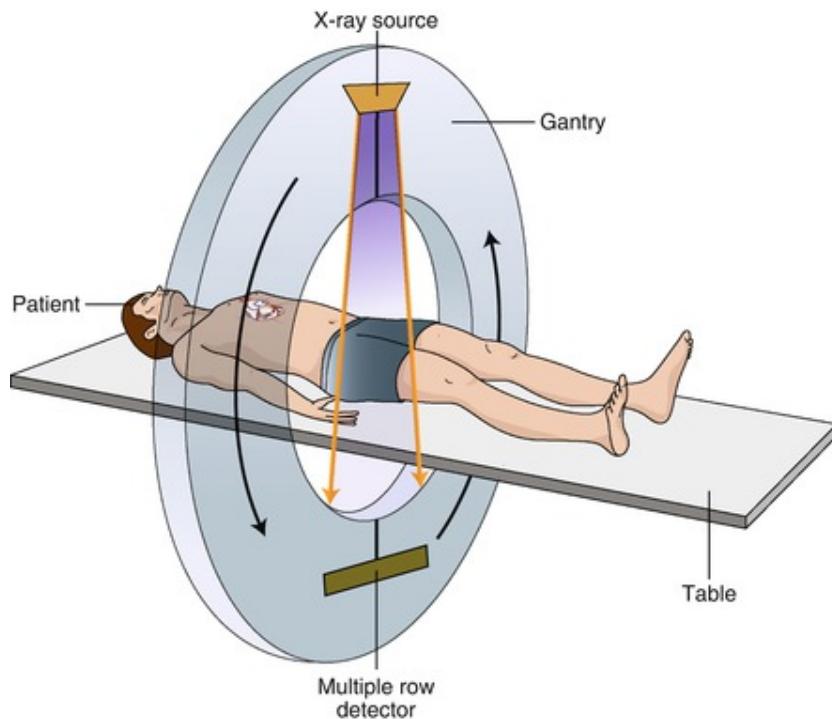


Figura 1: Principio de operación de un escáner de TC: la fuente y los sensores de rayos X giran alrededor del paciente, que se encuentra acostado atravesando el portal, o *gantry*.

Sin embargo, la radiación ionizante posee efectos biológicos que pueden resultar en la incidencia de cáncer, por lo que los estudios de tomografías computadas poseen un riesgo asociado a contraer esta enfermedad. Algunas cifras estiman que en EE.UU. entre el 1,5% y el 2% de las personas con

cáncer se deben al uso de TCs, y que 29.000 nuevos casos estarán relacionados a escaneos realizados en el año 2007 [1, 2].

Por estos motivos, la dosis de radiación incidente sobre el paciente debe ser minimizada tanto como sea razonable. Esto se puede lograr acortando el tiempo de exposición, así como reduciendo la corriente de tubo del escáner. No obstante, aunque disminuir la dosis de radiación resulte en un estudio menos nocivo para el paciente, el efecto producido en las imágenes TCs reconstruidas es el aumento de la presencia de ruido, por lo que se podría comprometer el diagnóstico.

Por otro lado, en el campo de las Ciencias de la Computación, en los últimos años se han producido grandes avances en tópicos de Aprendizaje Profundo, o *Deep Learning* [3], principalmente en Redes Neuronales (RNs) y Redes Neuronales de Convolución (RNCs) para Procesamiento de Imágenes. Actualmente estos algoritmos están presentes en tecnologías utilizadas en la vida cotidiana, y además son usados en diversas disciplinas científicas, así como lo es la Medicina y el Procesamiento de Imágenes Médicas. Sus principales aplicaciones en este campo son la detección de objetos (ej.: localizar lesiones), segmentación de objetos (ej.: contornear una lesión), y clasificar objetos (ej.: clasificar una lesión en maligna o benigna).

El objetivo de este trabajo es estudiar la aplicación de Redes Neuronales de Convolución para la reducción de ruido de imágenes de Tomografías Computadas Cuantitativas de Baja Resolución (TCCBR). Más concisamente, sobre el caso particular de TCs de vértebras humanas. La última aplicación de este concepto consistiría en mejorar el proceso de diagnóstico de la osteoporosis, una patología del esqueleto caracterizada por una baja densidad y calidad de la estructura interna del calcio dentro del hueso, predisponiendo

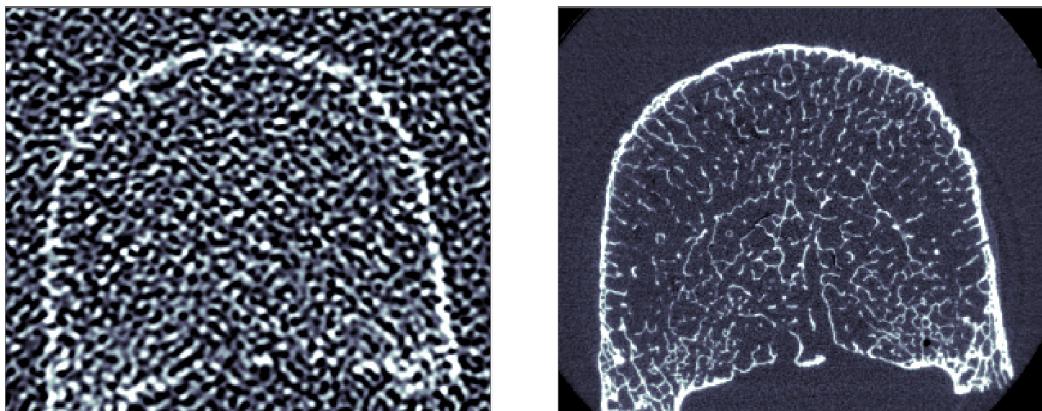


Figura 2: Comparación entre cortes transversales de una TC de baja resolución (izquierda) y TC de alta resolución (derecha) de una vértebra.

a la persona a un mayor riesgo de fractura. Poseer imágenes de tomografía de alta calidad a partir de baja dosis de radiación permitiría hacer un mejor análisis de la estructura del hueso del paciente, sin someterlo a dosis de radiación que podrían comprometer su salud.

Hasta aquí se ha introducido la temática de este trabajo. Los capítulos 2 y 3 tratan conceptos relacionados a las Tomografías Computadas, y a trabajos realizados en el tópico de la reducción de su ruido, respectivamente. El capítulo 4 introduce conceptos básicos de Redes Neuronales de Convolución, y en el capítulo 5 se trata con la RNC que se ha elaborado en este proyecto. Los resultados son mostrados en el capítulo 6, seguidos por la discusión en el capítulo 7, y la conclusión es elaborada en el capítulo 8.

2. Estudios de Tomografías Computadas

2.1. Instalaciones

La mayoría de los hospitales de grandes ciudades cuentan con escáneres de TC para ser utilizados en diagnósticos de varias condiciones. La instalación donde se encuentra el escáner debe estar preparada para realizar los estudios. Sus pisos deben ser capaces de soportar el peso del equipo, ser antiresbalante y antiestático, y las paredes deben ser construidas con material absorbente de rayos X. Una habitación separada del escáner debe existir para los radiólogos, y debe poseer una ventana con un vidrio especial que contiene plomo para proteger de la radiación. Las habitaciones utilizadas deben cumplir todas las normas de seguridad e higiene que son exigidas para la realización de estudios médicos que utilizan radiación ionizante.

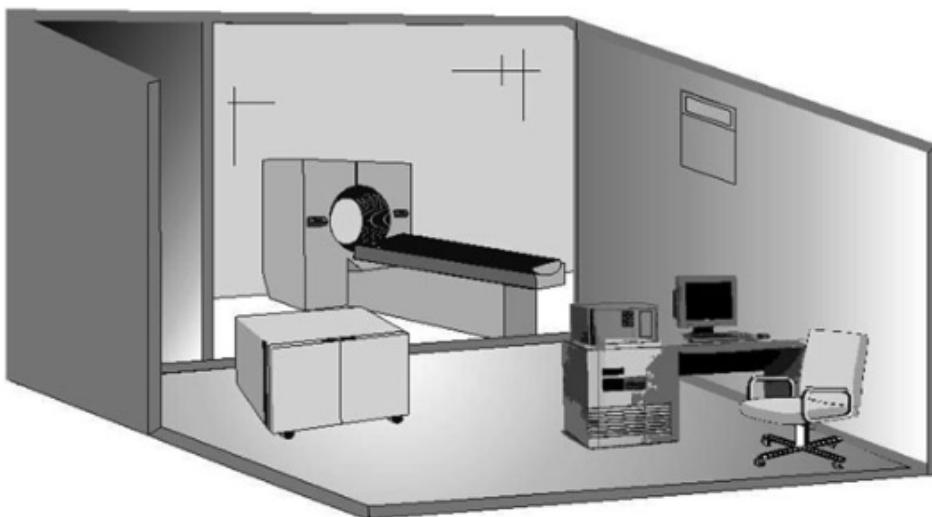


Figura 3: Ilustración de una instalación típica para estudios de TC
(Cierniak, [4]).

Un sistema completo de para la realización de Tomografías Computadas está constituido por los siguientes principales elementos:

- un sistema de adquisición de datos que lleva a cabo las proyecciones de rayos X,
- una computadora para reconstruir las imágenes desde las proyecciones y asistir en el análisis,

- una fuente eléctrica,
- un monitor para visualizar las imágenes producidas,
- una unidad para almacenar los datos producidos.

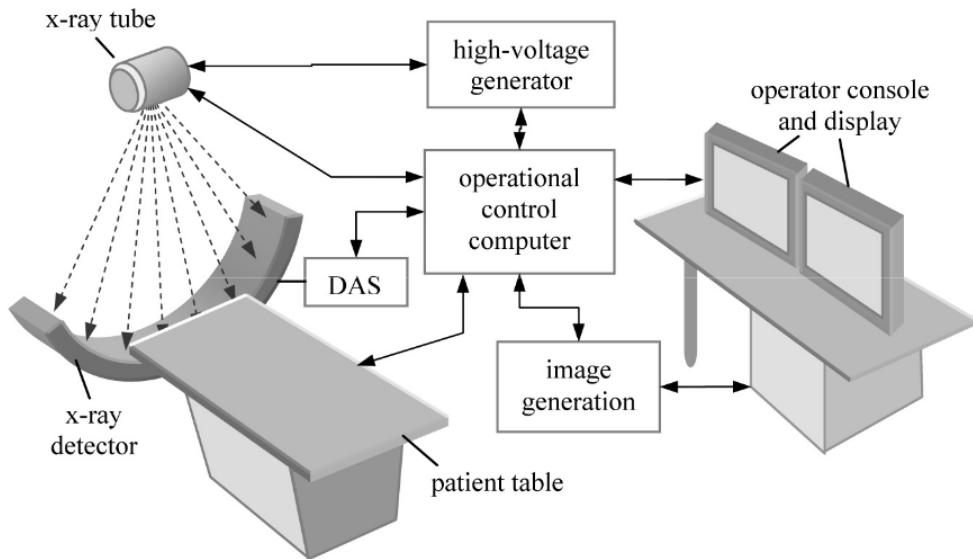


Figura 4: Diagrama en bloques de un sistema de TC (Hsieh, [5]).

2.2. Escáner

El escáner es el sistema de adquisición de datos. Sus principales componentes son los siguientes:

- El portal, o *gantry*. Es el anillo donde se sitúa al paciente. Dentro de él se hallan la fuente y detectores de rayos X, fuente de tensión de alto voltaje, y demás electrónica que estos dispositivos necesitan;
- Tubo de rayos X. Es la fuente de radiación ionizante que pasa a través del cuerpo del paciente;
- Arreglo de detectores de rayos X. Son los sensores que captan los rayos X y transducen los valores de intensidad de radiación en cantidades eléctricas;
- La camilla donde se sitúa al paciente [4, 5].

El mecanismo para realizar las proyecciones ha evolucionado a lo largo del tiempo. Se pueden distinguir distintas generaciones de escáneres de TC. Los escáneres de primera generación emplean un haz de rayos X paralelo y un único detector. Para un dado ángulo de rotación, el par fuente-detector es trasladado incrementalmente sobre una línea recta. La proyección es formada al medir la salida del detector en cada incremento en translación. Luego de completar una translación, el ángulo de rotación del par fuente-detector es incrementado y se repite el procedimiento para formar una proyección en este ángulo de radiación. El procedimiento es repetido para cada ángulo deseado en el rango de $[0^\circ, 180^\circ]$ para formar un corte transversal del paciente. Una vez terminado uno de estos escaneos, se puede mover al paciente de manera incremental atravesando el portal y repetir el proceso. La colección de cortes transversales del paciente es utilizada para formar un volumen tridimensional del mismo.

Los escáneres de segunda generación trabajan bajo el mismo principio de operación, pero un haz en forma de abanico es utilizado, por lo que se emplean más de un único detector y entonces se requieren menos translaciones del par fuente-arreglo de detectores. La tercera generación de escáneres emplean un haz de rayos X más amplio, y un arreglo de detectores lo suficientemente grande como para cubrir su campo de visión. Cada ángulo de rotación produce una proyección, eliminando la necesidad de trasladar el par fuente-arreglo de detectores. Los sistemas de cuarta generación emplean un arreglo circular de detectores en forma de anillo, por lo que sólo la fuente de radiación debe rotar.

Existen otros dispositivos más modernos que utilizan mecanismos más sofisticados. Los escáneres de quinta generación, llamados escáneres de Tomografía Computada de Haz de Electrones, eliminan el movimiento mecánico al emplear haces de electrones controlados electromagnéticamente. En los escáneres de TC en espiral o de sexta generación el paciente es movido en simultáneo a la rotación del par fuente-arreglo de detectores. Los escáneres multicorte o de séptima generación utilizan un haz de rayos X en forma de cono, y bancos de detectores paralelos que producen más de un corte transversal a la vez.

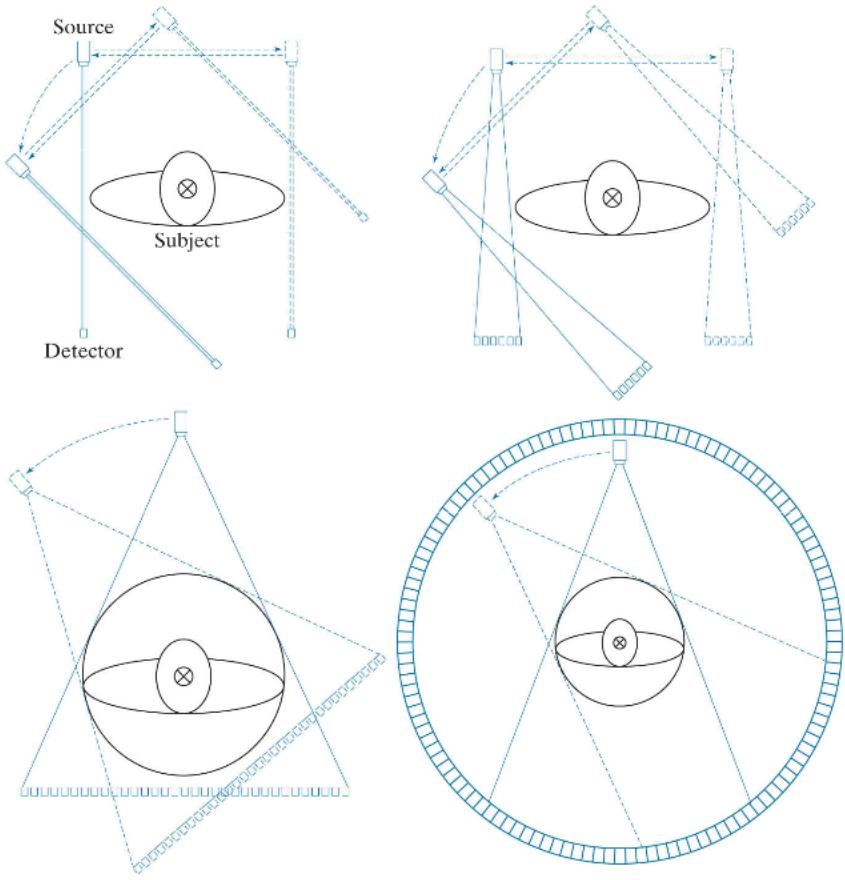


Figura 5: Movimientos de las fuentes de radiación y sensores en las primeras 4 generaciones de escáneres de TC (ordenadas de izquierda a derecha, de arriba a abajo). El movimiento del paciente se da perpendicular al plano de la página (Gonzalez-Woods, [6]).

2.3. Proyecciones y reconstrucción de imágenes

En esta sección se explica el proceso de proyección de rayos X y la reconstrucción de la imagen del objeto escaneado. Las matemáticas resultan más sencillas al explicar el caso de haces de rayos X paralelos. Es posible extender estos conceptos para haces en abanico o en cono.

Considérese la figura 6. La proyección producida al irradiar el objeto para cierto ángulo de rotación θ_k es $g(\rho, \theta_k)$. Los haces de rayos X están representados por las líneas L perpendiculares al plano de proyección. Estas líneas responden a la ecuación $L(\rho_j, \theta_k) = \{(x, y) \in R^2 : x \cos(\theta_k) + y \sin(\theta_k) = \rho_j\}$. El valor de la proyección $g(\rho, \theta_k)$ para un ángulo θ_k queda determinado por

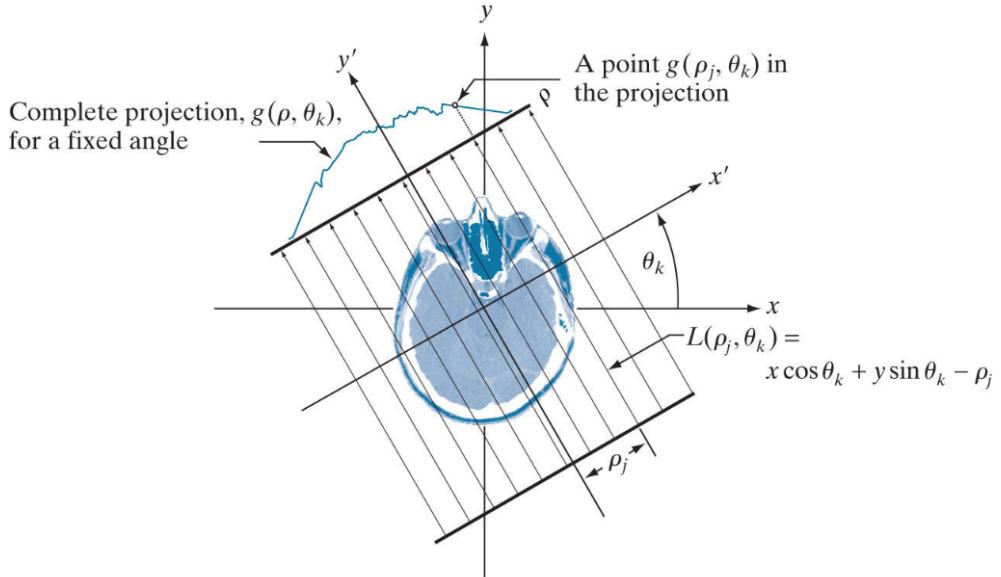


Figura 6: Proyección producida por un escáner de haces paralelos, para un ángulo θ_k (Gonzalez-Woods, [6]).

la integral de línea

$$g(\rho, \theta_k) = \int_{L(\rho, \theta_k)} f(x, y) ds. \quad (1)$$

Considerando todos los angulos posibles de rotación, se puede escribir entonces

$$g(\rho, \theta) = \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy, \quad (2)$$

donde la función impulso, δ indica que la integral es cero al menos que su argumento sea cero, es decir, la integral sólo se calcula sobre los puntos de la línea $L(\rho, \theta)$.

La función $g(\rho, \theta)$ es la denominada Transformada de Radon de $f(x, y)$. En la figura 7 se puede ver el resultado de la aplicación de la transformada de Radon sobre el fantoma Shepp-Logan. Este tipo de visualización de la transformada de Radon es conocido como sinograma.

En el caso discreto, la transformada de Radon de $f(x, y)$ se puede computar como

$$g(\rho, \theta) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \delta(x \cos \theta + y \sin \theta - \rho) dx dy, \quad (3)$$

donde x e y son ahora variables discretas, M y N son las dimensiones del área rectangular sobre la cual la transformada es calculada.

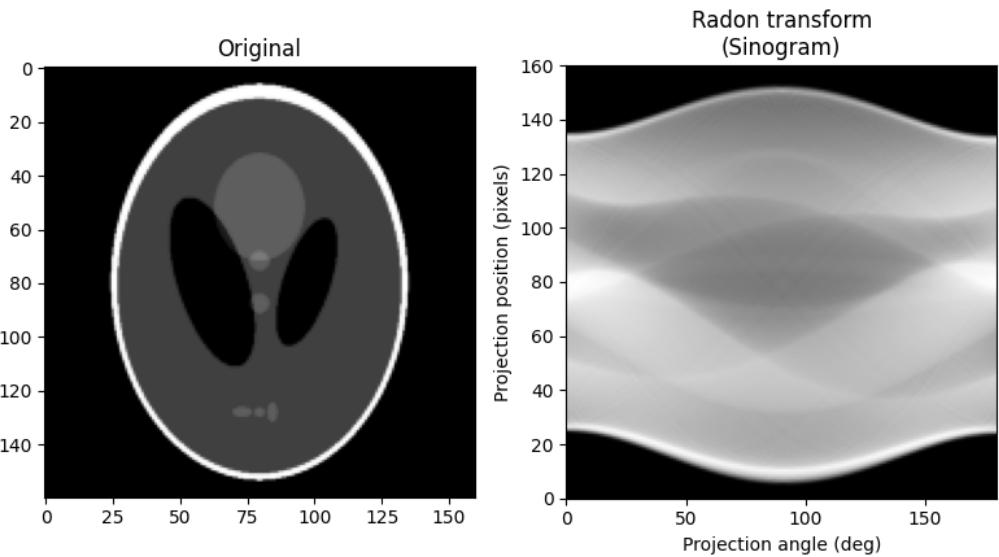


Figura 7: Izquierda: fantoma Shepp-Logan. Derecha: sinograma producido al aplicar la transformada de Radon.

Para reconstruir el objeto que se ha escaneado es necesario entonces realizar una transformada inversa. Dada una proyección $g(\rho, \theta_k)$, se puede calcular su retroproyección como

$$f_{\theta_k}(x, y) = g(\rho, \theta_k) = g(x \cos \theta_k + y \sin \theta_k, \theta_k). \quad (4)$$

Integrando sobre todos los posibles valores de ángulo θ se tiene entonces

$$f(x, y) = \int_0^\pi f_\theta(x, y) d\theta, \quad (5)$$

o su forma discreta

$$f(x, y) = \sum_{\theta=0}^{\pi} f_\theta(x, y), \quad (6)$$

donde ahora x, y, θ son cantidades discretas.

De la figura 7 se puede ver que una retroproyección resulta en una imagen difuminada. Se puede calcular entonces la Retroproyección Filtrada, o *Filtered Backprojection* por su nombre en inglés.

$$f(x, y) = \int_0^\pi [s(\rho) * g(\rho, \theta)]_{\rho=x \cos \theta + y \sin \theta} d\theta, \quad (7)$$

donde $*$ es el operador de convolución, y $s(\rho)$ un filtro rampa unidimensional. Por propiedad de la operación de convolución, es posible también realizar este cómputo en el campo transformado de Fourier [6].

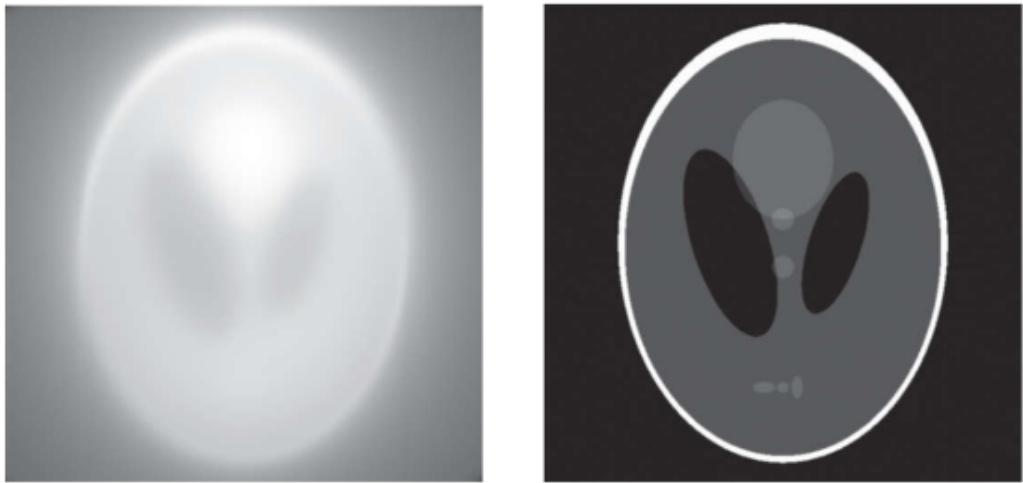


Figura 8: Reconstrucción del fantoma Shepp-Logan. Izquierda: retroproyección. Derecha: retroproyección filtrada (adaptada de Gonzalez-Woods, [6]).

2.4. Uso clínico de TCs

A diferencia de otras modalidades de imágenes médicas, las TCs permiten diferenciar distintos tipos de tejidos y órganos. Las TCs son útiles para buscar tumores y lesiones, y además permiten localizarlas e indicar su tamaño.

Las TCs se pueden utilizar en todas las regiones anatómicas. Por ejemplo, escaneos de la cabeza permiten mostrar tumores, coágulos de sangre, defectos en venas, así como anomalías en los nervios y músculos del ojo. TCs del tórax permiten revelar estructuras nodulares, infiltrados pulmonares, fibrosis, o efusiones pleurales.

Esta modalidad de imágenes es utilizada también para guiar procedimientos mínimamente invasivos como biopsias. Las TCs son también utilizadas

en el tratamiento para el cáncer, tanto para realizar planeamiento de radioterapia como para seguimiento de la evolución de la enfermedad [7].

2.5. Monitoreo de hueso

Cuando esta modalidad de imágenes es utilizada para el estudio de huesos, permite obtener valores de densidad ósea volumétrica que pueden ser calibrados en mg/cm^3 . Otras modalidades de rayos X como las radiografías convencionales o la Radioabsorciometría de Doble Energía, producen imágenes bidimensionales por lo que sólo permiten obtener valores de densidad de área.

La reproducibilidad y precisión de las mediciones de densidad ósea puede ser aumentada al introducir un cuerpo de calibración en la imagen. El cuerpo de calibración provee valores representativos de hueso I_B y tejido suave I_S . Con estos datos, la densidad ósea calibrada en miligramos de hidroxipatita por mililitro de volumen de hueso, D , puede ser calculada a partir del valor promedio de imagen $\langle I \rangle$ en la región de hueso y el valor conocido de densidad ósea, ρ_B :

$$D = \frac{\langle I \rangle - I_S}{I_B - I_S} \rho_B. \quad (8)$$

Esta modalidad de imágenes que arroja valores calibrados de densidad ósea es denominada Tomografía Computada Cuantitativa (TCC). Estudios clínicos *in vivo* utilizan dispositivos con baja resolución espacial, asociada a un amplio espesor de cortes transversales. Los dispositivos de este tipo poseen una resolución nominal de alrededor de $300 \times 300 \times 300 \mu m^3$ y son usados para mediciones de densidad ósea.

Dispositivos con mayor resolución espacial permiten capturar la microestructura ósea. Los dispositivos de Tomografía Computada Cuantitativa Periférica de Alta Resolución (TCCPAR) alcanzan una resolución nominal mayor ($61 \times 61 \times 61 \mu m^3$ en un Scanco XTreme CT II), pero son sólo aplicables a estudios *ex situ* de la espina o el calcáneo. *Micro CT* es una modalidad de imágenes preclínica con una resolución nominal de hasta $0,5 \times 0,5 \times 0,5 \mu m^3$, sin embargo dado que admite un tamaño reducido de muestra, y a la alta dosis de radiación, su uso queda limitado al de estudios *ex situ* [8, 9].

2.6. Ruido y artefactos presentes en las Tomografías Computadas

2.6.1. Artefacto de anillo

Un elemento detector descalibrado o defectuoso genera un anillo brillante u oscuro centrado en el centro de rotación, lo que puede ser confundido por una patología.



Figura 9: Artefacto de anillo en una TC de cráneo. Cortesía de Dr. Vikas Shah, Radiopaedia.org, rID 34473.

2.6.2. Endurecimiento y dispersión del haz de rayos X

El endurecimiento y la dispersión de los rayos X son dos mecanismos distintos causados por objetos de alta atenuación. El endurecimiento del haz ocurre al utilizarse rayos X policromáticos. Los materiales poseen mayor atenuación sobre fotones de menor energía, y este efecto es más visible en materiales densos como hueso o metal. El artefacto producido es de franjas oscurecidas entre dos materiales con alta atenuación.

La dispersión de los haces de rayos X causa que los haces cambien de dirección y entonces sean detectados por otros sensores. Este error se ve magnificado cuando un haz que ha sido dispersado es captado por un sensor que debería detectar una reducida cantidad de fotones. Por ejemplo, si un

implante metálico bloquea todos los fotones en una dirección, los detectores correspondientes solo recibirán fotones dispersados.

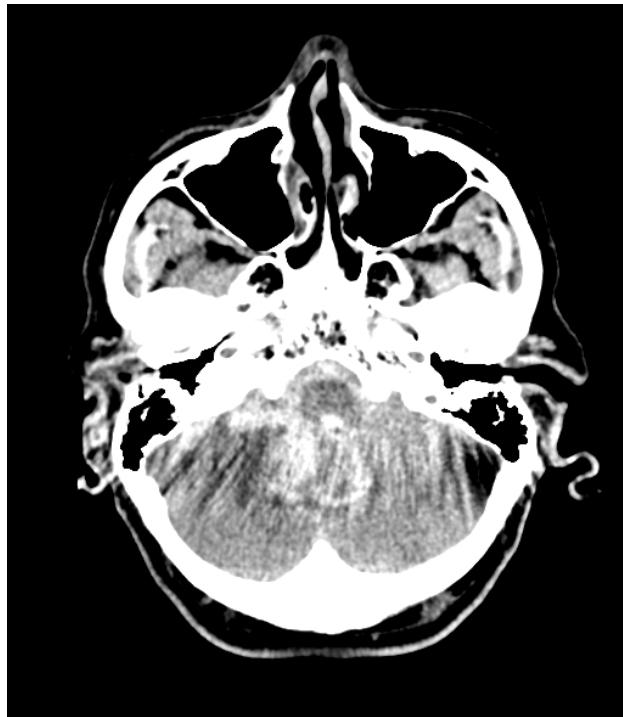


Figura 10: Endurecimiento del haz de rayos X. Cortesía de Dr Balint Botz , Radiopaedia.org, rID 63003.

2.6.3. Artefactos por movimiento

Los movimientos respiratorios, cardíacos, intestinales o del paciente en sí producen imágenes difuminadas y duplicadas, así como rayas prolongadas [10].

2.6.4. Ruido

La principal fuente de ruido es el ruido cuántico también llamado ruido shot o ruido Poisson dado que puede ser modelado por un proceso de Poisson. El ruido cuántico es una incertidumbre fundamental asociada al problema de medición de luz (en este caso, rayos X), debido a la naturaleza cuantizada de la misma.

Los detectores de rayos X (ya sean digitales o basados en películas fotosensibles) miden la intensidad de la señal contando el número de fotones incidentes en el sensor en unos intervalo de tiempo. Las llegadas de fotones

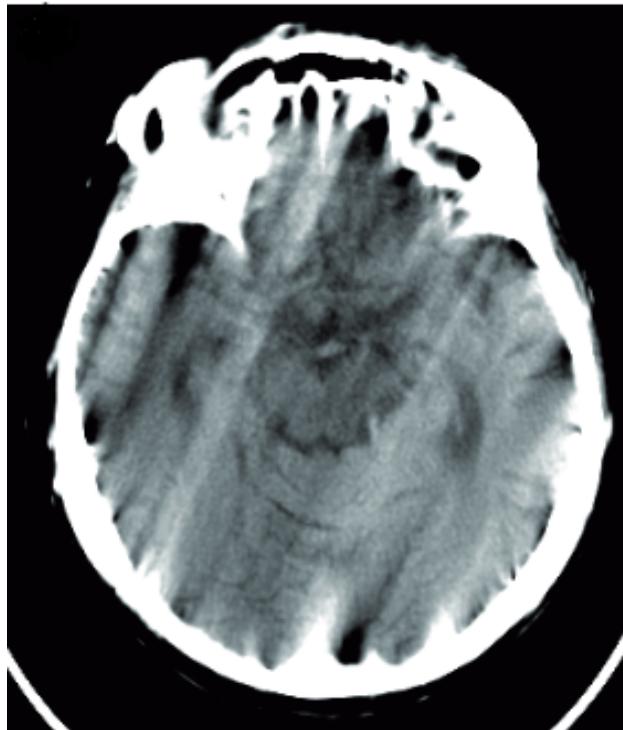


Figura 11: Artefacto por movimiento del paciente (Boas y Fleischmann, [10]).

individuales pueden ser consideradas como eventos independientes que siguen una distribución temporal aleatoria. El conteo de fotones es un proceso de Poisson, y el número k de fotones medido por un sensor en un intervalo de tiempo t es descrito por la distribución discreta de probabilidad

$$P(k) = \frac{e^{-\lambda t} (\lambda t)^k}{k!}, \quad (9)$$

donde λ es el número esperado de fotones por unidad de tiempo, proporcional a la irradiancia incidente.

Del proceso de Poisson, para reducir el nivel de ruido sería necesario capturar más fotones por parte del sensor. Esto requeriría un mayor tiempo de exposición o incrementar la intensidad de radiación de rayos X. Sin embargo, el sensor puede medir una cantidad limitada de fotones. Además, como ya se había mencionado, un elevado tiempo de exposición o una alta dosis de radiación podría tener efectos negativos para la salud del paciente [11].

3. Trabajos previos en reducción de ruido de TCs

Para abordar el problema de la reducción de ruido en imágenes médicas, distintos algoritmos han sido diseñados con el objetivo de mejorar la calidad de TCs de baja dosis de radiación. Estos métodos se pueden clasificar en tres grupos: procesamiento sobre sinogramas, reconstrucción iterativa, y procesamiento sobre imágenes reconstruidas [12, 13, 14, 15].

3.1. Procesamiento de sinogramas

Los métodos basados en el procesamiento de sinogramas trabajan sobre la representación visual de los datos brutos producidos por los escáneres previamente a la etapa de reconstrucción. La principal ventaja de estos métodos es que la distribución de ruido es conocida y caracterizada, y además son los métodos más eficientes computacionalmente [13].

Dado que los sinogramas son señales 2D, el procesamiento de imágenes tradicional ha sido estudiado en este tópico. Algunas de las técnicas estudiadas son filtrado adaptativo [16], filtrado bilateral [17], y *Penalized Weighted Least-Squares* [18].

3.2. Reconstrucción Iterativa

Los algoritmos de Reconstrucción Iterativa (RI) se basan en mejorar la calidad de imágenes reconstruidas usando etapas donde las imágenes se reconstruyen a partir de sinogramas, y donde se generan sinogramas a partir de imágenes reconstruidas. El objetivo de una RI completa es producir imágenes reconstruidas que se corresponden al objeto medido en cuestión. Para hacer esto, estos algoritmos incluyen modelos del escáner de TC, del objeto que se desea medir, y de la distribución de ruido en los sinogramas.

El primer paso en estos algoritmos es una reconstrucción por retroproyección filtrada para generar una imagen TC a partir de los datos brutos. Posteriormente, un sinograma es generado a partir de la imagen TC, simulando el proceso de adquisición de datos y usando información *a priori* de las características del escáner (ej.: la geometría de la máquina, la eficiencia de los detectores de rayos X, ruido electrónico). Luego de la comparación entre el sinograma original y el sinograma construido, se produce una nueva imagen TC con las correcciones realizadas. Estos pasos son repetidos un determinado número de veces, o hasta que las correcciones realizadas no produzcan modificaciones significativas.

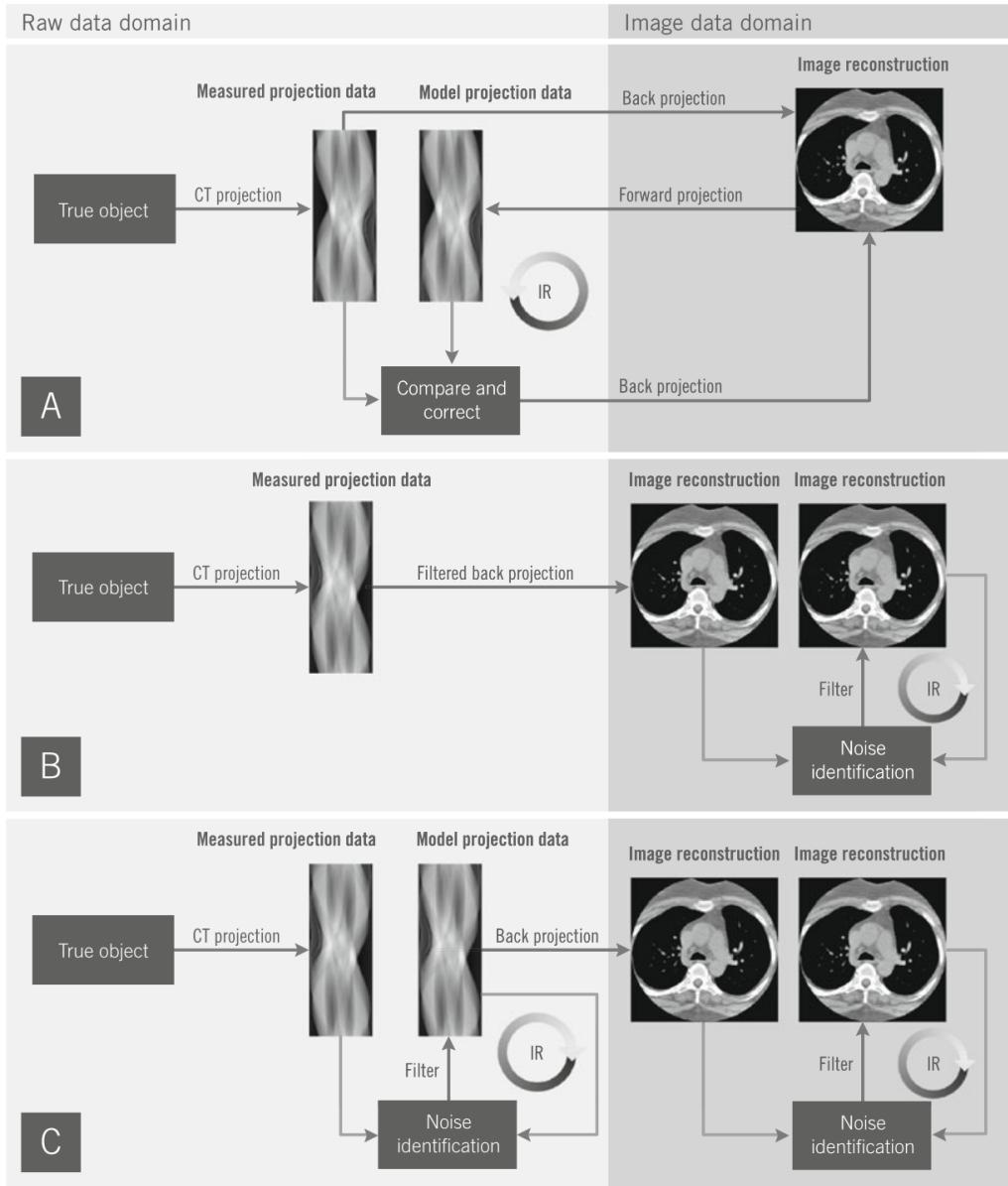


Figura 12: Distintas estrategias de reconstrucción iterativa. A) Iteraciones que utilizan reconstrucción de imágenes TC y construcción de sinogramas. B) Iteraciones sobre la imagen reconstruida. C) Iteraciones sobre el sinograma y sobre la imagen reconstruida (Willemink *et al.*, [19]).

Los métodos de reconstrucción iterativa son más costosos computacionalmente que los algoritmos que trabajan sobre los sinogramas directamente. Además de hacer uso de los datos de sinogramas, utilizan modelos de los

escáneres, cuyas características físicas no están disponibles por el público, por lo que estos algoritmos son implementados de distintas maneras según el fabricante y son propietarios (ej.: *IRIS, Iterative Reconstruction in Image Space*, por Siemens Medical Solutions; *AIDR 3D, Adaptive Iterative Dose Reduction 3D*, por Toshiba Medical Systems; *iDose*, por Philips Healthcare; etc.) [19, 20].

3.3. Procesamiento sobre imágenes reconstruidas

Con la premisa de la autosemejanza y características no locales de las imágenes, un algoritmo estudiado es el de *Non Local Means* (NLM). Con este algoritmo, el valor del píxel procesado se calcula como el promedio ponderado de píxeles centrados en regiones que son similares a la región donde se encuentra el píxel en cuestión [21]. Resultados en imágenes TC han sido obtenidos en [22].

Siguiendo con el mismo postulado de redundancia de información de la imagen, los algoritmos BM3D y BM4D extienden el concepto de NLM basándose en su cómputo en un campo transformado (siendo el primero utilizado en imágenes 2D y el último en imágenes 3D) [23, 24]. Ejemplos en aplicaciones en la reducción de ruido en TCs de baja dosis se encuentran en [25].

Algoritmos de aprendizaje de máquina también han sido estudiados en este tópico, por ejemplo K-SVD [26]. Este algoritmo es una extensión de *clustering k-means*, y busca representar a las imágenes como combinaciones lineales de un conjunto de elementos básicos. Una aplicación de este algoritmo se puede ver en [27].

Como se ha mencionado en la introducción, algoritmos de *Deep Learning* han recibido gran atención en el campo de las Imágenes Médicas, incluyendo la tarea de reducción de ruido. Las Redes que han obtenido mayor éxito en esta tarea son las arquitecturas basadas en RNCs. Chen *et al.* [28] propusieron el uso de RNCs en la reducción de ruido de TCs de baja dosis, y luego refinaron esta estrategia al implementar una estructura *Encoder-Decoder* con conexiones residuales [12]. También utilizando RNCs, Xiang *et al.* [29] estudiaron la estimación de tomografías PET de dosis normal a partir de tomografías PET de baja dosis. Por lo general, estos diseños no utilizan arquitecturas profundas, ya que la reducción de ruido es considerada una tarea de bajo nivel. Esto se contrasta con tareas como la clasificación de objetos, donde es necesario extraer atributos (*features*) de alto nivel en las imágenes.

El estado del arte de estos algoritmos está en las arquitecturas denominadas *Generative Adversarial Networks* (GANs) [30]. Estas arquitecturas cuen-

tan con dos redes: una Red generadora G entrenada para sintetizar imágenes verosímiles a partir de una entrada de ruido aleatorio; y una Red discriminadora D, entrenada para clasificar imágenes sintéticas de imágenes reales. Por lo general, las Redes G y D están basadas en arquitecturas RNC.

Resultan de particular interés las *conditional GANs* (cGANs), donde la síntesis de imágenes es condicionada utilizando información adicional. En este marco, se propone que la entrada de una cGAN no sea ruido aleatorio si no una imagen afectada por ruido. Arquitecturas de este tipo han sido implementadas para la reducción de ruido de TCs de baja dosis en [13, 14, 15] y para la reducción de ruido de tomografías PET en [31].

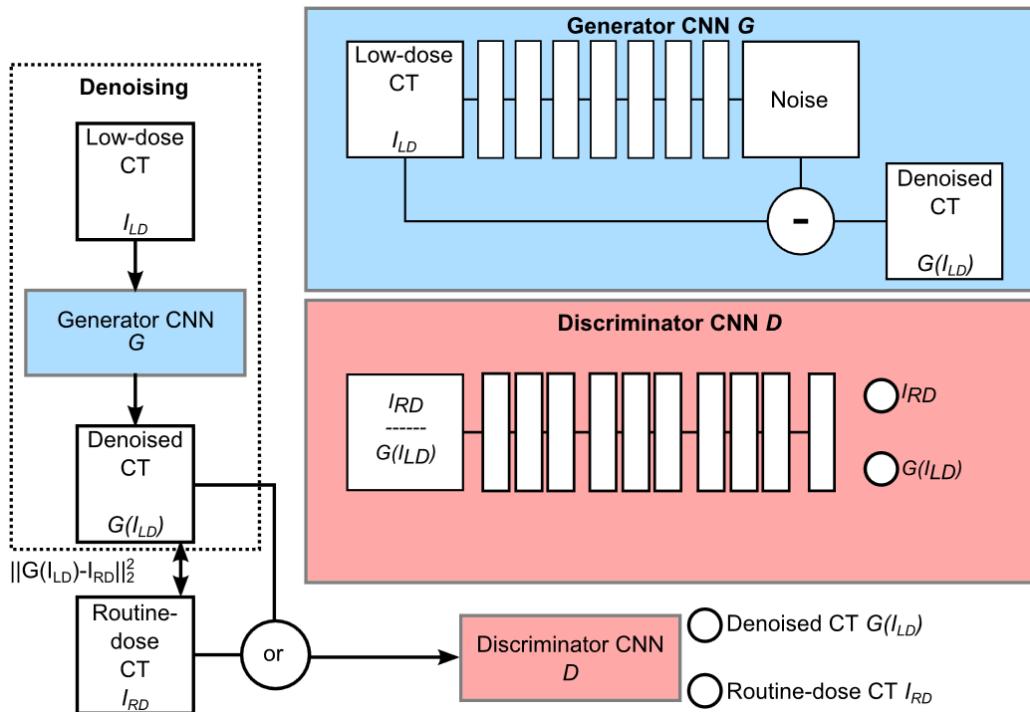


Figura 13: Arquitectura GAN utilizada por Wolterink *et al.*, [15].

4. Redes Neuronales de Convolución

4.1. Introducción

Las Redes Neuronales de Convolución son una clase de Redes Neuronales particularmente reconocidas en el campo de Procesamiento Digital de Imágenes, cuyo principio de operación se basa en el uso de filtros de convolución. Estas Redes tienen sus orígenes en el modelo *Neocognitron*, propuesto por Kunihiko Fukushima [32]. La arquitectura *Neocognitron* consiste en múltiples capas que automáticamente aprenden una jerarquía de *features* para reconocer patrones. Este modelo fue inspirado por el trabajo de Hubel y Wiesel [33] sobre la corteza visual primaria, que demostró que las neuronas en el cerebro se organizan en forma de capas. Estas capas de neuronas aprenden a reconocer patrones visuales al extraer *features* locales y posteriormente combinarlas para obtener representaciones de alto nivel. Basado en el *Neocognitron*, LeCun *et al.* propusieron el modelo *LeNet* [34], y lo entrenaron para reconocer dígitos escritos a mano.

Las RNCs son ampliamente utilizadas bajo el paradigma de aprendizaje supervisado, aunque también han sido estudiadas bajo aprendizaje no supervisado. Al utilizar aprendizaje supervisado, se conocen las salidas deseadas de la RN correspondientes a los datos de entrada, y el modelo debe aprender el mapeo entre ambas. En un esquema de aprendizaje no supervisado, las salidas deseadas no son conocidas, y el modelo debe aprender la distribución subjacente de las muestras de entrada. Las RNCs son empleadas en tareas de clasificación, reconocimiento de objetos, segmentación, reducción de ruido y en modelos generativos.

4.2. Arquitectura

Una RNC se encuentra ordenada en capas, que implementan distintas operaciones básicas. En esta sección se explican las capas que utilizan las RNCs convencionales.

4.2.1. Capas de convolución

Estas son las principales capas en una RNC. Las salidas de estas capas son el resultado de realizar la convolución entre los datos en su entrada y distintos filtros o *kernels* de convolución, que son los parámetros que la Red debe aprender a través de entrenamiento.

Para cada capa de convolución, es posible seleccionar la cantidad de filtros que posee así como los parámetros de la operación de convolución. Estos

parámetros son:

- el **tamaño** del filtro;
- el **paso** (*stride*) de desplazamiento del filtro sobre la imagen de entrada, que generalmente es de 1 píxel en cada dimensión (cuando se trata con imágenes);
- si a la imagen de entrada se le agrega un borde con píxeles de valor cero (normalmente denominado ***zero-padding*** en la literatura de Procesamiento Digital de Señales y Procesamiento Digital de Imágenes).

Estos tres parámetros, así como la cantidad de filtros empleados en la capa son denominados los *hiperparámetros* de la capa y son opciones de diseño, dejándose el término de *parámetros* reservado para los pesos de los filtros.

Dependiendo en el uso de *zero-padding*, las convoluciones se suelen clasificar en tres categorías.

- **Convolución válida.** Se usa este término cuando no se emplea *padding* (es decir, el filtro sólo opera sobre píxeles “válidos”);
- **Convolución de igual tamaño** (*same convolution*). Es el caso donde se usa *padding* tal que el tamaño de la imagen de salida es igual al tamaño de la imagen de entrada;
- **Convolución completa.** Es el caso donde se aplica la mayor cantidad posible de *zero-padding*, tal que en todos los pasos de la convolución se emplea al menos un píxel válido.

Convolución transpuesta o deconvolución

Supóngase la convolución de una imagen x de 4×4 con un filtro W de 3×3 , produciendo una salida y de 2×2 . Es posible representar esta operación como una multiplicación matricial, si la entrada y salida es expresada en forma vectorizada, y el filtro es expresado como una matriz de Toeplitz desenrollada \mathbf{C} :

$$\begin{bmatrix} w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} & 0 \\ 0 & 0 & 0 & 0 & w_{0,0} & w_{0,1} & w_{0,2} & 0 & w_{1,0} & w_{1,1} & w_{1,2} & 0 & w_{2,0} & w_{2,1} & w_{2,2} \end{bmatrix}$$

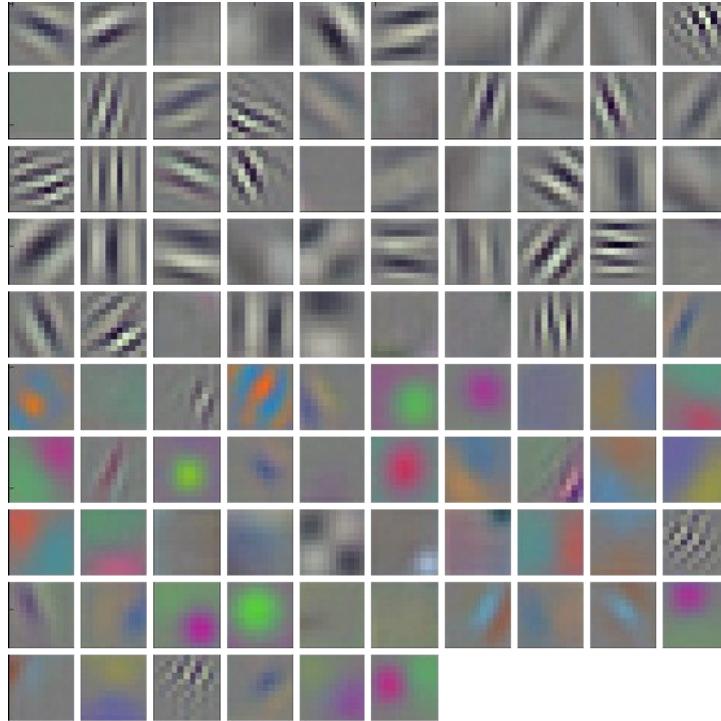


Figura 14: Filtros de convolución de la primera capa de la Red *AlexNet* entrada sobre el conjunto de datos (*dataset*) *ImageNet* (Krizhevsky *et al.*, [35]).

$$\mathbf{y} = \mathbf{Cx}, \quad (10)$$

donde $w_{i,j}$ son los elementos de W , con i, j los índices de fila y columna respectivamente.

La salida de una capa de convolución tiene menor resolución que su entrada (al menos que se use un *padding* suficientemente grande). En aplicaciones donde sea necesario aumentar la resolución de una entrada es posible emplear capas de convolución transpuesta. En el ejemplo anterior, el producto de \mathbf{C} con el vector \mathbf{x} resultó en un vector \mathbf{y} que representa una imagen de 2×2 . Es posible multiplicar la forma vectorizada de una imagen de 2×2 , \mathbf{x}' para producir una salida de 4×4 , \mathbf{y}' :

$$\mathbf{y}' = \mathbf{C}^T \mathbf{x}'. \quad (11)$$

Así como en las capas de convolución, para las capas de convolución transpuesta también es posible seleccionar el número de filtros, el tamaño de filtro, el paso, y la cantidad de *zero-padding*.

4.2.2. Capas de *pooling*

La operación de la capa de pooling reemplaza la salida de la Red en una determinada posición por un resumen estadístico en las posiciones cercanas. Por ejemplo, la operación de *max pool* devuelve el valor máximo en una vecindad rectangular. Otros tipos de funciones de *pooling* incluyen el promedio sobre un vecindario rectangular, el promedio ponderado sobre un vecindario rectangular basado en la distancia al píxel central, o la norma L2 de un vecindario rectangular.

Cualquiera sea la función implementada en la capa de *pooling*, esta operación hace que la salida de la capa sea invariante a pequeñas traslaciones en la entrada. En aplicaciones como clasificación de imágenes, resulta más importante saber la presencia de un *feature* en la imagen que la posición del mismo.

Además de realizar *pooling* sobre regiones espaciales, es posible aplicarlo sobre un conjunto de mapas de atributos (*feature maps*), de manera de detectar cuál de estos es el que se activa más fuertemente. De ambas maneras, las capas de *pooling* disminuyen la cantidad de entradas de la próxima capa de la Red, por lo que se mejora la eficiencia computacional del modelo.

Unpooling

La operación de *unpooling* fue introducida en [37]. Es utilizada para reconstruir el tamaño original de activaciones a las que se les ha aplicado *max pooling*. En arquitecturas que utilizan capas de convolución más *max pooling*, y capas de deconvolución más *unpooling*, las posiciones de las activaciones obtenidas de las capas de *max pool* son registradas. Luego, en las correspondientes capas de *unpool*, estas posiciones son utilizadas para colocar las activaciones producidas en sus posiciones originales. La figura 15 muestra la arquitectura DeconvNet para segmentación semántica, que emplea este tipo

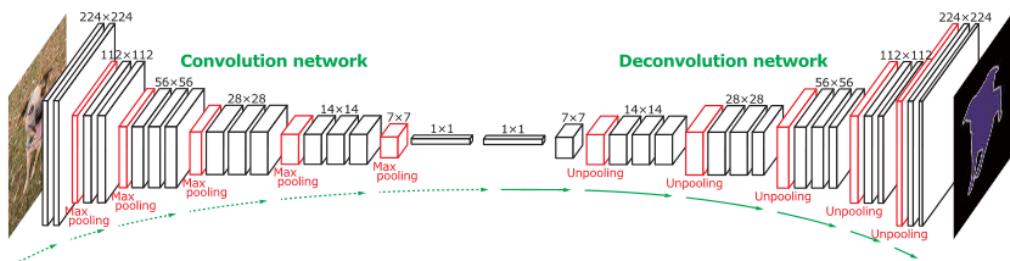


Figura 15: Arquitectura de la Red DeconvNet (Noh *et al.*, [36]).

de capas [36].

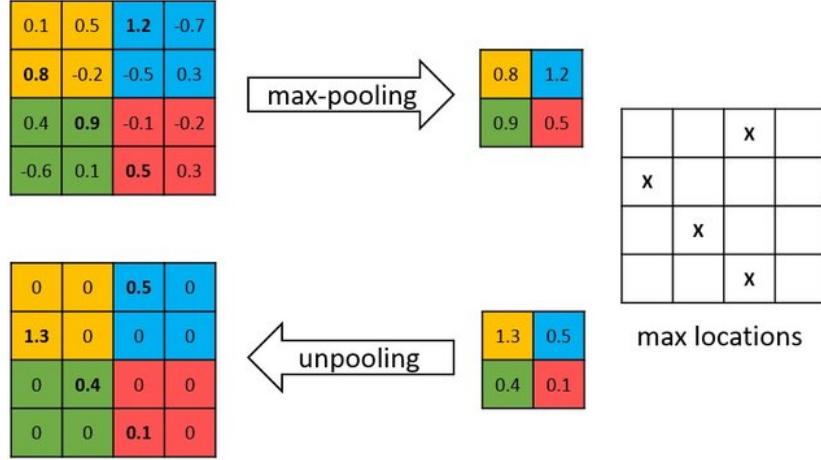


Figura 16: Ejemplo de operación de *max pool* de tamaño 2×2 y *stride* 1×1 , y su respectiva operación de *unpool* (David *et al.*, [38]).

4.2.3. Capas de activación

La capa de activación es una función no lineal aplicada a los valores de salida de la capa que la precede. La existencia de capas de activación es lo que permite a una Red Neuronal aprender mapeos no lineales. En ausencia de capas de activación, la cascada de capas de convolución o capas lineales es equivalente a una Red que aplica un mapeo lineal. Las funciones de activación más utilizadas se muestran en la figura 17.

4.2.4. Capas lineales

Las capas lineales, más conocidas en la literatura como *fully connected layers*, suelen ser las últimas capas en Redes utilizadas para tareas de clasificación o regresión. Una capa lineal está compuesta por unidades que computan una combinación lineal de todos los valores de la salida de la capa que la precede. La operación que realizan puede ser representada como una multiplicación matricial:

$$\mathbf{y} = \mathbf{W}\mathbf{x}, \quad (12)$$

donde \mathbf{x} es el vector de entrada, \mathbf{y} es el vector de salida, y \mathbf{W} la matriz conteniendo los pesos de la capa.

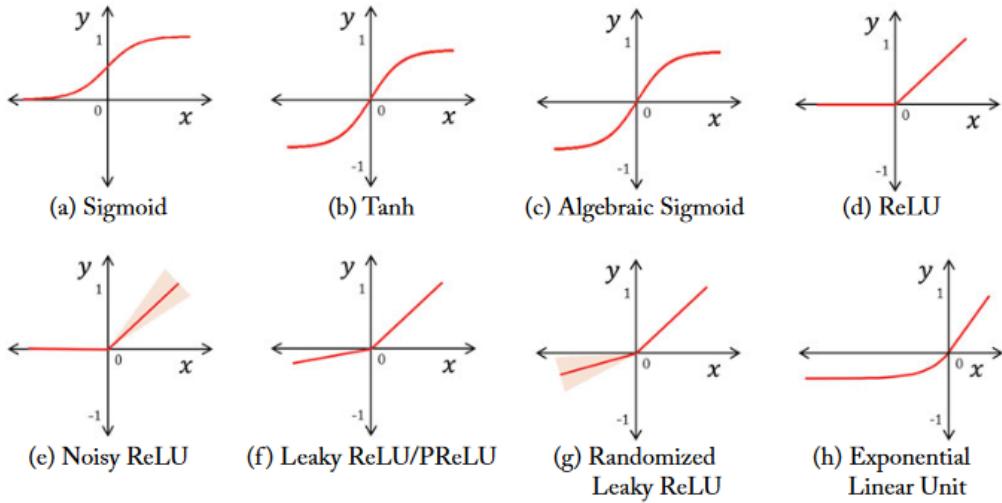


Figura 17: Funciones de activación más utilizadas (Khan *et al.*, [39]).

4.3. Entrenamiento de una Red Neuronal

La Red Neuronal computa una función $f(\mathbf{x}, \theta)$, donde \mathbf{x} es un vector de entrada y θ es el conjunto de todos los parámetros de la Red. Es posible cuantificar la diferencia entre $f(\mathbf{x}_i, \theta)$ la salida de la Red ante un vector de entrada \mathbf{x}_i y el resultado deseado \mathbf{y}_i utilizando alguna función pérdida $L(f(\mathbf{x}_i, \theta), \mathbf{y}_i)$. Teniéndose un *dataset* de m pares $(\mathbf{x}_i, \mathbf{y}_i)$, se define una función costo $J(\theta)$ como el promedio de L calculada sobre los datos disponibles:

$$J(\theta) = \frac{1}{m} \sum_i L(f(\mathbf{x}_i, \theta), \mathbf{y}_i). \quad (13)$$

El objetivo entonces es encontrar el conjunto de parámetros θ^*

$$\theta^* = \operatorname{argmin}_{\theta} J(\theta). \quad (14)$$

Debido a la complejidad de este problema de optimización no lineal, la búsqueda de parámetros se realiza de manera iterativa. Los algoritmos optimizadores utilizados en esta tarea actualizan los parámetros de la Red incrementándolos en la dirección del gradiente de J

$$\nabla J(\theta) = \frac{1}{m} \nabla_{\theta} \sum_i L(f(\mathbf{x}_i, \theta), \mathbf{y}_i). \quad (15)$$

Calcular el gradiente de esta manera resulta computacionalmente caro ya que requiere evaluar el modelo para cada muestra del *dataset*. En la práctica

se suele calcular el $\nabla J(\theta)$ de a mini lotes (*minibatches*), es decir, empleando un número de muestras menor a m . Los algoritmos optimizadores más mencionados en la literatura son *Gradient Descent*, *Stochastic Gradient Descent* y *Adaptive Moment Estimation* (más conocido como *Adam*) [40, 39, 41].

5. RNC para reducción de ruido en TCs

5.1. Estrategia propuesta

El objetivo de este trabajo consiste entonces en obtener imágenes TCC de alta calidad manteniendo un bajo nivel de corriente de tubo, a partir de un enfoque basado en *Deep Learning*. Basándose en los trabajos previos realizados sobre la temática, se propone el uso de una Red Neuronal con arquitectura *Encoder-Decoder*, empleando capas de convolución para la codificación de los datos de entrada, y capas de convolución transpuesta para su decodificación.

El flujo de trabajo de la estrategia propuesta comienza con el procesamiento de reconstrucciones de retroproyección filtrada de un escáner con baja corriente de tubo. Se plantea el siguiente problema:

Sea $\mathbf{X} \in R^{m \times n}$ un conjunto de TCCBRs, e $\mathbf{Y} \in R^{m \times n}$ su correspondiente conjunto de m imágenes CCPAR. Dichos conjuntos se relacionan de la siguiente manera:

$$\mathbf{X} = \sigma(\mathbf{Y}) \quad (16)$$

donde $\sigma : R^{m \times n} \rightarrow R^{m \times n}$ denota el proceso de degradación de calidad en la imagen. Se desea entonces entrenar la RNC elaborada para que implemente una aproximación óptima de σ^{-1} .

El camino codificador de la Red propuesta consta de sucesivas capas de convolución seguidas por funciones no lineales. La operación aplicada por la capa C_e^i de la Red codificadora puede ser notada como:

$$C_e^i(\mathbf{x}_i) = \text{ReLU}(\mathbf{W}_i * \mathbf{x}_i + \mathbf{b}_i) \quad i = 0, 1, \dots, N, \quad (17)$$

donde N es el número de capas, \mathbf{W}_i y \mathbf{b}_i los pesos y bias de la capa, respectivamente, $*$ el operador de convolución, ReLU la función no lineal *Rectified Linear Unit*, $\text{ReLU}(x) = \max(0, x)$, utilizada como función de activación, y \mathbf{x}_i el vector de datos de entrada de la capa. El vector \mathbf{x}_0 representa al vector de datos de entrada de la Red, siendo éste una imagen TCCBR, y los vectores \mathbf{x}_i ($1 \leq i \leq N$) son los *feature maps* producidos por la capa C_e^{i-1} , que actúan como entrada de la capa C_e^i . Los *feature maps* producidos por la última capa del camino codificador de la Red corresponden a una representación interna de los datos de entrada, denominada *espacio latente*.

El siguiente paso en el flujo de trabajo es recuperar los detalles estructurales que se han perdido en el proceso de degradación de calidad a partir del procesamiento de los *features* que ha extraído el camino codificador. Para ello se utiliza el camino decodificador de la Red, constituido por sucesivas

capas de convolución transpuesta seguidas por unidades ReLU. La operación aplicada por la capa D_d^i de la Red decodificadora puede ser notada como:

$$D_d^i(\mathbf{x}'_i) = \text{ReLU}(\mathbf{W}'_i *^T \mathbf{x}'_i + \mathbf{b}'_i) \quad i = 0, 1, \dots, N, \quad (18)$$

donde N es el número de capas, \mathbf{W}'_i y \mathbf{b}'_i los pesos y bias de la capa, respectivamente, $*^T$ el operador de convolución transpuesta e \mathbf{x}'_i el vector de datos de entrada de la capa. El vector $\mathbf{x}'_0 = \mathbf{x}_N$ representa al vector de datos de perteneciente al latent space, producido por la Red codificadora, y los vectores \mathbf{x}'_i ($1 \leq i \leq N$) son los *feature maps* producidos por la capa D_d^{i-1} , que actúan como entrada de la capa D_d^i . Finalmente, la imagen reconstruida corresponde al vector \mathbf{x}'_N .

5.2. Arquitectura

La Red en cuestión está formada por 8 capas: 4 capas de convolución, seguidas por 4 capas de convolución transpuesta. Todas las capas de la Red poseen 32 canales de entrada a excepción de la primer capa, que posee 1 canal de entrada; y 32 canales de salida, a excepción de la última capa, que posee 1 canal de salida.

Debido a que los volúmenes del *dataset* empleado poseen véxeles de dimensiones anisotrópicas, el tamaño de los *kernels* de las capas de convolución y convolución transpuesta se ha elegido de $5 \times 5 \times 3$ véxeles. El tamaño del *stride* se ha elegido de $1 \times 1 \times 1$ véxeles, y las convoluciones se realizan sin *padding*. Como función de activación se ha elegido *Rectified Linear Unit* (ReLU), presente a la salida de todas las convoluciones y deconvoluciones, exceptuando la última capa.

Con la configuración elegida, se ve que los volúmenes de salida de una capa de convolución poseen $4 \times 4 \times 2$ véxeles menos que los volúmenes de entrada de la misma capa. Los *feature maps* de la Red obtendrán un tamaño mínimo a la salida de la cuarta capa de convolución, siendo esta el cuello de botella de la Red. Posteriormente, los volúmenes de salida de cada capa de convolución transpuesta poseen $4 \times 4 \times 2$ véxeles más que los volúmenes de entrada de la misma capa. Se tiene entonces que el volumen de salida de la Red posee el mismo tamaño que el volumen de entrada, por lo que es posible comparar ambos directamente con alguna métrica de error.

Esta arquitectura no utiliza capas *fully connected*. Resulta más apropiado el uso capas de convolución, que permiten extraer *features* locales que se pueden repetir en todo el volúmen. Tampoco se utilizan capas de *pooling* ya que el submuestreo que realizan produce pérdida de información espacial.

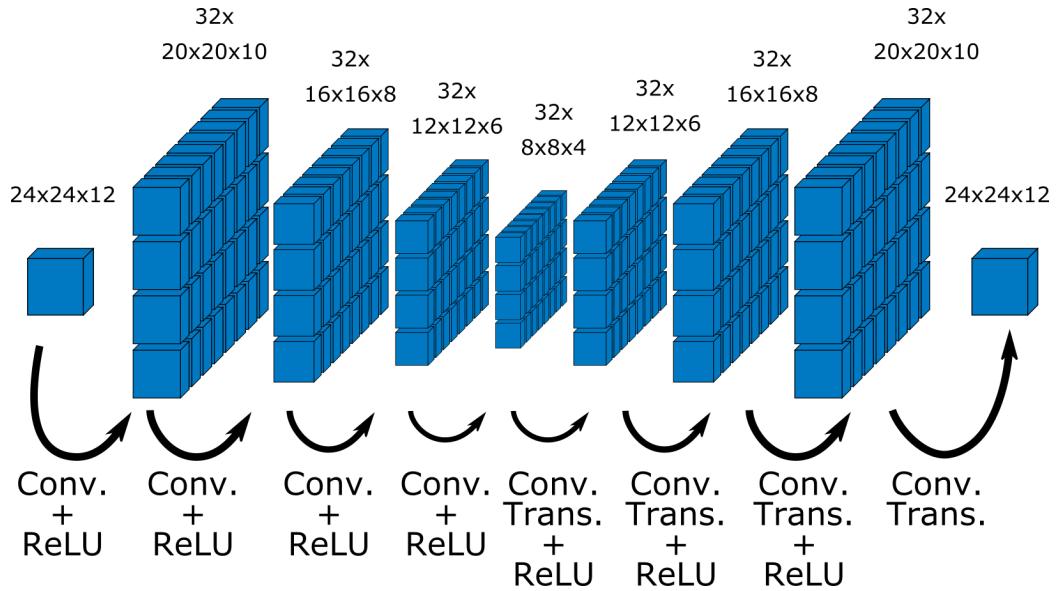


Figura 18: Arquitectura de la Red Neuronal.

5.3. Conjunto de datos

Los datos utilizados para el entrenamiento y testeo de la Red Neuronal provienen de cinco vértebras humanas que fueron embebidas en resina epoxy. A modo de obtener un conjunto de imágenes de alta resolución, y un conjunto con resolución clínica, dos dispositivos fueron utilizados.

El conjunto de escaneos de alta resolución fue obtenido usando un escáner de TCCPAR Scanco XtremeCT I con resolución isotrópica de $82\mu m$, $59,4kVp$ y $900\mu As$. Estos escaneos fueron remuestreados a una resolución de $172\mu m \times 172\mu m \times 340\mu m$.

Las imágenes de calidad clínica se adquirieron con un escáner Philips iCT 256. Para simular las condiciones de un *scan in vivo*, las vértebras fueron colocadas en un fantoma de abdomen (Model 235, Computerized Imaging Reference Systems Inc.). Con el objetivo de tener imágenes tomográficas con distinto nivel de ruido, se emplearon tres distintas corrientes de tubo: $100mA$ s, $250mA$ s y $360mA$ s, realizándose tres distintos escaneos para cada valor. De este modo, se poseen 45 TCCs de calidad clínica en total, correspondiendo 9 a cada vértebra. El resto de la configuración corresponde a $120kVp$, *pitch* 1, espesor de corte $0,67mm$, kernel estándar de hueso (YB), campo visual $88mm$, correspondiente a una resolución de $172\mu m \times 172\mu m \times 340\mu m$ y tamaño de voxel $0,01mm^3$.



Figura 19: *Scan* de una vértebra. Izquierda: escaneo de una vértebra situada dentro del fantoma de abdomen. Derecha: escaneo de una vértebra sin fantoma de abdomen (Thomsen, [9]).



Figura 20: Escáneres utilizados en la adquisición de datos. Izquierdo: Philips iCT256. Derecho: Scanco Xtreme CT I.

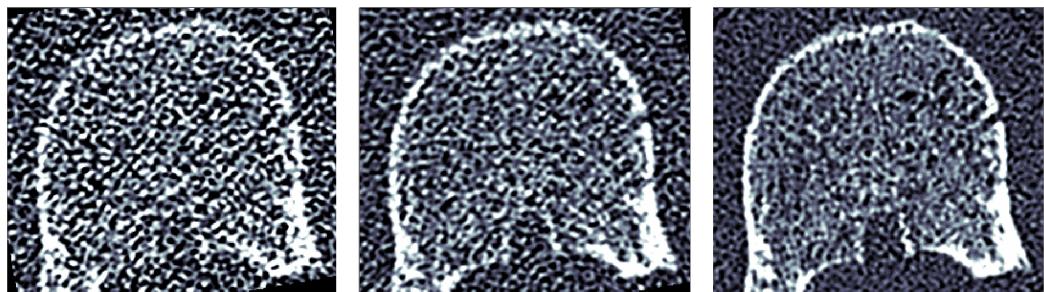


Figura 21: Cortes transversales de TCCs de diferentes dosis de una vértebra del set de datos. De izquierda a derecha: 100 mAs , 250 mAs , 360 mAs

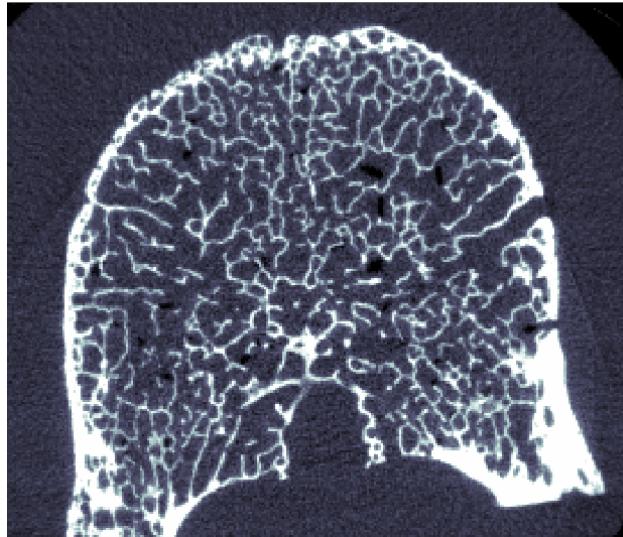


Figura 22: Corte transversal de la TCCPAR correspondiente a la vértebra de la figura 21.

5.4. Entrenamiento

Una vez que la Red es configurada, se debe estimar el conjunto de parámetros $\theta = \{\mathbf{W}, \mathbf{b}, \mathbf{W}', \mathbf{b}'\}$ de las capas de convolución y convolución transpuesta a través de entrenamiento progresivo para aproximar la función de mapeo σ . Se ha elegido como métrica de pérdida a la función Error Absoluto Medio (MAE, por sus siglas en inglés), o pérdida $L1$, con la intención de disminuir las pérdidas de información de alta frecuencia espacial. El algoritmo optimizador utilizado para minimizar la función pérdida es Adam.

Dado el reducido número de TCCs del *dataset*, para aumentar la cantidad de datos disponibles para entrenamiento se ha decidido extraer pequeños recortes de los volúmenes originales, únicamente de la espongiosa del cuerpo vertebral. Utilizando recortes de $24 \times 24 \times 12$ véxoles (equivalente a una caja isotrópica de $4,128mm$ de lado), y muestreando los volúmenes cada $20 \times 20 \times 10$ véxoles, se han extraído un total de 18423 recortes de las 45 TCCBR. Posteriormente, cada recorte fue registrado con su correspondiente recorte TCCPAR.

Se entrenaron 3 modelos, uno para cada valor de corriente de tubo utilizado en los scans. Habiendo entonces 6141 pares de recortes TCCBR-TCCPAR por corriente de tubo, para cada Red el 85 % se utilizó como conjunto de entrenamiento, y el 15 % se utilizó como conjunto de testeo.

Previo al entrenamiento final, se realizó la validación de los hiperparámetros de tamaño de lote (*batch size*) y tasa de aprendizaje (*learning rate*).

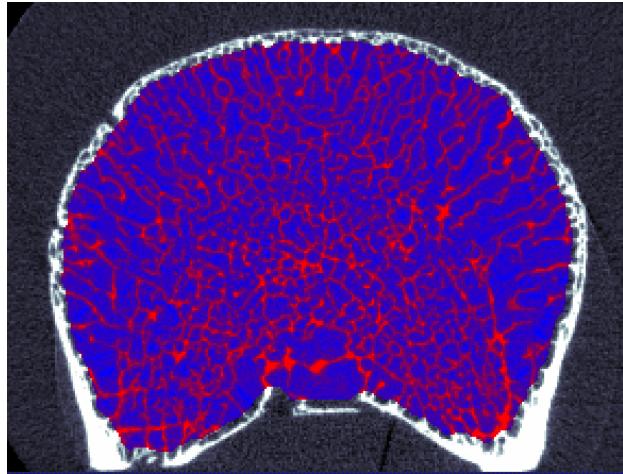


Figura 23: Corte transversal de una vértebra humana perteneciente al *dataset*. La región coloreada corresponde a la espongiosa.

Para esto, la Red correspondiente a corriente de tubo de $100mA_s$ fue entrenada múltiples veces, con distintas combinaciones de valores de *batch size* y *learning rate*, a modo de búsqueda exhaustiva. Estos entrenamientos fueron realizados con un 10 % de los recortes del conjunto de entrenamiento, y un conjunto de igual tamaño fue utilizado para la etapa de validación. La figura 25 muestra las distintas curvas de aprendizaje obtenidas en esta etapa de diseño. Finalmente, con los resultados obtenidos en la etapa de validación, se ha decidido por optar los valores de $learning\ rate = 10^{-5}$ y $batch\ size = 32$. Las curvas de aprendizaje resultantes se pueden ver en la figura 24.

Todos los experimentos fueron realizados con el servicio de cómputo en la nube gratuito Google Colab, que provee una GPU NVidia Tesla K80 de 12GB. En cuanto a la programación, el lenguaje Python fue utilizado. La implementación de la Red y su entrenamiento fue realizada con la librería PyTorch, desarrollada por el Laboratorio de Inteligencia Artificial de Facebook.

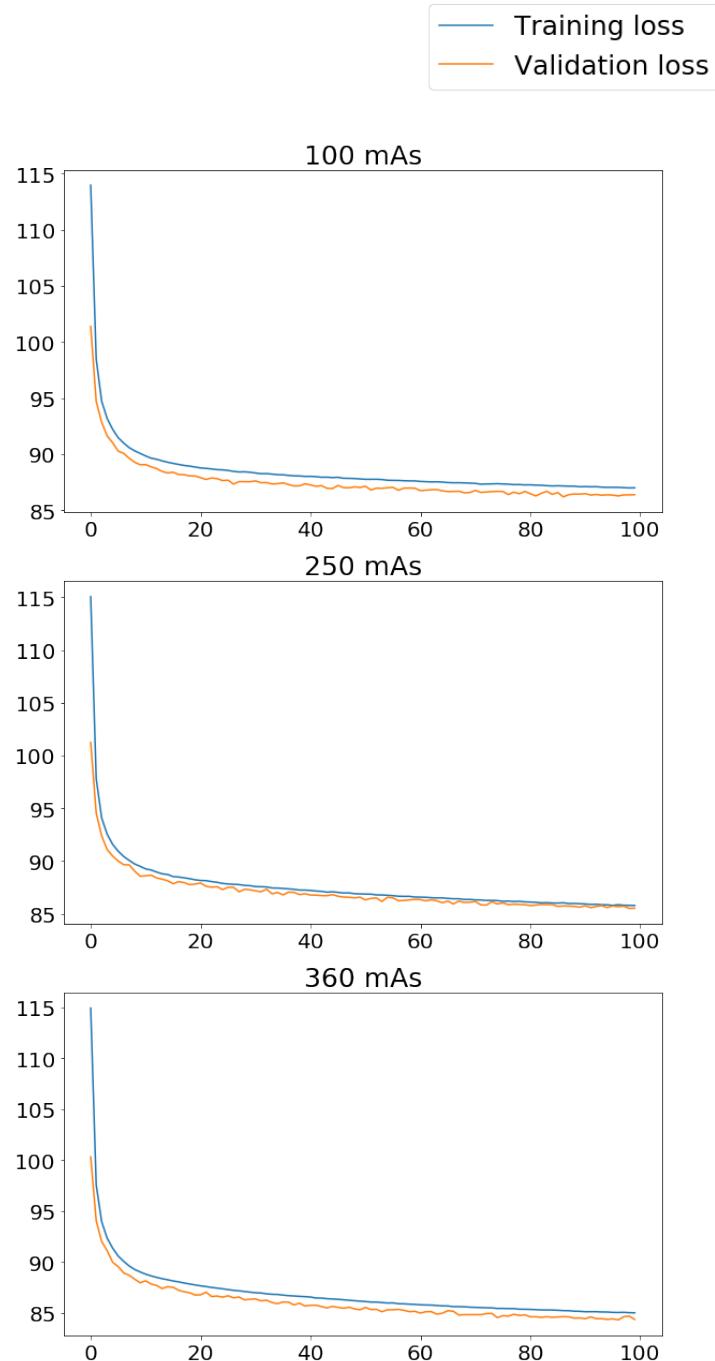


Figura 24: Curvas de aprendizaje de los tres modelos.

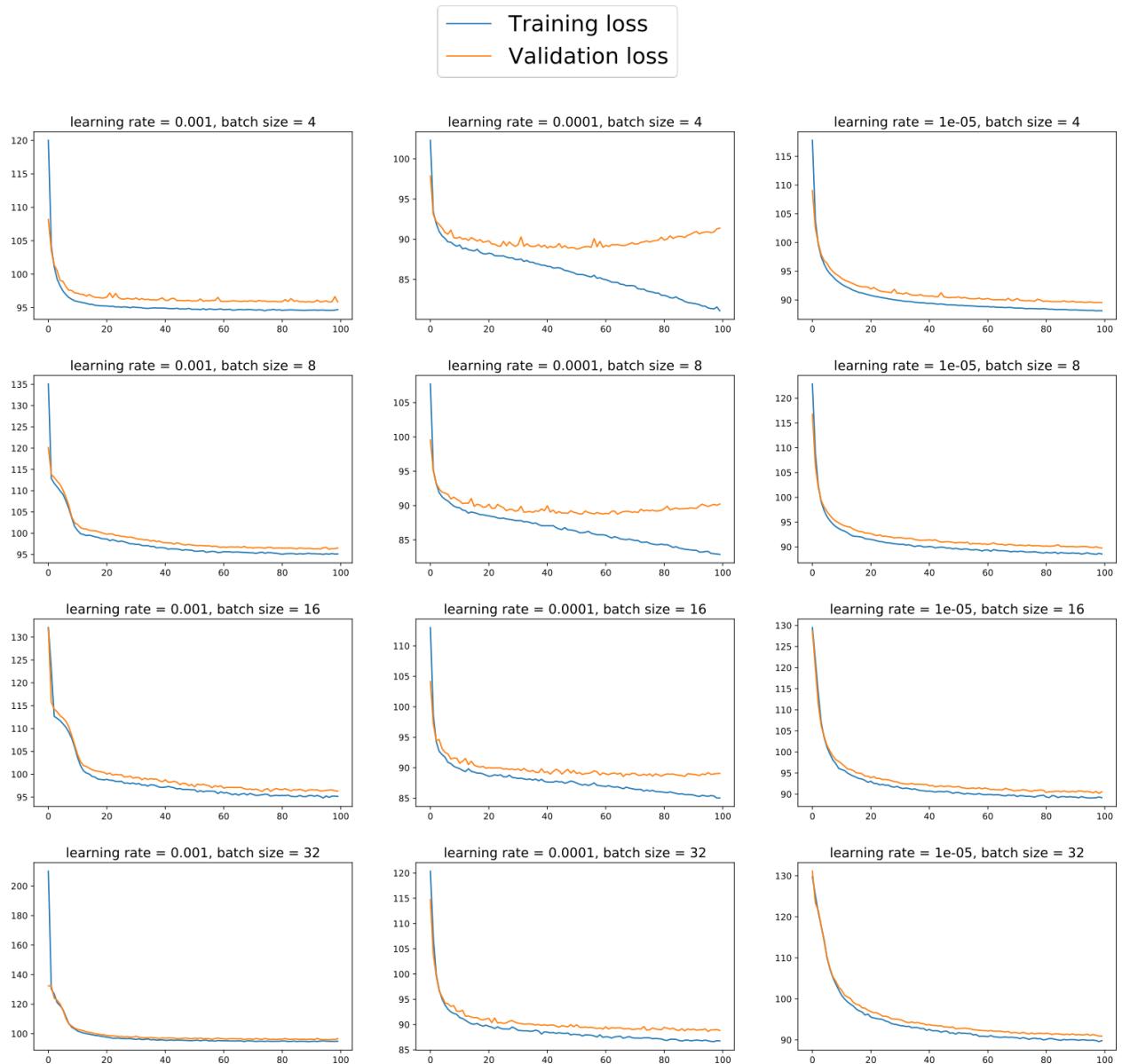


Figura 25: Curvas de aprendizaje de la etapa de validación de hiperparámetros, para distintos valores de *learning rate* y *batch size*.

6. Resultados

Las figuras 26 y 27 muestran cortes transversales de recortes de vértebras y de vértebras completas, respectivamente, producidos por las tres redes. Los resultados también fueron comparados con el filtrado BM4D de las TCCBR. La tabla 1 permite ver los valores de Error Absoluto Medio y Raíz del Error Cuadrático Medio (*Root Mean Squared Error*), Proporción Máxima de Señal a Ruido (*Peak Signal-to-Noise Ratio*) y Similitud Estructural (*Structural SIMilarity*) con respecto a los datos TCCPAR. Para los cálculos de PSNR y SSIM, se utilizó un rango dinámico de 6 desviaciones estándares de los valores de densidad ósea del *dataset* TCCPAR.

Cuadro 1: MAE y RMSE de las técnicas de reducción de ruido.

| C. de Tubo | Técnica | MAE | RMSE | PSNR[dB] | SSIM |
|------------|--------------|---------|---------|----------|-------|
| 100mAs | - | 271,333 | 339,988 | 19,831 | 0.054 |
| | BM4D | 257,496 | 322,549 | 20,428 | 0.060 |
| | Red Neuronal | 86,025 | 125,151 | 27,710 | 0.083 |
| 250mAs | - | 183,050 | 230,630 | 23,639 | 0.123 |
| | BM4D | 175,018 | 220,464 | 24,000 | 0.127 |
| | Red Neuronal | 85,078 | 124,013 | 27,745 | 0.108 |
| 360mAs | - | 158,554 | 199,889 | 24,735 | 0.156 |
| | BM4D | 152,749 | 192,679 | 25,004 | 0.162 |
| | Red Neuronal | 83,989 | 121,927 | 27,831 | 0.129 |

De la tabla 1 se observa que estos modelos disminuyeron los valores de RMSE y MAE con respecto a las TCCBR. Para los modelos de 100mAs, 250mAs y 360mAs, el RMSE disminuyó 63,19 %, 46,23 % y 39,01 %, y el MAE 68,3 %, 53,53 % y 47,03 %, respectivamente. Por otro lado, el PSNR aumentó para todos los modelos, con un incremento de 7,879dB, 4,106dB y 3,096dB para las Redes entrenadas con TCCBR de 100mAs, 250mAs y 360mAs, respectivamente. En cuanto al SSIM, para el modelo de 100mAs se registró un incremento de 0,029, y los modelos de 250mAs y 360mAs registraron una disminución de 0,015 y 0,027, respectivamente.

El filtro BM4D también produjo una disminución en los valores de MAE y RMSE, aunque de las figuras 26 y 27 el efecto visual producido por el mismo sea imperceptible. Para las corrientes de 100mAs, 250mAs y 360mAs, la disminución en RMSE fue de 5,13 %, 4,41 % y 3,67 %, respectivamente. De mismo modo, la disminución de MAE fue de 5,1 %, 4,39 % y 3,61 %. Para los modelos de 100mAs, 250mAs y 360mAs, los valores de PSNR aumentaron en 0,597dB, 0,361dB y 0,269dB, respectivamente, y los valores de SSIM

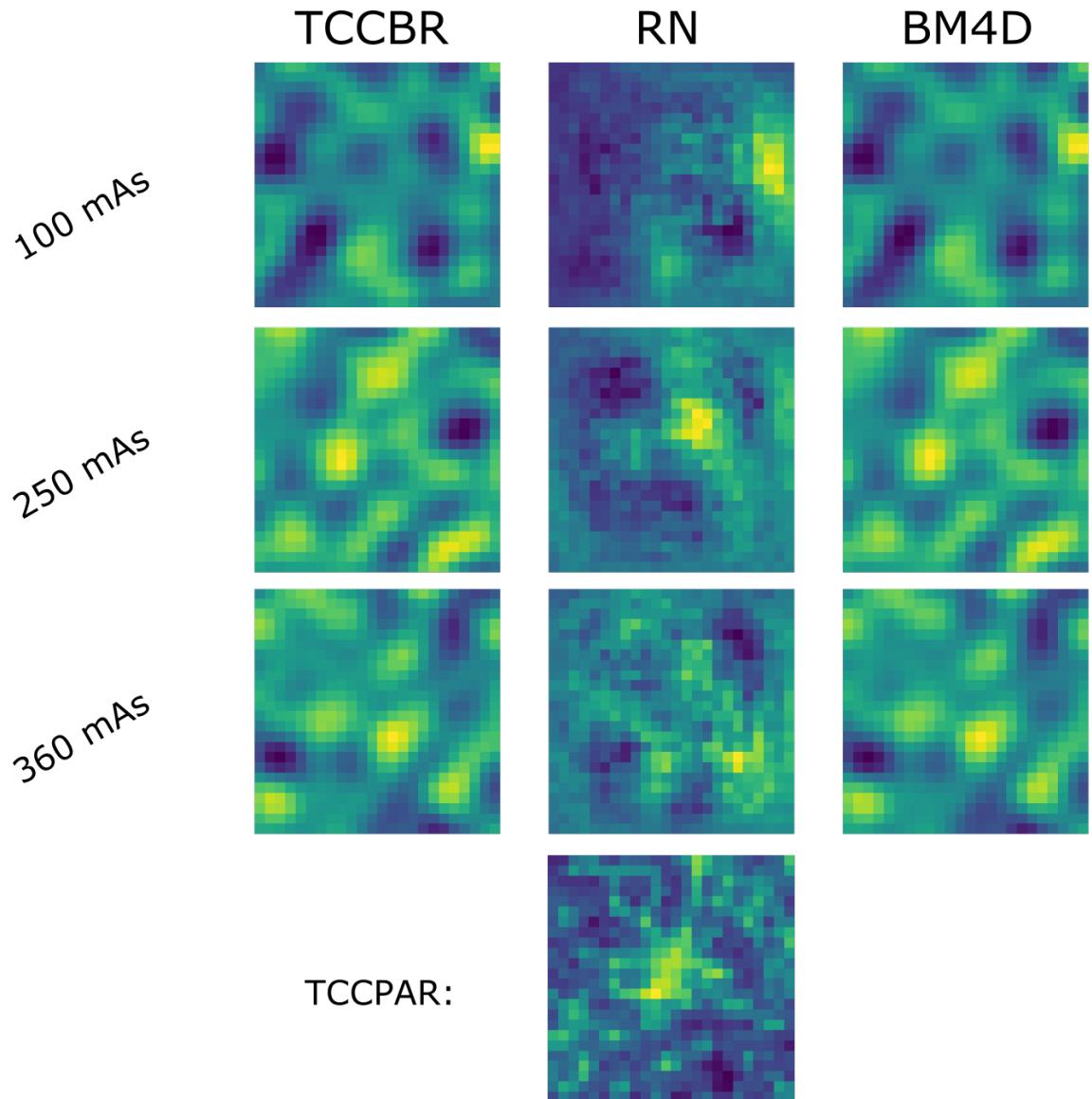


Figura 26: Comparación de cortes transversales de recortes.

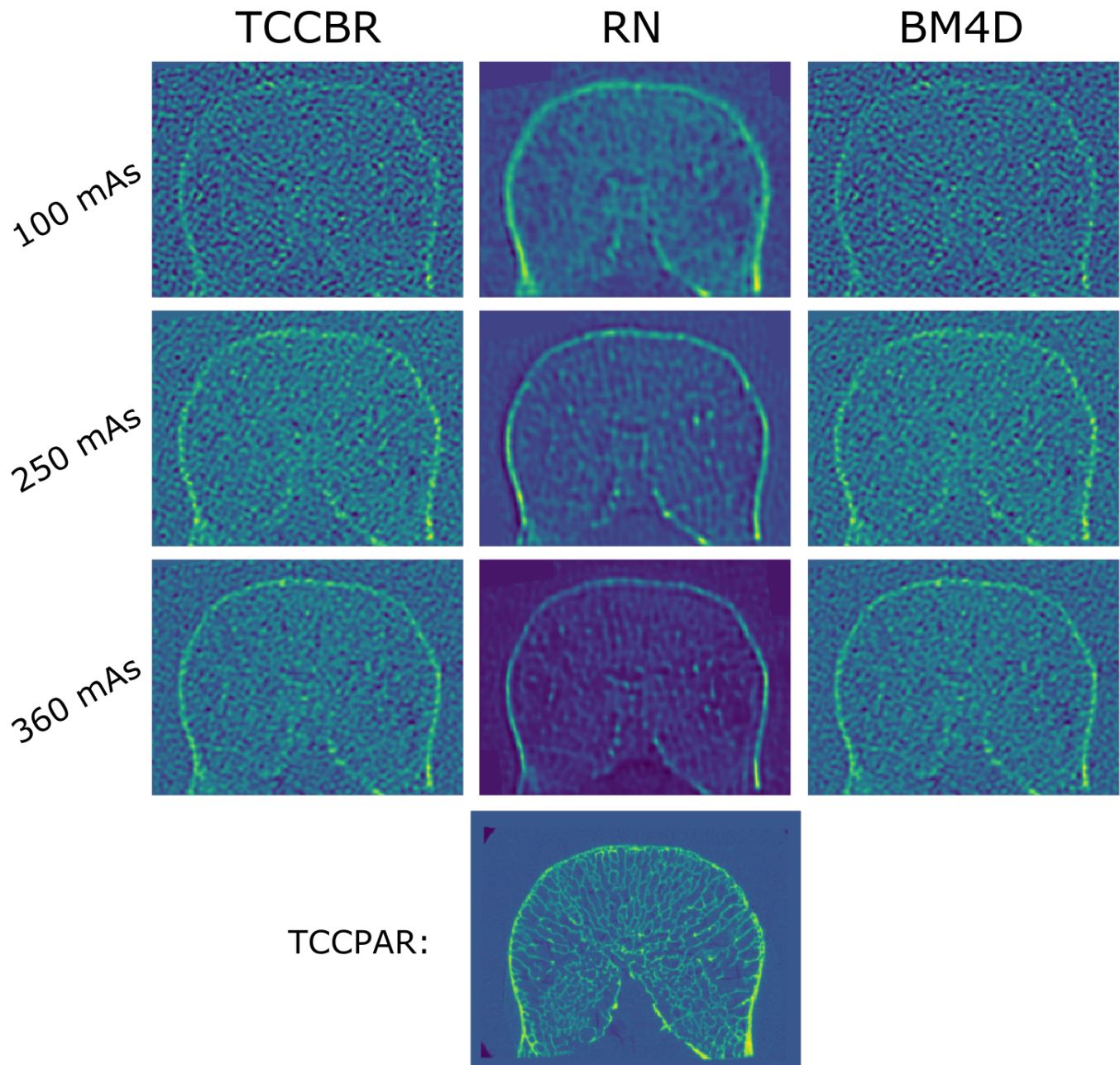


Figura 27: Comparación de cortes transversales de vértebras.

disminuyeron en 0,006, 0,004, y 0,006, respectivamente. Se observa que los resultados de las Redes Neuronales entrenadas superan a aquellos producidos por BM4D en todas las comparaciones de MAE, RMSE, PSNR y en el caso de SSIM sobre las TCCBR de $100mA_s$, siendo BM4D el método con mejores resultados para SSIM sobre TCCBR de $250mA_s$ y $360mA_s$.

7. Discusión

De las figuras 26 y 27 se observa que las imágenes producidas por las Redes Neuronales y por BM4D no poseen el mismo nivel de detalle en la estructura del hueso que las imágenes TCCPAR. Las Tomografías Computadas de vértebras de estudios clínicos poseen un nivel de ruido considerablemente mayor que estudios clínicos de otras partes del cuerpo, como por ejemplo de los huesos de la muñeca donde es posible utilizar un escáner de TCCPAR. Es por esto que los métodos analizados no logran recuperar la totalidad de la información de estructura ósea.

Los resultados producidos por la Red Neuronal también estuvieron influidos por las decisiones tomadas en el proceso de diseño. Por ejemplo, de la figura 25 se puede ver que una incorrecta elección de hiperparámetros de tasa de *learning rate* y *batch size* pueden causar un sobreajuste en los datos de entrenamiento. Esto ocasionaría que la Red Neuronal no aprenda a generalizar correctamente sobre nuevos datos, desempeñándose de un modo inferior. La búsqueda de hiperparámetros se puede hacer arbitrariamente compleja. En este trabajo no se trató con el análisis de número de capas de convolución, número de *feature maps*, función de activación, etc.

El tipo de arquitectura de Red Neuronal empleada también tiene efecto sobre los resultados. Previamente a la elección de la arquitectura *Encoder-Decoder*, se probaron arquitecturas de tipo U-Net, y Redes con conexiones residuales entre capas. Se decidió finalmente por la arquitectura utilizada ya que de todos los modelos probados, este resulta el más sencillo y por lo tanto el más fácil de depurar.

La cantidad de datos disponibles fue otro factor a tener en cuenta. Debido a la cantidad limitada de TCs, y a que los algoritmos de *Deep Learning* requieren un gran número de datos, fue necesario utilizar la estrategia de extracción de recortes. La división en conjuntos de entrenamiento y testeo tampoco es un detalle menor. Los conjuntos utilizados poseen recortes de las cinco vértebras disponibles y fueron elegidos de manera aleatoria. Se podría haber elegido, por ejemplo, el uso de 4 vértebras para entrenamiento, y la vértebra restante utilizarla para testeo. Sin embargo, cada vértebra posee una estructura con cualidades diferentes, la inferencia sobre la vértebra de testeo resultaría en un mayor error comparado con el obtenido.

8. Conclusión

En este trabajo se diseñó una Red Neuronal con el objetivo de la reducción de ruido en imágenes de Tomografías Computadas de Baja Resolución. Basándose en trabajos previos realizados en la temática, se eligió una arquitectura *Encoder-Decoder* utilizando capas de convolución y convolución transpuesta.

Este modelo fue entrenado en tres ocasiones separadas, sobre TCCBRs de $100mA_s$, $250mA_s$ y $360mA_s$ de corriente de tubo. Los resultados muestran una mejora en métricas de MAE, RMSE y PSNR, donde también superan a los resultados producidos por un filtro BM4D, un método de estado del arte en filtrado sin utilización de *Deep Learning*. Sin embargo, al comparar los resultados de la métrica SSIM, el modelo muestra un desempeño peor que las TCCBRs y que el filtro BM4D sobre las tomografías de $250mA_s$ y $360mA_s$. Además, de una inspección visual de los resultados, se puede ver que los detalles finos de estructura ósea que se ven en las imágenes TCCPAR no son recuperados por ambos métodos comparados. Intuitivamente, esto no resulta sorprendente debido a los niveles de ruido presentes en las imágenes TCCBR.

En base a los resultados obtenidos se propone continuar la investigación con el análisis del uso de la métrica SSIM como función de pérdida. También es posible avanzar con el desarrollo de otros tipos de arquitecturas, como lo es una Red cGAN. Estas arquitecturas son las que han obtenido mejor desempeño en este tópico, pero no han sido estudiadas sobre imágenes TCC de vértebras humanas. Además, para esta arquitectura se podría utilizar a la Red Neuronal de este trabajo como Red Generadora.

Por otro lado, este trabajo no trató con la medición de parámetros estructurales que indican la calidad ósea. Esto permite cuantificar la calidad del hueso y de esta manera obtener información que resulta relevante para el diagnóstico de osteoporosis. Se propone realizar este estudio en un desarrollo futuro.

El análisis de parámetros estructurales abre también otras líneas de trabajo que resultan interesantes de explorar. El cómputo de dichos parámetros podría ser realizado utilizando Redes Neuronales. Este cómputo se podría realizar de manera directa, con una Red que en lugar de generar imágenes, produzca valores escalares, pero se propone continuar con una estrategia de cómputo indirecto. Esto consistiría en una Red Neuronal que genere imágenes, pero optimice los valores de parámetros estructurales del hueso, en lugar de los valores de véxoles de la Tomografía. Con las nuevas imágenes, es posible calcular posteriormente sus parámetros estructurales, para luego utilizarlos en el proceso de diagnóstico.

Referencias

- [1] David J Brenner and Eric J Hall. Computed tomography - an increasing source of radiation exposure. *New England Journal of Medicine*, 357(22):2277–2284, 2007.
- [2] Amy Berrington De González, Mahadevappa Mahesh, Kwang-Pyo Kim, Mythreyi Bhargavan, Rebecca Lewis, Fred Mettler, and Charles Land. Projected cancer risks from computed tomographic scans performed in the united states in 2007. *Archives of internal medicine*, 169(22):2071–2077, 2009.
- [3] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.
- [4] Robert Cierniak. *X-ray computed tomography in biomedical engineering*. Springer Science & Business Media, 2011.
- [5] Jiang Hsieh. *Computed tomography: principles, design, artifacts, and recent advances*, volume 114. SPIE press, 2003.
- [6] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Pearson, USA, 4th edition, 2018.
- [7] Applications and clinical benefits of ct imaging. <https://www.imaginis.com/ct-scan/applications-and-clinical-benefits-of-ct-imaging>. Accessed: 2020-03-22.
- [8] Geoff Dougherty. *Medical image processing: techniques and applications*. Springer Science & Business Media, 2011.
- [9] Felix Sebastian Leo Thomsen. Medical 3d image processing applied to computed tomography and magnetic resonance imaging. 2017.
- [10] F Edward Boas and Dominik Fleischmann. Ct artifacts: causes and reduction techniques. *Imaging in medicine*, 4(2):229–240, 2012.
- [11] Dang Thanh, Prasath Surya, et al. A review on ct and x-ray images denoising methods. *Informatica*, 43(2), 2019.
- [12] Hu Chen, Yi Zhang, Mannudeep K Kalra, Feng Lin, Yang Chen, Peixi Liao, Jiliu Zhou, and Ge Wang. Low-dose ct with a residual encoder-decoder convolutional neural network. *IEEE transactions on medical imaging*, 36(12):2524–2535, 2017.

- [13] Chenyu You, Qingsong Yang, Lars Gjesteby, Guang Li, Shenghong Ju, Zhuiyang Zhang, Zhen Zhao, Yi Zhang, Wenxiang Cong, Ge Wang, et al. Structurally-sensitive multi-scale deep neural network for low-dose ct denoising. *IEEE Access*, 6:41839–41855, 2018.
- [14] Xin Yi and Paul Babyn. Sharpness-aware low-dose ct denoising using conditional generative adversarial network. *Journal of digital imaging*, 31(5):655–669, 2018.
- [15] Jelmer M Wolterink, Tim Leiner, Max A Viergever, and Ivana Isgum. Generative adversarial networks for noise reduction in low-dose ct. *IEEE transactions on medical imaging*, 36(12):2536–2545, 2017.
- [16] Michael Balda, Joachim Hornegger, and Bjoern Heismann. Ray contribution masks for structure adaptive sinogram filtering. *IEEE transactions on medical imaging*, 31(6):1228–1239, 2012.
- [17] Armando Manduca, Lifeng Yu, Joshua D Trzasko, Natalia Khaylova, James M Kofler, Cynthia M McCollough, and Joel G Fletcher. Projection space denoising with bilateral filtering and ct noise modeling for dose reduction in ct. *Medical physics*, 36(11):4911–4919, 2009.
- [18] Jing Wang, Tianfang Li, Hongbing Lu, and Zhengrong Liang. Penalized weighted least-squares approach to sinogram noise reduction and image reconstruction for low-dose x-ray computed tomography. *IEEE transactions on medical imaging*, 25(10):1272–1283, 2006.
- [19] Martin J Willemink, Pim A de Jong, Tim Leiner, Linda M de Heer, Rutger AJ Nievelstein, Ricardo PJ Budde, and Arnold MR Schilham. Iterative reconstruction techniques for computed tomography part 1: technical principles. *European radiology*, 23(6):1623–1631, 2013.
- [20] Lucas L Geyer, U Joseph Schoepf, Felix G Meinel, John W Nance Jr, Gorka Bastarrika, Jonathon A Leipsic, Narinder S Paul, Marco Rengo, Andrea Laghi, and Carlo N De Cecco. State of the art: iterative ct reconstruction techniques. *Radiology*, 276(2):339–357, 2015.
- [21] Antoni Buades, Bartomeu Coll, and J-M Morel. A non-local algorithm for image denoising. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, volume 2, pages 60–65. IEEE, 2005.

- [22] Zhoubo Li, Lifeng Yu, Joshua D Trzasko, David S Lake, Daniel J Blezek, Joel G Fletcher, Cynthia H McCollough, and Armando Manduca. Adaptive nonlocal means filtering based on local noise level for ct denoising. *Medical physics*, 41(1):011908, 2014.
- [23] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *IEEE Transactions on image processing*, 16(8):2080–2095, 2007.
- [24] Matteo Maggioni, Vladimir Katkovnik, Karen Egiazarian, and Alessandro Foi. Nonlocal transform-domain filter for volumetric data denoising and reconstruction. *IEEE transactions on image processing*, 22(1):119–133, 2012.
- [25] Dongwoo Kang, Piotr Slomka, Ryo Nakazato, Jonghye Woo, Daniel S Berman, C-C Jay Kuo, and Damini Dey. Image denoising of low-radiation dose coronary ct angiography by an adaptive block-matching 3d algorithm. In *Medical Imaging 2013: Image Processing*, volume 8669, page 86692G. International Society for Optics and Photonics, 2013.
- [26] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311–4322, 2006.
- [27] Yang Chen, Xindao Yin, Luyao Shi, Huazhong Shu, Limin Luo, Jean-Louis Coatrieux, and Christine Toumoulin. Improving abdomen tumor low-dose ct images using a fast dictionary learning based processing. *Physics in Medicine & Biology*, 58(16):5803, 2013.
- [28] Hu Chen, Yi Zhang, Weihua Zhang, Peixi Liao, Ke Li, Jiliu Zhou, and Ge Wang. Low-dose ct via convolutional neural network. *Biomedical optics express*, 8(2):679–694, 2017.
- [29] Lei Xiang, Yu Qiao, Dong Nie, Le An, Weili Lin, Qian Wang, and Dinggang Shen. Deep auto-context convolutional neural networks for standard-dose pet image estimation from low-dose pet/mri. *Neurocomputing*, 267:406–416, 2017.
- [30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680, 2014.

- [31] Yan Wang, Biting Yu, Lei Wang, Chen Zu, David S Lalush, Weili Lin, Xi Wu, Jiliu Zhou, Dinggang Shen, and Luping Zhou. 3d conditional generative adversarial networks for high-quality pet image estimation at low dose. *NeuroImage*, 174:550–562, 2018.
- [32] Kunihiko Fukushima. Neocognitron: A hierarchical neural network capable of visual pattern recognition. *Neural networks*, 1(2):119–130, 1988.
- [33] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat’s striate cortex. *The Journal of physiology*, 148(3):574–591, 1959.
- [34] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.
- [35] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [36] Hyeonwoo Noh, Seunghoon Hong, and Bohyung Han. Learning deconvolution network for semantic segmentation. In *Proceedings of the IEEE international conference on computer vision*, pages 1520–1528, 2015.
- [37] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *2011 International Conference on Computer Vision*, pages 2018–2025. IEEE, 2011.
- [38] Omid E David and Nathan S Netanyahu. Deeppainter: Painter classification using deep convolutional autoencoders. In *International conference on artificial neural networks*, pages 20–28. Springer, 2016.
- [39] Salman Khan, Hossein Rahmani, Syed Afaq Ali Shah, and Mohammed Bennamoun. A guide to convolutional neural networks for computer vision. *Synthesis Lectures on Computer Vision*, 8(1):1–207, 2018.
- [40] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.
- [41] Vincent Dumoulin and Francesco Visin. A guide to convolution arithmetic for deep learning. *arXiv preprint arXiv:1603.07285*, 2016.