



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIAS FÍSICAS

INSTITUTO DE FÍSICA

CORRECCIÓN POR DISPERSIÓN EN TOMOGRAFÍA DE HAZ CÓNICO
MEDIANTE REDES NEURONALES CONVOLUCIONALES

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN CIENCIAS (FÍSICA MÉDICA)

PRESENTA:
FERNANDO GONZALO MONCADA GUTIÉRREZ

TUTOR PRINCIPAL:
DR. ARNULFO MARTÍNEZ DÁVALOS
INSTITUTO DE FÍSICA, UNAM

MIEMBROS DEL COMITÉ TUTOR:
DRA. OLGA OLINCA GALVÁN DE LA CRUZ
INSTITUTO NACIONAL DE NEUROLOGÍA Y NEUROCIRUGÍA

DR. ARTURO AVENDAÑO ESTRADA
FACULTAD DE MEDICINA, UNAM

CIUDAD UNIVERSITARIA, CIUDAD DE MÉXICO, JULIO 2023



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

‘Other than the laws of physics, rules have never really worked out for me.’

Craig Ferguson

Dedicatoria

A la física, por permitirme vivir de lo que más me gusta y a mi física favorita: mi novia.

Agradecimientos

A mi tutor principal el Dr. Arnulfo Martínez, por su dedicación y apoyo en el desarrollo del presente proyecto de investigación, a mi comité tutor, sinodales y profesores de la maestría. Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por la beca otorgada en su Programa Nacional de Posgrados de Calidad en el periodo Agosto 2021-Agosto 2023. Al apoyos PAEP-UNAM 2022 por el apoyo económico para presentar el desarrollo de este proyecto en la Conferencia de Imágenes Médicas (MIC) de la IEEE, en Milán, Italia. Y a la Universidad Nacional Autónoma de México, por su alto nivel académico y por la oportunidad de venir a estudiar a México.

Agradecimientos personales

A Dios por la oportunidad de vivir esta hermosa experiencia de estudiar en otro país y viajar por el mundo. A mis padres por el apoyo a la distancia. A mis amigos de la maestría: Santi Ecuador, Mau, Tony, Jaz, Miriam, Claudio, Santi Colombia, Moi, Ana, Erick e Irene, por los viajes a Puebla, Querétaro, Tepoztlán, Xitle, Xochimilco, las salidas al Jaibol y en el depa; me han hecho sentir como en casa en México. Y sobretodo a mi novia Karen, quien me ha apoyado incondicionalmente en mis proyectos de vida.

Resumen

La tomografía computarizada es una de las principales herramientas de diagnóstico médico, mediante la cual se pueden obtener imágenes tridimensionales de alta resolución de las estructuras internas del cuerpo humano. En radioterapia guiada por imagen se ha empleado ampliamente la tomografía de haz cónico (CBCT) para optimizar la precisión y exactitud en el posicionamiento de los pacientes durante la impartición de tratamientos. Uno de los principales factores que producen artefactos en las imágenes tomográficas es la radiación dispersada. Esta radiación tiene su origen en las interacciones por efecto Compton o dispersión Rayleigh entre los fotones primarios y el paciente o su entorno. Cuando los fotones dispersados alcanzan el detector de imagen, su contribución a la formación de la misma constituye un ruido que afecta su calidad diagnóstica. Una imagen de buena calidad es vital para que el tratamiento impartido al paciente sea preciso, óptimo y la dosis adicional impartida al paciente sea tan baja como sea posible. Los métodos convencionales de corrección por radiación dispersada son computacionalmente demandantes o implican impartir una dosis adicional al paciente, por lo que un método de corrección de imágenes CBCT que sea eficiente y no implique irradiaciones adicionales al paciente es de vital importancia.

En el presente proyecto se desarrolló un método basado en aprendizaje profundo (*deep learning*), entrenando redes neuronales convolucionales para generar imágenes tomográficas corregidas por radiación dispersada. Los datos para el entrenamiento, validación y evaluación de los modelos de *deep learning* se generaron a partir de imágenes obtenidas mediante simulaciones Monte Carlo, validadas previamente, de los sistemas de imagen de un acelerador lineal Varian TrueBeam STx, basadas en la aplicación *egs_cbct* de *EGSnrc*. De esta manera, se determinaron las contribuciones de la radiación dispersada en imágenes CBCT de maniqués digitales geométricos y anatómicos.

Los algoritmos de *deep learning* desarrollados en esta tesis fueron evaluados mediante el error porcentual en las proyecciones corregidas y en los números CT, el cociente señal a ruido y el cociente contraste a ruido en las imágenes reconstruidas. Para analizar el efecto de la arquitectura empleada, se comparó el rendimiento de algoritmos basados en redes tipo U (UNets) y Redes Generatrices Adversariales (GANs). Se encontró que las UNets demostraron ser más eficientes y mostraron resultados más precisos. También se comparó el rendimiento de *funciones de pérdida* basadas en el error absoluto medio, error cuadrático medio, índice de similitud estructural y cociente máximo de señal a ruido. Además, se implementó un método de submuestreo para reducir los tiempos de entrenamiento sin comprometer la calidad de las imágenes, tomando ventaja de que la radiación dispersada se caracteriza por tener fundamen-

talmente componentes de baja frecuencia espacial. Adicionalmente se comparó la distribución de la radiación dispersada en imágenes de kilovoltaje y megavoltaje. Finalmente, se evaluaron la validez y las limitaciones de aplicar transferencia de aprendizaje entre diferentes conjuntos de datos.

El entrenamiento de los modelos se realiza por una única ocasión y su tiempo de entrenamiento depende de la cantidad de datos, el número de píxeles de las imágenes, el número de iteraciones (*epochs*) y las características de la Unidad de Procesamiento Gráfico (GPU) con la que se entrene. Para un conjunto de 250 proyecciones de 192×256 píxeles, entrenadas por 200 *epochs* en un GPU de 32 GB de RAM, el tiempo de entrenamiento fue de 5 minutos. De manera alternativa, los modelos desarrollados pueden entrenarse de manera gratuita en GPUs en la nube (p.ej. *Google Colab*) sin requerimientos especiales de *hardware*. En este caso, el tiempo de entrenamiento bajo las mismas condiciones anteriores es de aproximadamente 16 minutos. Una vez entrenado el modelo, la correcciones por radiación dispersada de un estudio típico de CBCT puede ejecutarse en aproximadamente 3 segundos, lo cual constituye una mejora en alrededor de 5 órdenes de magnitud en tiempo, en comparación con métodos basados en simulaciones Monte Carlo.

Índice general

Dedicatoria	III
Agradecimientos	IV
Resumen	v
1. Introducción	2
1.1. Antecedentes	2
1.2. Planteamiento del problema	5
1.3. Justificación	5
1.4. Objetivos	6
1.4.1. Objetivo general	6
1.4.2. Objetivos específicos	6
1.5. Estructura	7
2. Marco teórico	8
2.1. Tomografía Computarizada	8
2.1.1. Generaciones de tomógrafos	9
2.1.2. Tomografía Computarizada de Haz de Cono	12
2.1.3. Radioterapia guiada por imagen	13
2.2. Física de las imágenes médicas	14
2.2.1. Métricas de la calidad de la imagen tomográfica	15
2.2.2. Métricas de comparación de imágenes	16

2.2.3. Artefactos en las imágenes médicas	18
2.2.4. Dispersión incoherente de los fotones: Dispersión Compton	19
2.2.5. Radiación dispersada	21
2.2.6. Algoritmos de corrección por radiación dispersada	22
2.3. Inteligencia Artificial	23
2.3.1. Aplicaciones en la medicina	25
2.3.2. Aprendizaje profundo	25
2.3.3. Redes neuronales convolucionales	26
2.3.4. UNet	30
2.3.5. Redes Generativas Adversariales	31
2.3.6. Transferencia de aprendizaje	32
2.3.7. Corrección de imágenes médicas	33
3. Materiales y Métodos	36
3.1. Adquisición de datos simulados	36
3.1.1. Simulaciones Monte Carlo de transporte de la radiación: EGSnrc	36
3.2. Maniqués utilizados	38
3.2.1. Maniquí de Unidades Hounsfield (HUP)	38
3.2.2. Maniquí MOBY	39
3.3. Pre-procesamiento de datos	39
3.4. Entrenamiento	41
3.5. Modelos de deep learning	41
3.5.1. UNets	44
3.5.2. Redes Generatrices Adversariales	45
3.6. Análisis de datos	46

4. Resultados y Discusión	48
4.1. UNet entrenada solo con proyecciones del HUP	48
4.1.1. Estructura de los códigos empleados	48
4.1.2. Entrenamiento del modelo	49
4.1.3. Predicciones del modelo	50
4.1.4. Correcciones por radiación dispersada	51
4.1.5. Reconstrucción tomográfica	53
4.2. Conjunto de entrenamiento	55
4.3. Objetivo de predicción	60
4.4. Arquitecturas	63
4.5. Efectos de la Energía en la Radiación dispersada	65
4.6. Efecto de la función de pérdidas	71
4.7. Efecto del submuestreo en la distribución de la radiación dispersada	73
4.8. Transferencia de Aprendizaje	74
4.9. Rendimiento de los GPUs de los equipos empleados	83
5. Conclusiones	84
Bibliografía	85
A. Coeficientes lineales de atenuación	93
B. Números CT, SNR y CNR de los diferentes modelos implementados	96

Capítulo 1

Introducción

1.1. Antecedentes

La tomografía computarizada (CT, por sus siglas en inglés) es una herramienta esencial para la visualización de estructuras anatómicas internas en procedimientos clínicos rutinarios. Su desarrollo comenzó en la década de 1960, generando enormes avances en el diagnóstico médico. Actualmente, es la tecnología de imagenología médica más usada en radiología. Esto se debe a que las imágenes que se adquieren en un equipo de CT permiten identificar, sin ambigüedades, estructuras anatómicas que se superponen en diferentes planos y tienen un mejor contraste que las imágenes radiográficas [1].

La tomografía computarizada de haz de cono (CBCT, por sus siglas en inglés) es una modalidad de imagen avanzada que tiene una amplia variedad de aplicaciones clínicas, que incluyen imagenología dental, maxilofacial, cardíaca, de mama y generación de imágenes para el posicionamiento de pacientes en radioterapia guiada por imagen (IGRT, por sus siglas en inglés). Sin embargo, las dosis de radiación a las que se exponen los pacientes en imágenes CBCT son hasta diez veces mayor que aquellas a las que se exponen en imágenes convencionales de CT [2,3].

En radioterapia, la radiación ionizante se dirige a un volumen blanco en el paciente, principalmente con propósitos paliativos o de erradicación del cáncer. La precisión en el direccionamiento, colimación y modulación del haz de radiación es de primordial importancia para evitar dañar estructuras circundantes y asegurar el cumplimiento de los objetivos del tratamiento. Debido a la naturaleza altamente localizada de la radioterapia, existe una estrecha relación entre esta y la imagenología médica [3]. La necesidad de mejorar los resultados en tratamientos de radioterapia ha llevado a buscar metodologías para disminuir las incertidumbres

en las dosis entregadas a los pacientes en el proceso de planificación. El uso de márgenes de dosis apropiados es crítico para asegurar tratar de manera satisfactoria los volúmenes blanco y proteger adecuadamente el tejido sano y los órganos de riesgo; sin embargo, si estos márgenes pudieran reducirse, sería posible incrementar la dosis prescrita a los volúmenes blanco. Debido a factores como el movimiento de los órganos durante un tratamiento o las incertidumbres de posicionamiento, para alcanzar reducciones significativas en las incertidumbres geométricas se requiere una visualización de las estructuras internas del paciente desde la máquina del tratamiento. Por tal motivo, el desarrollo de sistemas para guiar el tratamiento por imágenes fue el centro de la investigación en CBCT desde finales de la década de 1990. [4]

En la IGRT se adquiere un conjunto de imágenes CBCT, las cuales son comparadas con aquellas de la planificación realizada en la simulación CT, al inicio del curso del tratamiento. De esta comparación se determina el ajuste final necesario para que el paciente sea ubicado exactamente en la misma posición y configuración como lo estuvo durante la planificación original [5]. En IGRT se ha empleado ampliamente la CBCT, modalidad de imagen que permite optimizar la precisión y exactitud en la impartición de tratamientos. Sin embargo, existen varios factores que producen artefactos en las imágenes tomográficas, afectando la calidad de éstas; entre ellos, uno de los más importantes es la radiación dispersada.

En CBCT, la radiación dispersada es la principal fuente de disminución en la calidad de la imagen. En el proceso de formación de la imagen, los detectores que se encuentran ubicados en la trayectoria directa del haz, no son capaces de diferenciar el mecanismo físico de las interacciones por las cuales se redujo la intensidad. Fundamentalmente, estos mecanismos pueden ser efecto fotoeléctrico o efecto Compton. Cuando ocurre el efecto Compton, el fotón incidente es absorbido y se produce un fotón dispersado, el cual puede llegar al detector. Estos fotones producidos como consecuencia del efecto Compton constituyen la radiación dispersada. En ciertos procedimientos, la magnitud de la fluencia de radiación dispersada puede exceder a la de los fotones primarios en el plano del detector. Se sabe también que al tener una mayor cantidad de detectores, existe un mayor aporte de la radiación dispersada debido a que esta sale en distintos ángulos y por tanto puede ser detectada en más regiones, lo cual es el caso en imágenes de CBCT [3]. La principal fuente de radiación dispersada es el paciente, pero también puede ser generada por la camilla, filtros del haz y colimadores. Por tal motivo, los esfuerzos para mejorar la calidad de imagen en CBCT se enfocan principalmente en suprimir los efectos de la radiación dispersada [3].

Los efectos de la radiación dispersada pueden ser *corregidos* mediante métodos de supresión y/o estimación de la radiación dispersada. Entre los métodos de supresión de la radiación dispersada se incluye la incorporación de una rejilla antidispersión, el uso de grandes espacios

de aire (incrementar distancia paciente-detector) y la minimización del campo de vista. Estos métodos tienen como objetivo disminuir la proporción de radiación dispersada que llega al arreglo de detectores [3,6]. Por otro lado, los métodos de estimación de la radiación dispersada se basan en simulaciones Monte Carlo (MC), modelos analíticos y/o empíricos, kernels de deconvolución y algoritmos de inteligencia artificial. Estas estimaciones son realizadas con base en mediciones directas, calibraciones, cálculos o simulaciones. El objetivo de estos métodos es definir algoritmos para identificar el aporte de la radiación dispersada y suprimirlo de la fluencia medida (primarios + dispersados) [3,6].

La Inteligencia Artificial (AI, por sus siglas en inglés) se define como la habilidad de un sistema para interpretar correctamente datos externos, aprender de dichos datos y emplear su aprendizaje para cumplir tareas y metas específicas [7]. El aprendizaje de máquina o *machine learning* es una técnica cuyo objetivo consiste en entrenar al algoritmo con un conjunto de datos dado, para que este encuentre el modelo que resuelve el problema en estudio [8]. Los algoritmos de aprendizaje de máquina son capaces de realizar predicciones o tomar decisiones con base en los modelos construidos a partir de los datos de entrada del algoritmo y son empleados en aplicaciones donde los datos históricos son capaces de predecir eventos futuros [9]. El aprendizaje profundo (*deep learning*) es un campo reciente con un crecimiento acelerado dentro del área de aprendizaje de máquina, que se encarga de buscar modelos a partir de la abstracción de datos a gran escala, empleando múltiples capas de ‘redes neuronales profundas’. De esta manera, estos algoritmos adquieren y procesan información relevante a partir de imágenes, sonidos o textos, de manera supervisada o no supervisada. Las primeras estructuras de aprendizaje profundo fueron construidas sobre redes neuronales artificiales en los 80s; sin embargo, el impacto real del aprendizaje profundo se hizo notorio a partir de mediados de los 2000s, con la comercialización y disponibilidad de unidades procesadoras gráficas [10]. Una de las principales ventajas del aprendizaje profundo frente al aprendizaje de máquina es que mientras los algoritmos de aprendizaje de máquina alcanzan un máximo en su rendimiento, más allá del cual, aunque se disponga una suficiente cantidad de datos, no se puede mejorar el algoritmo; en el aprendizaje profundo, el rendimiento puede ser ampliamente superior al de los algoritmos de aprendizaje de máquina, siempre y cuando se disponga de la suficiente cantidad de datos. En el contexto de aprendizaje de máquina, las redes neuronales convolucionales (CNN, por sus siglas en inglés) son un tipo de redes neuronales artificiales en las cuales el patrón de conectividad entre las neuronas está inspirado en la organización de la corteza visual de los animales [9].

Durante los últimos años, la AI ha sido ampliamente usada en la medicina. Sus aplicaciones incluyen algoritmos de aprendizaje, desarrollo de dispositivos médicos y robots. Los algo-

ritmos de AI, han impulsado grandes descubrimientos en genética, medicina molecular y pronóstico médico [11]. Sin embargo, la rama de la medicina en la cual la AI se ha aplicado más ampliamente es la radiología. Esto se debe fundamentalmente al enorme progreso desarrollado en tareas de reconocimientos de imágenes, la digitalización de datos y el incremento del poder computacional [12].

1.2. Planteamiento del problema

La CT es una de las principales herramientas de diagnóstico médico, mediante la cual se pueden obtener imágenes detalladas de las estructuras internas del cuerpo humano. En IGRT se ha empleado ampliamente la CBCT para optimizar la precisión y exactitud en el posicionamiento de los pacientes para la impartición de tratamientos. Uno de los principales factores que producen artefactos en las imágenes tomográficas, afectando la calidad de las mismas, es la radiación dispersada.

En el proyecto de tesis de maestría del M. en C. Brian Zapien [13] se caracterizaron los equipos CBCT de kilo- y megavoltaje de un acelerador lineal Varian TrueBeam STx usando simulaciones MC. En aquel proyecto se implementó un método iterativo de corrección de imágenes por radiación dispersada. El presente proyecto plantea desarrollar un método basado en aprendizaje profundo, entrenando CNN, para generar imágenes tomográficas corregidas por artefactos de radiación dispersa. A partir de las simulaciones MC desarrolladas en la tesis de Brian Zapien se generarán imágenes corregidas y no corregidas por radiación dispersada. Estas imágenes serán empleadas para conformar conjuntos de pares de imágenes con los que se entrenarán, validarán y evaluarán los algoritmos de aprendizaje profundo [6, 14, 15]. Una vez entrenados, estos algoritmos permitirán ejecutar correcciones por radiación dispersada de una forma más rápida que con los métodos MC.

1.3. Justificación

Uno de los principales parámetros de interés en la evaluación de las imágenes tomográficas es la calidad de imagen. Una imagen de buena calidad es vital para que el tratamiento impartido al paciente sea preciso, óptimo y la dosis adicional sea tan baja como sea posible. Una imagen con baja dosis generalmente es ruidosa y de pobre calidad, mientras que una imagen de buena calidad requiere impartir al paciente una mayor dosis. De esta manera, existe una competencia entre la calidad de la imagen y la dosis impartida durante la adquisición de la misma.

Uno de los principales factores responsables de artefactos en imágenes tomográficas es la radiación dispersa. Este efecto es de particular interés en CBCT, debido al amplio campo de

radiación que posee, donde los fotones de radiación dispersada pueden corresponder incluso a más de un 50 % del total de fotones que se detectan para la obtención de la imagen. De esta manera, el presente proyecto plantea el desarrollo de un algoritmo de aprendizaje profundo que sea capaz de corregir de manera automática los efectos de la radiación dispersada en imágenes tomográficas. Esto permitiría incorporar técnicas que mejoren la calidad de la imagen sin necesidad de incrementar la dosis impartida al paciente.

En proyectos de investigación previos ya se han separado los efectos de la radiación dispersada mediante simulaciones MC, las cuales a pesar de requerir pocas iteraciones, son computacionalmente demandantes dado que requieren tiempos de cálculo muy extensos (algunas horas) como para una implementación en un ambiente clínico. En particular, en CBCT, donde las imágenes tomográficas se toman para guiar tratamientos de IGRT, es de gran interés un algoritmo que pueda realizar la corrección de la imagen tomográfica en pocos segundos, de tal manera que no se extienda significativamente el tiempo del tratamiento del paciente. Se estima que los algoritmos de aprendizaje profundo a desarrollar en el presente proyecto identifiquen y resten las componentes de radiación dispersada a partir de una imagen tomográfica dada, de manera eficaz y eficiente.

1.4. Objetivos

1.4.1. Objetivo general

Diseñar, entrenar y optimizar redes neuronales convolucionales mediante el uso de conjuntos de datos generados con simulaciones Monte Carlo para identificar y suprimir el aporte de la radiación dispersada en imágenes tomográficas producidas por CBCT.

1.4.2. Objetivos específicos

- Comprender las bases físicas de la CBCT y artefactos en las imágenes tomográficas debido a la radiación dispersada.
- Usar simulaciones Monte Carlo de transporte de radiación en materia para la generación de imágenes médicas y su uso en radioterapia.
- Entender los métodos de reconstrucción y corrección de imágenes por radiación dispersada.
- Aplicar algoritmos de aprendizaje profundo para aplicaciones de interés en física médica.

1.5. Estructura

El presente trabajo de tesis está estructurado por capítulos. En el primer capítulo se presenta la introducción, antecedentes teóricos, planteamiento del problema, justificación y objetivos. En el segundo capítulo, se presentan el marco teórico sobre el que se basa el presente trabajo de investigación; en particular, se hace énfasis sobre la tomografía de haz de cono, los efectos de la radiación dispersada en la formación de imágenes y el aprendizaje profundo. En el tercer capítulo se presenta la metodología de la investigación. En el cuarto capítulo se presentan y discuten los resultados más relevantes. En el quinto capítulo se presentan las conclusiones del trabajo.

Capítulo 2

Marco teórico

2.1. Tomografía Computarizada

La tomografía computarizada (CT) es una herramienta esencial en la visualización de estructuras anatómicas internas de los pacientes en procedimientos clínicos rutinarios. Entre los principales antecedentes históricos que contribuyeron al desarrollo de la CT, se puede destacar el artículo “*Sobre la determinación de funciones desde sus integrales de línea a lo largo de ciertas variedades*” publicado en 1917, por el matemático Johann Radon, trabajo en el que se plantea la transformada de Radon, que es la base matemática para la reconstrucción de una imagen tomográfica [16, 17]. La CT es un *problema inverso*, cuyo objetivo central es encontrar la distribución espacial de los coeficientes lineales de atenuación de un objeto que producen el conjunto de proyecciones medidas. En 1961, la solución a este problema inverso fue aplicada por primera vez a una secuencia de proyecciones de rayos X de un objeto, medidas desde diferentes direcciones [1]. En 1963, el físico Allen MacLeod Cormack realizó la primera implementación matemática para una reconstrucción tomográfica. En 1969 el ingeniero eléctrico Godfrey Hounsfield demostró la validez de estos principios matemáticos con el primer escáner CT basado en una fuente radioactiva en los laboratorios de investigación de la *Electric and Musical Industries* (EMI). En 1972 Hounsfield y el radiólogo Jamie Ambrose publicaron los primeros escaneos clínicos con un escáner EMI de cabeza [1]. En la Fig. 2.1 se muestra el prototipo clínico de escáner de cerebro instalado en el Hospital Atkinson Morley’s, en Londres y su primera imagen producida. En 1975 se instaló el primer escáner de cuerpo completo con un sistema de haz de abanico [1]. Finalmente, Cormack y Hounsfield recibieron el premio Nobel en Medicina en 1979 por “*sus contribuciones en el desarrollo de la tomografía asistida por computadora, un método radiológico revolucionario, particularmente*

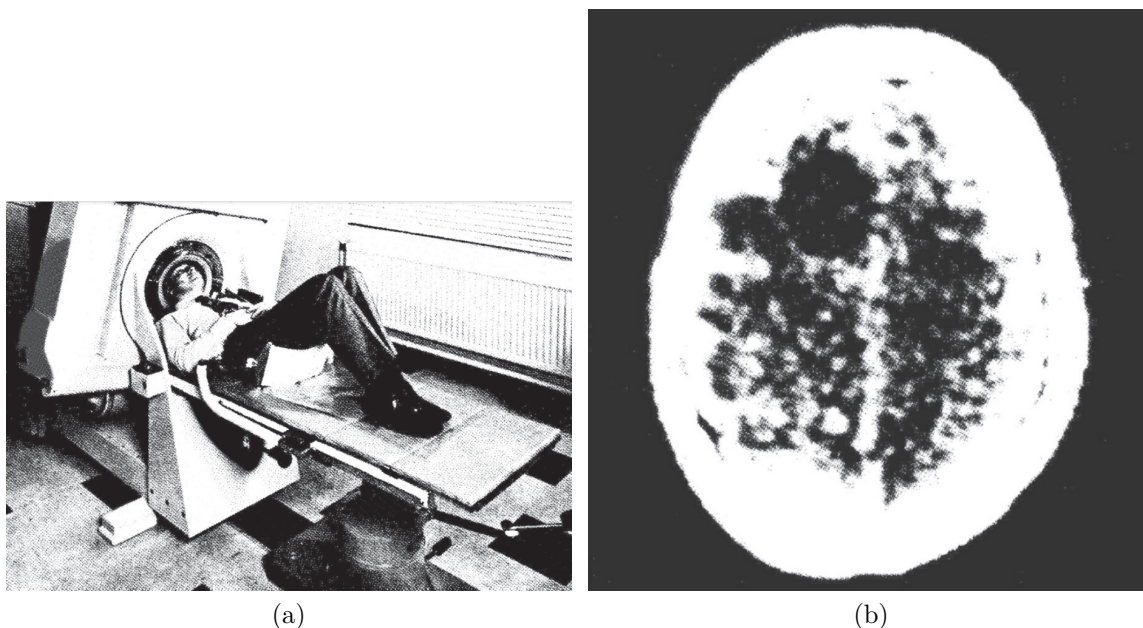


Figura 2.1: (a) Primer prototipo clínico de escáner de cabeza instalado en el hospital Atkinson Morley's, en Londres. (b) Primera imagen producida por este prototipo. Imágenes tomadas de [19].

por la investigación de enfermedades del sistema nervioso.” [18]

2.1.1. Generaciones de tomógrafos

A partir de los primeros escaneos CT, se han realizado múltiples esfuerzos por mejorar el proceso, tanto en calidad de imagen como en tiempo de adquisición de las proyecciones. Estas mejoras se han clasificado en cuatro generaciones de tomógrafos claramente identificadas y posteriormente se han desarrollado varios tipos de tomografía que se detallan en los siguientes párrafos. Su clasificación se debe a la forma en la que están contruidos, ubicación y movimientos, tanto del tubo de rayos X como del arreglo de detectores [1].

La primera generación de CT corresponde a un tubo de rayos X y un detector individual, situado en el lado opuesto del campo de medición. Esta geometría se conoce como “haz de lápiz” y es generada mediante un colimador con un agujero circular pequeño (*pinhole*). El detector se mueve de manera sincrónica junto con el tubo de rayos X , mediante un desplazamiento lineal, el cual es repetido para los diferentes ángulos de proyección (ver Fig. 2.2a) [20]. En estos tomógrafos prácticamente no se detecta radiación dispersada debido a que solo se tiene un detector individual y de área pequeña en la dirección del haz incidente y al producirse radiación dispersada, por Efecto Compton, esta tiene una dirección distinta a la del haz incidente, por lo que no llega al detector. Sin embargo, el proceso de adquisición

de imágenes es muy lento.

La segunda generación de tomógrafos se caracterizó por una fuente de rayos X con un haz angosto de abanico (alrededor de 10°) y un arreglo de detectores colimados en el extremo opuesto (alrededor de 30°). Debido al pequeño ángulo de apertura del haz de abanico, el tubo de rayos X y el arreglo de detectores aún deben ser trasladados linealmente para los diferentes ángulos de proyección (ver Fig. 2.2b). A pesar de los desplazamientos requeridos, el tiempo de adquisición en esta generación se redujo a pocos minutos por corte [21].

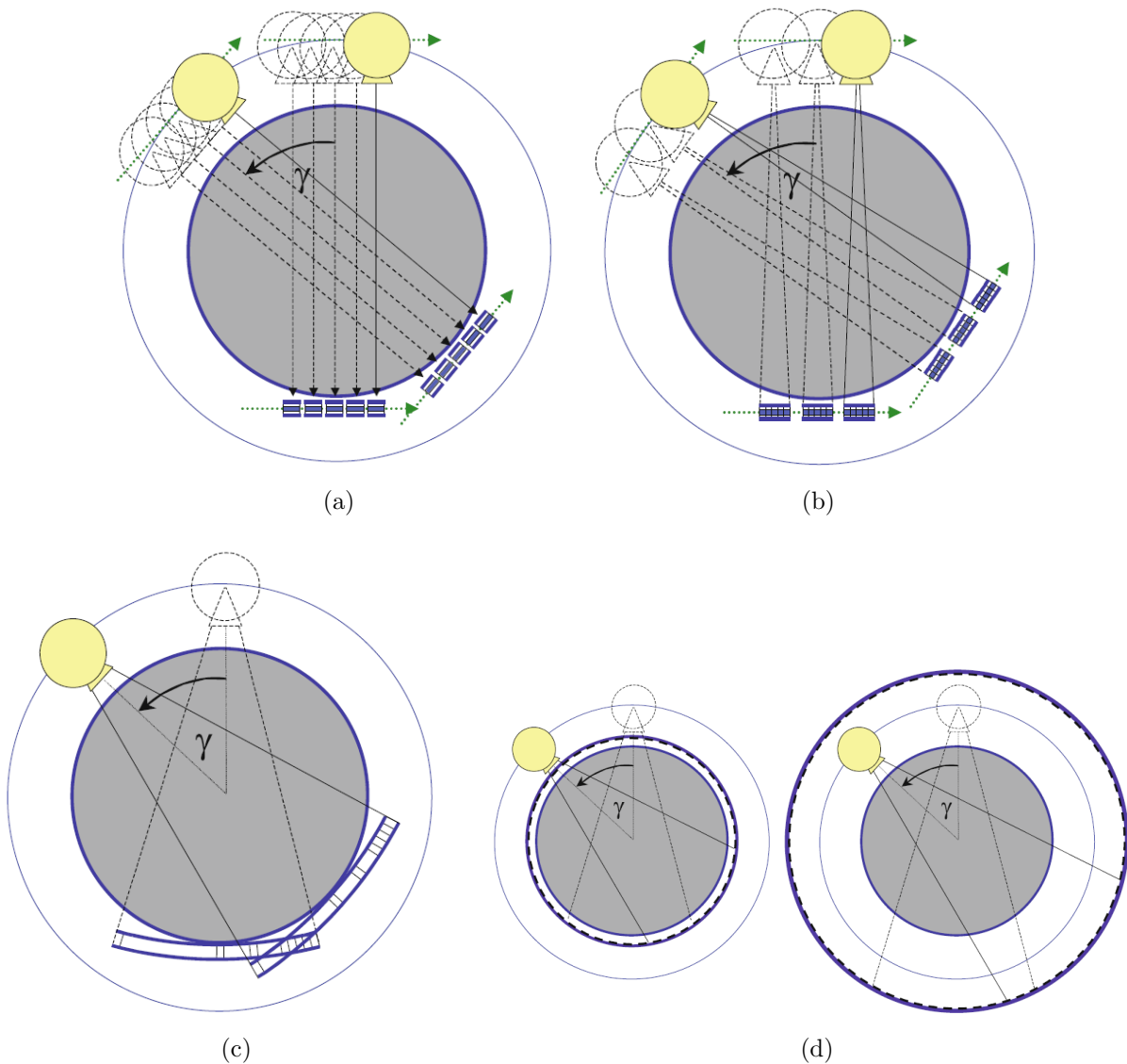


Figura 2.2: Esquema de las cuatro primeras generaciones de CT. Imágenes tomadas de [1].

La tercera generación se caracterizó por la introducción de ángulos más grandes para el haz

de abanico (alrededor de 40° - 60°) y arreglos de detectores colimados más extensos (entre 400-1000). Como resultado, los tiempos de adquisición se redujeron considerablemente debido a que en esta generación fue posible realizar una rotación completa sin interrupciones para los desplazamientos lineales (ver Fig. 2.2c) [1].

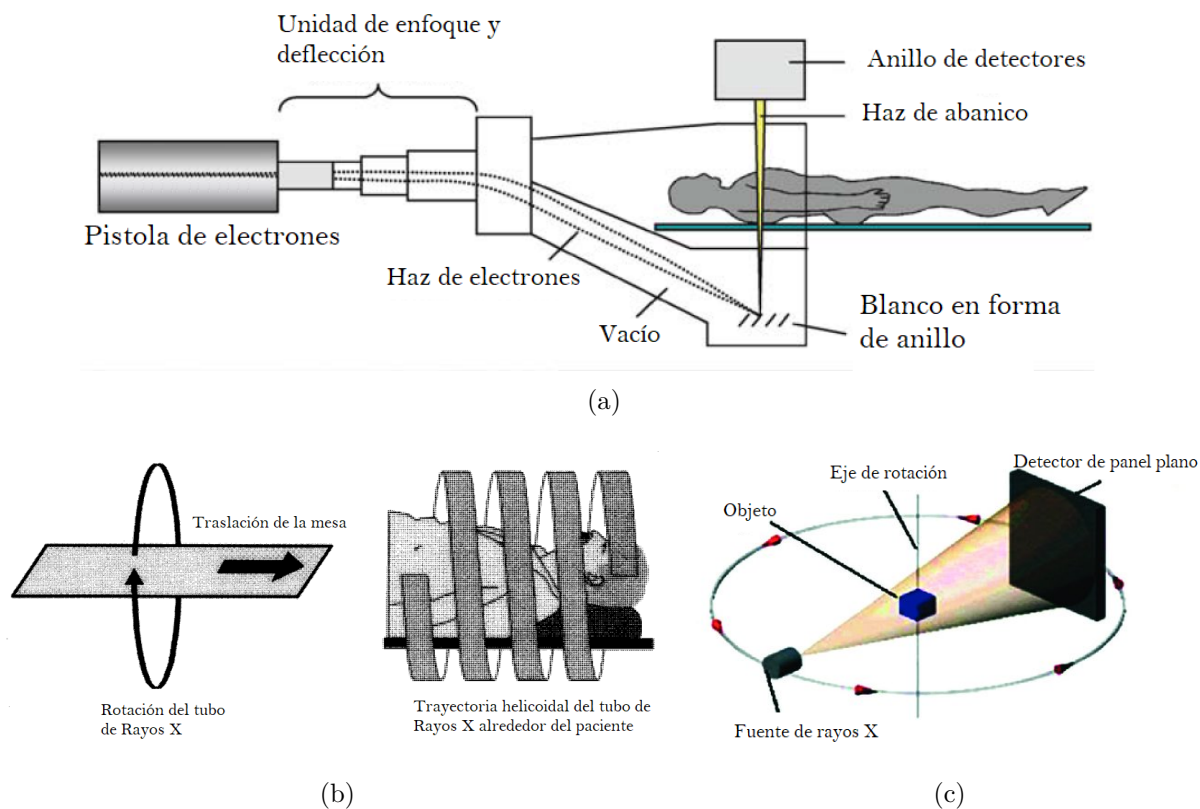


Figura 2.3: Diferentes modalidades de CT. Se muestran los esquemas de EBCT (a), CT helicoidal (b) y CBCT (c). Imágenes adaptadas de [1], [22] y [23], respectivamente.

En la cuarta generación de escáneres de CT se introdujeron arreglos circulares continuos de hasta 5000 detectores. En esta generación solo se mueve el tubo de rayos X, el cual puede estar ubicado adentro o afuera del anillo de detectores (ver Fig. 2.2d). Debido a que los detectores son estáticos, no pueden ser colimados por lo que en esta generación se tiene una mayor contribución de radiación dispersada [21].

La tomografía computarizada por haz de electrones (EBCT, por sus siglas en inglés) se caracterizó por una mejora en los tiempos de adquisición llegando a obtener imágenes hasta en 50 ms . En esta modalidad, un haz de electrones es enfocado sobre blancos de tungsteno en forma de anillo los cuales se arreglan en forma de semicírculo alrededor del paciente. Esta geometría permite que tras el impacto de los electrones con el tungsteno, los rayos X



Figura 2.4: Acelerador lineal *Varian True Beam STX* que incorpora la tecnología CBCT [24].

producidos formen el haz de abanico deseado (ver Fig. 2.3e). De igual manera, en este caso no hay colimadores. Aunque no hay una clara definición de las generaciones posteriores a la cuarta, de acuerdo con Bushberg [22], se ha identificado a estos tomógrafos como la quinta generación.

Luego se introdujo una tecnología de anillos deslizantes que permitió el muestreo *helicoidal* mostrado en Fig. 2.3f. Estos tomógrafos se caracterizan por tiempos de adquisición muy cortos y elevados costos [20]. Según la clasificación de Bushberg [22], estos tomógrafos corresponden a la sexta generación.

Posteriormente se desarrollaron tomógrafos equipados con un haz de cono (CBCT) y un arreglo de detectores en panel como lo indica la Fig. 2.3g. La motivación para el desarrollo de CBCT fue la misma que el avance de haz de lápiz a haz de abanico, disminuir el tiempo de adquisición [1]. El tener un arreglo plano de detectores implicó un incremento en el aporte de la radiación dispersada. De acuerdo con la clasificación de Bushberg [22], estos tomógrafos corresponden a la séptima generación.

2.1.2. Tomografía Computarizada de Haz de Cono

La CBCT representó la esperada evolución desde un haz de abanico hacia un haz de cono. Para su desarrollo, se introdujo un arreglo de detectores planos para reemplazar los arreglos lineales de detectores y se incrementó el ancho de banda necesario para transferir la gran cantidad de datos crudos recolectados por los detectores. Para la reconstrucción de la imagen se desarrollaron algoritmos más sofisticados en comparación con los métodos bidimensionales [1].

La forma convencional de obtener una imagen tridimensional de un objeto consiste en ilumi-

nar a este con un haz angosto de rayos X y emplear algoritmos bidimensionales de reconstrucción, como el de retroproyección filtrada (FBP, por sus siglas en inglés). La reconstrucción tridimensional se obtiene iluminando planos sucesivos del objeto y apilando las reconstrucciones bidimensionales resultantes. Por otro lado, una aproximación más eficiente corresponde a una generalización de los algoritmos bidimensionales de haz de abanico. En este caso, en lugar de iluminar un corte del objeto con un haz de abanico se ilumina todo el objeto con una fuente puntual cuyo haz toma la forma de un cono y su transmisión se mide en un plano opuesto. El algoritmo de reconstrucción de haz de cono más utilizado fue propuesto por Feldkamp, Davis, y Kress y se lo conoce como FDK [25]. La principal ventaja de los algoritmos de haz de cono es la reducción en el tiempo de adquisición [26].

A partir de la introducción comercial de la CBCT en 2005, esta se ha convertido rápidamente en la principal tecnología que ha permitido tener imágenes como guía durante un tratamiento de RT. De hecho, su desarrollo ha significado el comienzo de la era de la Radioterapia Guiada por Imagen (IGRT) [3]. En la Fig. 2.4 se muestra un acelerador lineal *Varian TrueBeam STx* que incorpora la tecnología CBCT.

2.1.3. Radioterapia guiada por imagen

La posibilidad de emplear el haz de tratamiento para ejecutar CT de megavoltajes (MV) del paciente en la posición del tratamiento fue propuesto por primera vez en 1983 por Swindell et al. [27] y su implementación con CBCT fue desarrollada por Mosleh-Shirazi [28]. De esta manera, fue posible adquirir imágenes portales que permiten asegurar tanto la correcta apertura del haz, como la ubicación del paciente; sin embargo, esto causa un incremento en la dosis que recibe el paciente.

Desde la década de 1980 hasta la actualidad han estado comercialmente disponibles los dispositivos electrónicos portales de imagen (EPIDs, por sus siglas en inglés), montados sobre los aceleradores lineales, lo cual ha mejorado la calidad de imagen y precisión del tratamiento. Estos dispositivos han permitido el incremento en el número de imágenes portales para análisis y en la velocidad a la cual se obtienen las mismas, mientras el paciente aún se encuentra en la mesa del tratamiento. La baja calidad de imágenes portales adquiridas con rayos X de MV, debido a la falta de contraste y resolución espacial, cuestionó la precisión de la metodología. Una aproximación lógica a este problema fue emplear rayos X de kilovoltaje (kV) para verificar la configuración del tratamiento. A mediados de la década de 1990, investigadores del hospital William Beaumont usaron un sistema de EPID para adquirir imágenes desde una fuente de kV. La marcada mejora en la calidad de las imágenes portales motivó la instalación de sistemas de imágenes de kV en aceleradores Elekta, los cuales ya contaban con sistemas

EPID de MV. En 1997, el sistema de imagen de kV fue instalado en posición ortogonal al sistema de MV con la fuente de rayos X de kV montada sobre el tambor del cabezal del acelerador [3]. El desarrollo de sistemas de imágenes volumétricos para guiar el tratamiento por imágenes fue el centro de la investigación desde finales de la década de 1990. [4]

La CBCT ha permitido disponer de grandes cantidades de imágenes como guías, previo a un tratamiento. La mayoría de imágenes CBCT se obtienen con un tubo de rayos X de kV y un panel plano de detectores de silicio acoplados a un material centelleador. Esta técnica de tomografía permite obtener imágenes volumétricas en alrededor de un minuto [5].

En resumen, el proceso de IGRT consiste en obtener un conjunto de imágenes CBCT, las cuales son comparadas con las adquiridas mediante CT durante la simulación virtual al inicio del proceso de tratamiento. De esta manera se determina el ajuste final necesario para que el paciente sea ubicado exactamente en la misma posición y configuración como lo estuvo durante la planeación [5].

2.2. Física de las imágenes médicas

La medicina humana se caracteriza por buscar comprender el cuerpo humano, su estructura y funcionamiento tanto bajo condiciones de buena salud como de enfermedad. Debido a la gran complejidad de las estructuras del cuerpo humano, las imágenes médicas se han convertido en una herramienta indispensable para el diagnóstico médico [29].

Existen diferentes técnicas médicas que permiten obtener imágenes del cuerpo humano, entre las que destacan radiografías, tomografías, resonancias magnéticas, ultrasonido, imágenes de medicina nuclear, tomografías por emisión de positrones, entre otras. Estas imágenes médicas revelan características como la transmisividad, opacidad, emisividad, reflectividad, conductividad, etc.; lo cual a su vez brinda información sobre propiedades intrínsecas como el número atómico efectivo, la densidad física o la densidad electrónica. Dependiendo en la aplicación específica, los datos de las imágenes pueden ser interpretados en términos anatómicos o de procesos fisiológicos [29].

A principios de los 1970's, una enorme innovación fue introducida en imagenología diagnóstica, la CT. La introducción de la CT constituye el evento más significativo en imagenología médica desde el descubrimiento de los rayos X . La calidad de sus resultados dependen de múltiples factores como los parámetros de adquisición de la imagen (kV, tiempo de exposición, corriente, etc.), la dosis impartida al paciente y los algoritmos de reconstrucción de imágenes. Debido a la importancia de los algoritmos de reconstrucción de imágenes, la CT ha abierto las puertas a la posibilidad de manipulación computacional de las imágenes. Así,

la física es la ciencia que se encuentra detrás de la producción de estas imágenes y está en estrecha relación con la radiología desde el descubrimiento de los rayos X [29].

Es fundamental que el sistema diseñado para la adquisición de imágenes funcione correctamente, cumpla con los estándares en la calidad de la imagen y a la vez, minimice la exposición del paciente y radiólogos. Para cumplir estos objetivos se deben medir de forma periódica una serie de parámetros que afectan la calidad en las imágenes; esto como parte de los programas de aseguramiento de la calidad (QA, por sus siglas en inglés) [30].

2.2.1. Métricas de la calidad de la imagen tomográfica

Cociente señal a ruido

El *ruido* en una imagen médica corresponde a las fluctuaciones estocásticas en los valores de intensidad, las cuales no están asociadas con variaciones en la magnitud física que se mide en las estructuras de interés. La principal fuente de estas fluctuaciones es el ruido cuántico, que tiene su origen en la distribución aleatoria de los fotones individuales de rayos X que inciden en cada detector. Este ruido puede ser caracterizado en términos de su magnitud y correlación espacial. La magnitud del ruido se puede describir como la desviación estándar de los valores de los vóxeles de regiones uniformes. La correlación del ruido se refiere a la extensión en la cual la fluctuación en un vóxel afecta la de sus vecinos. El ruido en CBCT es *no-estacionario*, es decir, depende de la posición en la imagen. La dependencia espacial del ruido se debe a la variación en el número promedio de rayos X que atraviesan el detector. Por ejemplo, la media y varianza de la fluencia en el centro del detector es diferente que en sus bordes. Como consecuencia, en general el ruido es mayor en el centro que en los bordes de la imagen [3]. Por lo tanto, como un indicador de la calidad de una imagen, es útil definir un cociente entre el nivel de la señal y el rango de fluctuación, lo cual se conoce como *cociente señal a ruido* (SNR, por sus siglas en inglés) y se define según (2.1), la cual para fotones que siguen una distribución de Poisson se simplifica a (2.2) [1].

$$SNR = \frac{\text{señal}}{\text{ruido}} = \frac{\mu}{\sigma} \quad (2.1)$$

$$SNR = \frac{\langle n \rangle}{\sqrt{\langle n \rangle}} = \sqrt{\langle n \rangle} \propto \sqrt{\text{dosis}} \quad (2.2)$$

donde μ y σ son la media y desviación estándar de la señal y $\langle n \rangle$ es el valor de expectación del número de fotones en una distribución de Poisson. De esta manera, al incrementar la fluencia de fotones se puede mejorar la SNR ; sin embargo, esto implica un incremento en la dosis impartida al paciente por lo que no se la puede incrementar ilimitadamente. Para dosis

bajas, el ruido cuántico es el factor dominante que influye en la calidad de imagen [1].

Contraste

El contraste es la diferencia en la señal entre dos regiones dadas de una imagen. Las unidades del contraste son las mismas de la imagen; sin embargo, también es posible definirlo normalizado con respecto a la señal promedio. Debido a que el coeficiente de atenuación depende de la energía del haz, el contraste también.

2.2.2. Métricas de comparación de imágenes

Error Cuadrático Medio

El error cuadrático es la métrica más simple y comúnmente usada para comparar un valor medido/calculado con un valor esperado. Matemáticamente, corresponde a la distancia entre dos elementos de un espacio vectorial euclídeo con norma L^2 . En el caso de imágenes esta métrica se extiende al error cuadrático medio (MSE, por sus siglas en inglés), donde se calcula el error cuadrático píxel a píxel y se promedian estos valores. Sea X_{ij} una matriz de píxeles de $M \times N$ que representa la imagen de la cual deseamos evaluar su calidad y R_{ij} la de la imagen de referencia, el MSE se define mediante la ec. (2.3).

$$MSE = \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (X_{ij} - R_{ij})^2 \quad (2.3)$$

Error Absoluto Medio

Una forma alternativa y comúnmente empleada para comparar un valor medido/calculado con un valor esperado es el error absoluto medio (MAE, por sus siglas en inglés). Matemáticamente, corresponde a la distancia entre dos elementos de un espacio vectorial euclídeo con norma L^1 . El MAE se define mediante la ec. (2.4).

$$MAE = \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |X_{ij} - R_{ij}| \quad (2.4)$$

Error Absoluto Porcentual Medio

El error absoluto porcentual medio (MAPE, por sus siglas en inglés) determina las discrepancias entre los valores medidos y esperados en forma de porcentaje. En el caso de imágenes el MAPE se define mediante la ec. (2.5). Una desventaja de esta métrica es que diverge en

el caso en el que la imagen de referencia tenga píxeles con valores nulos o cercanos a cero.

$$MAPE = \frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \frac{|X_{ij} - R_{ij}|}{R_{ij}} \quad (2.5)$$

Proporción Máxima de Señal a Ruido

En el área de procesamiento de imágenes se define la proporción máxima de señal a ruido (PSNR, por sus siglas en inglés) para cuantificar la calidad de una imagen y se la calcula mediante (2.6).

$$PSNR = 10 \cdot \log_{10} \frac{L^2}{MSE} \quad (2.6)$$

Donde L es el rango dinámico de las intensidades permitidas para un píxel de la imagen. El PSNR usa una escala logarítmica debido al amplio rango de variaciones que se puede tener entre imágenes y sus unidades son los decibeles. Se ha reportado que el MSE y, por consiguiente, el PSNR no permiten discriminar contenido estructural en imágenes [31]. Mientras mayor valor sea el PSNR, la similitud entre las imágenes comparadas es mejor.

Índice de Similitud Estructural

El Índice de Similitud Estructural (SSIM, por sus siglas en inglés) es una métrica que determina la calidad de una imagen, midiendo la similitud estructural de esta con respecto a una referencia [31]. Este índice considera la luminosidad, l , contraste c y similitud de las estructuras s ; se lo determina mediante la ec. (2.7).

$$\begin{aligned} SSIM &= l(X_{i,j}, R_{i,j}) \cdot c(X_{i,j}, R_{i,j}) \cdot s(X_{i,j}, R_{i,j}) \\ &= \frac{2\mu_X \mu_R + C_1}{\mu_X^2 + \mu_R^2 + C_1} \cdot \frac{2\sigma_X \sigma_R + C_2}{\sigma_X^2 + \sigma_R^2 + C_2} \cdot \frac{\sigma_{X,R} + C_3}{\sigma_X \sigma_R + C_3} \end{aligned} \quad (2.7)$$

Donde μ_X y μ_R son el promedio en la intensidad de los píxeles de X_{ij} y R_{ij} , respectivamente; σ_X y σ_R son las correspondientes desviaciones estándar y $\sigma_{X,R}$ es la covarianza entre X_{ij} y R_{ij} . Las constantes positivas $C_1 = (k_1 \cdot L)^2$, $C_2 = (k_2 \cdot L)^2$ y $C_3 = C_2/2$ son empleadas para evitar que los denominadores se anulen y se tengan inestabilidades numérica y L es el rango dinámico de las intensidades permitidas para un píxel de la imagen. Este índice fue propuesto originalmente por Wang, et al. (2004) y los valores propuestos en el artículo original para k_1 y k_2 son 0.01 y 0.03, respectivamente [32]. Cada una de estas medias, desviaciones estándar y covarianzas son calculadas con una ‘ventana deslizante’. De esta manera, se tiene un valor de SSIM para cada región de la imagen, centrada en cada píxel. Todos estos valores se promedian

para determinar el valor del SSIM de la imagen total. El SSIM toma valores entre -1 y 1 , donde 1 corresponde al caso en el que las dos imágenes en comparación son idénticas. En contraste con el PSNR, este índice es poco sensible a cambios en la luminosidad y contraste que no degradan la estructura de la misma [33].

2.2.3. Artefactos en las imágenes médicas

Se conoce como *artefactos* a los errores obtenidos en las imágenes tomográficas, los cuales pueden ser atribuidos a diferentes factores como simplificaciones en los métodos de reconstrucción y fallas en los detectores. Entre las principales simplificaciones de los algoritmos de reconstrucción, destacan la consideración de un haz monoenergético y la adquisición de proyecciones en forma discreta y no continua. Es esencial corregir estos artefactos debido a que su presencia reduce el valor diagnóstico de las imágenes [1]. A continuación se describen los artefactos más comunes en imágenes CBCT.

El *artefacto de copa* está asociado al endurecimiento del haz y se caracteriza por producir un mayor oscurecimiento hacia el centro de regiones homogéneas de un objeto con una frontera de alto contraste como se observa en la Fig. 2.5a. Por lo tanto, el valor medido en las proyecciones es siempre inferior al valor esperado (“verdadero”) [1].

La dispersión de rayos X es una fuente importante de artefactos, en especial en haces volumétricos, como es el caso del haz de cono en CBCT. Esto se debe a las elevadas fracciones de radiación dispersada que producen este tipo de haces. De hecho, en CBCT, es común tener un cociente radiación dispersada a primaria superior al 100%. En estos casos, la mayoría de fotones que arribaron al detector han sufrido al menos un evento de dispersión. El cociente de dispersión a primarios depende de la geometría del sistema, tamaño del objeto y campo de visión. Los artefactos producidos por la radiación dispersada se muestran principalmente como sombras oscuras y manchas como se indica en la Fig. 2.5b. Estas son el resultado de una señal incrementada en el detector debido al aporte de la radiación dispersada y por tanto causan una subestimación del coeficiente de atenuación. Debido a la radiación dispersada también se puede encontrar artefactos de copa o manchas brillantes [3].

Los artefactos de *endurecimiento del haz* resultan debido a los cambios que experimenta el espectro de rayos X a medida que se transmite por las estructuras del objeto que está formando la imagen. Por ejemplo, cuando un haz atraviesa trayectorias de gran longitud en un objeto, existe un desplazamiento, en su espectro, hacia una energía media mayor (endurecimiento). Entonces se mide un coeficiente de atenuación menor al valor real y por tanto se observan estructuras más oscuras en la reconstrucción, como se muestra en la Fig.

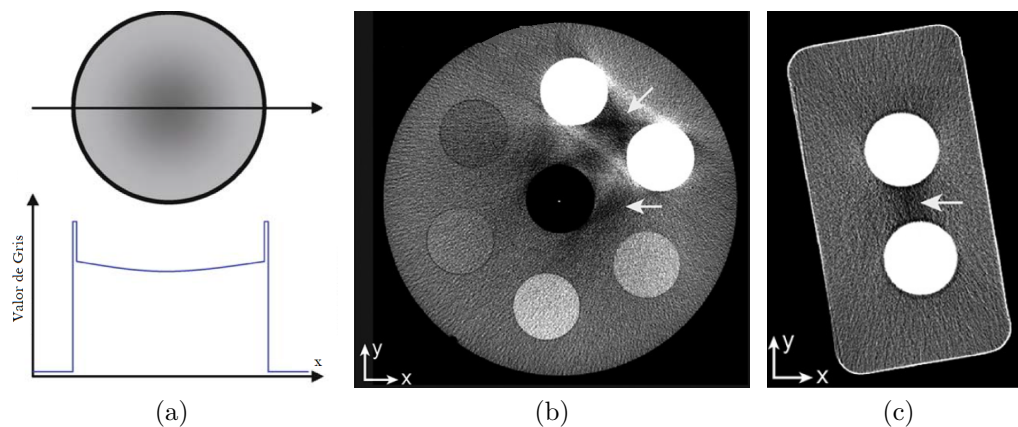


Figura 2.5: Diferentes artefactos en imágenes CT. (a) Efecto Copa, (b) artefactos de radiación dispersada, (c) endurecimiento del haz. Imágenes tomadas de [1] y [3].

2.5b. Estos artefactos tienen la forma de un sombreado a lo largo del centro de objetos grandes, brillo en la periferia del objeto y manchas en estructuras altamente atenuantes en el interior del objeto. Debido a esta forma, son difíciles de distinguir de los artefactos producidos por la radiación dispersada. Los métodos para mitigar los efectos del endurecimiento del haz incluyen aproximaciones físicas y mediante algoritmos; entre ellos destacan el uso de un filtro *tipo moño*, de filtros adicionales de cobre, aluminio y/o estaño para pre-endurecer el haz y la calibración de la respuesta en energía mediante fantasmas [3].

2.2.4. Dispersión incoherente de los fotones: Dispersión Compton

La dispersión de fotones puede ser clasificada como coherente e incoherente. Se conoce como dispersión coherente al fenómeno en el cual la longitud de onda de la luz dispersada es la misma que la de la luz incidente, como ocurre en el caso de la dispersión Rayleigh. Mientras que la dispersión incoherente es el fenómeno en el que la luz dispersada contiene longitudes de onda diferentes a las de la luz incidente, como es el caso de la dispersión Compton. La dispersión Rayleigh es un proceso electrodinámico clásico en el que no existe transferencia de energía al medio, mientras que la dispersión Compton es una interacción electrodinámica cuántica que resulta en una transferencia de energía a los electrones del medio con el que interactúa [34].

Se conoce como Efecto Compton a la interacción entre un fotón incidente, con longitud de onda λ , y un electrón en órbitas superficiales de los átomos del medio sobre el que incide el fotón. Debido a que en estas órbitas el electrón tiene una energía cinética muy baja (en comparación con su masa en reposo) se lo considera en reposo para simplificar el modelo.

Como resultado de esta interacción, el fotón incidente entrega energía al electrón del medio, el cual sale expulsado de su órbita y se produce un fotón con una longitud de onda λ' , la cual es mayor que la incidente. De la conservación de momento y energía se deducen las ecuaciones (2.8) y (2.9), las cuales indican la relación entre la longitud de onda y energía de los fotones incidente y dispersado.

$$\lambda' - \lambda = \frac{h}{m_0 c} (1 - \cos \theta) \quad (2.8)$$

$$E_{\gamma'} = \frac{E_{\gamma}}{1 + \frac{E_{\gamma}}{m_0 c^2} (1 - \cos \theta)} \quad (2.9)$$

Donde h es la constante de Planck, m_0 es la masa en reposo del electrón, c es la velocidad de la luz, θ es el ángulo formado entre la dirección en la que se dispersó el fotón y la dirección de incidencia y E_{γ} y $E_{\gamma'}$ son las energías de los fotones incidente y dispersado, respectivamente.

La radiación puede interactuar con la materia mediante diferentes procesos como Dispersión Rayleigh, Efecto Fotoeléctrico, Efecto Compton, Producción de pares y Fotodesintegración [35]. Cada uno de estos procesos tiene una probabilidad de ocurrir, la cual depende de la energía de la radiación incidente y de características como el número atómico y masa atómica del material sobre el que incide. En procesos como la producción de pares, existe umbrales de energía en la radiación incidente, a partir de los cuales puede ocurrir este proceso. En el caso del Efecto Compton, su probabilidad está descrita por la sección eficaz de Klein-Nishina.

La sección eficaz diferencial de Klein-Nishina, $\frac{d\sigma_{KN}}{d\Omega}$, mide la probabilidad de que ocurra un evento de dispersión Compton entre un fotón y un electrón, describe la distribución angular de los fotones dispersados y se calcula mediante la ec. (2.10) [34].

$$\frac{d\sigma_{KN}}{d\Omega} = \frac{1}{2} \left(\frac{1}{4\pi\epsilon_0} \frac{e^2}{m_0 c^2} \right)^2 \left(\frac{\lambda}{\lambda'} \right)^2 \left[\frac{\lambda}{\lambda'} + \frac{\lambda'}{\lambda} - \sin^2 \theta \right] \quad (2.10)$$

Donde ϵ_0 es la permitividad eléctrica del vacío y e es la carga elemental. La fórmula de la ec. (2.10) fue derivada en 1928 por Oskar Klein y Yoshio Nishina [36] a partir de la ecuación de Dirac, siendo calculada al orden más bajo en electrodinámica cuántica.

Esta sección eficaz es particularmente importante en dosimetría de radiaciones debido a que los procedimientos de imagenología médica con rayos X ocurren a energías de entre 80 y 150 keV. A estas energías, el Efecto Compton es el proceso más probable de interacción radiación-materia [37]. Y por lo tanto, la sección eficaz de Klein-Nishina puede ser empleada para estimar la radiación absorbida por el paciente durante procedimientos de imagenología

con rayos X.

La dispersión Compton es un proceso fundamental en física de radiaciones, que es crucial para comprender las interacciones entre la radiación ionizante y los tejidos biológicos. Además, este fenómeno es responsable de la producción de radiación dispersada, la cual puede afectar significativamente las dosis que reciben los pacientes durante procedimientos de imagenología con radiaciones ionizantes y puede contribuir al ruido en las imágenes médicas, degradando la calidad de éstas.

2.2.5. Radiación dispersada

El principio fundamental para obtener una imagen radiográfica o tomográfica consiste en determinar la atenuación que producen las diferentes estructuras anatómicas de un paciente. Se conoce como haz primario a los fotones que inciden sobre el paciente. De estos fotones, algunos son absorbidos por efecto fotoeléctrico y otros son dispersados por efecto Compton o dispersión Rayleigh. Para formar la imagen médica se tiene un detector de radiación opuesto al haz primario, detrás del paciente. Los fotones que llegan al detector son los que aportan a la formación de la imagen. El número de fotones de rayos X que atraviesan el paciente experimenta una atenuación exponencial, de acuerdo con la Ley de Lambert-Beer. El nivel de atenuación depende del coeficiente másico de atenuación de las diferentes estructuras anatómicas. Los fotones del haz primario que han sido dispersados en el paciente se conocen como radiación dispersada. Parte de esta radiación dispersada también llega a los detectores desde posiciones arbitrarias del paciente, afectando al proceso de formación de imagen. En particular, la radiación dispersada degrada el contraste y añade ruido de fondo.

En el proceso de formación de la imagen, el elemento (o píxel) del detector que se encuentra ubicado en la trayectoria directa del haz primario no es capaz de diferenciar el mecanismo físico de las interacciones por las cuales se redujo la intensidad. De hecho, otros elementos detectores ubicados fuera de la trayectoria del haz primario pueden ser afectados por interacciones de dispersión. Estos efectos son más marcados en objetos anatómicos altamente atenuantes como hombros, abdomen y pelvis. Esta radiación dispersada aporta inconsistencias en las proyecciones, las cuales al ser retroproyectadas se traducen en artefactos en forma de manchas en la imagen [38]. Se conoce también que al tener una mayor cantidad de detectores, existe un mayor aporte de la radiación dispersada debido a que esta sale en distintos ángulos y por tanto puede ser detectada en más regiones; en particular, en CBCT su aporte es considerable. Para rayos X empleados en imagenología diagnóstica (de hasta 140 keV), la dispersión coherente e incoherente son de los principales mecanismos de interacción en tejido suave. En CBCT, la radiación dispersada causa artefactos de copa y reduce la precisión en

los números CT [3].

2.2.6. Algoritmos de corrección por radiación dispersada

Para corregir imágenes por radiación dispersada existen métodos de supresión y de estimación de radiación dispersada. Los métodos de supresión de radiación dispersada incluyen el uso de rejillas anti-dispersoras, incremento de la distancia objeto-detector y disminución del campo de vista [39]. Todos estos métodos tienen por objetivo evitar que la radiación dispersada llegue a los detectores; sin embargo, estos métodos no la eliminan completamente. Por tal motivo, los algoritmos de estimación de radiación dispersada son esenciales para ejecutar estas correcciones de una manera más eficaz [40].

Los métodos de estimación de la radiación dispersada se basan en mediciones, núcleos o *kernels* de dispersión, simulaciones MC, modelos analíticos y algoritmos de aprendizaje profundo. Las estimaciones de la dispersión pueden ser realizadas en el dominio de las proyecciones o de las imágenes CBCT. En general, los métodos aplicados sobre el dominio de las proyecciones son más rápidos pero contienen información limitada acerca de la forma y composición del objeto. Por otro lado, los métodos aplicados directamente en el dominio de la imagen CBCT son más precisos debido a que se conoce la forma tridimensional y composición del objeto, lo cual mejora la estimación de la dispersión, pero el costo computacional es más elevado [3].

Las correcciones basadas en mediciones usan arreglos bidimensionales *enrejados* de plomo para detener el haz primario en determinadas regiones. Este arreglo es ubicado entre la fuente y el objeto del cual se obtiene la imagen. Las señales de la imagen detectadas detrás de las porciones sólidas del arreglo de detención del haz primario son empleadas para estimar la radiación dispersada. Mediante interpolación se determina la componente de la radiación dispersada en las demás regiones. Este método requiere mediciones con y sin el arreglo de bloqueo del haz. Sin embargo, un doble conjunto de proyecciones implica duplicar la dosis impartida sobre el paciente e incrementar el tiempo de adquisición de la imagen [41].

Los métodos basados en *kernels* de dispersión fueron las primeras técnicas, basadas en software, implementadas para corregir los efectos de la radiación dispersada. Este método, al aplicarse en el espacio de las proyecciones, es computacionalmente eficiente por lo que permite realizar correcciones en tiempo real. Un *kernel* de dispersión es una función de transferencia que relaciona la fluencia primaria en un punto del objeto con la correspondiente intensidad de dispersión en el plano del detector. En el caso ideal, para determinar completamente la distribución de la radiación dispersada, basta con conocer la fluencia primaria de rayos X y el *kernel* de dispersión para cada punto del volumen del objeto. Los *kernels* de dispersión se

obtienen de simulaciones MC o de métodos experimentales [3].

Las simulaciones MC de transporte de la radiación permiten estudiar los procesos físicos y parámetros asociados a la radiación dispersada. Con estos métodos, es posible identificar la naturaleza (primarios o dispersados) de los fotones que aportan a la formación de la imagen. En un trabajo de tesis previo [13], se implementó un método de corrección iterativo basado en simulaciones MC y se redujeron los errores en la estimación de los números CT [13]. Sin embargo, debido a su elevado costo computacional, estos cálculos no pueden ser ejecutados dentro de tiempos clínicamente aceptables.

También se han desarrollado *métodos analíticos* para correcciones por radiación dispersada, basados en las secciones eficaces diferenciales de Klein–Nishina y Rayleigh. Estos son más rápidos que las simulaciones MC y permiten cálculos precisos a primer orden de las señales debido a dispersiones incoherentes y coherentes en el dominio de las proyecciones. Sin embargo, su aplicación no considera las dispersiones múltiples [41].

Recientemente, se han propuesto alternativas interesantes capaces de combinar la precisión de los métodos MC con una eficiencia computacional apropiada para aplicaciones clínicas. Estos métodos emplean algoritmos de aprendizaje profundo con *redes neuronales* entrenadas en la detección del aporte de la radiación dispersada en imágenes médicas. Los conjuntos de entrenamiento consisten en pares de imágenes con y sin aporte de la radiación dispersada, obtenidas mediante simulaciones MC. La precisión de estos métodos está limitada por la precisión de los conjuntos de datos empleados para entrenar las redes neuronales. Se ha reportado que estos métodos han mostrado mejores resultados que los basados en *kernels* de dispersión [42–44].

2.3. Inteligencia Artificial

La Inteligencia Artificial (AI) es un campo de estudio que se dedica al desarrollo de modelos computacionales con la habilidad de interpretar correctamente datos externos, aprender de dichos datos y emplear su aprendizaje para cumplir tareas que normalmente requieren inteligencia humana [7]. El término “inteligencia” es aplicado debido a que la máquina imita funciones cognitivas asociadas con la mente, como el aprendizaje, resolución de problemas, toma de decisiones y optimizaciones [9]. El término de AI fue empleado por primera vez por un grupo de científicos de computación en las conferencias de Dartmouth, en 1956. Hasta aproximadamente el año 2012 este tipo de técnicas estaban limitadas a compañías de tecnología, gobiernos y agencias de investigación; sin embargo, gracias al desarrollo y disponibilidad de unidades de procesamiento gráfico (GPUs, por sus siglas en inglés), que han permitido el de-

sarrollo de la computación paralela, la AI se ha vuelto más accesible, económica, rápida y poderosa. Los problemas que actualmente se resuelven con AI incluyen, reconocimiento de voz, juegos de estrategias de alto nivel como el ajedrez, vehículos autónomos, procesamiento de lenguaje natural, reconocimiento y corrección de imágenes, entre otros. Los algoritmos de AI se basan en ramas de la matemática, como la estadística, optimización, lógica, probabilidad, álgebra, cálculo multivariante, entre otras. Sus aplicaciones se expanden a la neurociencia, economía, psicología, medicina, radiología, física, entre otras.

El *aprendizaje de máquina* es un subcampo de la ciencia de computación que se encarga de generar modelos que den a las computadoras la habilidad para aprender a resolver tareas, sin ser explícitamente programadas para las mismas. Se explora el estudio y construcción de algoritmos que pueden aprender de los datos y hacer predicciones sobre los mismos. El aprendizaje de máquina es empleado en un amplio rango de tareas de computación donde diseñar y programar explícitamente un algoritmo para resolver dicho problema sería muy difícil o imposible. Como ejemplos de sus aplicaciones tenemos los filtros de correos electrónicos, visión de computadoras, tareas de clasificación, etc [45].

El aprendizaje de máquina puede ser supervisado o no-supervisado. El aprendizaje no-supervisado se encarga de establecer perfiles de comportamientos base para diferentes entidades y emplearlo para encontrar anomalías importantes [45]. Para el aprendizaje supervisado, el algoritmo es entrenado con un conjunto de datos y sus correspondientes rótulos, y aprende comparando su salida en cada iteración con los rótulos correctos y cuantificando su error. De esa manera, el algoritmo va modificando iterativamente su modelo de tal modo que mediante herramientas de cálculo multivariante es capaz de minimizar su error. Los patrones encontrados por el algoritmo son empleados posteriormente para predecir los rótulos de un conjunto adicional de datos no rotuladas. Por ejemplo, se puede entrenar un algoritmo de aprendizaje de máquina con un conjunto de imágenes rotuladas como “perro”, “gato” y “ave” y luego ingresar una imagen de alguno de estos animales y el algoritmo será capaz de identificar el rótulo correcto. El conjunto de datos con el que inicialmente se entrena el algoritmo se conoce como *conjunto de entrenamiento*, mientras que el conjunto de datos con el que se verifica la validez del modelo encontrado se llama conjunto de validación. En general, se acostumbra a dividir este conjunto en dos, uno llamado *conjunto de validación* y otro propiamente llamado *conjunto de evaluación*. El conjunto de validación se diferencia del de evaluación en que el primero es empleado para modificar hiperparámetros del algoritmo, de tal manera que se maximice la eficiencia del mismo, y por tanto sus resultados van a ser sesgados hacia este conjunto de datos y de ahí la necesidad de un último conjunto de “evaluación”, del cual el algoritmo no tenga ningún conocimiento previo [9].

2.3.1. Aplicaciones en la medicina

La AI ha sido ampliamente empleada en la medicina, desde algoritmos de aprendizaje hasta el desarrollo de dispositivos médicos [11]. Gracias a los avances en tecnologías médicas y biológicas ha sido posible obtener enormes volúmenes de imágenes médicas, electroencefalogramas, genomas y secuenciamiento de proteínas. Múltiples investigadores han desarrollado algoritmos de procesamiento de imágenes y visión de computadoras para permitir diagnósticos más rápidos, mejorando la visualización de patologías y alertando en situaciones de emergencia [12]. Es más, en la actualidad, múltiples algoritmos de aprendizaje profundo constituyen la tecnología de punta en análisis de imágenes médicas [46]. Por tal motivo, los algoritmos de aprendizaje profundo muestran un enorme potencial en la extracción de características y aprendizaje de patrones a partir de datos complejos, como lo son los asociados a la salud humana.

2.3.2. Aprendizaje profundo

El aprendizaje profundo o *deep learning* es un campo del aprendizaje de máquina que se encarga de buscar modelos a partir de la abstracción de datos a gran escala. Ha sido aplicado en una amplia variedad de campos, incluyendo reconocimiento automático de voz, reconocimiento de imágenes, procesamiento de lenguaje natural y bioinformática. En el campo de la medicina, los algoritmos de *deep learning* han demostrado ser una herramienta efectiva para el manejo, almacenamiento, análisis e interpretación de datos biomédicos [10]. Este tipo de aprendizaje se conoce como aprendizaje “profundo” debido a la profundidad de las redes neuronales utilizadas, lo cual se refiere a que estas redes tienen múltiples capas ocultas entre la capa de entrada y la capa de salida.

Los algoritmos de *deep learning* se enfocan en el aprendizaje de *representaciones* de datos. Por ejemplo, una imagen puede representarse como un arreglo numérico, con las intensidades por píxel en cada canal. Sin embargo, también puede ser representada de manera más abstracta como un conjunto de bordes o regiones de una forma particular. Algunas de estas *representaciones* son ideales para simplificar ciertas tareas de aprendizaje. Por ejemplo, los bordes de una imagen son de gran utilidad para el reconocimiento facial. El potencial del *deep learning* se encuentra en su capacidad para extraer de manera jerárquica este tipo de características y aprender de ellas. Investigaciones en *deep learning* buscan conseguir *representaciones* inspiradas de avances en neurociencia, procesamiento de información y patrones de comunicación en el sistema nervioso. De acuerdo con el enfoque en ciertas *representaciones*, se pueden desarrollar diferentes *arquitecturas* de *deep learning*. Se conoce como arquitectura

a un diseño específico de una red neuronal artificial, su estructura y organización de capas. Las principales arquitecturas de *deep learning* son las redes neuronales profundas, redes neuronales convolucionales (CNN), redes neuronales recurrentes, redes generatrices adversariales (GANs, por sus siglas en inglés) y redes neuronales residuales (ResNets) [9].

2.3.3. Redes neuronales convolucionales

Una CNN es un tipo de arquitectura de *deep learning* inspirada en procesos biológicos. En la corteza visual de los animales, las neuronas individuales responden a estímulos en una región restringida del espacio, conocida como campo receptivo. El campo receptivo de diferentes neuronas se superpone parcialmente, de tal manera que forman el campo visual. La respuesta individual de una neurona a un estímulo dentro de su campo receptivo puede ser modelada matemáticamente mediante una operación de *convolución* [9]. Las CNN se caracterizan por agrupar una secuencia de operaciones de convolución seguidas por una función de activación y luego una operación de agrupación o *pooling*. La convolución es aplicada a la imagen de entrada para *filtrar* la información y de esta manera ir adquiriendo un *mapa de características*. Las capas de *pooling* mejoran la estabilidad de la CNN, disminuyendo la propagación de fluctuaciones durante el proceso de extracción del mapa de características [47].

Operaciones de convolución, *pooling* y *un-pooling*

La *convolución* es una operación matemática que es caracterizada por un filtro y un término de desviación o *bias*. El filtro es una matriz de dimensiones $f_x \times f_y$ y el *bias* es un escalar, b . La convolución de una imagen I , de $n_x \times n_y$ píxeles, donde $n_i > f_i$, con $i \in \{x, y\}$, da como resultado una imagen I' , de $m_x \times m_y$ píxeles. Para las operaciones de convolución en *deep learning*, los filtros usualmente son cuadrados ($f \times f$) con f impar. De esta manera, para determinar el valor del píxel de la imagen I' en la posición (i, j) se ubica el centro del filtro en la posición (i, j) de la imagen I y se realiza una multiplicación elemento a elemento entre los elementos del filtro y los elementos de la imagen I sobre los que algún elemento del filtro se superpone y se suman esos valores y el valor de *bias*. Sin embargo, de esta manera no se puede calcular los valores de los píxeles en los bordes de la imagen debido a que si centramos el filtro en un borde de la imagen, algunos de sus elementos no se pueden superponer con ningún píxel de la imagen de entrada. Esto se puede resolver omitiendo los píxeles del borde o usando estrategias como *relleno* o *padding*. Cuando se omiten los bordes de la imagen, la relación entre el tamaño de la imagen de entrada y la de salida viene determinada por las dimensiones del filtro y se cumple que $m_i = n_i - f_i + 1$. La estrategia de *padding* consiste en agregar p_x columnas de píxeles con valor cero, distribuidas simétricamente al extremo derecho e izquierdo de la imagen de entrada y p_y filas de ceros, distribuidas simétricamente

arriba y abajo de la imagen. Para poder aplicar el filtro en todos los bordes de la imagen original I , se debe cumplir $p_i = f_i - 1$; de esta manera, cuando se aplica *padding*, se tiene que $m_i = n_i$. Así, mediante una convolución se obtiene una *mezcla* entre la información del filtro y de la imagen de entrada. Al aplicar múltiples filtros de convolución a una imagen bidimensional se obtiene un tensor tridimensional [47].

Las operaciones de *pooling* generalmente son caracterizadas por un filtro de 2×2 que se encarga de *comprimir* la información de cada cuadrado de cuatro píxeles en un solo píxel, disminuyendo el tamaño de la matriz de entrada a la mitad en cada dimensión. Estas operaciones pueden tomar el promedio (*average pooling*) o el máximo (*max pooling*) de los cuatro píxeles. Por el contrario, las operaciones de *un-pooling* o *upsampling* se encargan de *expandir* la imagen, usualmente en un factor de 2; usando estrategias como la repetición de los elementos de la matriz de entrada [48].

En la Fig. 2.6a se ejemplifica de convolución entre una *imagen* de 4×4 píxeles (celeste), usando la estrategia de *padding* (gris) con un filtro compuesto de una matriz de 3×3 (verde) y un factor de *bias* de 1. En la imagen se suma los productos entre los elementos en rosado de la matriz de entrada por los correspondientes elementos del filtro y a estos se les suma el factor de *bias* para obtener el primer elemento de la convolución (rojo). De la misma manera se recorre el filtro sobre la matriz de entrada para determinar el resto de elementos de la convolución. En la Fig. 2.6b Se ejemplifica el proceso de *max-pooling* sobre la imagen resultante del proceso de convolución de la parte a). En la Fig. 2.6c Se ejemplifica un proceso de *un-pooling* o *upsampling* usando una estrategia de repetición de los elementos de la matriz de entrada.

Función de activación

Una función de activación es una operación matemática aplicada a la salida de cada neurona de cada capa de una red neuronal. Su objetivo es agregar no linealidad a la red, permitiéndole modelar relaciones complejas en los datos de entrada. Algunos ejemplos de funciones de activación son la función sigmoide, tangente hiperbólica, Softmax, unidad rectificadora lineal (ReLU), softplus y la rectificación de valor absoluto. En la Fig. 2.7 se muestra la forma de las funciones de activación más comunes.

Función de pérdidas

El objetivo del proceso de entrenamiento es minimizar una *función de pérdidas*, la cual mide la discrepancia entre la salida de la red y el resultado esperado o *ground truth*. Esta salida depende de los parámetros del modelo en cada *epoch* o iteración. El MSE es una de las funciones de pérdidas más comúnmente empleadas; sin embargo, esta función tiende a

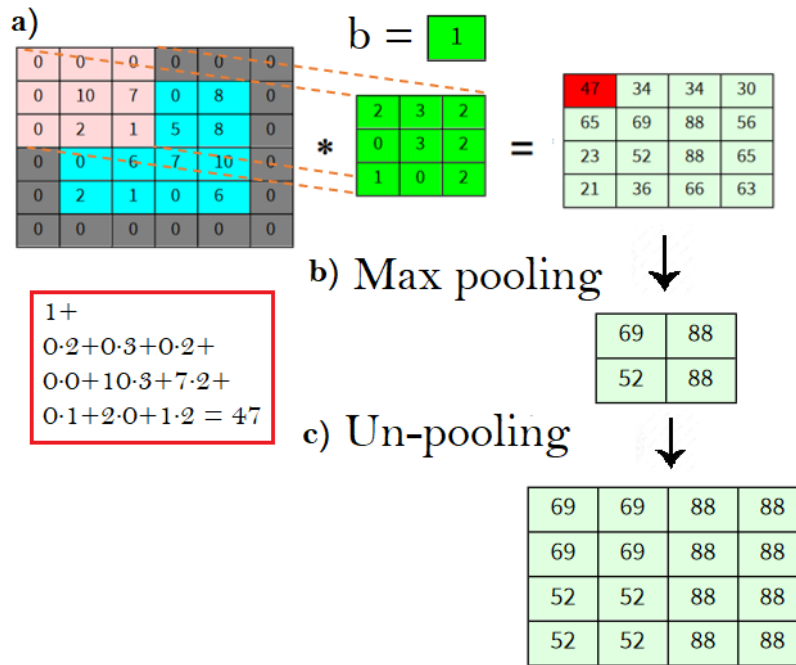


Figura 2.6: a) Ejemplo el proceso de convolución para una imagen de 4×4 píxeles con un filtro de 3×3 , operación en la que se ha incluido la estrategia de *padding*. b) Se muestra el ejemplo de la operación de *max pooling*. c) Ejemplo de la operación de *un-pooling* mediante la repetición de los elementos de la matriz de entrada.

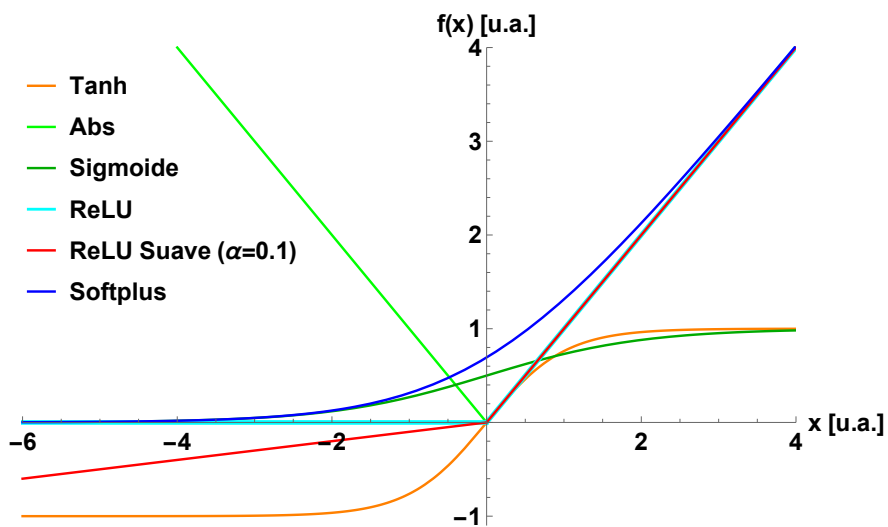


Figura 2.7: Algunas de las principales funciones de activación empleadas en *deep learning*.

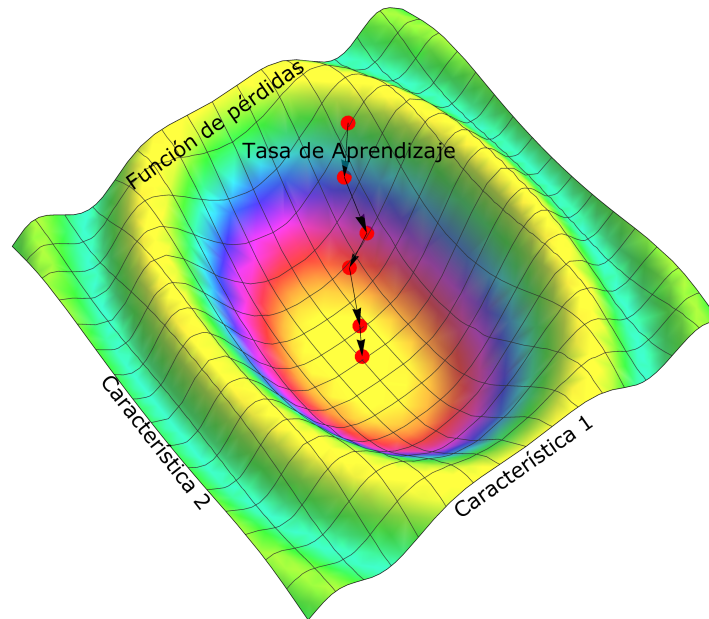


Figura 2.8: Proceso de minimización de una función de pérdidas usando una tasa de aprendizaje variable.

penalizar excesivamente los datos atípicos, lo cual disminuye las tasas de convergencia. En problemas de procesamiento de imágenes con CNN, su eficacia ha sido cuestionada como un buen indicador de calidad de imagen [49]. Por otra parte, usar el MAE como función de pérdidas penaliza a todos los errores en una escala lineal, por lo que no se amplifica la supresión de valores atípicos. En problemas de procesamiento de imágenes, esta métrica ha dado resultados superiores que con el MSE; sin embargo, recientemente se ha encontrado mejores resultados con funciones de pérdidas basadas en indicadores perceptuales, como el SSIM debido a que toma en cuenta la estructura de la entrada [49, 50]. Cuando la salida de un algoritmo de *deep learning* es otra imagen, se pueden emplear métricas como el PSNR o SSIM para determinar la fidelidad de la salida del algoritmo con los valores de referencia, *ground truth*.

Proceso de optimización y tasa de aprendizaje El *aprendizaje* en un algoritmo de *deep learning* consiste en un problema de optimización: alcanzar un mínimo en la función de pérdidas, mediante una apropiada selección de parámetros de los filtros de convolución. Primero, se calcula la salida del algoritmo para un conjunto o *batch* de imágenes y la función de pérdidas del modelo con el conjunto inicial de parámetros. Luego, se ajustan los valores de los parámetros desde la última capa hasta la primera capa, de tal manera que se reduzca

el valor de la función de pérdidas. Este ajuste se realiza de manera numérica, modificando los parámetros en un valor caracterizado por la *tasa de aprendizaje* en la dirección contraria al gradiente de la hipersuperficie formada por la función de pérdidas, en función de los parámetros del modelo. Este proceso se repite iterativamente hasta que se alcanza el modelo deseado. En la Fig. 2.8 se ilustra un proceso de minimización de una función de pérdidas usando una tasa de aprendizaje variable. Entre los principales métodos de optimización destacan el descenso de gradiente estocástico, RMSProp y el estimador adaptativo de momentos (ADAM). Este último es el optimizador más comúnmente empleado en *deep learning* [10, 51].

2.3.4. UNet

Una de las arquitecturas de CNN más ampliamente utilizadas es la UNet. Esta red fue propuesta por primera vez en 2015 por Ronneberger et al. [52] en aplicaciones de segmentación de imágenes biomédicas.

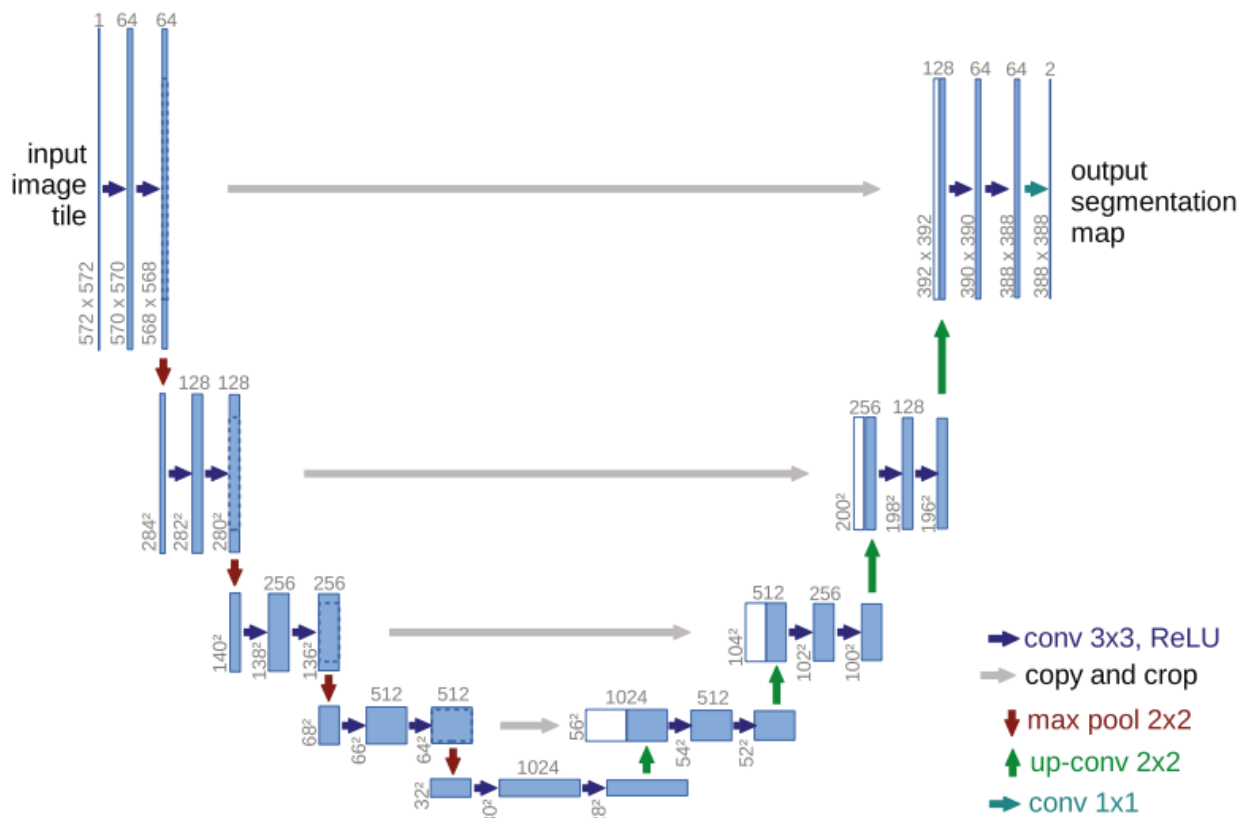


Figura 2.9: Arquitectura de la versión original de la UNet, implementada para segmentar imágenes biomédicas. En las descripciones se muestran las dimensiones de las imágenes y la cantidad de filtros empleados en cada etapa [52].

Esta red consiste en una combinación de un ‘camino’ de *contracción* y un ‘camino’ de *expansión*, los cuales le dan su característica forma de ‘U’. La contracción consiste en una serie de operaciones de convolución y *max pooling* mientras que la expansión consiste en un conjunto de operaciones de convolución y desagrupación o *un-pooling*. Además, a lo largo del camino de contracción, se almacenan los resultados para *concatenarlos* con los correspondientes resultados a lo largo del camino de expansión. La operación de *concatenación* consiste en *juntar contiguamente* dos tensores, en su dimensión de profundidad. De esta manera, el camino de contracción captura el contexto, reduciendo la información en las dimensiones espaciales pero incrementando la información en el *espacio de características* y el camino de expansión asegura una localización precisa al combinar la información espacial y de *características*.

Ronneberger et al. [52] mostraron que con muy pocas imágenes, un modelo entrenado con esta arquitectura puede superar a la *red neuronal de ventana deslizante* en la segmentación de estructuras neuronales en conjuntos de imágenes de microscopía electrónica, el cual era previamente el *estado-del-arte* en estas aplicaciones. En la Fig. 2.9 se presenta la arquitectura de la versión original de la UNet.

Posteriormente, múltiples investigaciones han demostrado resultados exitosos con el uso de UNets en problemas de supresión de ruido [53–56]. En comparación con otros métodos de supresión de ruido, los principales retos que han encontrado estas arquitecturas en aplicaciones de supresión de ruido son el costo computacional asociado a la cantidad de memoria requerida durante el procesamiento, los datos rotulados y el tiempo asociado al entrenamiento de la red [53, 57]. En comparación con otras arquitecturas de *deep learning*, las UNets son menos eficientes en presencia de ruido correlacionado con la señal [58] y la falta de generalizabilidad, requiriéndose modelos específicos para cada tarea de supresión de ruido [57].

2.3.5. Redes Generativas Adversariales

Las redes generativas adversariales, *GANs*, fueron propuestas en 2014 por Goodfellow et al. [59]. Esta estructura permite estimar modelos generativos mediante un proceso adversarial, en el cual se entrena simultáneamente dos modelos. El *generador* (G) que captura la distribución de los datos y el *discriminador* (D) que estima la probabilidad de que una muestra corresponda al conjunto de entrenamiento o sea generada por G. El generador es entrenado para maximizar la probabilidad de que el discriminador cometa errores [59]. Estas arquitecturas de *deep learning* permiten un *aprendizaje* de representaciones profundas sin necesidad de grandes conjuntos de datos de entrenamiento [60].

El proceso competitivo entre la red generadora y la red discriminadora puede describirse

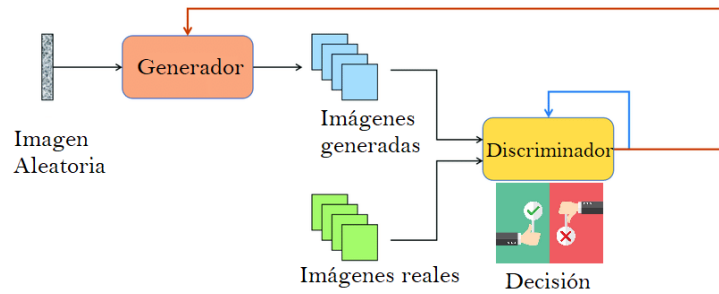


Figura 2.10: Esquema ilustrativo del proceso de entrenamiento adversarial en las GANs [64].

mediante la siguiente analogía. La red *generadora* puede verse como un *engañador* que crea imágenes *falsas* con el objetivo de hacerlas lucir realistas. La red *discriminadora* puede verse como un *detective* que recibe las imágenes *falsas* y *auténticas* e intenta distinguirlos. De este modo, durante el proceso de entrenamiento simultáneo, la red generadora aprende a producir imágenes cada vez *más realistas*, mediante un proceso iterativo, hasta que logra *engañar* a la red *discriminadora*. Esto significa que la red generadora no es entrenada para minimizar la distancia a una imagen específica sino que es entrenada para *engañar* al discriminador. Esto permite que se apliquen a modelos de entrenamiento no supervisado. En la Fig. 2.10 se muestra un esquema ilustrativo del proceso de entrenamiento adversarial en las GANs.

A pesar de que originalmente las GANs fueron propuestas como una forma de aprendizaje no supervisado, también se ha demostrado su utilidad para problemas de aprendizaje semisupervisado, supervisado, aprendizaje por refuerzo y supresión de ruido [61–63]. Las GANs han sido ampliamente utilizadas en múltiples aplicaciones incluyendo generación de arte, transferencia de estilo, colorización de imágenes, reconstrucción de objetos a partir de bordes, incremento de resolución en imágenes y problemas de clasificación [60, 62].

2.3.6. Transferencia de aprendizaje

La transferencia de aprendizaje o *transfer learning* es una poderosa técnica de Aprendizaje de Máquina en la que un modelo entrenado en una tarea es reutilizado como punto de partida para un modelo que se busca aplicar en una tarea relacionada.

Esta idea se basa en aprovechar el *conocimiento* adquirido por el modelo en la resolución de un problema determinado y aplicarlo en un problema diferente, pero relacionado. La principal utilidad de aplicar esta técnica es para tareas en las que se tiene una cantidad limitada de datos. Las primeras capas de una red se especializan en *aprender* características generales de las imágenes por lo que pueden ser re-utilizadas. Las últimas capas de una red se especializan

en *aprender* características específicas de las tareas por lo que pueden ser re-entrenadas y así, mediante un ajuste fino, especializarse en la nueva tarea [65].

Esta herramienta ha sido aplicada exitosamente en áreas como visión de computadoras, procesamiento de lenguaje natural y reconocimiento de voz [66].

2.3.7. Corrección de imágenes médicas

Los algoritmos de inteligencia artificial, en particular de *deep learning* han sido ampliamente empleados para realizar diferentes tareas sobre imágenes médicas. Se han usado de manera satisfactoria este tipo de algoritmos para tareas de clasificación, reconstrucción, detección, reconocimiento, segmentación y corrección de imágenes; diagnósticos apoyados por computadoras; y simulación de fenómenos físicos que afectan significativamente la calidad de las imágenes médicas, como es el caso de la radiación dispersada [46].

La radiación dispersada es uno de los principales limitantes de la calidad en una imagen tomográfica de CBCT, siendo responsable de artefactos de copa o manchas oscuras en las reconstrucciones CT, los cuales afectan la calidad de la imagen médica. Las estrategias convencionales para reducir su efecto incrementan la dosis sobre el paciente o se basan en algoritmos computacionalmente demandantes. De hecho, el método más eficaz para predecir la radiación dispersada es emplear simulaciones MC para sustraer la correspondiente componente dispersada de los datos crudos. Sin embargo, las simulaciones MC son demasiado lentas para ser empleadas rutinariamente en la clínica. En la literatura se han propuesto métodos de corrección basados en AI, específicamente en algoritmos de *deep learning* que emplean CNN. Desafortunadamente, para obtener datos reales con los cuales se pueda entrenar la red, se requeriría tener un método para separar las componentes de radiación primaria y radiación dispersada de un sistema de imagen de rayos X, lo cual es justamente el problema a resolver. Ante ello, la principal propuesta ha sido ejecutar simulaciones MC para generar los datos necesarios para entrenar el algoritmo [43,44]. A continuación se presenta los resultados reportados en algunas investigaciones en el área.

En la investigación de Lee et al. [44] simularon pares de imágenes CBCT de tórax. Una producida únicamente por la componente del haz primario (es decir, libre del aporte de la radiación dispersada) y otra imagen que incluye la componente de la radiación dispersada. En dicho trabajo, la componente dispersada fue calculada tomando la diferencia entre los pares y empleada como el objetivo para el entrenamiento de las CNN. En comparación con los métodos iterativos basados en MC, los métodos basados en *deep learning* mostraron mejores resultados. En dicho trabajo consiguieron una reducción de hasta un 58.5% en el MSE de

las proyecciones y un incremento de 18.1 % en el PSNR y de 3.4 % en el SSIM.

Para corregir la radiación dispersada en tiempo real, Maier et al. [43], desarrollaron un método de corrección basado en CNN entrenadas para reproducir la salida de algoritmos MC a partir de tener proyecciones reales como datos de entrada. Se entrenó una variación de la arquitectura UNet por 80 épocas, usando un optimizador de ADAM y con un tamaño de lotes de 16. La función de pérdidas empleada fue el MSE entre la salida de la red y la componente dispersada obtenida de la simulación MC. Para mejorar el rendimiento computacional durante el entrenamiento de la red se disminuyó el tamaño de las imágenes de entrada a 256×256 . Esta reducción en el tamaño de las imágenes no representó una disminución significativa en el rendimiento del algoritmo debido a que se conoce que la radiación dispersada tiene componentes de bajas frecuencias espaciales. Se reportó que este método es más preciso que las estimaciones basadas en *kernels* y en métodos híbridos. Este algoritmo fue empleado tanto con imágenes simuladas como con datos medidos y se encontró que en ambos casos, las estimaciones de la componente dispersada difieren en menos de 3 % que las predicciones MC. El tiempo que le toma al algoritmo realizar la corrección a cada proyección es de 20 *ms*.

Nomura et al. [42] propusieron un método de corrección de radiación por dispersión en imágenes CBCT mediante el uso de CNN con una arquitectura UNet de 25 capas, con el optimizador de ADAM. En su investigación se entrenó el modelo con 1800 pares de proyecciones de rayos *X* y sus correspondientes distribuciones de radiación dispersada. Las imágenes empleadas fueron de fantasmas no antropomórficos, generados mediante simulaciones MC de un escáner CBCT de un sistema de protonterapia. Para el entrenamiento de las CNN implementaron dos funciones de pérdidas para evaluar el impacto de estas en los resultados. Una basada en el MSE y otra basada en el MAE. Se encontró mejores resultados cuando se empleó la función de pérdida basada en el MAE. El entrenamiento se realizó para 100 épocas, usando un optimizador de ADAM y con un tamaño de lotes de 10. Con el objetivo de realizar un *aumento de los datos*, rotaciones e inversiones de las imágenes fueron aplicadas al conjunto de datos de entrenamiento, incrementando los datos de entrenamiento a 14400 pares, cuyo entrenamiento tomó 10 *h* en una computadora de procesador core i7 y con GPU NVIDIA GTX 1070. En la validación se usaron 200 proyecciones digitales de un fantoma de cabeza. Las reconstrucciones de las proyecciones corregidas mediante las CNN mostraron una mejor cuantificación de HU, especialmente en regiones cercanas a las interfases de aire o hueso. La calidad de las imágenes corregidas mediante CNN fue superior a la de las imágenes corregidas con un método rápido basado en superposición de kernels de dispersión (fAKS, por sus siglas en inglés) de acuerdo con varios índices de calidad de imagen, incluyendo el MAE, el MSE, la PSNR y SSIM. Se encontró que el método basado en CNN es computacionalmente

eficiente dado que empleando un GPU le toma 5 s en corregir 360 proyecciones. También evaluaron de manera satisfactoria la técnica de *transferencia de aprendizaje*; para ello, el modelo original, que fue entrenado con proyecciones de abanico completo fue ajustado para correcciones de escaneos de medio abanico, empleando únicamente 360 pares de proyecciones de medio abanico [42].

Capítulo 3

Materiales y Métodos

3.1. Adquisición de datos simulados

El objetivo central de este trabajo de investigación es el entrenamiento de un algoritmo de *deep learning* que sea capaz de identificar la componente de radiación dispersada en imágenes de tomografía de haz de cono (CBCT). Para ello, lo primero que se necesita es adquirir un conjunto de datos lo suficientemente grande para entrenar, validar y evaluar el modelo a desarrollar. Este conjunto de datos corresponde a pares de proyecciones de imágenes CBCT, donde una de las imágenes tenga solo la contribución de la radiación dispersada y la otra la contribución total de los fotones (primarios+dispersados). Sin embargo, esta discriminación en el origen de los fotones que forman la imagen no es posible realizarla en adquisiciones reales de tomografías de haz de cono. Afortunadamente, en simulaciones Monte Carlo (MC) sí es posible rastrear el origen de los fotones que forman la imagen y separar la componente primaria y la de la radiación dispersada.

3.1.1. Simulaciones Monte Carlo de transporte de la radiación: EGSnrc

En múltiples trabajos de investigación en los que se han analizado los efectos de la radiación dispersada mediante algoritmos de inteligencia artificial se han empleado simulaciones MC para la generación de conjuntos sintéticos de datos con los que se puede entrenar el modelo [\[42-44\]](#).

La herramienta computacional *Cascada de electrones y rayos gamma* o *Electron Gamma Shower* (EGS) es uno de los principales paquetes de simulaciones MC de transporte de

radiación en materia. Consiste en una colección de rutinas genéricas que permiten modelar las interacciones de fotones, electrones y positrones con la materia en diferentes geometrías, para partículas con energías desde unos pocos keV hasta centenas de GeV. Originalmente fue desarrollado por Centro de Aceleradores Lineales de Stanford (SLAC, por sus siglas en inglés) [67] y desde sus comienzos, en la década de 1980, ha recibido significativos aportes en el desarrollo del código, por parte del Consejo Nacional de Investigación de Canadá (NRC, por sus siglas en inglés) y de la Organización de Investigación en Aceleradores de Alta Energía. La versión del NRC es conocida como *EGSnrc*.

EGSnrc es un código de acceso libre que puede ser instalado en Linux, macOS y Windows. Fue publicado en el año 2000, como una revisión completa de la versión original de EGS. Esta versión incorpora refinaciones cruciales en el transporte de partículas cargadas, mejores valores para las secciones eficaces a bajas energías y una librería para la elaboración de geometrías complicadas y fuentes de partículas. Ha sido ampliamente empleado en múltiples trabajos de investigación en el área de física médica, incluyendo investigaciones en desarrollo de dispositivos de detección de radiación, imagenología de pacientes con rayos X, estimación de la dosis entregada a tumores y tejido sano en procedimientos de radioterapia y simulación de aceleradores lineales (Linacs) [68].

Dentro de las múltiples aplicaciones de *EGSnrc*, la que es de interés para el presente proyecto de tesis es *egs.cbct*. Esta aplicación es parte de la librería *egs++* que ha sido escrita en *C++* y permite simular la geometría de una CBCT. El principal objetivo de esta aplicación es la estimación de la contribución de radiación dispersada en un detector ideal, bajo condiciones de CBCT. Utiliza técnicas sofisticadas de reducción de varianza y un algoritmo de suavizado para mayor eficiencia en los cálculos MC. Permite estimar la señal total y sus componentes (primaria y dispersada) en un detector ideal. También se pueden modelar las configuraciones de escáneres CT convencionales. Los escaneos CBCT se obtienen mediante el registro de las contribuciones de los fotones (kerma en aire) en un detector plano ideal, luego de atravesar el maniquí en un ángulo determinado de su órbita circular [68]. Cada iteración produce las proyecciones a un único ángulo y mediante *scripts* se puede extraer de manera simultánea un conjunto completo de proyecciones de una adquisición *CBCT*. En estas adquisiciones se extrae de manera separada las imágenes formadas por las contribuciones de los fotones primarios y de la radiación dispersada. En estas simulaciones se pueden emplear tanto *maniquíes* geométricos como voxelizados.

En este proyecto se trabajó con una versión modificada y validada de *egs.cbct* [69]. En esta versión se simuló un Linac Varian TrueBeam STx, equipo que está diseñado para tratamientos de radioterapia no convencionales como IMRT, VMAT y radiocirugía. Este linac es de

relevancia debido a que incorpora dos sistemas de formación de imagen, un sistema EPID y un sistema OBI v1.5. El sistema EPID es un sistema de formación de imágenes con detector *PortalVision as1200*, que genera imágenes planas para la reconstrucción tomográfica con geometría de haz de cono, formadas con un haz de la misma geometría del haz de terapia. Este sistema puede ser usado con haces de fotones de 6 MV para verificaciones dosimétricas de control de calidad y de 2.5 MV sin filtro de aplanado (FFF, por sus siglas en inglés) para imágenes de verificación usadas en radioterapia guiada por imagen (IGRT). El Sistema OBI v1.5 con el que cuenta el acelerador consta de un tubo de rayos X modelo *Varian GS 1542*, el cual produce un haz de rayos X en forma de cono, perpendicular al haz principal y trabaja a una diferencia de potencial entre 40 y 140 kV. Este sistema puede producir proyecciones 2D e imágenes CBCT, las cuales pueden ser adquiridas en haz completo (FF, por sus siglas en inglés) cuando se hace uso del filtro de moño completo y medio haz (HF, por sus siglas en inglés) cuando se utiliza medio filtro. La primera modalidad es empleada para imágenes de cabeza y cuello y la última en protocolos de imagen de tórax, abdomen y pelvis [13, 70].

3.2. Maniqués utilizados

El conjunto de datos empleado para el entrenamiento de los algoritmos fue adquirido de un maniquí geométrico y de un voxelizado. El maniquí geométrico utilizado es un Maniquí de Unidades Hounsfield que será llamado *HUP*, con imágenes adquiridas a 125 kV y a 2.5 MV. El maniquí voxelizado es un maniquí anatómico de una sección del tórax de ratón, llamado *MOBY* [71]; con imágenes adquiridas a 125 kV. El conjunto de datos de cada maniquí consiste en 360 pares de imágenes, originados de las proyecciones tomadas en una órbita completa, en pasos de 1° . Cada par de imágenes consiste en un proyección generada solo con aporte de la radiación dispersada y otra generada con el aporte de la radiación primaria y dispersada.

3.2.1. Maniquí de Unidades Hounsfield (HUP)

El maniquí aquí denominado HUP es un maniquí geométrico digital, basado en el modelo 062M del maniquí de densidad electrónica de *CIRS* [72]. El maniquí HUP consiste en un cilindro de 18 cm de diámetro y 10 cm de alto y contiene cuatro insertos cilíndricos de 3 cm de diámetro. El cilindro está hecho de agua sólida, con una densidad de 1.00 g/cm^3 . Los insertos son de hueso cortical con una densidad de 2.150 g/cm^3 , tejido adiposo con una densidad de 0.960 g/cm^3 , hueso trabecular con una densidad de 1.161 g/cm^3 y pulmón (en exhalación) con una densidad de 0.510 g/cm^3 [13, 72]. En la Fig. 3.1 se muestra un esquema de este fantoma. La composición empleada para simular los insertos fue proporcionada por el fabricante [72].

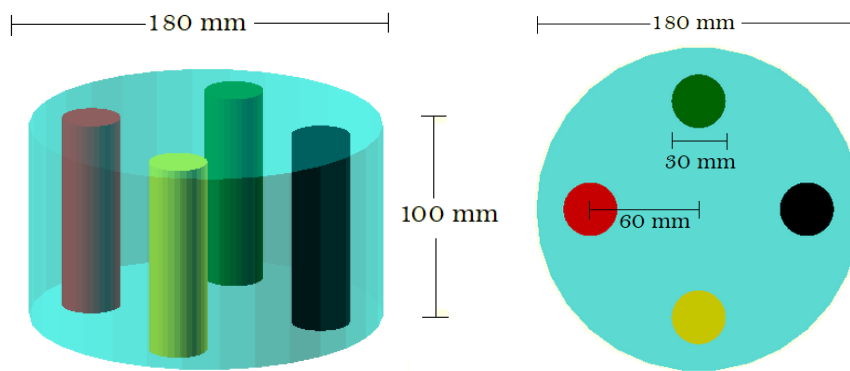


Figura 3.1: Esquema y dimensiones del maniquí de Unidades Hounsfield. El material del cilindro es agua sólida y los insertos son de hueso cortical (amarillo), tejido adiposo (rojo), hueso trabecular (verde) y pulmón en exhalación (negro). El diseño fue realizado con OpenSCAD [73].

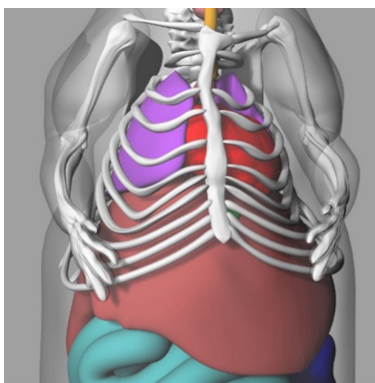


Figura 3.2: Estructura esquemática de MOBY, el maniquí digital voxelizado del tórax de un ratón, diseñado por el Centro de Ensayos de Imágenes Virtuales de la Escuela de Medicina de la Universidad de Duke [71].

3.2.2. Maniquí MOBY

El maniquí denominado MOBY es un maniquí voxelizado digital de una sección del tórax de un ratón. Este maniquí es de aproximadamente 3 cm de diámetro, siendo más pequeño en su porción superior. Está constituido de hueso cortical, hueso trabecular, tejido adiposo y pulmón. Este maniquí fue desarrollado por el Centro de Ensayos de Imágenes Virtuales de la Escuela de Medicina de la Universidad de Duke [71]. En la Fig. 3.2 se muestra un esquema de este fantoma.

3.3. Pre-procesamiento de datos

El pre-procesamiento de datos es una parte esencial en el desarrollo de algoritmos de inteligencia artificial. Tiene por objetivo preparar los datos de tal manera que se resalte las

características más relevantes, previo al entrenamiento [74]. A continuación se detallan las técnicas empleadas en esta investigación.

Tamaño de las imágenes.- Para poder entrenar en un mismo algoritmo con imágenes provenientes de diversos maniqués se aseguró que todas ellas sean del mismo tamaño, 192×256 píxeles. En el caso de las proyecciones del maniquí *MOBY*, se requirió recortar las imágenes para que cumplan esta condición. Esto se realizó teniendo en cuenta que se conserve el centro de la imagen y verificando que las partes recortadas no tengan aporte de radiación dispersada. Esto debido a que si se cambia el centro de imagen, la reconstrucción tomográfica sería errónea. Para asegurar conservar el centro de la imagen, se recortaron las imágenes de manera simétrica a cada lado de la misma. La relevancia de conservar el centro de la imagen se debe a que esta información es esencial para el proceso de reconstrucción de imágenes.

Píxeles con valores atípicos.- Debido a las fluctuaciones estadísticas y procesos inherentes al algoritmo de MC, en algunas imágenes se detectó la presencia de píxeles con valores atípicos. Para evitar que estos píxeles afecten al entrenamiento del modelo de IA, se identificó estos píxeles al graficar un histograma de 256 *bins* con ImageJ. Se consideró atípicos a aquellos píxeles que se encontraban fuera de la componente principal del histograma y tenían frecuencias inferiores a 0.05%. Se reemplazó el valor de estos píxeles atípicos por el valor del píxel más cercano que no era atípico.

Normalización.- El orden de magnitud del aporte de la radiación dispersada entre diferentes maniqués no es el mismo debido a las diversas características de estos. Por ejemplo, en el HUP, los píxeles asociados a la imagen formada solo con la componente de la radiación dispersada son de un orden de magnitud de 10^{-6} u.a., mientras que en el maniquí *MOBY*, estos son de un orden de magnitud de 10^{-8} u.a. Esta diferencia en el orden de magnitud del aporte de la radiación dispersada está asociada a las dimensiones geométricas de los maniqués, dado que el HUP es un cilindro de 18 cm de diámetro y *MOBY* tiene un diámetro de aproximadamente 3 cm. Sin embargo, al entrenar estas imágenes con un mismo algoritmo, es importante que estas se presenten al algoritmo con un mismo orden de magnitud, para evitar sesgos en las predicciones. Por tal motivo, para cada maniqué se identificó cuál es el valor máximo de los píxeles de la imagen formada solo con la componente de la radiación dispersada y se dividió cada una de estas imágenes para su correspondiente máximo, de tal manera que todas estas imágenes tenían píxeles con valores entre 0 y 1. Para cada maniqué se almacenó el factor de normalización de tal modo que cuando se realice las predicciones de la componente de dispersión, se emplee nuevamente este factor de normalización para entregar la componente de radiación dispersada en la escala correcta.

Proyecciones log-norm.- En este proyecto, para el entrenamiento del algoritmo de *deep learning* se preparó pares de imágenes de la siguiente manera: una imagen con el aporte de la radiación dispersada (I_{sc}) y otra con el aporte, tanto de fotones primarios como de fotones dispersados (I_{tot}). La imagen I_{sc} fue normalizada, tal como se describió en la sección previa. Mientras que la imagen I_{tot} fue transformada a una proyección *log-norm* (p_{tot}) usando la correspondiente imagen *plana* (I_b) y la ec. (3.1). La imagen I_b es una proyección tomada en ausencia de cualquier maniquí, por lo que en cada píxel debe tener valores superiores a I_{tot} . Sin embargo, debido a las fluctuaciones estadísticas, es posible encontrar píxeles con valores mayores en I_{tot} , lo que causaría valores negativos en la proyección *log-norm*, sobretodo en los bordes de la imagen. Para evitar la presencia de estos píxeles que no tienen significado físico se tomó, píxel a píxel, el cociente I_{tot}/I_b y se la normalizó de tal modo que todos sus elementos estén entre 0 y 1. De este modo se aseguró que la proyección p_{tot} tenga los valores de sus píxeles entre 0 y unas pocas unidades.

$$p_{tot} = \ln \frac{I_b}{I_{tot}} \quad (3.1)$$

3.4. Entrenamiento

En la Fig. 3.3 se presenta un diagrama de flujo que resume el proceso de entrenamiento de los algoritmos. Con un cuadro rojo se ha encerrado las imágenes que representan a los datos obtenidos de las simulaciones MC. Estas simulaciones entregan 360 tripletes de imágenes (I_{pr} , I_{tot} e I_{sc}) para cada maniquí, donde se cumple que $I_{tot} = I_{pr} + I_{sc}$. La operación marcada en azul indica el proceso para la obtención de la proyección *log-norm* (p_{tot}) de las imágenes I_{tot} . Se conforma un conjunto de 360 pares de imágenes formados como (p_{tot}, I_{sc}) , los cuales se separan en conjuntos de entrenamiento (cuadro naranja) y validación (cuadro morado) en los porcentajes indicados. Con las imágenes del cuadro naranja se entrena el algoritmo para que aprenda a identificar la componente I_{sc} a partir de la proyección p_{tot} .

3.5. Modelos de deep learning

En los algoritmos desarrollados se mantuvo constantes los siguientes parámetros:

- **Entrada.** La entrada para estos algoritmos son las proyecciones *log-norm*, formadas por el aporte de los fotones primarios y de la radiación dispersada; es decir, p_{tot} .
- **Conjunto de entrenamiento, evaluación y validación.** Se tomó aleatoriamente el

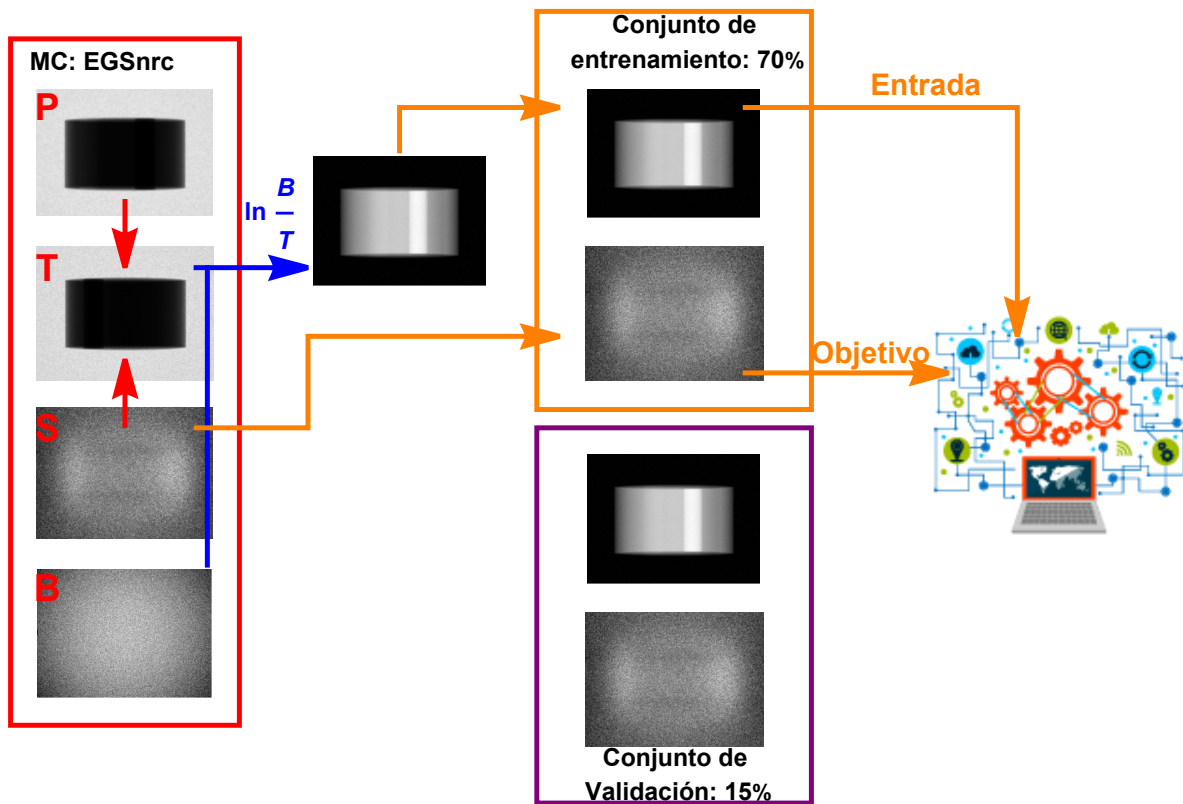


Figura 3.3: Diagrama de flujo que resume el proceso de entrenamiento de un modelo de *deep learning* entrenado con el maniquí MOBY para predecir I_{sc} . Por simplicidad, en esta figura se ha usado la notación $P = I_{pr}$, $T = I_{tot}$, $S = I_{sc}$, $B = I_b$ y $S' = I_{sc,1}$.

70 % de los datos de entrada para formar el conjunto de entrenamiento, el 15 % para el conjunto de validación y el 15 % restante para el de evaluación.

- **Optimizador.** Se usó el optimizador ADAM
- **Función de activación.** Se empleó la función ReLU
- **Tasa de aprendizaje.** Se tomó el valor de $1 \cdot 10^{-4}$.
- **Tamaño de los lotes o *batches*.** Se consideró un tamaño de *batch* de 18.
- **Número de *epochs*.** Las UNets se entrenaron por 200 epochs y las GANs por 40 mil pasos.

Además, se programó varios algoritmos con el objetivo de evaluar los siguientes aspectos:

- **Tipo de maniqués incluidos en el conjunto de entrenamiento.** Se entrenó un algoritmo solo con proyecciones del maniquí geométrico (HUP), otro solo con proyecciones del maniquí anatómico (MOBY) y otro con proyecciones de ambos maniqués.
- **Estrategia de predicción.** Se diseñó un algoritmo entrenado para identificar I_{sc} , y otro para identificar el *scatter ratio*, I_{sc}/I_{tot} .
- **Arquitecturas.** Se evaluó el efecto de emplear redes neuronales convolucionales con las arquitecturas de UNets y GANs.
- **Efectos de la Energía en la radiación dispersada.** Se entrenó un algoritmo con proyecciones del HUP en kV y otro en MV. Las imágenes de kV fueron simuladas a 125 kVp y a 2.5 MVp.
- **Función de pérdidas.** Se consideró funciones basadas en el MAE, MSE, PSNR y SSIM.
- **Efectos del sub-muestreo.** Se comparó el rendimiento de los algoritmos cuando las entradas son imágenes de 192×256 píxeles y cuando son de 96×128 píxeles.
- **Transferencia de aprendizaje.** Se consideró cuatro casos. En el caso *a*) se entrenó un modelo con proyecciones del HUP y se aplicó TL para evaluarlo en proyecciones de MOBY. En el caso *b*) se tomó un modelo entrenado con proyecciones de MOBY y se aplicó TL para evaluarlo en las proyecciones del HUP. En el caso *c*) se tomó un modelo entrenado con proyecciones del HUP en kV y mediante TL se aplicó el modelo a proyecciones del HUP en MV. En el caso *d*) se partió de un modelo entrenado en proyecciones de MV del HUP y se usó TL para evaluarlo en proyecciones de kV del HUP.

Los algoritmos fueron programados en *Python 3.7*, con la librería *Keras 2.8* de *tensorflow 2.8*. Debido a la complejidad computacional asociada al entrenamiento de este tipo de arquitecturas, los modelos fueron implementados y entrenados en unidades de procesamiento gráfico (GPUs). Se empleó los GPUs de *Google Colab*, de un clúster de la UNAM llamado *Holiday* y de una estación de trabajo del laboratorio de imágenes biomédicas de la UNAM, llamada *Alice*. El GPU de *Holiday* es un modelo *NVIDIA Volta V100* con 32 GB de RAM [75]. El GPU de *Alice* es un modelo *NVIDIA RTX 3060* con 12 GB de RAM [76]. También se midió el tiempo de entrenamiento para evaluar la eficiencia de los GPUs empleados, tanto en el

proceso de entrenamiento, como en el proceso de ejecución de las correcciones por radiación dispersada.

3.5.1. UNets

En la Fig. 3.4 se muestra una representación esquemática de la arquitectura usada como UNet. Su entrada es una imagen de 192×256 píxeles, imagen que es transformada mediante diversas operaciones de convolución y *max-pooling* que reducen su tamaño hasta el punto de tener 1024 imágenes de 3×4 píxeles en el *cuello de botella* de la red. Posteriormente se incrementa el tamaño de la imagen mediante operaciones de *up-sampling*, concatenaciones y convoluciones hasta tener como salida de una única imagen de 192×256 píxeles. La red implementada tuvo 71 capas con 53.5 millones de parámetros entrenables.

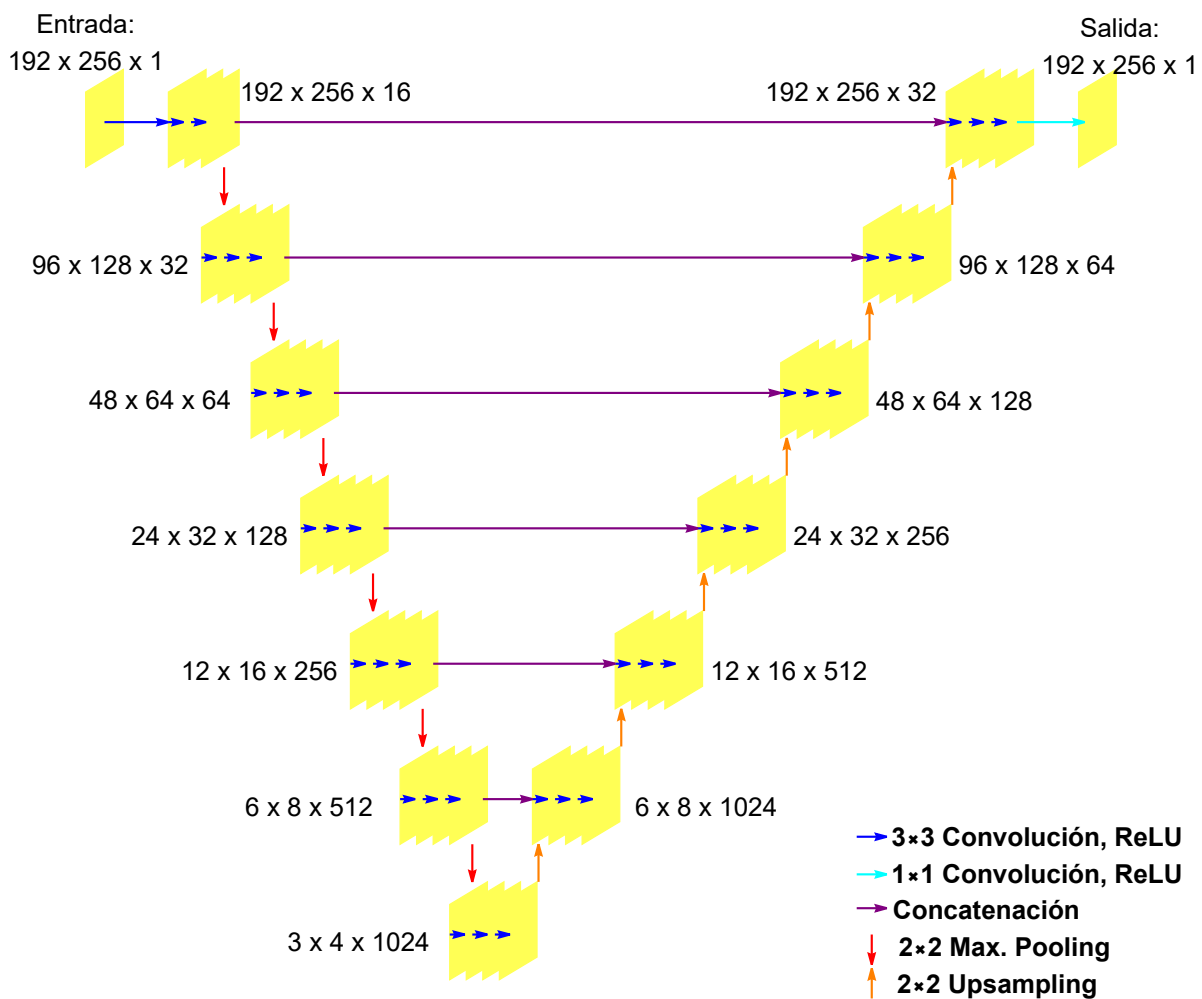


Figura 3.4: Representación esquemática de la arquitectura empleada para las UNets.

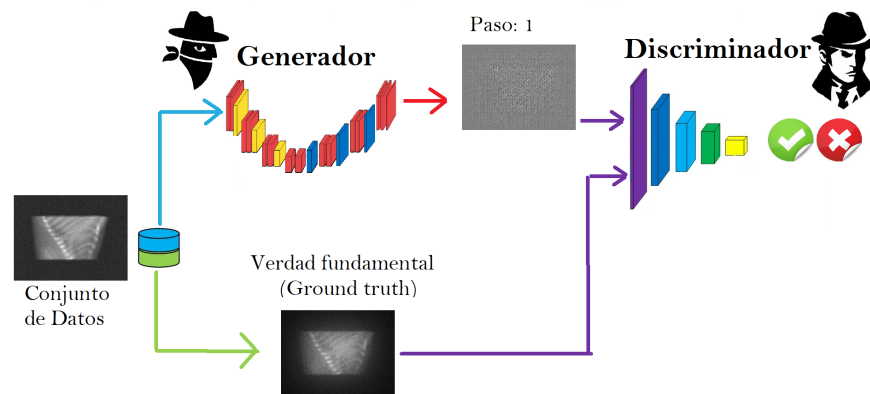


Figura 3.5: Esquema del entrenamiento de una GAN. Se muestra una imagen producida por la red G en el paso 1.

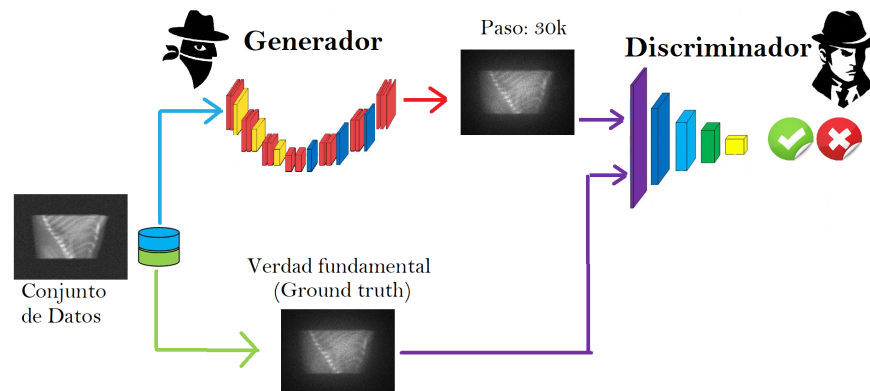


Figura 3.6: Esquema del entrenamiento de una GAN. Se muestra una imagen producida por la red G en el paso 30 mil.

3.5.2. Redes Generatrices Adversariales

En la Fig. 3.5 se muestra un esquema del proceso de entrenamiento en las GANs. La red generadora (G) recibe las imágenes del conjunto de datos de entrenamiento y en el primer paso/iteración genera una imagen con píxeles aleatorios. Por otra parte, la red discriminadora, recibe de forma aleatoria las imágenes esperadas o *ground truth* y las producidas por la red G ; su tarea es distinguir cuáles son las auténticas y cuales son las falsas/generadas. En la primera iteración se observa que fácilmente se distingue que la imagen generada es una imagen ruidosa; sin embargo, en la Fig. 3.6 se observa que luego de 30 mil iteraciones, la red G ha aprendido a producir imágenes prácticamente indistinguibles del *ground truth*.

La estructura de una red generadora es prácticamente la misma de una UNet, conformada por

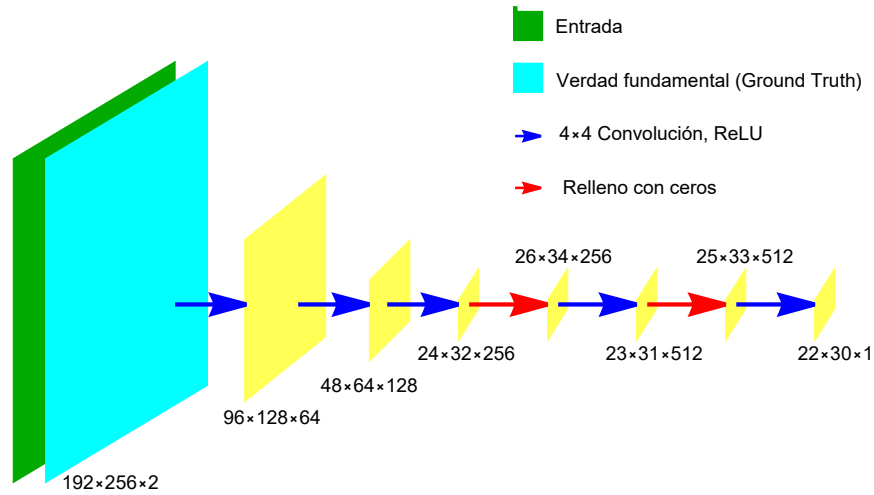


Figura 3.7: Arquitectura de la red discriminadora empleada en las GANs usadas en el presente trabajo.

un camino de contracción y uno de expansión con concatenaciones para no perder información espacial. La red discriminadora es una red neuronal convolucional que tiene dos salidas, *aceptar* o *rechazar* la imagen que recibe. La arquitectura de la red generadora en el modelo de GANs implementado en este trabajo fue una UNet equivalente a la mostrada en la Fig. 3.4, que contó con 18 capas. En la Fig. 3.7 se muestra la arquitectura empleada para la red discriminadora, la cual contó con 12 capas.

3.6. Análisis de datos

La evaluación de los modelos implementados fue realizada con base en el tiempo de entrenamiento, en el tiempo que se tarda en corregir conjuntos completos de proyecciones y en la calidad de imagen. La calidad de imagen se evaluó en las proyecciones y en la imagen tomográfica reconstruida. En las proyecciones se consideró el error relativo, SSIM, PSNR y comparó perfiles. En las reconstrucciones se analizó el número CT, el cociente señal a ruido (SNR) y el cociente contraste a ruido (CNR).

En la Fig. 3.8 se presenta un diagrama de flujo que resume el proceso de comparación de resultados. Una vez entrenado el algoritmo, se evalúa dándole proyecciones p_{tot} del conjunto de validación y determinando la salida. En este caso, se ejemplifica la predicción de imágenes formadas únicamente por la componente de radiación dispersada, I'_{sc} (en esta notación, el apóstrofe indica que la imagen es producida por un modelo de *deep learning*). Estas imágenes tienen píxeles con valores entre 0 y 1 aproximadamente por lo que deben ser multiplicadas por su correspondiente factor de normalización. Luego son empleadas para generar las pro-

yecciones p'_{pr} de acuerdo con la ec. (3.2). Finalmente, las proyecciones p'_{pr} y las imágenes I'_{sc} son comparadas con los valores asumidos como verdaderos (*ground truth*) que corresponden a las proyecciones p_{pr} e imágenes I_{sc} del conjunto de evaluación, respectivamente.

$$p'_{pr,i} = \ln \frac{I_b}{I_{tot} - I'_{sc,i}} \quad (3.2)$$

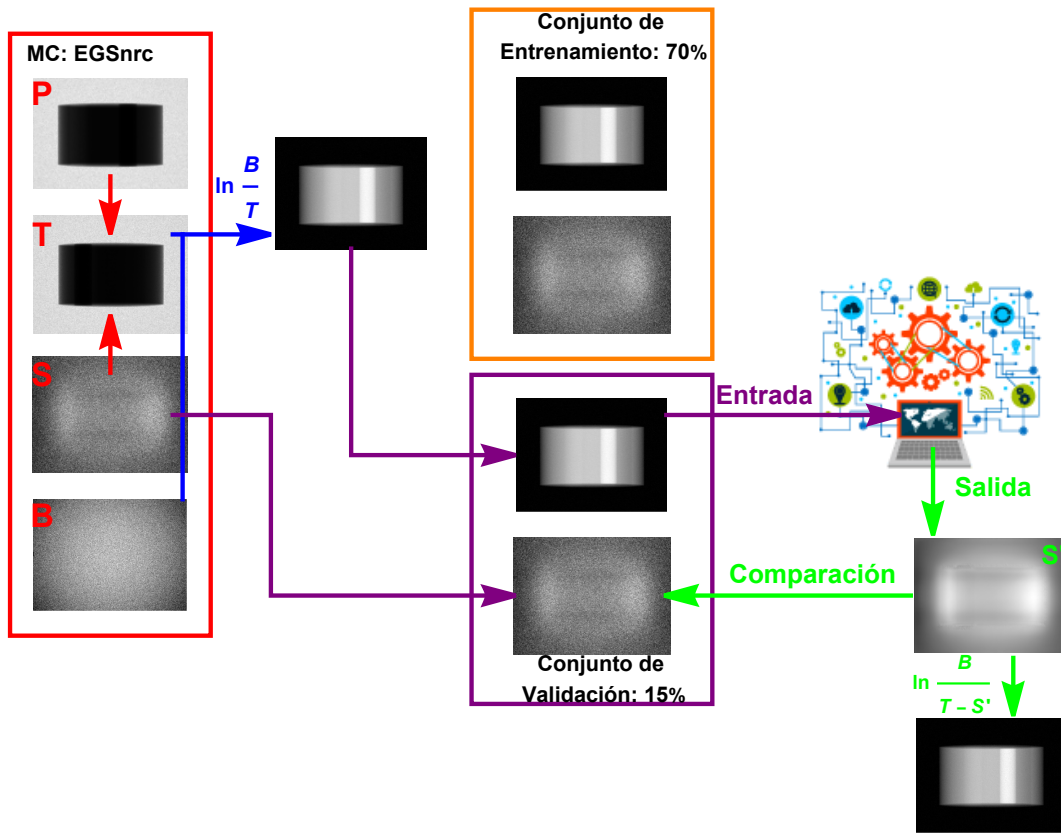


Figura 3.8: Diagrama de flujo que resume el proceso de corrección por radiación dispersada con un modelo de deep learning entrenado y la evaluación de sus resultados. Se ejemplifica con un modelo entrenado con MOBY para predecir I_{sc} . Por simplicidad, en esta figura se ha usado la notación $P = I_{pr}$, $T = I_{tot}$, $S = I_{sc}$, $B = I_b$ y $S' = I_{sc,1}$.

Capítulo 4

Resultados y Discusión

Este capítulo se ha estructurado de la siguiente manera. En la sección [4.1](#) se presenta los resultados de un “modelo base”, que es una UNet entrenada sin transferencia de aprendizaje, con proyecciones del HUP en kV, para predecir la componente de radiación dispersada I_{sc} . El modelo tiene una función de pérdidas basada en el error absoluto medio y su entrada son imágenes de 192×256 píxeles. En las secciones [4.2](#)[4.8](#) se presenta cómo varían estos resultados a medida que se modifica el conjunto de entrenamiento, el objetivo de predicción, la arquitectura de la red, la energía de adquisición de las imágenes, la función de pérdidas, el tamaño de la imagen de entrada y se aplica transferencia de aprendizaje, respectivamente. Finalmente, en la sección [4.9](#) se presenta un análisis asociado a los tiempos de entrenamiento y corrección de un conjunto completo de proyecciones, en función del equipo empleado.

4.1. UNet entrenada solo con proyecciones del HUP

4.1.1. Estructura de los códigos empleados

El modelo descrito en la esta sección fue desarrollado en Google Colab utilizando el lenguaje de programación *Python v3.8*, junto con las librerías *Tensorflow v2.8* y *Keras v2.8*; el programa realiza las siguientes tareas:

- Carga las librerías requeridas.
- Definición de las funciones y parámetros que se usan en el código.
- Importación de los datos, lo cual cuando se trabaja en *Google Colab* se realiza directamente desde un repositorio de *google drive*.

- Clasificación de los pares de imágenes en los conjuntos de entrenamiento, validación y evaluación.
- Almacenamiento de las proyecciones *log-norm* calculadas.
- Implementación y entrenamiento de la red neuronal convolucional.
- Generación de gráficas de la función de pérdidas del conjunto de entrenamiento y validación.
- Almacenamiento del modelo en formato *.h5* para emplearlo posteriormente al aplicar transferencia de aprendizaje.
- Correcciones de conjuntos completos de proyecciones de todo el maniquí.
- El almacenamiento de los resultados.
- La comparación de perfiles de las predicciones frente a los de las imágenes verdaderas. Este análisis se realiza tanto a nivel de imágenes formadas por la componente de la radiación dispersada como en proyecciones *log-norm* formadas únicamente por fotones primarios.
- La determinación del error porcentual absoluto medio entre las imágenes predichas y las esperadas.

Las proyecciones corregidas son empleadas para ejecutar la reconstrucción tomográfica utilizando las rutinas de ASTRA Toolbox, con interfaz en *Python*. Para el análisis del número CT de los insertos, del cociente señal a ruido (SNR) y del cociente contraste a ruido (CNR) se ha diseñado un cuaderno de *Google Colab* en el que se definen Regiones de Interés (ROIs, por sus siglas en inglés) circulares de 1 cm de radio centradas en cada inserto y en la región homogénea del cilindro y se calcula valores medios, desviación estándar entre cortes y error relativo.

4.1.2. Entrenamiento del modelo

La Fig. [4.1](#) muestra la función de pérdidas evaluada en el conjunto de entrenamiento y en el de validación en función del número de *epochs*. Normalmente, la función de pérdida es menor en el conjunto de entrenamiento que en el de validación debido a que el modelo conoce las soluciones del conjunto de entrenamiento. Para que el modelo sea generalizable, se requiere que el rendimiento del modelo en datos con los que no fue entrenado (conjunto

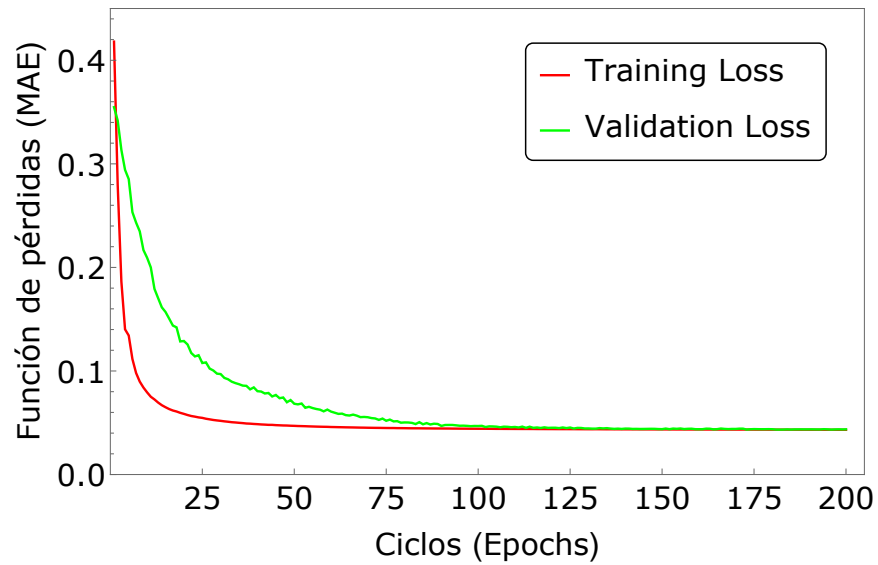


Figura 4.1: Función de pérdidas evaluada en el conjunto de entrenamiento y en el de validación, en función del número de *epochs*.

de validación) sea tan bueno como su rendimiento en datos que ya conocía (conjunto de entrenamiento). Mientras más bajos sean los valores de la función de pérdidas en ambos conjuntos, los resultados del modelo serán mejores.

Este modelo alcanza una solución estable en alrededor de *50 epochs*. Al evaluarlo en el conjunto de evaluación se obtuvo una función de pérdidas de *0.04*, la cual difiere solo en *0.37%* de la función de pérdidas en el conjunto de entrenamiento. Esto significa que el rendimiento del modelo en datos desconocidos es tan bueno como su rendimiento en los datos con los que fue entrenado; lo cual valida su capacidad para identificar la componente de la radiación dispersada.

4.1.3. Predicciones del modelo

La Fig. 4.2 compara las proyecciones del HUP formadas únicamente por radiación dispersada. La imagen de la parte (a) fue obtenida directamente de la simulación MC por lo que es el *ground truth*. La imagen de la parte (b) es la predicción de la componente de radiación dispersada de acuerdo con una UNet entrenada. Ambas imágenes se presentan con una misma ventana de contrastes. Las líneas en azul y rojo indican la ubicación de la que se adquirieron los perfiles que se comparan en la Fig. 4.3.

El error porcentual medio de comparar todas las proyecciones predichas por la UNet con sus respectivas imágenes del *ground truth* es de *8.62%*. Las predicciones de la UNet aprenden la tendencia de la radiación dispersada sin incluir su ruido. Esto es muy favorable para una

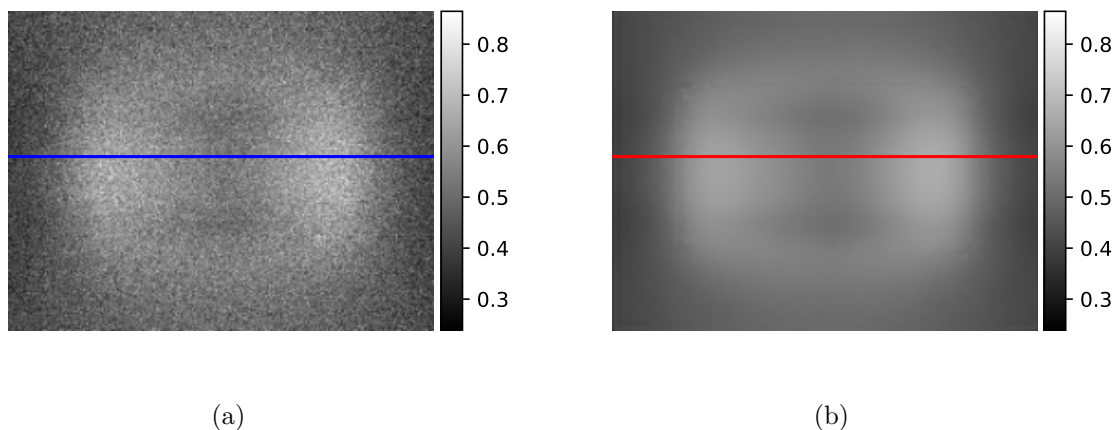


Figura 4.2: Comparación de una imagen del HUP, formada únicamente por la componente de la radiación dispersada, cuando esta es generada por las simulaciones MC (a) y cuando es predicha por la UNet (b). Las líneas azul y roja indican la región de la que se tomó los perfiles que se muestran en la Fig. 4.3.

corrección eficiente ya que el ruido en los perfiles de las simulaciones MC está asociado a las fluctuaciones estadísticas y no al comportamiento de la radiación dispersada [13].

4.1.4. Correcciones por radiación dispersada

La Fig. 4.4 compara proyecciones *log-norm* formadas únicamente por radiación primaria de acuerdo con lo predicho por las simulaciones MC y de acuerdo con una UNet. La UNet ha sido entrenada para predecir la componente de la radiación dispersada, I'_{sc} ; por lo que para encontrar la proyección *log-norm* corregida, p'_{pr} se usó la ec. (3.2). La Fig. 4.5 compara los perfiles mostrados en las proyecciones *log-norm* de la Fig. 4.4. Además, en color verde se ha incluido una proyección *log-norm* formada por la componente total de la radiación, para notar el efecto de la radiación dispersada en las imágenes de tomografía de haz de cono (CBCT) y la eficiencia de las UNets entrenadas en la supresión de este aporte.

El error porcentual medio de comparar todas las proyecciones corregidas por la UNet con sus respectivas proyecciones corregidas mediante MC es de 1.73%; el mayor aporte a este valor proviene de los píxeles con valores cercanos a cero que están fuera del fantoma, por lo que si se restringe el cálculo a píxeles mayores a 0.1, el error porcentual medio disminuye a 0.08%. Además, el error porcentual medio entre las proyecciones corregidas con la UNet y las proyecciones no corregidas (totales) es de 4.00%. Esto indica el efecto de las correcciones por radiación dispersada.

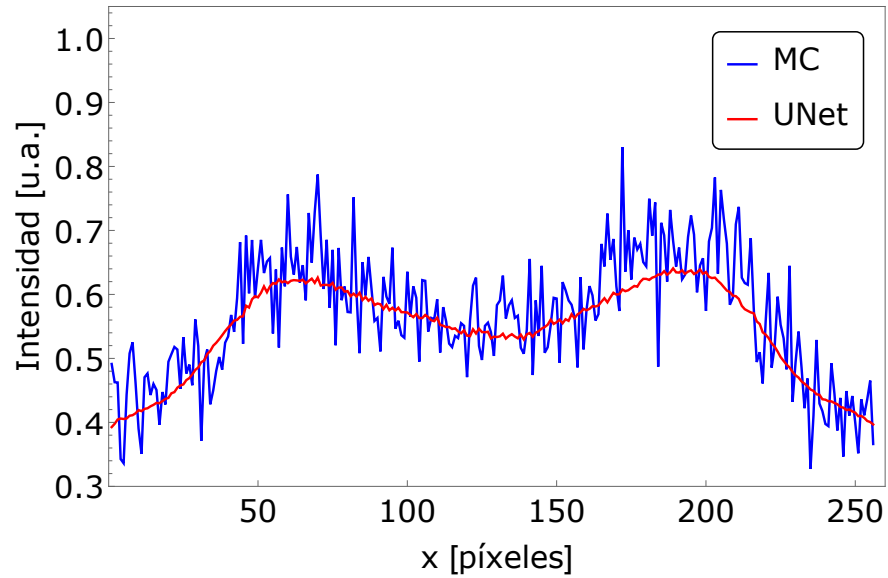
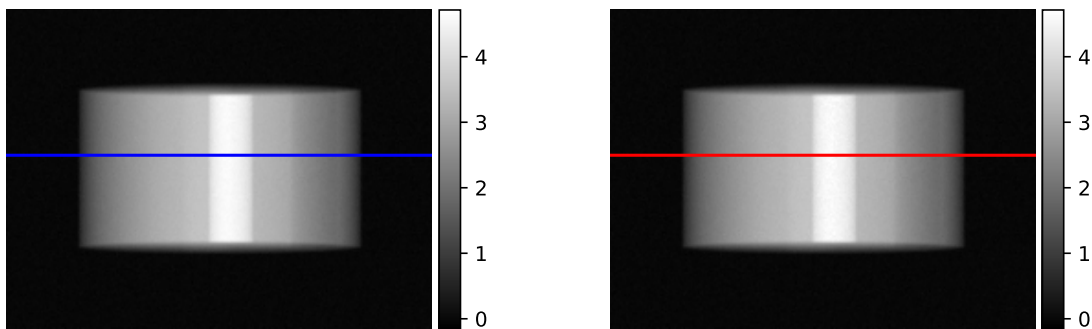


Figura 4.3: Comparación de un perfil de una proyección del HUP formada únicamente por la componente de la radiación dispersada, cuando esta es generada por las simulaciones MC (azul) y cuando es predicha por la UNet (rojo). Estos perfiles son los mostrados en las imágenes de la Fig. [4.2](#).



(a)

(b)

Figura 4.4: Comparación de una proyección *log-norm* del HUP, formada únicamente por fotones primarios, cuando esta es generada por las simulaciones MC (a) y cuando es predicha por la UNet (b).

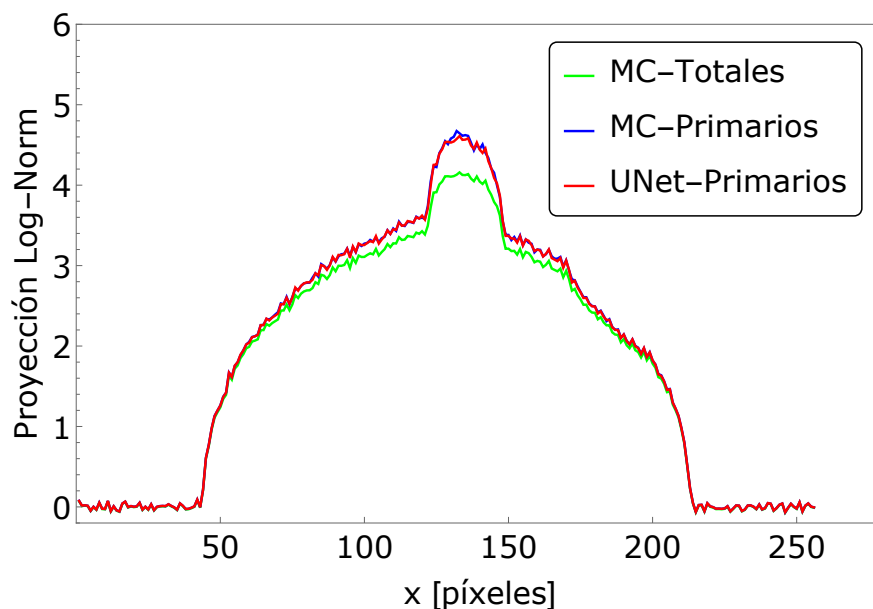


Figura 4.5: En rojo se muestra el perfil de una proyección *log-norm* formada por la componente primaria de la radiación según la UNet, en azul el perfil generado por la simulación MC solo con el aporte de la radiación primaria y en verde el perfil generado por la simulación MC con el aporte de la radiación total. Los perfiles en rojo y azul son los de las imágenes mostradas en la Fig. 4.4

4.1.5. Reconstrucción tomográfica

Luego de encontrar la contribución de la radiación dispersada y suprimirla de las proyecciones se realizó la reconstrucción tomográfica empleando las rutinas de ASTRA Toolbox, con interfaz en *Python*. La reconstrucción fue realizada con el algoritmo de Feldkamp Davis y Kress (FDK), con un filtro *Ram-Lak* y frecuencia de corte 1. Las reconstrucciones tomográficas obtenidas de Astra son un mapa de coeficientes de atenuación lineal μ^{H} . Con una ROI circular de 1 cm de radio, centrada en el centro del cilindro de agua se determinó el $\mu_{\text{H}_2\text{O}}$ y con la ec. (4.1) se obtuvo un mapa de números CT ($\#CT$).

$$\#CT = 1000 \cdot \frac{\mu - \mu_{\text{H}_2\text{O}}}{\mu_{\text{H}_2\text{O}}} \quad (4.1)$$

La Fig. 4.6 presenta las imágenes tomográficas del HUP, reconstruidas con proyecciones totales, primarias según MC y primarias corregidas mediante una UNet. En la parte superior está el inserto de hueso cortical, en la inferior el de hueso trabecular, a la izquierda el de

¹En el Anexo A se presenta un cálculo teórico de los valores de μ correspondientes a la energía efectiva y promedio de los espectros empleados, para todos los materiales del HUP, utilizando la composición indicada por el fabricante. Estos valores se comparan con el promedio de las intensidades de píxel, medidas en cada inserto en una reconstrucción realizada solo con proyecciones primarias.

tejido adiposo y a la derecha el de pulmón exhalado.

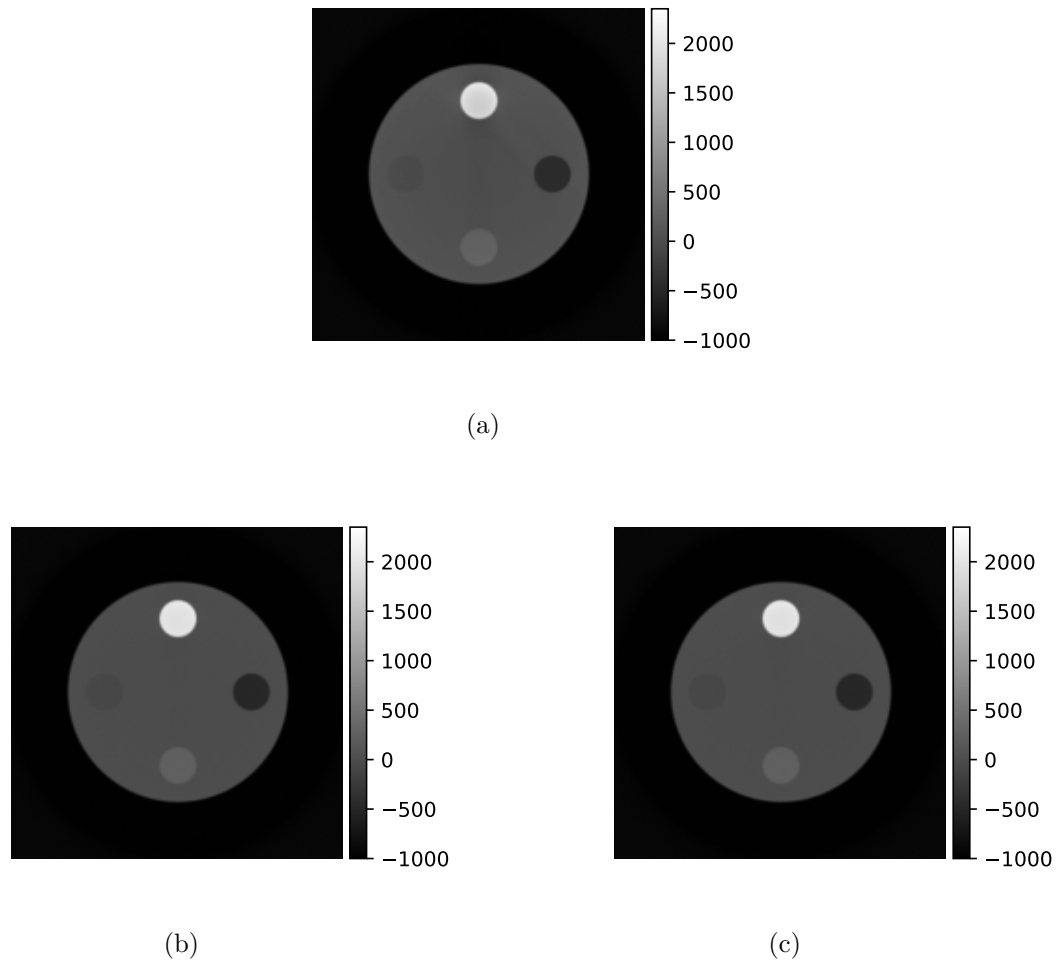


Figura 4.6: Corte central de la reconstrucción tomográfica del HUP empleando proyecciones con el aporte de total de la radiación, según las simulaciones MC (a), usando solo el aporte la radiación primaria, según las correcciones basadas en métodos MC (b) y solo con el aporte la radiación primaria, según las correcciones de una UNet (c).

De forma cualitativa, al comparar las gráficas mostradas en la Fig. 4.6 no se visualizan diferencias significativas. Levemente se distingue un artefacto en forma de rayas oscuras en la sección homogénea del fantoma, justo debajo del inserto de hueso cortical en la Fig. 4.6a; artefacto que desaparece tanto en las correcciones con las simulaciones MC como con la UNet.

Para realizar un análisis cuantitativo en las reconstrucciones tomográficas del HUP se calculó el número CT del material del cilindro y de los insertos, la SNR y la CNR. En la tabla 4.1 se presentan los valores obtenidos con las reconstrucciones mostradas en la Fig. 4.6. Los errores porcentuales en los números CT han sido calculados con respecto al *ground*

truth, que corresponde a la reconstrucción realizada con correcciones basadas en métodos MC [13]. El signo en el error porcentual indica una sobre-estimación (+) o sub-estimación (-) en los números CT. Las incertidumbres se reportan como una desviación estándar de las fluctuaciones estadísticas entre cortes.

La Tabla 4.1 muestra que la radiación dispersada causa importantes imprecisiones en los números CT. En promedio, el error porcentual en la determinación de los $\#CT$ en una reconstrucción realizada sin correcciones por radiación dispersada es 25.88%, con errores de hasta el 72.31% en el tejido adiposo. Al suprimir el aporte de la radiación dispersada con una UNet entrenada, la precisión de los $\#CT$ mejora notablemente, reduciendo el error porcentual medio a 0.48%. Los números CT calculados para los insertos del HUP, considerando su composición reportada por el fabricante y la energía efectiva del espectro de kV son 0, 232, -73, 1967 y -501 HU para el agua, hueso trabecular, tejido adiposo, hueso cortical y pulmón exhalado, respectivamente. Estos valores difieren de los obtenidos de proyecciones primarias con simulaciones MC debido a que la energía efectiva no captura completamente el comportamiento de un haz polienergético. En cuanto a la SNR y CNR, en general, se detecta una mejoría al corregir las proyecciones por radiación dispersada mediante los métodos MC. Se encontró que al realizar las correcciones con la UNet, la SNR y CNR tiene un comportamiento similar a la de la medida en la reconstrucción con proyecciones primarias.

4.2. Conjunto de entrenamiento

En esta sección se detallan las principales diferencias que se encuentran cuando al modelo de la sección 4.1 se lo entrena solo con proyecciones de MOBY o con un conjunto mezclado de proyecciones (HUP+MOBY).

UNet entrenada solo con proyecciones de MOBY

La Fig. 4.7 compara la distribución de la radiación dispersada medida con la esperada para las proyecciones de MOBY. La imagen (a) fue obtenida directamente de la simulación MC y la imagen (b) es la predicción de la componente de radiación dispersada de acuerdo con una UNet. Ambas imágenes se presentan con una misma ventana de contrastes. Las líneas en azul y rojo indican la ubicación de la que se adquirieron los perfiles que se comparan en la Fig. 4.8. El error porcentual medio de comparar todas las proyecciones predichas por la UNet con sus respectivas imágenes del *ground truth* es de 10.53%.

En la Fig. 4.9 se muestra una proyección *log-norm* formada únicamente por radiación primaria de acuerdo con lo predicho por la UNet. En la Fig. 4.10 se muestra el perfil de la proyección *log-norm* de la Fig. 4.9, junto con el perfil de la proyección *log-norm* total (en verde) y

Tabla 4.1: Comparación de los números CT, CNR y SNR en los insertos del HUP de acuerdo con la imagen reconstruida con proyecciones totales, primaria y corregidas por radiación dispersada con una UNet.

Material	$\#CT_{kV}$ [HU]	$\#CT_{ref}$ [HU]	Error [%]	SNR	CNR
Proyecciones primarias					
Agua	0 ± 3	—	—	17.09 ± 1.22	—
Hueso Trabecular	257 ± 11	—	—	25.51 ± 1.62	4.40 ± 0.37
Tejido Adiposo	-65 ± 11	—	—	18.92 ± 1.25	1.12 ± 0.22
Hueso Cortical	1957 ± 8	—	—	51.52 ± 2.87	33.45 ± 2.38
Pulmón Exhalado	-488 ± 11	—	—	10.35 ± 0.66	8.35 ± 0.63
Proyecciones totales					
Agua	0 ± 4	0 ± 3	—	16.21 ± 1.09	—
Hueso Trabecular	284 ± 9	257 ± 11	10.51	24.70 ± 1.49	4.62 ± 0.35
Tejido Adiposo	-18 ± 10	-65 ± 11	-72.31	19.39 ± 1.17	0.30 ± 0.19
Hueso Cortical	1788 ± 7	1957 ± 8	-8.64	29.96 ± 1.05	28.99 ± 1.94
Pulmón Exhalado	-429 ± 11	-488 ± 11	-12.09	11.31 ± 0.74	6.96 ± 0.51
Proyecciones corregidas con una UNet					
Agua	0 ± 4	0 ± 3	—	14.49 ± 1.04	—
Hueso Trabecular	258 ± 11	257 ± 11	0.39	22.84 ± 1.51	3.74 ± 0.32
Tejido Adiposo	-64 ± 11	-65 ± 11	-1.54	17.59 ± 1.09	0.94 ± 0.18
Hueso Cortical	1957 ± 9	1957 ± 8	0.00	40.24 ± 2.61	28.36 ± 2.03
Pulmón Exhalado	-488 ± 11	-488 ± 11	0.00	9.87 ± 0.66	7.07 ± 0.53

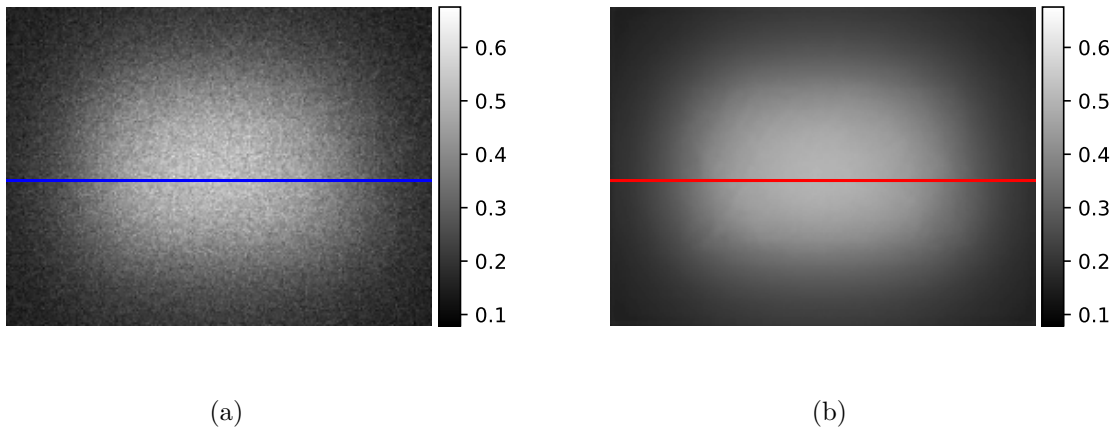


Figura 4.7: Comparación de una imagen del maniquí MOBY, formada únicamente por la componente de la radiación dispersada, cuando esta es generada por las simulaciones MC (a) y cuando es predicha por la UNet (b). Las líneas azul y roja indican la región de la que se tomó los perfiles que se muestran en la Fig. 4.8.

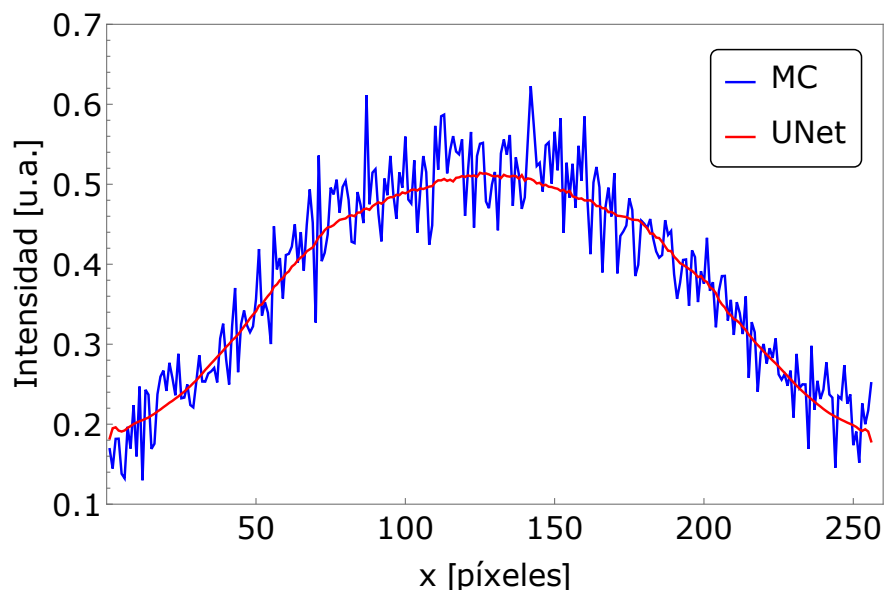


Figura 4.8: Comparación de un perfil de una proyección de MOBY, formada únicamente por la componente de la radiación dispersada, cuando esta es generada por las simulaciones MC (azul) y cuando es predicha por la UNet (rojo). Estos perfiles son los mostrados en las imágenes de la Fig. 4.7.

primaria (en azul), de acuerdo con las simulaciones MC. El aporte de la radiación dispersada es muy bajo y visualmente imperceptible, lo cual se debe a las dimensiones del maniquí. Sin embargo, la diferencia entre la curva verde y azul es de 2.67%, mientras que entre la curva roja y verde hay una diferencia de 0.22%. Esto indica la capacidad de la red neuronal para identificar el aporte de la radiación dispersada, incluso cuando es insignificante.

La Fig. 4.11 muestra el corte central de la reconstrucción tomográfica de MOBY, adquirida a partir de las proyecciones corregidas con la UNet.

UNet entrenada con proyecciones del HUP y de MOBY

Para evaluar la generalizabilidad de los algoritmos, también se entrenó un modelo con un conjunto de datos conformado por proyecciones, tanto de maniquíes geométricos como anatómicos (HUP+MOBY). La principal diferencia es que el modelo requiere de un mayor número de *epochs* para que la función de pérdidas del conjunto de validación sea tan baja como la del conjunto de entrenamiento. Esto se observa en la Fig. 4.12, que contrasta con el comportamiento mostrado en la Fig. 4.1. La función de pérdida evaluada en el conjunto de evaluación fue de 0.04 y la diferencia con su valor en el conjunto de entrenamiento fue de solo 0.04%, lo cual indica la validez del proceso de entrenamiento.

Las diferencias en el comportamiento de la función de pérdidas se debe a que para el modelo,

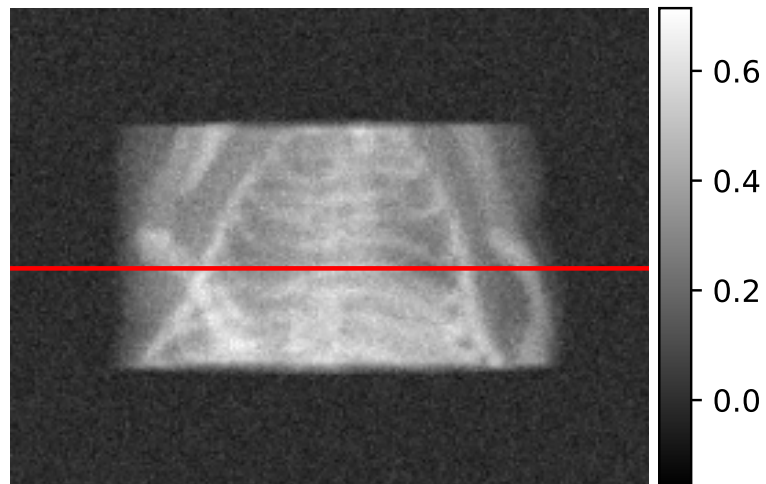


Figura 4.9: Imagen de una proyección *log-norm* del maniquí MOBY corregida por radiación dispersada, con la UNet.

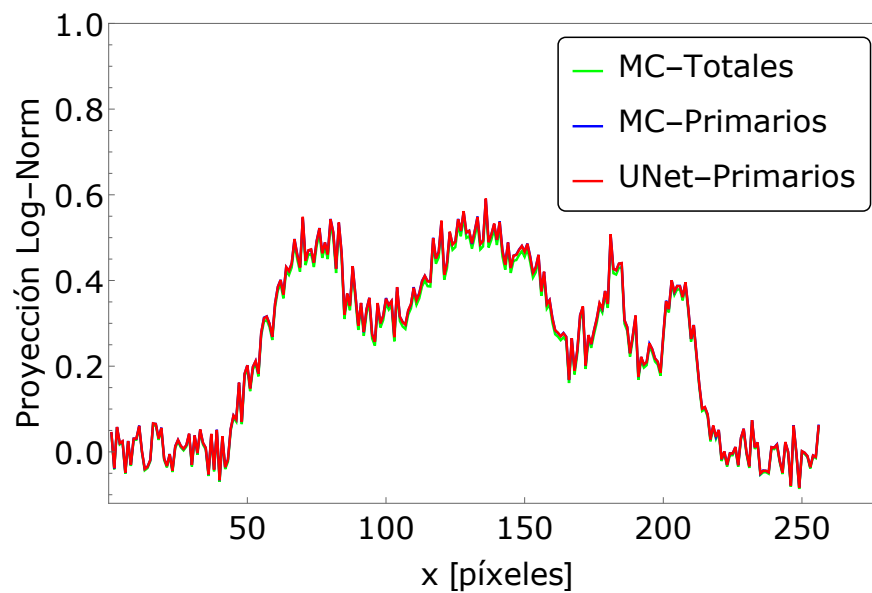


Figura 4.10: Comparación de los perfiles de las proyecciones *log-norm* no corregidas (verde) y corregidas con MC (azul) y con la UNet (rojo).

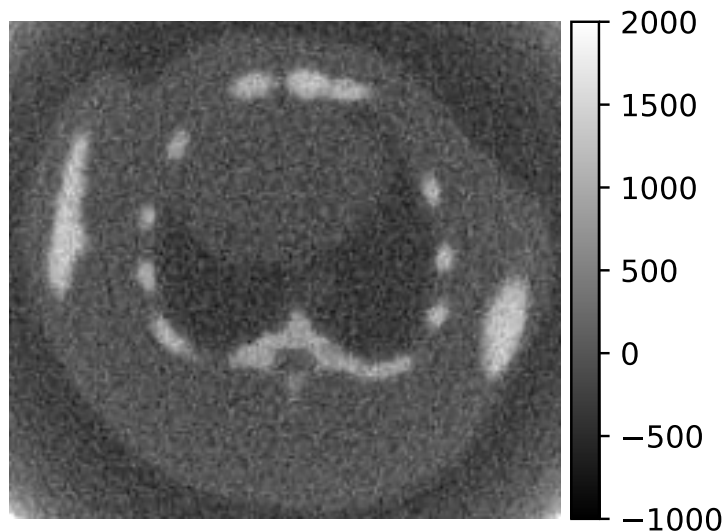


Figura 4.11: Corte central de la reconstrucción tomográfica de MOBY, obtenida a partir de las proyecciones corregidas con la UNet.

es más *complicado* identificar el comportamiento que tiene la radiación dispersada en maniqués de diferente naturaleza. Esto debido a que el HUP es un cilindro de agua de 20 cm de diámetro, en el cual el cociente radiación dispersada a total es 4.1 %, en promedio. Por otro lado, en el fantoma de MOBY, dado que se simula el tórax de un ratón, sus dimensiones son mucho menores (~ 3 cm) y por tanto el aporte de la radiación dispersada a la formación de la imagen es menor, detectándose un cociente radiación dispersada a total de 0.5 %, que es significativamente menor que el aporte en el HUP. Además, la estructura de un cilindro con insertos es diferente a la del tórax de un ratón, que tiene elementos anatómicos con distintas formas. Esto genera distintos patrones en la distribución de la radiación dispersada.

El error porcentual medio calculado al comparar las proyecciones predichas por esta UNet, con sus respectivas imágenes de *ground truth* es de 8.98 %. El error porcentual de comparar todas las proyecciones corregidas por la UNet con las corregidas mediante MC (excluyendo los píxeles en los que no está el objeto, i.e., menores que 0.1) es de 0.32 %. Estos resultados nos indican que un modelo entrenado con un conjunto mezclado de datos tarda más en el proceso de entrenamiento, pero la eficiencia de sus resultados es comparable con la de un modelo entrenado solo con proyecciones del HUP.

Al reconstruir la imagen tomográfica el error porcentual medio en los números CT fue de 1.19 %. Este es un error levemente mayor que el encontrado con las correcciones realizadas

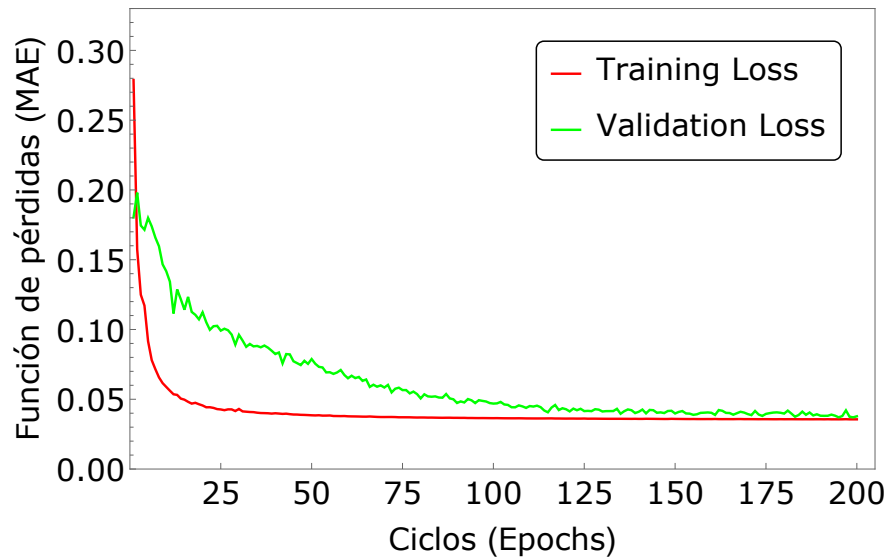


Figura 4.12: Función de pérdidas del conjunto de entrenamiento y validación en función del número de *epochs* para un modelo entrenado simultáneamente con proyecciones del HUP y de MOBY.

con una UNet entrenada solo con proyecciones del HUP. Por lo tanto, entrenar un modelo con un datos de diversa geometría puede afectar levemente la precisión de los números CT. También se detectó una leve mejoría en la SNR, con respecto a las imágenes no corregidas por dispersión. Además, esta UNet no muestra diferencias significativas, en la SNR, ni CNR, con respecto a la UNet entrenada solo con proyecciones del HUP.

4.3. Objetivo de predicción

El *target* u objetivo de predicción fue otro factor que se evaluó en los diferentes modelos de esta investigación. El objetivo de predicción es la imagen que se requiere que el modelo de *deep learning* la prediga a partir de una imagen de entrada, formada por fotones de radiación primaria y dispersada. Se consideró como objetivos a la imagen formada únicamente por la radiación dispersada y al *scatter ratio*. Se conoce como *scatter ratio* a la división, píxel a píxel, entre la imagen formada únicamente por radiación dispersada y la imagen formada por todos los fotones que arriban al detector. Esta sección se enfoca en el *scatter ratio* como objetivo de predicción y se comparan sus resultados con los de la sección [4.1](#).

La Fig. [4.13](#) compara el *scatter ratio* esperado y predicho para una proyección del HUP y la Fig. [4.14](#) presenta los respectivos perfiles de *scatter ratio* para las imágenes de la Fig. [4.13](#).

Cuando la imagen objetivo es la imagen formada solo por la radiación dispersada, junto con el modelo entrenado, se requiere almacenar un factor de normalización. Este factor permite recuperar la escala de una imagen corregida; sin embargo, si esta es originada por diferentes

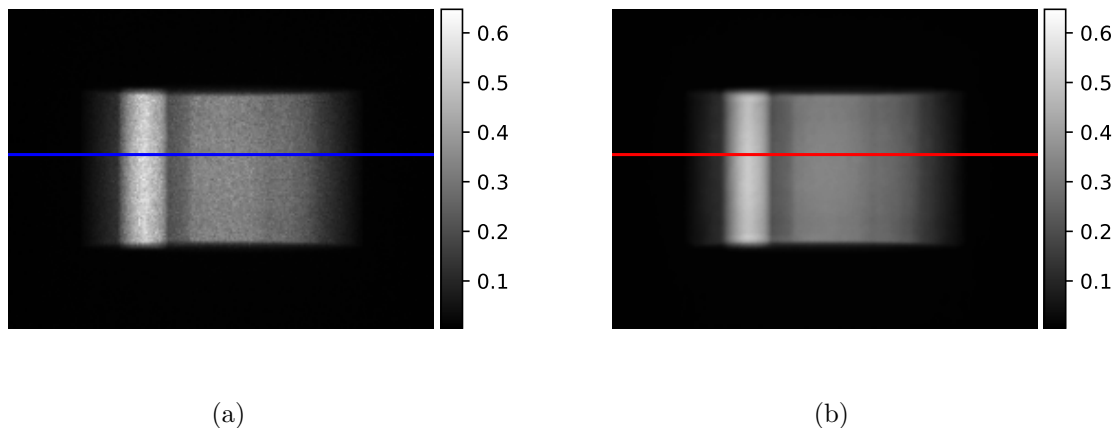


Figura 4.13: Comparación entre el *scatter ratio* esperado de acuerdo con las simulaciones MC (a) y el predicho por la UNet (b) para una proyección del HUP.

programas de simulaciones MC, puede producir resultados incompatibles. Por lo tanto, predecir el *scatter ratio* es una mejor estrategia dado que no requiere factores de normalización, pues por su definición, la imagen ya tiene valores escalados entre 0 y 1.

La Fig. 4.15 muestra la imagen formada solo por la radiación dispersada. Esta fue obtenida al multiplicar el *scatter ratio* predicho por la UNet por la proyección total. La imagen esperada era la mostrada en la Fig. 4.3 (izquierda); sin embargo, en la Fig. 4.15 se muestra una forma distinta, en la que resaltan los bordes del objeto. Por lo tanto, las predicciones realizadas con un modelo entrenado para predecir la componente de la radiación dispersada (ver Fig. 4.3, derecha) son mejores que las de un modelo entrenado para predecir *scatter ratio*.

La Fig. 4.16 compara el perfil predicho bajo esta estrategia con el esperado. Al usar el *scatter ratio* como objetivo de predicción, las predicciones son más ruidosas. Además, el error porcentual medio de las proyecciones para este modelo es de 9.82%, en contraste con un error del 8.62%, que fue el encontrado para un modelo entrenado en predecir directamente la componente de la radiación dispersada. El error porcentual de comparar todas las proyecciones corregidas por esta UNet con las corregidas mediante MC (excluyendo los píxeles menores que 0.1) es de 0.36%. Por lo tanto, la estrategia de predecir el *scatter ratio* permite evitar problemas de normalización, pero genera mayores imprecisiones en la determinación del aporte de la radiación dispersada.

Al reconstruir la imagen tomográfica se encontró que el error porcentual medio en los números CT fue de 4.97%. Este valor es significativamente más alto que el encontrado con el modelo descrito en la sección 4.1, por lo que las reconstrucciones con una UNet para pre-

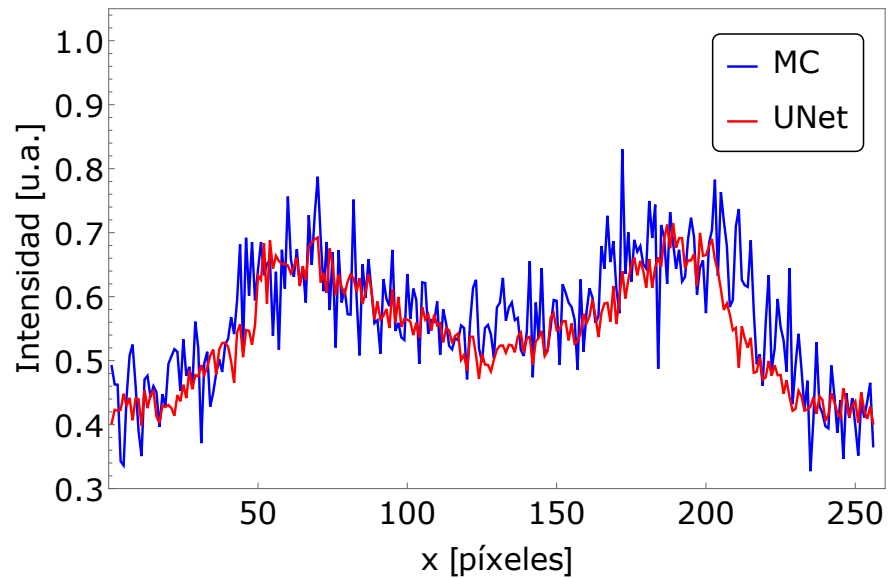


Figura 4.14: Comparación entre los perfiles del *scatter ratio*, de una proyección del HUP, generados por las simulaciones MC (azul) y predichos por la UNet (rojo). Estos perfiles son los de las imágenes mostradas en la Fig. [4.13](#).

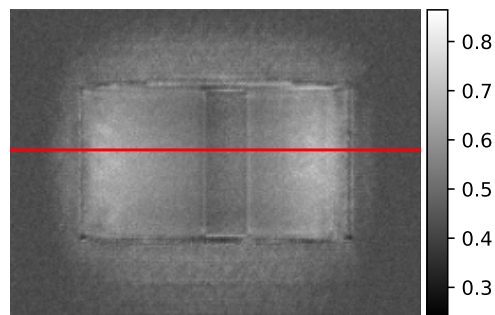


Figura 4.15: Predicción de una imagen formada por la componente de dispersión, obtenida de multiplicar una predicción de *scatter ratio* por una imagen formada por todos los fotones que llegan al detector.

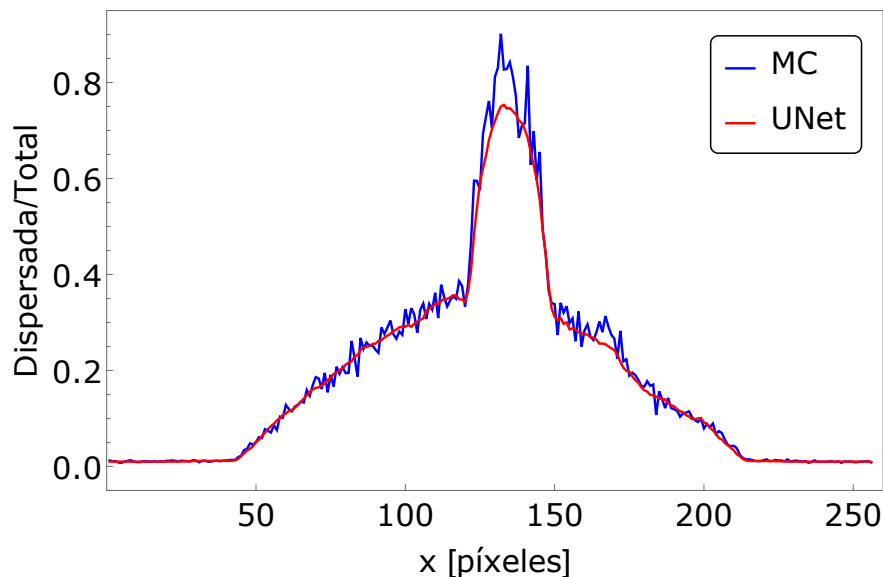


Figura 4.16: Comparación entre un perfil tomado de la imagen de la Fig. 4.15 (rojo) y la Fig. 4.3 izquierda (azul).

decir directamente *scatter* son más exactas para determinar los números CT que las de un modelo entrenado para predecir *scatter ratio*. Por otra parte, en este modelo se encontró que la SNR y el CNR mejoran significativamente, tanto en comparación con la reconstrucción de las proyecciones sin corregir, como en comparación con la de las proyecciones corregidas con la UNet descrita en la sección 4.1. Los valores de SNR y CNR para este modelo se muestran en el Anexo B, en la Tabla B.1.

4.4. Arquitecturas

Las arquitecturas implementadas fueron UNets y GANs. La principal diferencia que se encontró fue que las GANs capturan el comportamiento ruidoso de las simulaciones MC. Este comportamiento ruidoso se debe a las fluctuaciones estadísticas asociadas al número de historias de las simulaciones y no al comportamiento de la radiación dispersada [13].

La Fig. 4.17 muestra una proyección del HUP generada solo por el aporte de la radiación dispersada, de acuerdo con las predicciones de una GAN (b) y las simulaciones MC (a). La Fig. 4.18 compara los perfiles de la componente de la radiación dispersada predicha por las simulaciones MC (en azul), por una UNet (rojo) y por una GAN (verde).

La Fig. 4.18 muestra que la predicción de la GAN tiene un comportamiento ruidoso al igual que la simulación MC. Es decir, un modelo entrenado con una GAN *aprende* también las fluctuaciones estadísticas de las simulaciones MC. Esto se atribuye a que la red *generadora*

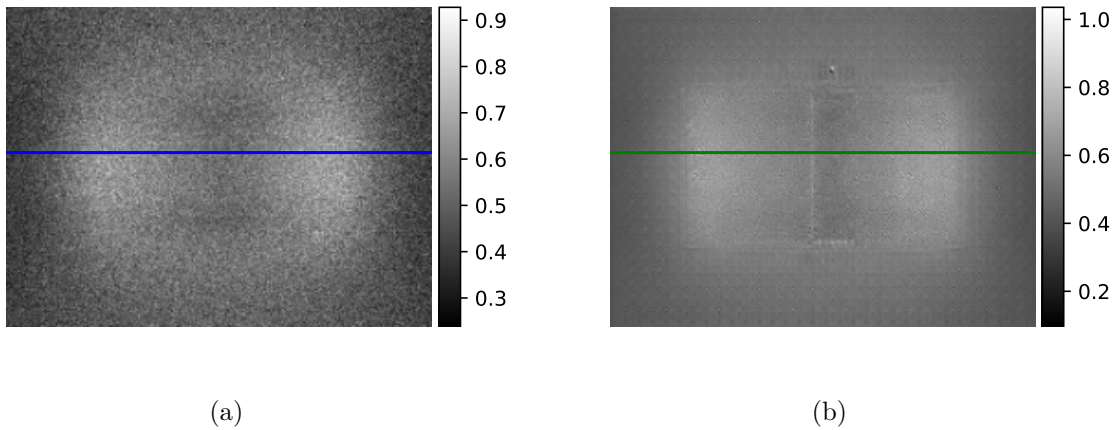


Figura 4.17: Comparación de una imagen del HUP, formada únicamente por la componente de la radiación dispersada, cuando esta es generada por las simulaciones MC (a) y cuando es predicha por la GAN (b). Las líneas azul y verde indican la región de la que se tomó los perfiles que se muestran en la Fig. 4.18.

parte de una imagen aleatoria y la va *mejorando* a lo largo de las diferentes iteraciones. Sin embargo, dado que partió de una imagen aleatoria/ruidosa, se espera que en las diferentes iteraciones aún mantenga esta característica en la imagen. Debido a que el ruido que predicen las GANs no está asociado al comportamiento físico de la radiación dispersada, se concluye que las UNets son más apropiadas para la predicción de la radiación dispersada dado que no *aprenden* el comportamiento ruidoso de las simulaciones MC.

La GAN fue entrenada durante 40 mil pasos debido a que de esta manera, su tiempo de entrenamiento fue de 15.92 min., que es comparable a los 16.43 min que tarda en entrenarse la UNet por 200 *epochs*, con el mismo número de imágenes. El corregir por radiación dispersada un estudio completo de CBCT le toma de 6 a 7 segundos a una GAN entrenada; mientras que a una UNet le toma de 2 a 3 segundos.

El error porcentual medio de las predicciones de la GAN para la componente de la radiación dispersada es de 9.19%; en comparación con 8.62% que fue el error porcentual de las predicciones de la UNet. El error porcentual de comparar todas las proyecciones corregidas por la GAN con las corregidas mediante MC (excluyendo los píxeles en los que no está el objeto, i.e., menores que 0.1) es de 0.36%.

El error porcentual medio en los números CT de la reconstrucción tomográfica corregida por este modelo es 1.14%. Esto significa que las correcciones de las UNets son más eficientes en la correcta determinación del número CT. Por otro lado, se encontró que las reconstruccio-

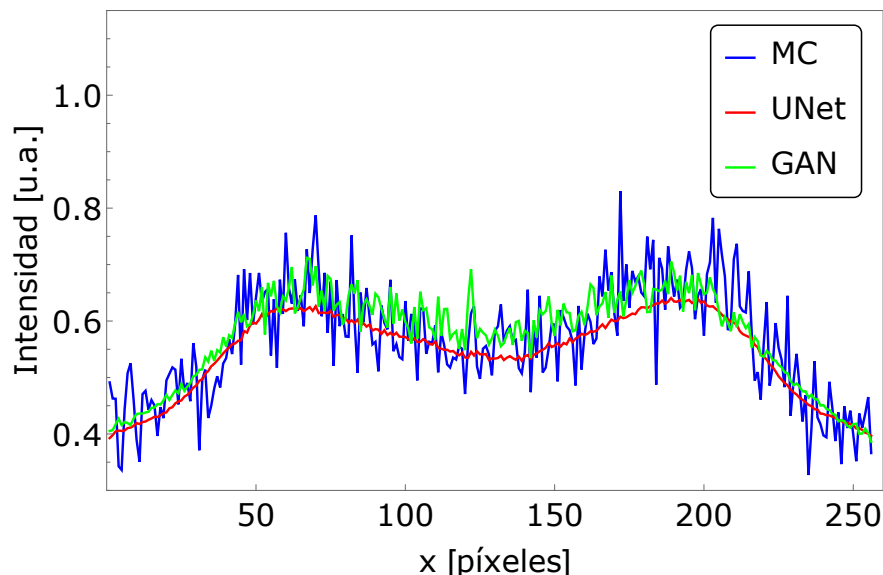


Figura 4.18: Comparación de un perfil de una imagen del HUP, formada únicamente por la componente de la radiación dispersada, cuando esta es generada por las simulaciones MC (azul) y cuando es predicha por la GAN (verde). Estos perfiles son los de las imágenes mostradas en la Fig. 4.17. Para efectos comparativos se ha incluido el perfil de las predicciones de una UNet (en rojo).

nes realizadas con las proyecciones corregidas por una GAN tienen una mejor SNR que las corregidas con una UNet. Los valores de SNR y CNR para todos los insertos se presentan en la Tabla B.2 del Anexo B.

Se concluye que las UNets son superiores a las GANs en efectividad (calculada mediante perfiles, error porcentual en las proyecciones y números CT) y eficiencia (tiempos de correcciones).

4.5. Efectos de la Energía en la Radiación dispersada

Para analizar el efecto de la radiación dispersada en función de la energía se trabajó con imágenes CBCT de kilovoltaje y de megavoltaje. Las primeras fueron obtenidas de las simulaciones MC del sistema de imagen OBI de un True Beam STX, con un espectro de 125 kV y las imágenes de MV fueron obtenidas con una simulación MC del haz de 2.5 MV, sin filtro de aplanado, del mismo equipo [13]. La Fig. 4.19 compara los espectros de kV y de MV empleados en la adquisición de los dos conjuntos de imágenes.

La Fig. 4.20 compara la distribución de la radiación dispersada en imágenes CBCT de MV del HUP, entre las predicciones MC y las predicciones de una UNet. Existe una leve diferencia entre estas imágenes y sus equivalentes en kV, mostradas en la Fig. 4.2. La Fig. 4.21 compara los perfiles de las imágenes de la Fig. 4.20. La radiación dispersada tiende a distribuirse en

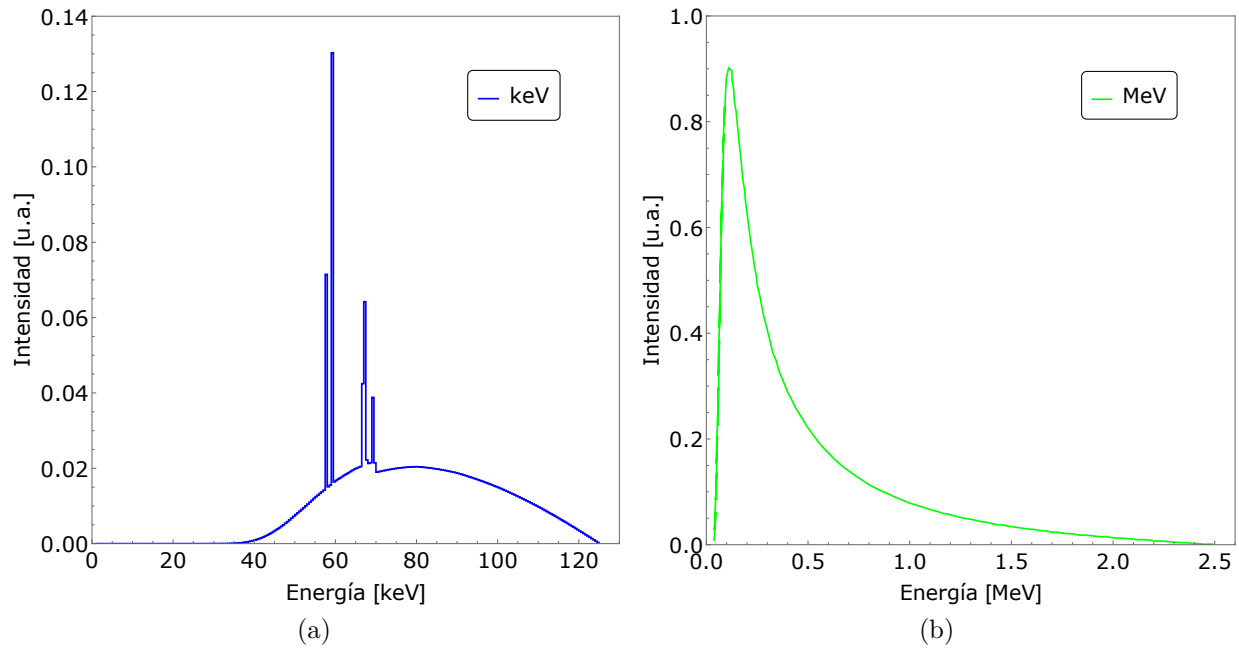


Figura 4.19: Comparación de los espectros de kV [77] (a) y MV [13] (b) utilizados en las simulaciones MC para la generación de imágenes CBCT.

mayor medida hacia el centro del cilindro.

Para efectos comparativos, en la Fig. 4.22 se muestra un perfil del comportamiento de la radiación dispersada en kV (azul) y en MV (verde). Se observan diferencias tanto en la distribución de la radiación dispersada como en su magnitud. En las imágenes de kV, la energía depositada por la radiación dispersada sobre el detector de imagen fue el 0.79% de la energía total depositada para la formación de la imagen; mientras que en las imágenes de MV, este valor fue de 1.86%.

La diferencia entre la distribución de la radiación dispersada en imágenes de kV y de MV se atribuye fundamentalmente a la dependencia direccional en las secciones eficaces de Klein-Nishina, que describen la distribución angular de fotones dispersados por Efecto Compton. La Fig. 4.23 muestra una gráfica polar que describe esta distribución para haces monoenergéticos con energía igual a la energía efectiva de los haces de kV y MV empleados. Para fotones de MV, la probabilidad de que sean dispersados en ángulos pequeños, es decir *hacia adelante*, es mucho mayor que en fotones de kV. Por lo tanto, aunque el cilindro sea más ancho en el centro, debido a la predominancia de las dispersiones *hacia adelante* se observa una mayor dispersión en el centro.

La diferencia en la magnitud de la radiación dispersada se atribuye a que los fotones del

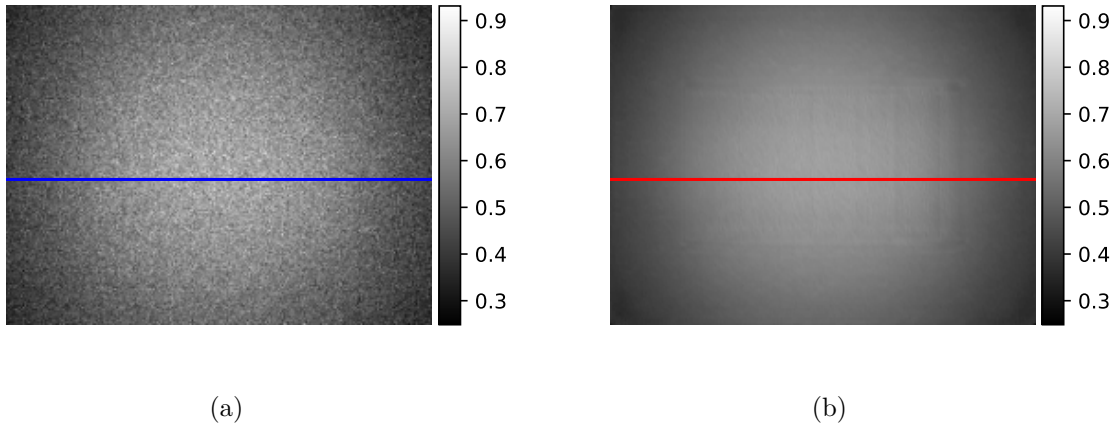


Figura 4.20: Comparación de una imagen del HUP, formada únicamente por la componente de la radiación dispersada, cuando esta es generada por las simulaciones MC (a) y cuando es predicha por la UNet (b). Las líneas azul y roja indican la región de la que se tomó los perfiles que se muestran en la Fig. 4.21.

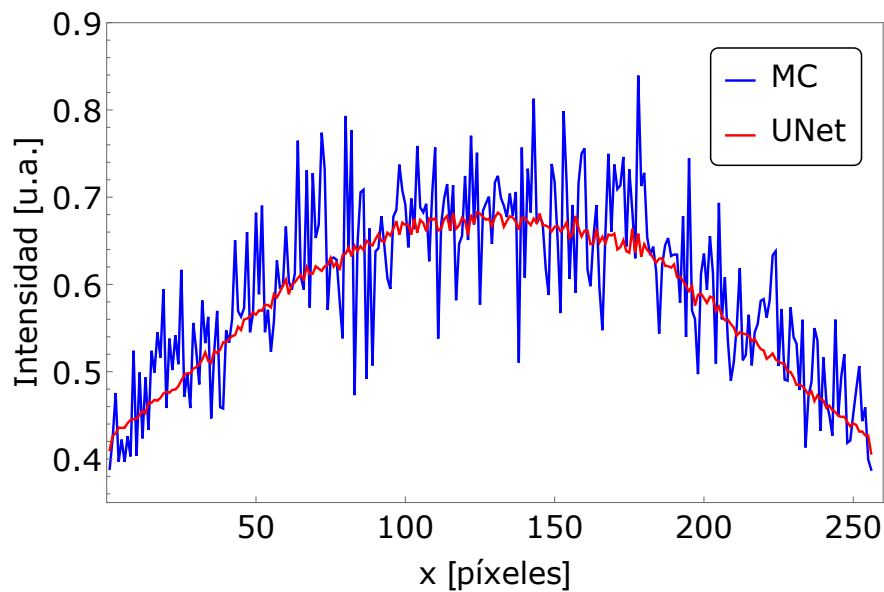


Figura 4.21: Comparación de un perfil de una proyección del HUP, adquirida con el haz de MV, formada únicamente por la componente de la radiación dispersada, cuando esta es generada por las simulaciones MC (azul) y cuando es predicha por la UNet (rojo). Estos perfiles son los de las imágenes mostradas en la Fig. 4.20.

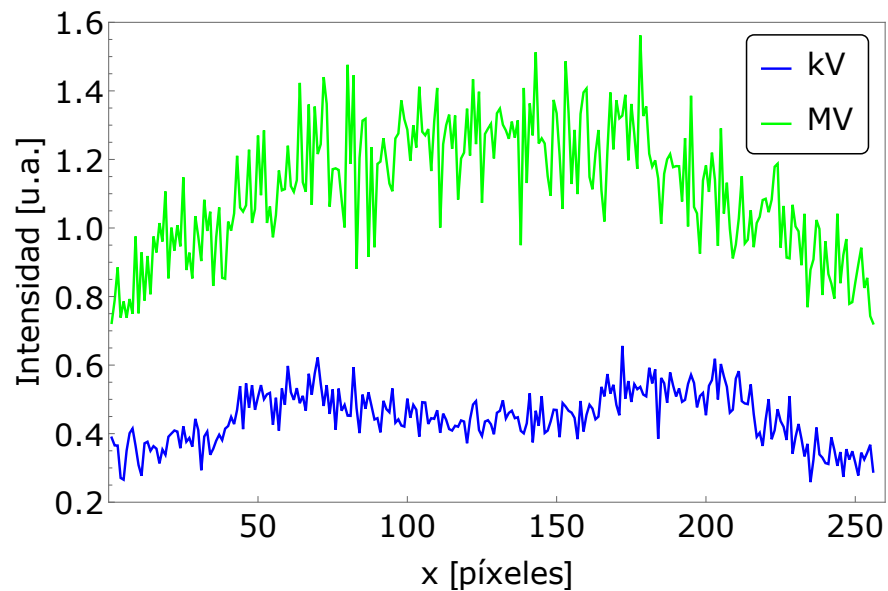


Figura 4.22: Comparación de un perfil de una proyección del HUP, adquirida con el haz de kV (en azul) y con el haz de MV (en verde), formada únicamente por la componente de la radiación dispersada, cuando esta es generada por las simulaciones MC. Estos perfiles son los de las imágenes mostradas en la parte izquierda de las Fig. 4.2 y 4.20.

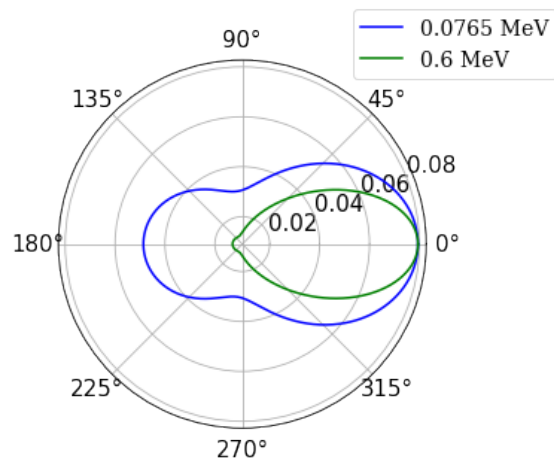


Figura 4.23: Gráfica de la sección eficaz diferencial, que describe la probabilidad de la distribución angular de fotones dispersados por Efecto Compton, en función de la energía del haz incidente. Las curvas presentadas son las de la energía efectiva de los haces empleados en la adquisición de las imágenes.

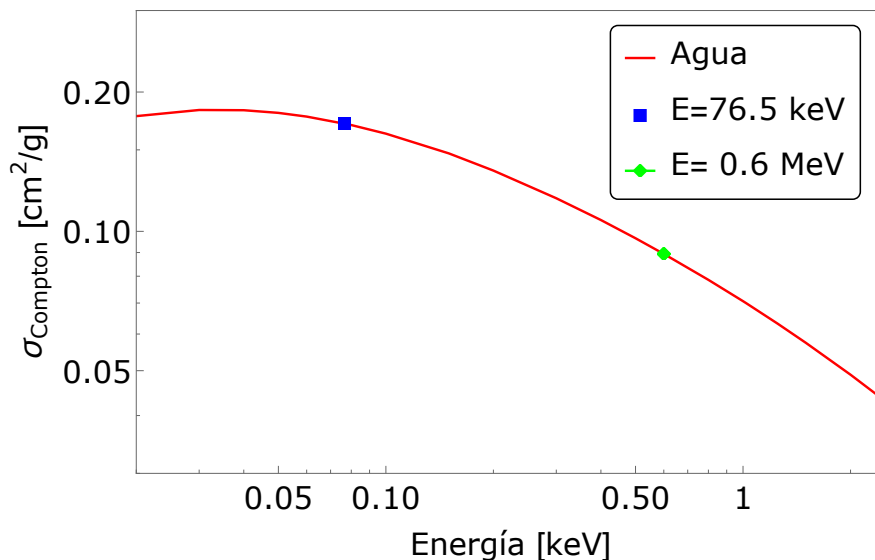


Figura 4.24: Comportamiento de la sección eficaz para el Efecto Compton en agua, en función de la energía. Se resalta los valores a la energía efectiva de los haces empleados en la adquisición de imágenes.

haz de MV tienen una mayor probabilidad de dispersión frontal, por lo que la mayoría de los dispersados son detectados, mientras muchos fotones de kV son retrodispersados o dispersados en ángulos que no les permiten llegar al detector. Sin embargo, este efecto es contrarrestado por la probabilidad de interacción por Efecto Compton entre la radiación y la materia para haces de MV. La Fig. 4.24 indica el comportamiento de la sección eficaz para Efecto Compton, σ_{Compton} , del agua en función de la energía. Existe una diferencia de casi un orden de magnitud entre el valor de μ a la energía efectiva del haz de MV y del haz de kV.

El error porcentual medio calculado al comparar la componente de la radiación dispersada predicha por esta UNet, con sus respectivas imágenes del *ground truth* es de 7.82%. El error porcentual de comparar todas las proyecciones corregidas por la UNet con las corregidas mediante MC (excluyendo los píxeles donde no está el objeto, i.e., menores que 0.2) es de 0.35%. La diferencia porcentual entre las proyecciones corregidas por la UNet y las proyecciones sin corregir es de 4.66%, lo cual indica el efecto de ejecutar las correcciones por radiación dispersada con esta UNet. A pesar de que el error porcentual encontrado entre las predicciones de la UNet y las simulaciones MC es levemente inferior en MV que en kV, este es un valor alto. Esto se debe fundamentalmente a la estadística de las simulaciones MC, como se observa en la Fig. 4.3

Tabla 4.2: Comparación de los números CT, CNR y SNR en los insertos del HUP en MV, de acuerdo con la imagen reconstruida con proyecciones totales.

Material	$\#CT_{MV}$ [HU]	$\#CT_{ref}$ [HU]	Error [%]	SNR	CNR
Proyecciones primarias					
Agua	0 ± 3	—	—	10.21 ± 0.69	—
Hueso Trabecular	138 ± 10	—	—	12.52 ± 0.87	0.18 ± 0.11
Tejido Adiposo	-17 ± 10	—	—	14.26 ± 1.01	1.42 ± 0.15
Hueso Cortical	939 ± 8	—	—	7.58 ± 0.55	4.44 ± 0.32
Pulmón Exhalado	-435 ± 10	—	—	22.15 ± 1.60	9.59 ± 0.65
Proyecciones totales					
Agua	0 ± 3	0 ± 3	—	10.10 ± 0.65	—
Hueso Trabecular	154 ± 10	138 ± 10	11.59	12.44 ± 0.84	0.01 ± 0.11
Tejido Adiposo	0 ± 10	-17 ± 10	-100.00	14.14 ± 1.05	1.57 ± 0.15
Hueso Cortical	946 ± 7	939 ± 8	0.75	7.65 ± 0.56	4.22 ± 0.30
Pulmón Exhalado	-417 ± 10	-435 ± 10	-4.14	21.62 ± 1.47	9.56 ± 0.62
Proyecciones corregidas con una UNet					
Agua	0 ± 3	0 ± 3	—	9.97 ± 0.66	—
Hueso Trabecular	139 ± 10	138 ± 10	0.72	12.23 ± 0.85	0.21 ± 0.11
Tejido Adiposo	-20 ± 10	-17 ± 10	17.65	13.99 ± 1.07	1.39 ± 0.14
Hueso Cortical	956 ± 8	939 ± 8	1.81	7.30 ± 0.54	4.45 ± 0.32
Pulmón Exhalado	-446 ± 10	-435 ± 10	2.53	21.91 ± 1.57	9.54 ± 0.64

La Tabla [4.2](#) presenta los números CT de referencia (de la reconstrucción con primarios), los determinados en una reconstrucción realizada sin correcciones por radiación dispersada y los determinados en una reconstrucción realizada con proyecciones corregidas por radiación dispersada mediante una UNet. Además se presentan los valores de SNR y CNR para los diferentes materiales del HUP. En el caso de la reconstrucción realizada sin correcciones por radiación dispersada se identificó un error porcentual medio es de 29.12%, con respecto a los valores de referencia. En las reconstrucciones realizadas con proyecciones corregidas por radiación dispersada mediante una UNet, el error porcentual medio en la determinación de los números CT se redujo a 5.68%. Este error tiene un valor alto debido al error medido en el tejido adiposo, el cual tiene un número CT relativamente cercano a cero y que en el caso de una reconstrucción realizada sin correcciones por radiación dispersada, casi sería indistinguible del agua. Los números CT calculados para los insertos del HUP, considerando la composición reportada por el fabricante y la energía efectiva del espectro de MV son

0, 117, -51 , 979 y -501 para el agua, hueso trabecular, tejido adiposo, hueso cortical y pulmón exhalado, respectivamente. Estos valores difieren de los obtenidos de proyecciones primarias con simulaciones MC debido a que la energía efectiva no captura completamente el comportamiento de un haz polienergético. No se encontraron diferencias significativas en la SNR ni en el CNR de las reconstrucciones corregidas con respecto a las reconstrucciones realizadas sin correcciones.

La Fig. 4.25 presenta el promedio de los cortes de la reconstrucción tomográfica del HUP en MV, obtenida con proyecciones corregidas con una UNet. Las correcciones con la UNet no inducen ningún tipo de artefactos en la imagen reconstruida.

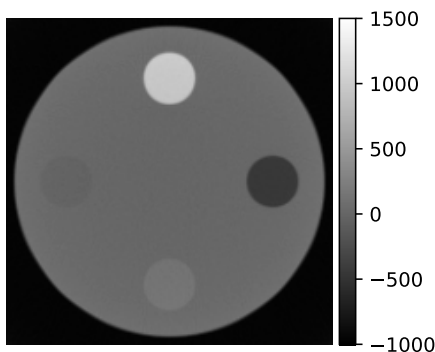


Figura 4.25: Promedio de los cortes de la reconstrucción tomográfica del HUP en MV, obtenida con proyecciones corregidas con una UNet.

4.6. Efecto de la función de pérdidas

La función de pérdidas cuantifica las discrepancias entre las predicciones de un modelo y el *ground truth* a medida que se lo entrena y determina la dirección en la que se deben modificar los parámetros del modelo para minimizarla. Se consideran cuatro indicadores como función de pérdidas y se analiza la diferencia en los modelos entrenados en cada caso. Los indicadores que han sido empleados como función de pérdidas son el error absoluto medio (MAE), el error cuadrático medio (MSE), el índice de similitud estructural (SSIM) y el cociente señal máxima a ruido (PSNR).

El MAE o MSE son valores que se desea minimizar para tener mejores resultados por lo que su función de pérdidas se define igual a estos valores. En contraste, el SSIM y PSNR son índices que crecen cuando la similitud entre las imágenes es mayor, por lo que se desea maximizarlos. Debido a que el SSIM toma valores entre -1 y 1 , su función de pérdidas fue

Tabla 4.3: Comparación del MAE, MSE, SSIM, PSNR y error porcentual en el conjunto de evaluación con los modelos I, II, III y IV. Se ha resaltado en negrita los mejores resultados.

Modelo	MAE	MSE	SSIM	PSNR	$E_{\%}$ [%]
I	0.0434	0.0030	0.2743	25.2626	8.54
II	0.0439	0.0031	0.2701	25.1822	8.68
III	0.0570	0.0053	0.3150	22.7759	11.29
IV	0.0430	0.0030	0.2817	25.3688	8.61

definida mediante la ecuación (4.2). El PSNR típicamente toma valores entre 20 y 60 por lo que su función de pérdidas fue definida mediante la ec. (4.3).

$$LOSS_{SSIM} = 1 - SSIM \quad (4.2)$$

$$LOSS_{PSNR} = 1/PSNR \quad (4.3)$$

Para analizar el efecto de la función de pérdidas se comparó el rendimiento entre los modelos entrenados para minimizar el MAE, MSE, SSIM y PSNR, a los cuales se los llamó modelos *I*, *II*, *III* y *IV*, respectivamente. En todos los casos se consideró un modelo entrenado con proyecciones del HUP. En la tabla 4.3 se presenta los valores de MAE, MSE, SSIM, PSNR y error porcentual medio ($E_{\%}$), medidos en el conjunto de evaluación o *test set*, para todos los modelos considerados. Los modelos entrenados con una función de pérdidas basada en el MAE (I) y en el PSNR (IV) son los que muestran los mejores resultados.

Las Tablas 4.1 y B.3 presentan los valores encontrados para la SNR, CNR, los números CT de los materiales del HUP y su error porcentual con respecto a las reconstrucciones corregidas con métodos MC [13], para los casos en que se realiza las reconstrucciones de las proyecciones corregidas con los modelos I-IV. En las imágenes reconstruidas con las proyecciones corregidas con los modelos I, II, III y IV se determinó que el error porcentual medio en los números CT es de 0.48 %, 1.65 %, 4.30 % y 1.60 %, respectivamente. Por lo tanto, el modelo que determina los números CT con una mayor exactitud es la UNet que tiene una función de pérdidas basada en el MAE, mientras que un modelo con función de pérdidas basado en el SSIM produce reconstrucciones con una menor exactitud en los números CT. También se destaca que una UNet con una función de pérdida basada en el SSIM mejora significativamente la SNR y CNR con respecto a una UNet basada en el MAE, como se muestra en la Tabla B.3. Esto se atribuye a que el SSIM es una métrica basada en la estructura de las imágenes.

4.7. Efecto del submuestreo en la distribución de la radiación dispersada

Para evaluar los efectos del submuestreo en la distribución de la radiación dispersada, primero se redujo el tamaño de las proyecciones con las que se entrenó el algoritmo de 192×256 píxeles a 96×128 píxeles. Esta operación fue realizada con la función *Image.resize* de la Librería de Imágenes de *Python* (*PIL*, por sus siglas en inglés), la cual utiliza el método de ‘vecinos más cercanos’ o *nearest neighbors* para el re-muestreo. Luego, se modificó la arquitectura de la UNet de tal manera que en lugar de la mostrada en la Fig. 3.4 se tuvo la arquitectura mostrada en la Fig. 4.26. Una vez que se ejecuta el algoritmo, la salida son imágenes de 96×128 píxeles compuestas únicamente por el aporte de la radiación dispersada. Posteriormente estas imágenes fueron ampliadas a 192×256 píxeles con la función *Image.resize* de PIL. Con estas imágenes se efectuó las correcciones por radiación dispersada empleando la ec. (3.2).

De esta manera, al cambiar las imágenes de entrada de 192×256 por imágenes de 96×128 , el número de parámetros entrenables en la UNet se redujo de 53.48 millones a 13.37 millones, esto es a un 25 %. Además, el tiempo de entrenamiento se redujo de 16.43 min. a 4.42 min., esto es a un 26.90 %. Y el número de capas del modelo se redujo de 71 a 60. Por otro lado, el error porcentual en las componentes de la radiación dispersada cambió de 8.62 % a 8.59 %; mientras que en las proyecciones corregidas cambió de 0.08 % a 0.05 %. Por lo tanto, usar submuestreo reduce los tiempos de entrenamiento en un factor proporcional a la reducción en la cantidad de píxeles. Con una reducción del 75 % de los píxeles en las imágenes usadas, no se detectó ninguna afectación en la precisión de las correcciones por radiación dispersada. Esto comprueba que la radiación dispersada está compuesta por información de baja frecuencia espacial [43], por lo que trabajar con versiones reducidas de las proyecciones en la entrada del algoritmo, no reduce la información sobre la distribución espacial de la radiación dispersada.

La Tabla B.4 presenta la SNR, CNR, números CT y su error porcentual de la imagen reconstruida a partir de las proyecciones corregidas con la UNet entrenada con submuestreo. El error porcentual medio encontrado en los números CT en la imagen tomográfica reconstruida fue de 0.51 %, valor que es comparable con el error porcentual (0.48 %) encontrado cuando la UNet es entrenada con imágenes de 192×256 píxeles. En la SNR y CNR no se encontraron diferencias significativas con respecto a la imagen reconstruida sin correcciones.

Por lo tanto, disminuir las dimensiones de las imágenes de entrada prácticamente no afecta al rendimiento de los algoritmos de *deep learning* en la determinación de la radiación dispersada y tiene la ventaja de reducir los tiempos de entrenamiento en un factor prácticamente

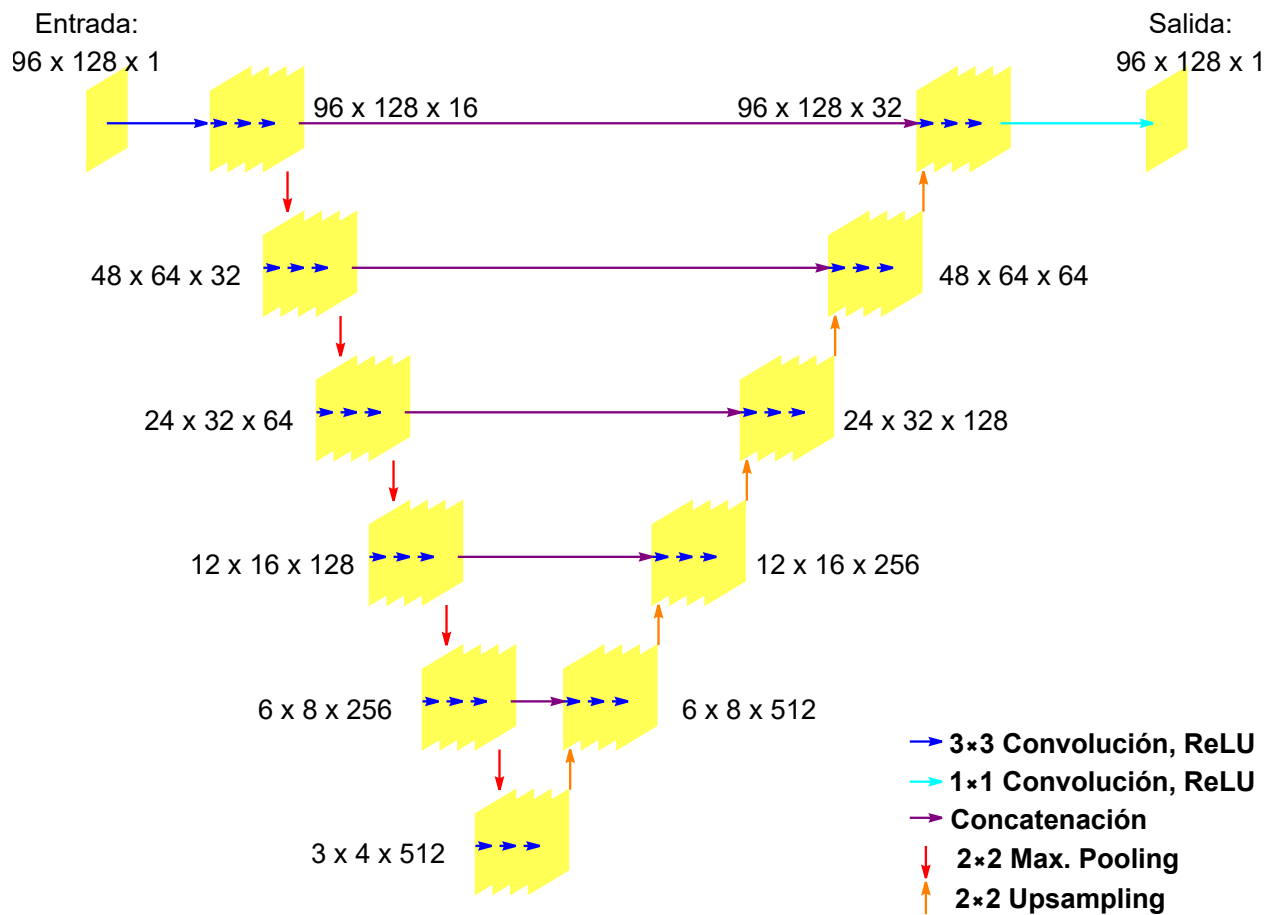


Figura 4.26: Representación esquemática de la arquitectura empleada para las UNets cuando la entrada son imágenes de 96×128 píxeles.

proporcional a la reducción en número de píxeles.

4.8. Transferencia de Aprendizaje

La transferencia de aprendizaje o *transfer learning* es una técnica ampliamente utilizada en algoritmos de inteligencia artificial con el objetivo de tomar ventaja de un modelo entrenado. La idea es partir de un modelo base que ha sido entrenado para resolver un problema X y se desea re-entrenarlo y aplicarlo en la resolución de un problema Y . En general, el modelo base es entrenado con un conjunto genérico de datos y al aplicar *transfer learning* el re-entrenamiento ocurre con un conjunto específico de datos. En el re-entrenamiento solo se entrena los parámetros de las capas finales de red neuronal convolucional, reduciendo el tiempo de entrenamiento y manteniendo el *conocimiento* de la red sobre el problema genérico X . Además, al prescindir de un conjunto de datos extenso, los tiempos de entrenamiento

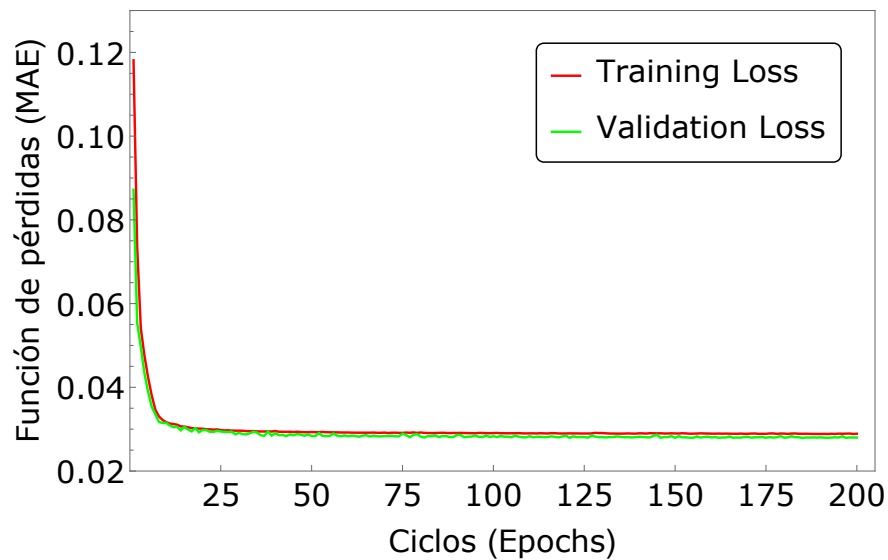


Figura 4.27: Comportamiento de la función de pérdidas para los conjuntos de entrenamiento y validación en el caso *A*.

pueden reducirse más.

Se evaluó la técnica de *transfer learning* tomando como base a un modelo entrenado con las proyecciones de uno de los fantasmas y tras re-entrenar las últimas 15 de las 71 capas del modelo original, el nuevo modelo fue aplicado a las proyecciones del otro fantoma. De igual manera se procedió entre modelos entrenados con datos del HUP en kV y en MV. Se encontró que los resultados son distintos según el conjunto de datos que se tome para el modelo base y el conjunto de datos al que se aplique *transfer learning*.

En específico se trabajó con cuatro casos. El caso *A* toma como base un modelo entrenado con proyecciones del HUP y se lo emplea para corregir las proyecciones de MOBY. El caso *B* toma como base un modelo entrenado con proyecciones de MOBY y se lo emplea para corregir las proyecciones del HUP. El caso *C* toma como base un modelo entrenado con proyecciones en kV del HUP y se lo emplea para corregir sus proyecciones en MV. Y el caso *D* toma como base el modelo entrenado con proyecciones en MV y se lo emplea para corregir proyecciones en kV.

Las Figs. [4.27](#) - [4.30](#) comparan la función de pérdidas del proceso de re-entrenamiento de los casos *A-D*, respectivamente. En todos los casos, la función de pérdidas tiene valores bajos desde el comienzo del entrenamiento, en comparación con los valores de la Fig. [4.1](#).

En el caso *A*, el modelo entrenado con *transfer learning* encuentra una solución estable en alrededor de *25 epochs*. En el caso *B*, la función de pérdidas en el conjunto de validación es significativamente mayor que en el conjunto de entrenamiento incluso tras *200 epochs*. En

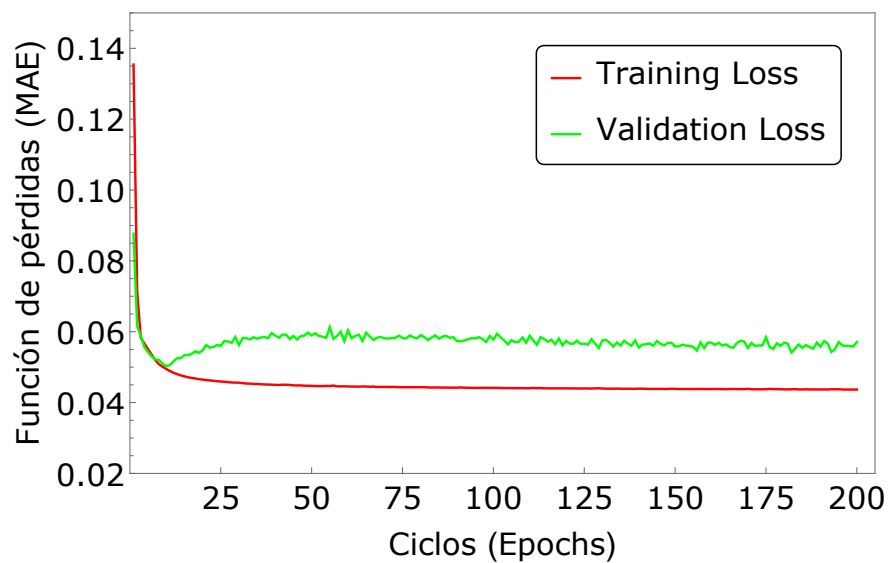


Figura 4.28: Comportamiento de la función de pérdidas para los conjuntos de entrenamiento y validación en el caso *B*.

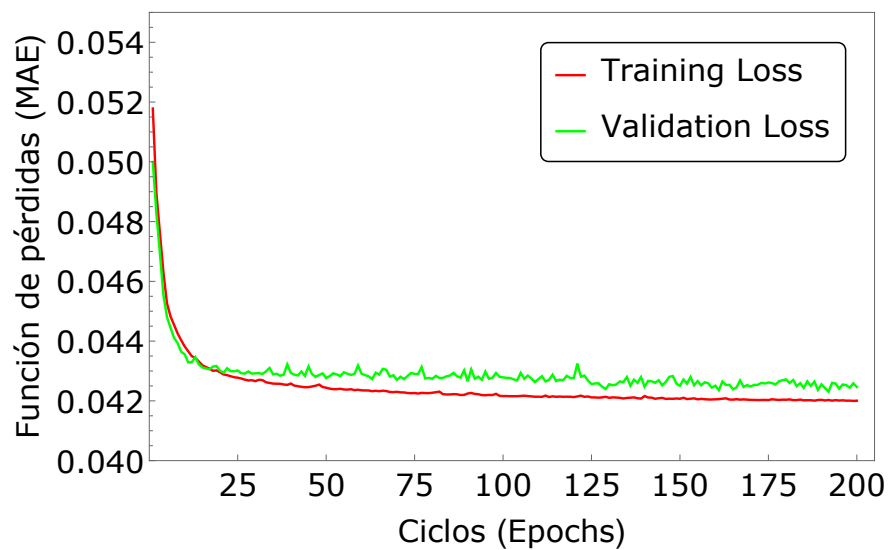


Figura 4.29: Comportamiento de la función de pérdidas para los conjuntos de entrenamiento y validación en el caso *C*.

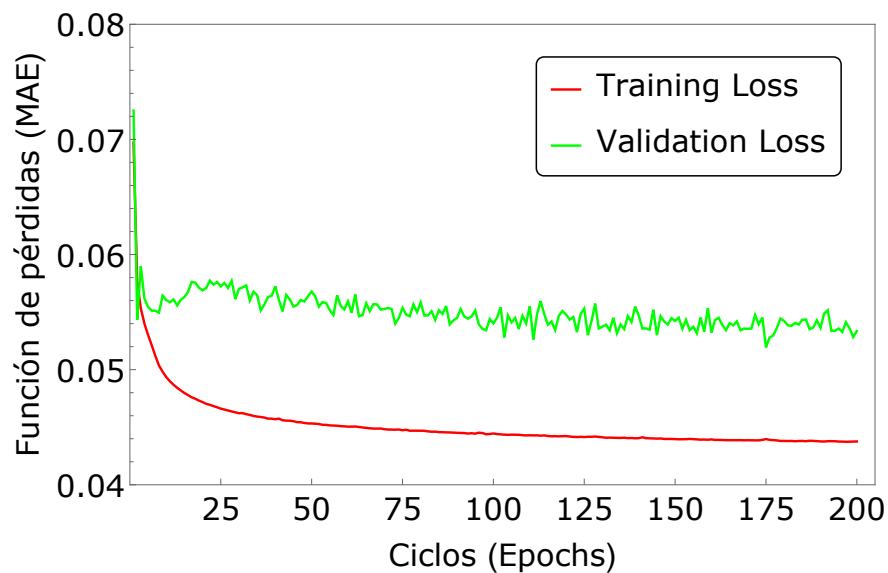


Figura 4.30: Comportamiento de la función de pérdidas para los conjuntos de entrenamiento y validación en el caso D .

los casos C y D se observa que desde las primeras iteraciones, la función de pérdidas tiene valores pequeños, tanto en el conjunto de entrenamiento, como en el conjunto de validación. Esto se debe a que la distribución de la radiación dispersada en kV y MV es relativamente similar para un mismo objeto.

De los tres conjuntos de datos, solo las componentes de *scatter* del HUP en kV muestran una estructura similar al objeto; mientras que las componentes de *scatter* de MOBY o las del HUP en MV no muestran una estructura definida, sino que muestran una forma como de *mancha* en forma de sombra del objeto. De esta manera, se ha encontrado que al re-entrenarse las últimas capas, un modelo entrenado para predecir imágenes con estructura puede adaptarse a predecir imágenes sin estructura. Sin embargo, si se parte de un modelo entrenado para predecir imágenes sin estructura, es más *difícil* para el modelo *aprender* a predecir imágenes con estructura solo re-entrenando las últimas capas.

Las Figs. [4.31](#)-[4.34](#) presentan la imagen esperada (a), predicha por el modelo (b) y una comparación de sus perfiles (c), para los casos A , B , C y D , respectivamente. La Tabla [4.4](#) compara el error porcentual promedio medido del aporte de la radiación dispersada en un conjunto completo de proyecciones, el SSIM medido en el conjunto de evaluación o *test set* y el tiempo de entrenamiento entre los modelos entrenados con *transfer learning* en los cuatro casos considerados y los modelos equivalentes entrenados desde cero.

En los casos A y C no existen diferencias significativas en los índices de calidad de imagen

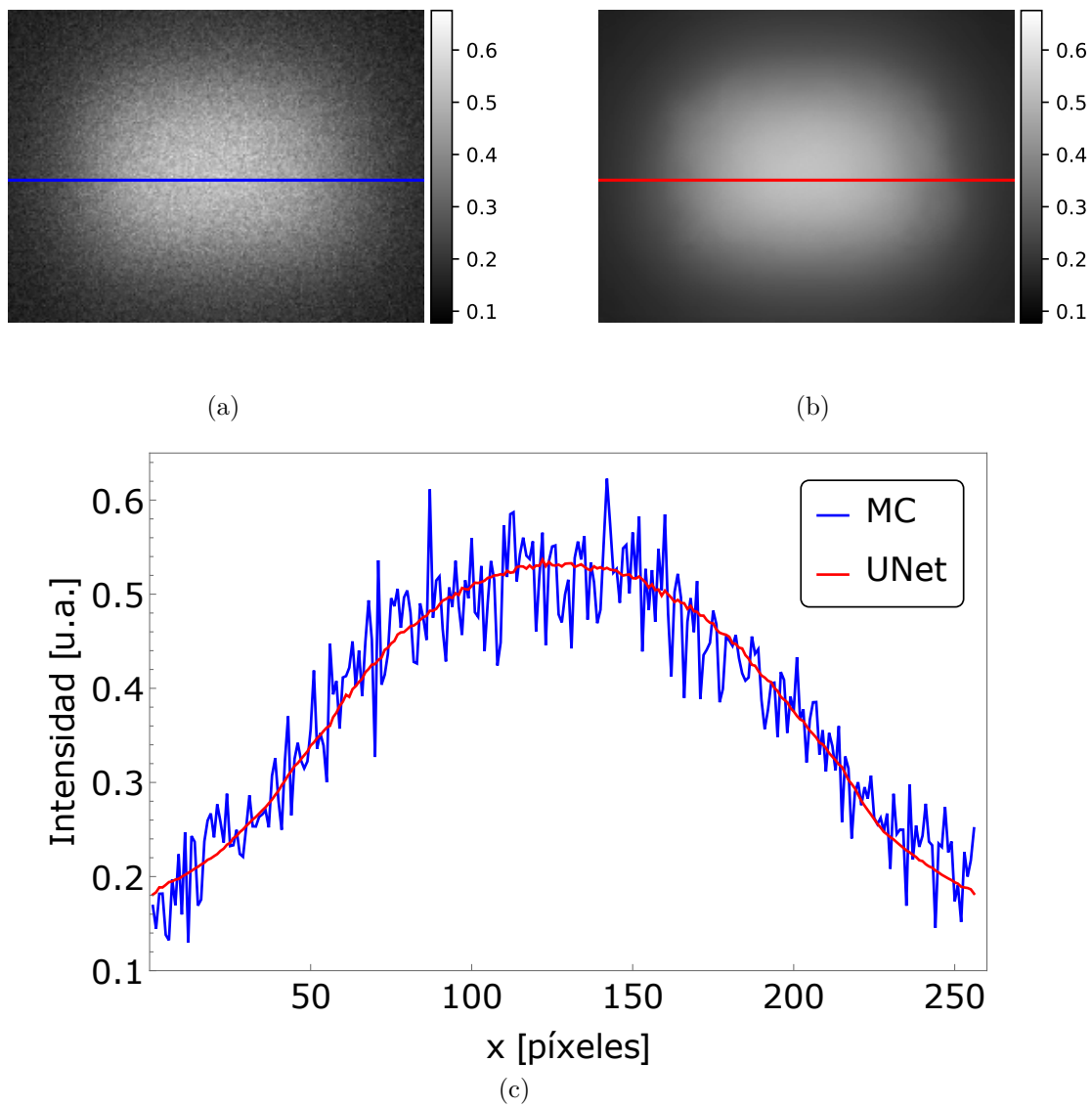


Figura 4.31: Comparación de imágenes formadas únicamente con radiación dispersada de acuerdo con las simulaciones MC (a) y las predichas por el modelo entrenado con *transfer learning* en el caso A (b) y una comparación de sus perfiles (c).

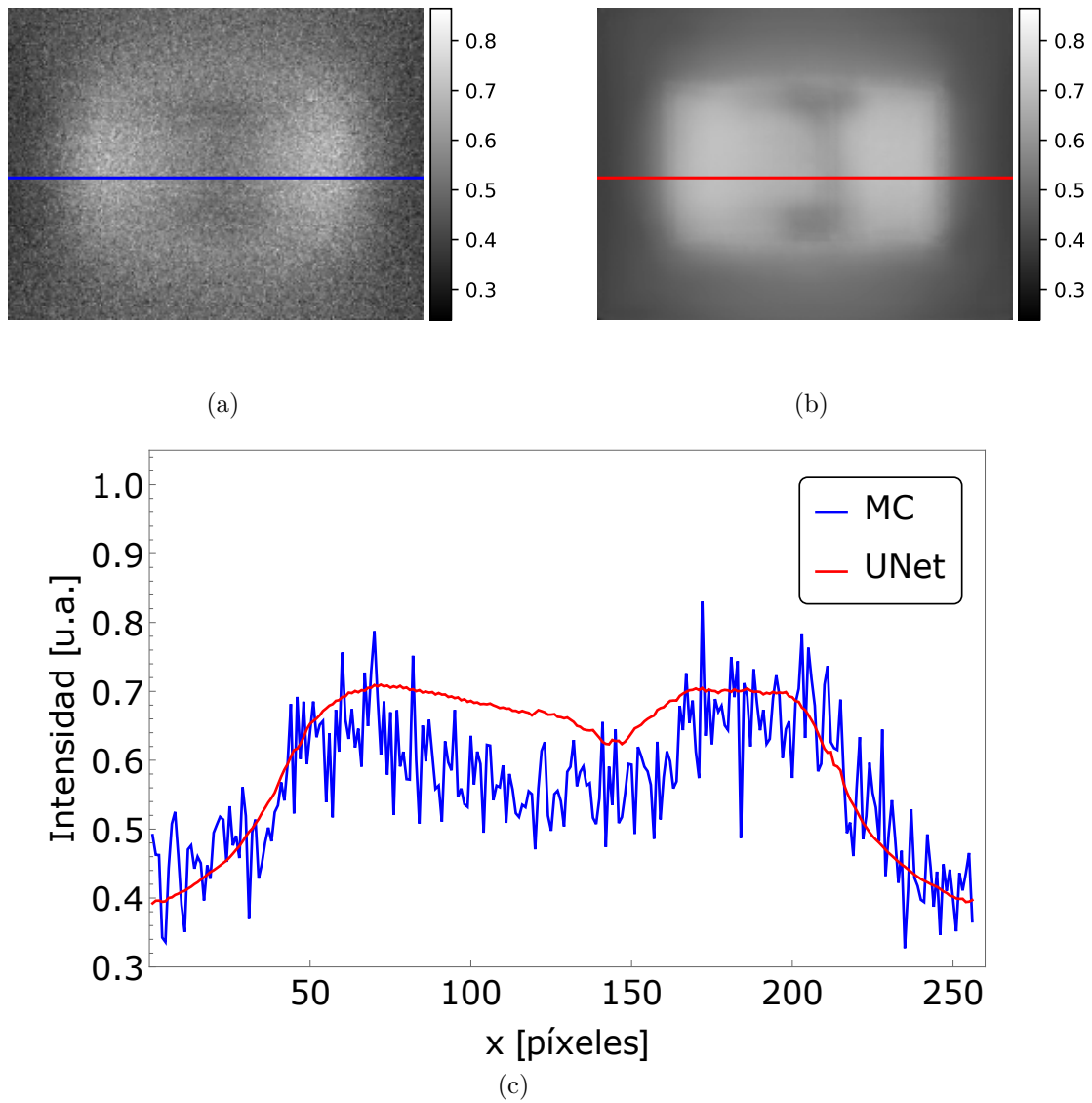


Figura 4.32: Comparación de imágenes formadas únicamente con radiación dispersada de acuerdo con las simulaciones MC (a) y las predichas por el modelo entrenado con *transfer learning* en el caso *B* (b) y una comparación de sus perfiles (c).

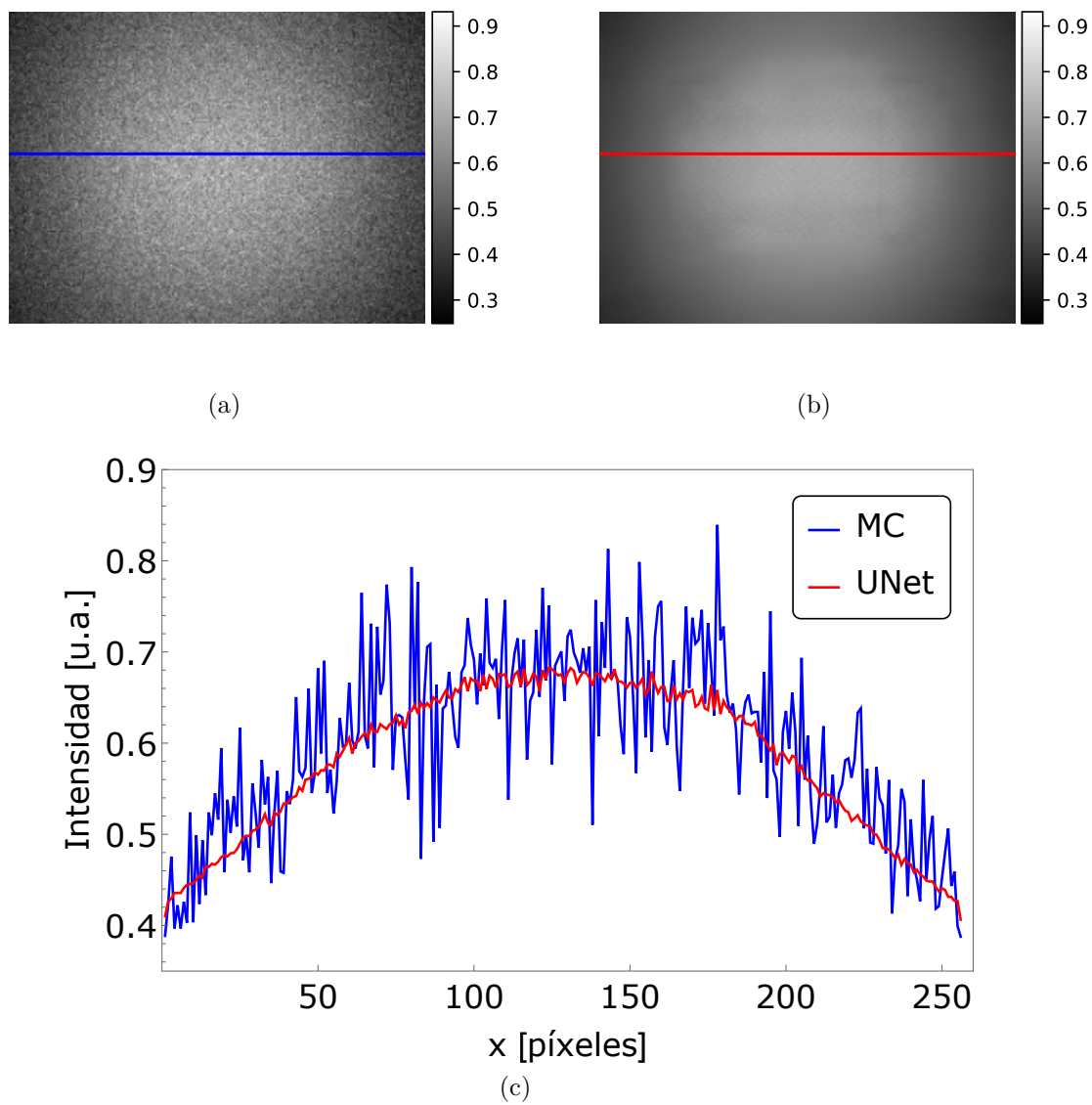


Figura 4.33: Comparación de imágenes formadas únicamente con radiación dispersada de acuerdo con las simulaciones MC (a) y las predichas por el modelo entrenado con *transfer learning* en el caso *C* (b) y una comparación de sus perfiles (c).

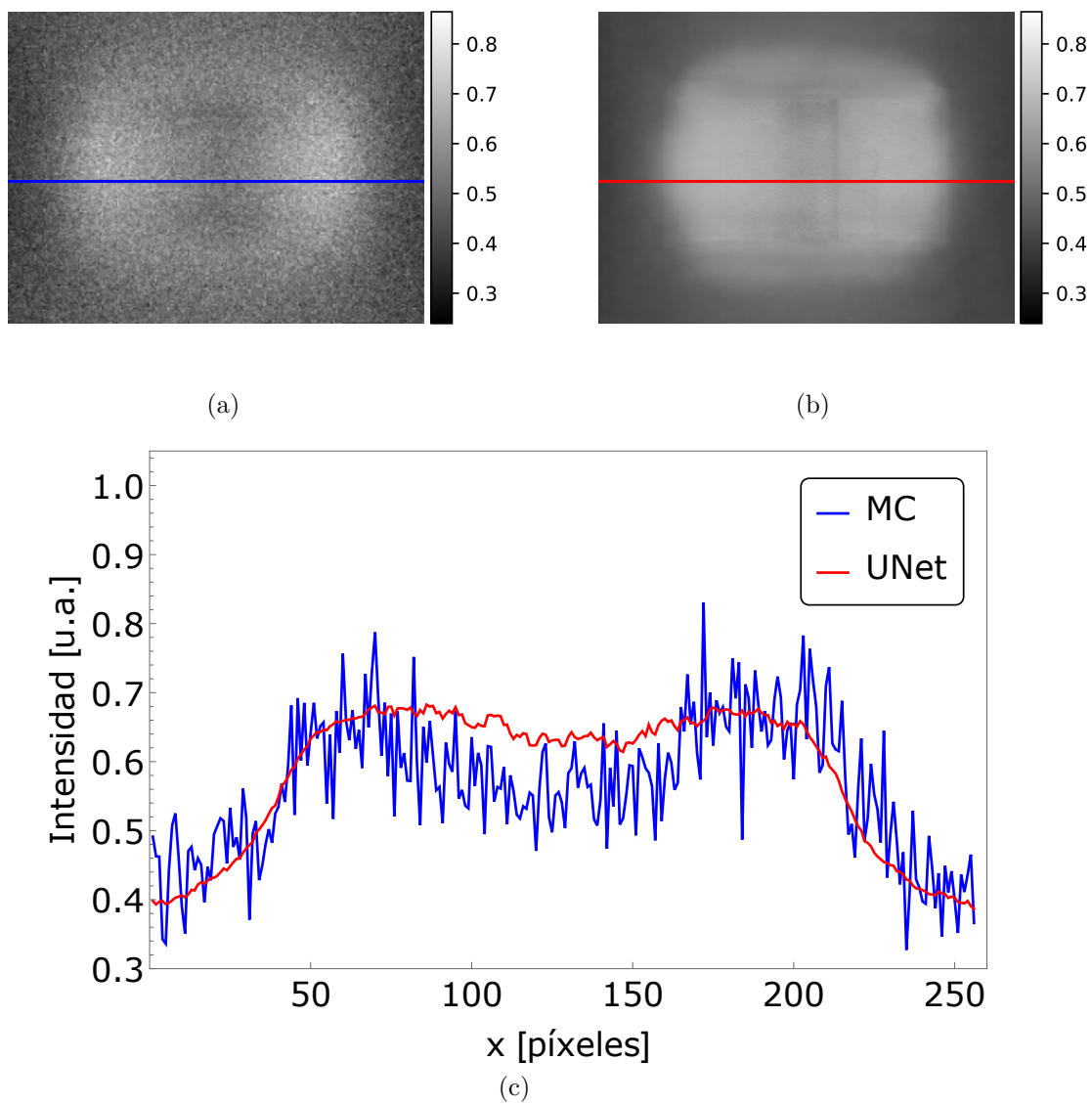


Figura 4.34: Comparación de imágenes formadas únicamente con radiación dispersada de acuerdo con las simulaciones MC (a) y las predichas por el modelo entrenado con *transfer learning* en el caso D (b) y una comparación de sus perfiles (c).

Tabla 4.4: Comparación de error porcentual en las proyecciones de un estudio completo, el SSIM en el *test set* y los tiempos de entrenamiento para *200 epochs* entre modelos entrenados con *transfer learning* y entrenados desde cero, para los casos *A*, *B*, *C* y *D*.

Entrenados con <i>Transfer Learning</i>				Entrenados desde cero		
Caso	$E_{\%}$ [%]	SSIM	t_{train} [min]	$E_{\%}$ [%]	SSIM	t_{train} [min]
A	10.83	0.4714	9.42	10.53	0.4665	15.42
B	11.29	0.2427	9.42	8.62	0.2743	16.43
C	7.90	0.2800	10.43	7.82	0.2789	17.49
D	10.52	0.2809	10.44	8.62	0.2743	16.43

y existen reducciones de hasta un 37.92% en el tiempo de entrenamiento. En los casos *B* y *D* el error porcentual en las predicciones de la componente de la radiación dispersada con respecto al *ground truth* se incrementa en 2.67% y 1.90%, respectivamente. En ambos casos, las discrepancias entre los perfiles de la salida del algoritmo y el *ground truth* son notorias como se muestran en las Figs. 4.32 y 4.34.

Además se encontró que en las reconstrucciones de los casos *B* y *D*, el error porcentual medio en la determinación de los números CT fue de 6.61% y 5.92% con respecto a las reconstrucciones de primarios, respectivamente. Un modelo equivalente, entrenado sin *transfer learning*, tuvo un error porcentual medio de 0.48% en la determinación de los números CT. Esto significa que un modelo entrenado con *transfer learning* puede producir reconstrucciones con una determinación menos exacta en los números CT. Sin embargo, estos son más exactos que los de una reconstrucción sin correcciones por radiación dispersada (su error porcentual medio fue de 25.88%).

En el caso *C*, el error porcentual medio en los números CT de las reconstrucciones fue de 1.94%, valor que es incluso menor al error porcentual detectado en un modelo equivalente, entrenado desde cero (5.68%). Esto se atribuye a que el modelo, al ser primero entrenado con proyecciones en kV y luego en MV adquiere información adicional sobre el comportamiento de la radiación dispersada, por lo que realiza mejores predicciones.

En ninguno de estos modelos se encontró cambios significativos en la SNR ni CNR con respecto a reconstrucciones sin corrección por radiación dispersada. La Tabla B.5 presenta la SNR, CNR, números CT y sus errores porcentuales en las reconstrucciones de los casos *B*, *C* y *D*, para todos los materiales del HUP.

La técnica de *transfer learning* es una herramienta muy útil para tomar ventaja de la información almacenada en un modelo previamente entrenado y así reducir los tiempos de entrenamiento, mejorando la exactitud en la determinación de los números CT con respec-

to a reconstrucciones sin corrección por radiación dispersada. Sin embargo, sus resultados pueden ser menos exactos en la determinación de los números CT en comparación con un modelo entrenado desde cero.

4.9. Rendimiento de los GPUs de los equipos empleados

Finalmente se evaluó la diferencia en tiempos de corrección y entrenamiento con diferentes equipos. Los tiempos reportados en tablas previas son los tiempos medidos en Google Colab, con GPU. La Tabla 4.5 compara la diferencia en los tiempos de corrección de un estudio completo y en los tiempos de entrenamiento de un mismo modelo. Los tiempos reportados en esta comparación son los de una UNet entrenada para predecir *scatter*, por 200 epochs, con 252 proyecciones en kV del HUP.

En el clúster *Holiday* del Instituto de Física de la UNAM se cuenta con un GPU NVIDIA Volta V100 con 32 GB de RAM [75]. En la estación de trabajo (*Workstation*) *Alice* del Laboratorio de Imágenes Biomédicas se cuenta con un GPU NVIDIA RTX 3060 con 12 GB de RAM [76].

En un GPU con 32 GB de RAM, los algoritmos de *deep learning* implementados pueden ser entrenados incluso en menos de 5 minutos por 200 epochs con más de 250 proyecciones. En un GPU de 12 GB de RAM, como el de *Alice*, el entrenamiento comienza a tomar mucho tiempo, sobretodo si se busca trabajar con un conjunto de datos más extenso. Sin embargo, estos modelos también pueden ser entrenados en GPUs de manera remota mediante *Google Colab*, una plataforma a la que se puede acceder desde cualquier computador, tablet o celular sin costo por múltiples horas al día.

En conclusión, estos modelos pueden ser entrenados con una impresionante rapidez en equipos sofisticados, pero a su vez también pueden ser entrenados de manera gratuita, sin requerir ningún *hardware* avanzado y tener una rapidez de entrenamiento muy buena.

Tabla 4.5: Comparación del tiempo en el que se ejecuta la corrección por radiación dispersada de un estudio completo y en el tiempo de entrenamiento de una misma red con diferentes equipos.

Equipo	$t_{correction}$ [s]	t_{train} [min]
Holiday	1.1	4.7
Alice	16.8	124.8
Google Colab	2.88	16.43

Capítulo 5

Conclusiones

- ✓ Un programa de simulación Monte Carlo (MC) previamente validado, basado en el código `egs_cbct` de EGSnrc, fue empleado para generar proyecciones de tomografía de haz de cono (CBCT) de maniqués geométricos y anatómicos separando las componentes de radiación primaria y dispersada. Estos datos se usaron para entrenar un algoritmo de *deep learning* y así producir imágenes libres de radiación dispersada a partir del equivalente a proyecciones medidas.
- ✓ Los modelos de *deep learning* entrenados predicen la componente de la radiación dispersada con una precisión de hasta 93.56 % en comparación con las imágenes esperadas o *ground truth*, generadas por las simulaciones MC. Las discrepancias se deben fundamentalmente al ruido que tienen las imágenes producidas por las simulaciones MC. Este ruido es causado por el número de historias de las simulaciones MC y no se debe a un comportamiento físico de la radiación dispersada.
- ✓ Los modelos entrenados con la arquitectura de Redes Generatrices Adversariales (GANs) son capaces de capturar tanto la tendencia de la radiación dispersada como sus fluctuaciones estadísticas. Las UNets, tienen la ventaja de solo capturar únicamente el comportamiento de la radiación dispersada, pero no su ruido. Por lo tanto, los modelos basados en UNets son más eficientes que las GANs para identificar la distribución de la radiación dispersada.
- ✓ El comportamiento de la radiación dispersada en MV es distinto a su comportamiento en kV. Esto se debe a la dependencia direccional en las secciones eficaces de Klein-Nishina y al cambio en la sección eficaz de Efecto Compton en función de la energía, para los diferentes materiales del fantoma. Estas diferencias hacen necesario entrenar

modelos específicos para cada equipo y energía. No obstante, es posible acelerar el proceso de entrenamiento utilizando técnicas de transferencia de aprendizaje.

- ✓ El algoritmo que se entrenó para minimizar una función de pérdida basada en el Error Absoluto Medio (MAE) mostró mejores resultados en los índices de calidad de imagen empleados. El algoritmo entrenado con una función de pérdidas basado en el Índice de Similitud Estructural (SSIM) se caracterizó por mejorar el Cociente Señal a Ruido (SNR) y el Cociente Contraste a Ruido (CNR) de los insertos del HUP, en la imagen tomográfica reconstruida.
- ✓ Emplear submuestreo para entrenar los algoritmos de *deep learning* puede reducir los tiempos de entrenamiento en un factor proporcional a la reducción en el número de píxeles. Debido a que la radiación dispersada tiene componentes de baja frecuencia espacial, el submuestreo prácticamente no afecta la calidad de las predicciones.
- ✓ La técnica de *transferencia de aprendizaje* permite tomar ventaja de la información almacenada en un modelo previamente entrenado. De esta manera es posible reducir los tiempos de entrenamiento incluso a la mitad. Sin embargo, la determinación de los números CT en las imágenes reconstruidas en algunos casos puede ser menos exacta con respecto a modelos entrenados sin transferencia de aprendizaje.
- ✓ Los modelos de *deep learning* fueron entrenados en menos de 5 minutos con más de 250 imágenes por 200 *epochs* en un *GPU* NVIDIA Tesla V100 de 32 GB de RAM. En *Google Colab* este entrenamiento tomó aproximadamente 16 minutos.
- ✓ Las correcciones por dispersión mediante las UNets toman de 2 a 3 segundos para un estudio tomográfico completo con 360 proyecciones y con las GANs toman de 6 a 7 segundos. Estas correcciones tardan varias horas con métodos basados en simulaciones MC.
- ✓ Para futuros trabajos, se puede usar estrategias de *transferencia de aprendizaje* para ejecutar correcciones sobre datos experimentales. También se puede trabajar en implementar algoritmos que corrijan en simultáneo la radiación dispersada y el endurecimiento del haz en aquellos casos en que no es posible emplear filtros tipo moño, como por ejemplo en sistemas de microtomografía.

Bibliografía

- [1] T. M. Buzug, “Computed tomography: from photon statistics to modern cone-beam CT,” 2009.
- [2] M. Kumar, M. Shanavas, A. Sidappa, and M. Kiran, “Cone beam computed tomography-know its secrets,” *Journal of international oral health*, vol. 7, no. 2, p. 64, 2015.
- [3] C. C. Shaw, *Cone beam computed tomography*. Taylor & Francis, 2014.
- [4] D. A. Jaffray, J. H. Siewerdsen, J. W. Wong, and A. A. Martinez, “Flat-panel cone-beam computed tomography for image-guided radiation therapy,” *International Journal of Radiation Oncology* Biology* Physics*, vol. 53, no. 5, pp. 1337–1349, 2002.
- [5] B. A. Fraass, A. Eisbruch, and M. Feng, “Intensity-modulated and image-guided radiation therapy,” in *Clinical Radiation Oncology*, pp. 294–324, Elsevier, 2016.
- [6] H. Li, R. Mohan, and X. R. Zhu, “Scatter kernel estimation with an edge-spread function method for cone-beam computed tomography imaging,” *Physics in Medicine & Biology*, vol. 53, no. 23, p. 6729, 2008.
- [7] M. Haenlein and A. Kaplan, “A brief history of artificial intelligence: On the past, present, and future of artificial intelligence,” *California management review*, vol. 61, no. 4, pp. 5–14, 2019.
- [8] K. Phil, “Matlab deep learning with machine learning, neural networks and artificial intelligence,” *Apress, New York*, 2017.
- [9] P. Ongsulee, “Artificial intelligence, machine learning and deep learning,” in *2017 15th International Conference on ICT and Knowledge Engineering (ICT&KE)*, pp. 1–6, IEEE, 2017.

- [10] C. Cao, F. Liu, H. Tan, D. Song, W. Shu, W. Li, Y. Zhou, X. Bo, and Z. Xie, “Deep learning and its applications in biomedicine,” *Genomics, proteomics & bioinformatics*, vol. 16, no. 1, pp. 17–32, 2018.
- [11] P. Hamet and J. Tremblay, “Artificial intelligence in medicine,” *Metabolism*, vol. 69, pp. S36–S40, 2017.
- [12] Y. Mintz and R. Brodie, “Introduction to artificial intelligence in medicine,” *Minimally Invasive Therapy & Allied Technologies*, vol. 28, no. 2, pp. 73–81, 2019.
- [13] B. Zapien, “Simulación Monte Carlo para corrección por dispersión en tomografía de haz de cono para radioterapia guiada por imagen,” Master’s thesis, Universidad Nacional Autónoma de México, UNAM, Ciudad de México, 11 2020.
- [14] W. Van Aarle, W. J. Palenstijn, J. De Beenhouwer, T. Altantzis, S. Bals, K. J. Batenburg, and J. Sijbers, “The ASTRA Toolbox: A platform for advanced algorithm development in electron tomography,” *Ultramicroscopy*, vol. 157, pp. 35–47, 2015.
- [15] W. Van Aarle, W. J. Palenstijn, J. Cant, E. Janssens, F. Bleichrodt, A. Dabrovolski, J. De Beenhouwer, K. J. Batenburg, and J. Sijbers, “Fast and flexible X-ray tomography using the ASTRA toolbox,” *Optics express*, vol. 24, no. 22, pp. 25129–25147, 2016.
- [16] J. Radon, “On the determination of functions from their integrals along certain manifolds,” *Ber. Verh. Sachs Akad Wiss.*, vol. 69, pp. 262–277, 1917.
- [17] R. Ramlau and O. Scherzer, “The first 100 years of the radon transform,” *Inverse Problems*, vol. 34, no. 9, p. 090201, 2018.
- [18] H. G. N. Cormack Allan M, “The nobel prize in physiology or medicine 1979,” *Nobel Prize*, 1979.
- [19] E. Bosch, “Sir Godfrey Newbold Hounsfield y la tomografía computada, su contribución a la medicina moderna,” *Revista chilena de radiología*, vol. 10, no. 4, pp. 183–185, 2004.
- [20] A. C. Kak and M. Slaney, *Principles of computerized tomographic imaging*. SIAM, 2001.
- [21] E. Seeram, *Computed Tomography-E-Book: Physical Principles, Clinical Applications, and Quality Control*. Elsevier Health Sciences, 2015.
- [22] J. T. Bushberg and J. M. Boone, *The essential physics of medical imaging*. Lippincott Williams & Wilkins, 2011.

- [23] K. Srinivasan, M. Mohammadi, and J. Shepherd, “Applications of linac-mounted kilovoltage cone-beam computed tomography in modern radiation therapy: A review,” *Polish journal of radiology*, vol. 79, p. 181, 2014.
- [24] G. Recoletas, “Vanguardia tecnológica para combatir el cáncer mama: Descubre TrueBeam STx.” <https://www.gruporecoletas.com/noticias/vanguardia-tecnologica-truebeam-stx/>, 2019.
- [25] L. A. Feldkamp, L. C. Davis, and J. W. Kress, “Practical cone-beam algorithm,” *Josa a*, vol. 1, no. 6, pp. 612–619, 1984.
- [26] A. C. Kak and M. Slaney, *Principles of computerized tomographic imaging*. SIAM, 2001.
- [27] W. Swindell, R. G. Simpson, J. R. Oleson, C.-T. Chen, and E. A. Grubbs, “Computed tomography with a linear accelerator with radiotherapy applications,” *Medical physics*, vol. 10, no. 4, pp. 416–420, 1983.
- [28] M. A. Mosleh-Shirazi, P. M. Evans, W. Swindell, S. Webb, and M. Partridge, “A cone-beam megavoltage CT scanner for treatment verification in conformal radiotherapy,” *Radiotherapy and Oncology*, vol. 48, no. 3, pp. 319–328, 1998.
- [29] W. R. Hendee and E. R. Ritenour, *Medical imaging physics*. John Wiley & Sons, 2003.
- [30] M. A. Flower, *Webb’s physics of medical imaging*. CRC Press, 2012.
- [31] Z. Wang and A. C. Bovik, “Mean squared error: Love it or leave it? A new look at signal fidelity measures,” *IEEE signal processing magazine*, vol. 26, no. 1, pp. 98–117, 2009.
- [32] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [33] E. C. Serrano, “Evaluación de calidad de imagen en tomografía óptica de lumniscencia por excitación selectiva con rayos X a través de PSNR Y SSIM,” Master’s thesis, Universidad Nacional Autónoma de México, UNAM, Ciudad de México, 2017.
- [34] B. J. McParland, *Nuclear medicine radiation dosimetry: advanced theoretical principles*. Springer Science & Business Media, 2010.
- [35] H. Nikjoo, S. Uehara, and D. Emfietzoglou, *Interaction of radiation with matter*. CRC press, 2012.

- [36] O. Klein and Y. Nishina, “Über die Streuung von Strahlung durch freie Elektronen nach der neuen relativistischen Quantendynamik von Dirac,” *Zeitschrift für Physik*, vol. 52, no. 11-12, pp. 853–868, 1929.
- [37] R. D. Evans, *Compton effect*. Springer, 1958.
- [38] L. T. Niklason, J. A. Sorenson, and J. A. Nelson, “Scattered radiation in chest radiography,” *Medical Physics*, vol. 8, no. 5, pp. 677–681, 1981.
- [39] E.-P. Rührnschopf and K. Klingenberg, “A general framework and review of scatter correction methods in X-ray cone-beam computerized tomography. Part 1: Scatter compensation approaches,” *Medical physics*, vol. 38, no. 7, pp. 4296–4311, 2011.
- [40] E.-P. Rührnschopf and K. Klingenberg, “A general framework and review of scatter correction methods in cone beam ct. part 2: scatter estimation approaches,” *Medical physics*, vol. 38, no. 9, pp. 5186–5199, 2011.
- [41] G. Poludniowski, P. Evans, V. Hansen, and S. Webb, “An efficient Monte Carlo-based algorithm for scatter correction in keV cone-beam CT,” *Physics in Medicine & Biology*, vol. 54, no. 12, p. 3847, 2009.
- [42] Y. Nomura, Q. Xu, H. Shirato, S. Shimizu, and L. Xing, “Projection-domain scatter correction for cone beam computed tomography using a residual convolutional neural network,” *Medical physics*, vol. 46, no. 7, pp. 3142–3155, 2019.
- [43] J. Maier, S. Sawall, M. Knaup, and M. Kachelrieß, “Deep scatter estimation (DSE): accurate real-time scatter estimation for X-ray CT using a deep convolutional neural network,” *Journal of Nondestructive Evaluation*, vol. 37, no. 3, pp. 1–9, 2018.
- [44] H. Lee and J. Lee, “A deep learning-based scatter correction of simulated X-ray images,” *Electronics*, vol. 8, no. 9, p. 944, 2019.
- [45] T. M. Mitchell *et al.*, *Machine learning*, vol. 1. McGraw-hill New York, 2007.
- [46] A. Maier, C. Syben, T. Lasser, and C. Riess, “A gentle introduction to deep learning in medical image processing,” *Zeitschrift für Medizinische Physik*, vol. 29, no. 2, pp. 86–101, 2019.
- [47] K. O’Shea and R. Nash, “An introduction to convolutional neural networks,” *arXiv preprint arXiv:1511.08458*, 2015.

- [48] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 international conference on engineering and technology (ICET)*, pp. 1–6, Ieee, 2017.
- [49] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Is L2 a good loss function for neural networks for image processing? arxiv 2015,” *arXiv preprint arXiv:1511.08861*.
- [50] H. Zhao, O. Gallo, I. Frosio, and J. Kautz, “Loss functions for image restoration with neural networks,” *IEEE Transactions on computational imaging*, vol. 3, no. 1, pp. 47–57, 2016.
- [51] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.
- [52] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: convolutional networks for biomedical image segmentation,” in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.
- [53] A. E. Ilesanmi and T. O. Ilesanmi, “Methods for image denoising using convolutional neural network: a review,” *Complex & Intelligent Systems*, vol. 7, no. 5, pp. 2179–2198, 2021.
- [54] M. P. Reymann, T. Würfl, P. Ritt, B. Stimpel, M. Cachovan, A. H. Vija, and A. Maier, “U-net for spect image denoising,” in *2019 IEEE Nuclear Science Symposium and Medical Imaging Conference (NSS/MIC)*, pp. 1–2, IEEE, 2019.
- [55] J. Zhang, Y. Niu, Z. Shangguan, W. Gong, and Y. Cheng, “A novel denoising method for ct images based on u-net and multi-attention,” *Computers in Biology and Medicine*, vol. 152, p. 106387, 2023.
- [56] M. P. Heinrich, M. Stille, and T. M. Buzug, “Residual u-net convolutional neural network architecture for low-dose ct denoising,” *Current Directions in Biomedical Engineering*, vol. 4, no. 1, pp. 297–300, 2018.
- [57] R. Komatsu and T. Gonsalves, “Comparing u-net based models for denoising color images,” *AI*, vol. 1, no. 4, pp. 465–486, 2020.
- [58] W. Zhu, S. M. Mousavi, and G. C. Beroza, “Seismic signal denoising and decomposition using deep neural networks,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 11, pp. 9476–9488, 2019.

- [59] G. Ian, J. Pouget-Abadie, M. Mirza, B. Xu, and D. Warde-Farley, “Generative adversarial nets.” in *advances in neural information processing systems*,” 2014.
- [60] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” *IEEE Signal Processing Magazine*, vol. 35, no. 1, pp. 53–65, 2018.
- [61] J. Chi, C. Wu, X. Yu, P. Ji, and H. Chu, “Single low-dose ct image denoising using a generative adversarial network with modified u-net generator and multi-level discriminator,” *IEEE Access*, vol. 8, pp. 133470–133487, 2020.
- [62] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, “Image-to-image translation with conditional adversarial networks,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1125–1134, 2017.
- [63] J. Ho and S. Ermon, “Generative adversarial imitation learning,” *Advances in neural information processing systems*, vol. 29, 2016.
- [64] D. Vint, M. Anderson, Y. Yang, C. Ilioudis, G. Di Caterina, and C. Clemente, “Automatic target recognition for low resolution foliage penetrating sar images using cnns and gans,” *Remote Sensing*, vol. 13, no. 4, p. 596, 2021.
- [65] L. Torrey and J. Shavlik, “Transfer learning,” in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pp. 242–264, IGI global, 2010.
- [66] K. Weiss, T. M. Khoshgoftaar, and D. Wang, “A survey of transfer learning,” *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.
- [67] W. R. Nelson, H. Hirayama, and D. W. Rogers, “EGS4 code system,” tech. rep., Stanford Linear Accelerator Center, Menlo Park, CA (USA), 1985.
- [68] I. Kawrakow, D. Rogers, E. Mainegra-Hing, F. Tessier, R. Townson, and B. Walters, “EGSnrc toolkit for Monte Carlo simulation of ionizing radiation transport.” [doi:10.4224/40001303](https://doi.org/10.4224/40001303) [release v2021], 2000.
- [69] B. H. Zapien-Campos, H. Alva-Sanchez, M. Rodríguez-Villafuerte, F. P. Herrera-Martínez, and A. Martínez-Dávalos, “Monte Carlo modelling of the kV and MV imaging systems on the Varian TrueBeam STx Linac,” *AIP Conference Proceedings*, vol. 2348, no. 1, p. 040004, 2021.

- [70] Varian Medical Systems, “Memoria documental del acelerador lineal marca Varian, modelo TrueBeam STx,” 2018.
- [71] Center for virtual imaging trials. Duke University School of Medicine., “MOBY/ROBY phantoms, <https://cvit.duke.edu/resource/moby-roby-phantoms/>,” 2022.
- [72] Sun Nuclear. A Mirion Medical Company , “Electron Density Phantom. Model 062M, <https://www.cirsinc.com/products/radiation-therapy/electron-density-phantom/>,” 2023.
- [73] M. Kintel and C. Wolf, “OpenSCAD,” *GNU General Public License*, vol. 2, no. 10, 2014.
- [74] J. Han, J. Pei, and H. Tong, *Data mining: concepts and techniques*. Morgan kaufmann, 2022.
- [75] NVIDIA, “NVIDIA V100 Tensor Core GPU,” <https://www.nvidia.com/en-us/data-center/v100/>, 2023.
- [76] NVIDIA, “GeForce RTX 3060 Family,” <https://www.nvidia.com/en-us/geforce/graphics-cards/30-series/rtx-3060-3060ti/>, 2022.
- [77] R. Bujila, A. Omar, and G. Poludniowski, “A validation of SpekPy: A software toolkit for modelling X-ray tube spectra,” *Physica Medica*, vol. 75, pp. 44–54, 2020.
- [78] M. Rafei and S. Parsaei, “A database of photon mass-energy transfer coefficients for dosimetric materials from 1 keV to 100 MeV,” *Journal of Instrumentation*, vol. 16, no. 10, p. T10005, 2021.
- [79] M.J. Berger, J.H. Hubbell, S.M. Seltzer, J. Chang, J.S. Coursey, R. Sukumar, D.S. Zucker, and K. Olsen, “NIST XCOM: photon cross sections database,” <https://physics.nist.gov/PhysRefData/Xcom/html/xcom1.html>, 2010.

Anexo A

Coeficientes lineales de atenuación

El propósito de este Anexo es realizar una comparación entre los coeficientes lineales de atenuación, μ , obtenidos para los diferentes materiales del HUP con los valores correspondientes a la energía efectiva y promedio de los haces de kV y MV. Para esta comparación se consideró las reconstrucciones realizadas con proyecciones primarias, las cuales constituyen los valores de referencia (*ground truth*) para este estudio.

Para determinar la energía promedio, $\langle E \rangle$, de un espectro de fluencia se empleó la ec. (A.1).

$$\langle E \rangle = \frac{\int E\Phi(E)dE}{\int \Phi(E)dE} \quad (\text{A.1})$$

donde $\Phi(E)$ es el espectro de fluencia y la integración se realizó de manera numérica, utilizando el método del trapecio.

Para determinar la energía efectiva, primero se requiere obtener el espectro de Kerma en aire, $K(E)$, a partir del espectro de fluencia, lo cual se realiza con la ec. (A.2).

$$K(E) = E \cdot \Phi(E) \cdot \left(\frac{\mu_{tr}}{\rho} \right)_{aire} \quad (\text{A.2})$$

donde $\left(\frac{\mu_{tr}}{\rho} \right)_{aire}$ es el coeficiente másico de transferencia de energía en aire y depende de la energía. Este coeficiente fue obtenido mediante interpolaciones logarítmicas de los datos publicados por Rafiei & Parsaei (2021) [78].

El Kerma total, K_0 , es el área bajo la curva del espectro de Kerma $K(E)$. El espectro de kerma atenuado $K_{att}(E)$, por una lámina de espesor t , de material X se determina mediante

la ec. (A.3).

$$K_{att}(E) = K(E) \cdot e^{-\left(\frac{\mu}{\rho}\right)_X \rho_X t} \quad (\text{A.3})$$

donde $\left(\frac{\mu}{\rho}\right)_X$ y ρ_X son el coeficiente másico de atenuación y la densidad del material X , respectivamente, los cuales se los obtiene de la base de datos del NIST [79]. Para determinar la capa hemirreductora (HVL, por sus siglas en inglés) de un espectro se busca el espesor $t = HVL$ del material X , tal que el área bajo la curva de $K_{att}(E)$ sea igual a $K_0/2$. Generalmente, en haces de kV el material usado es Aluminio y en haces de MV es Cobre.

Para determinar la energía efectiva del haz, primero se determina el μ correspondiente a la HVL, el cual se determina a partir de la ec. (A.4). Y luego, mediante interpolaciones logarítmicas de los datos de la base de datos XCOM [79] se determina la energía correspondiente a ese coeficiente μ para el material X , la cual es la energía efectiva del espectro.

$$\mu = \frac{\ln 2}{HVL} \quad (\text{A.4})$$

En la Tabla A.1 se muestra la energía promedio, efectiva y HVL para los espectros de kV y MV empleados en este proyecto.

Tabla A.1: Capa hemirreductora y energías promedio y efectiva de los espectros de kV y MV empleados en este estudio.

Magnitud	Energía promedio [keV]	HVL	Energía efectiva [keV]
Espectro kV	80.5	1.22 cm-Al	76.6
Espectro MV	476	1.01 cm-Cu	598

Con las energías promedio y efectiva y la composición y densidad proporcionada por el fabricante para los materiales del HUP, se determinaron los correspondientes coeficientes de atenuación μ a partir de los datos de XCOM [79]. En la Tabla A.2 se comparan estos coeficientes con los medidos en las reconstrucciones realizadas con proyecciones primarias.

Las intensidades de píxel de las reconstrucciones realizadas con las rutinas de ASTRA Toolbox tienen unidades de inverso de tamaño de píxel por lo que hay que dividir su valor por el tamaño de píxel (0.106 cm, considerando que la imagen tiene una magnificación de 1.5 por la geometría de la adquisición). Las diferencias porcentuales medias entre los coeficientes de atenuación de la reconstrucción con primarios y los correspondientes a la energía efectiva son 3.48 % y 6.88 % para los espectros de kV y MV, respectivamente. Si estos valores se comparan

con los coeficientes de atenuación a la energía media, las diferencias son de 6.08 % y 16.08 %, para los espectros de kV y MV, respectivamente; lo cual es consistente con el hecho de que la energía efectiva caracteriza de mejor manera un espectro.

Tabla A.2: Comparación entre los coeficientes de atenuación medidos para los materiales del HUP en la reconstrucción con primarios y los correspondientes a las energías promedio y efectivas de los espectros empleados en este estudio.

Material	$\mu_{primarios} [cm^{-1}]$	$\mu(E_{eff}) [cm^{-1}]$	$\mu(\langle E \rangle) [cm^{-1}]$
Espectro de kV			
Agua	0.1913 ± 0.0007	0.1865	0.1833
Hueso Cortical	0.5659 ± 0.0017	0.5534	0.5218
Hueso Trabecular	0.2406 ± 0.0022	0.2298	0.2233
Tejido Adiposo	0.1788 ± 0.0021	0.1729	0.1704
Pulmón Exhalado	0.0979 ± 0.0021	0.0931	0.0915
Espectro de MV			
Agua	0.0827 ± 0.0003	0.0897	0.0989
Hueso Cortical	0.1604 ± 0.0007	0.1775	0.1961
Hueso Trabecular	0.0942 ± 0.0009	0.1002	0.1105
Tejido Adiposo	0.0812 ± 0.0009	0.0851	0.0938
Pulmón Exhalado	0.0467 ± 0.0009	0.0448	0.0494

Anexo B

Números CT, SNR y CNR de los diferentes modelos implementados

En este anexo se presenta los Números CT, SNR, CNR y sus errores, medidos en las reconstrucciones tomográficas realizadas con los diferentes modelos implementados.

Tabla B.1: Comparación de los números CT, CNR y SNR en los insertos del HUP de acuerdo con la imagen reconstruida con proyecciones corregidas por radiación dispersada con una UNet entrenada con proyecciones del HUP y de Moby y con la imagen reconstruida con una UNet.

Material	$\#CT_{UNet}$ [HU]	$\#CT_{ref}$ [HU]	Error [%]	SNR	CNR
Modelo entrenado con proyecciones de HUP y MOBY					
Agua	0 ± 4	0 ± 3	–	15.12 ± 1.06	–
Hueso Trabecular	255 ± 11	257 ± 11	–0.78	23.44 ± 1.54	3.86 ± 0.33
Tejido Adiposo	-67 ± 10	-65 ± 11	3.08	17.94 ± 1.11	1.02 ± 0.19
Hueso Cortical	1963 ± 11	1957 ± 8	0.31	43.52 ± 2.90	29.68 ± 2.08
Pulmón Exhalado	-491 ± 10	-488 ± 11	0.61	10.00 ± 0.67	7.43 ± 0.55
Modelo entrenado para predecir <i>scatter ratio</i>					
Agua	0 ± 4	0 ± 3	–	17.78 ± 1.19	–
Hueso Trabecular	250 ± 10	257 ± 11	–2.72	26.82 ± 1.67	4.46 ± 0.36
Tejido Adiposo	-59 ± 11	-65 ± 11	–9.23	20.38 ± 1.24	1.06 ± 0.22
Hueso Cortical	1922 ± 11	1957 ± 8	–1.79	55.49 ± 3.46	34.19 ± 2.30
Pulmón Exhalado	-458 ± 11	-488 ± 11	–6.15	11.74 ± 0.77	8.15 ± 0.59

Tabla B.2: Comparación de los números CT, CNR y SNR en los insertos del HUP de acuerdo con la imagen reconstruida con proyecciones corregidas por radiación dispersada con una GAN.

Material	$\#CT_{GAN}$ [HU]	$\#CT_{ref}$ [HU]	Error [%]	SNR	CNR
Agua	0 ± 5	0 ± 3	–	12.52 ± 1.87	–
Hueso Trabecular	260 ± 11	257 ± 11	1.17	26.24 ± 1.64	3.26 ± 0.51
Tejido Adiposo	-63 ± 11	-65 ± 11	-3.08	19.62 ± 1.25	0.79 ± 0.19
Hueso Cortical	1963 ± 8	1957 ± 8	0.31	48.49 ± 3.10	24.59 ± 3.66
Pulmón Exhalado	-488 ± 11	-488 ± 11	0.00	10.77 ± 0.71	6.12 ± 0.92

Tabla B.3: Comparación de los números CT, CNR y SNR en los insertos del HUP de acuerdo con la imagen reconstruida con proyecciones corregidas por radiación dispersada con una UNet entrenada para minimizar el MSE, SSIM y PSNR.

Material	$\#CT_{UNet}$ [HU]	$\#CT_{ref}$ [HU]	Error [%]	SNR	CNR
MSE					
Agua	0 ± 4	0 ± 3	–	14.85 ± 1.07	–
Hueso Trabecular	260 ± 11	257 ± 11	1.17	23.31 ± 1.54	3.87 ± 0.33
Tejido Adiposo	-62 ± 11	-65 ± 11	-4.62	17.88 ± 1.12	0.92 ± 0.19
Hueso Cortical	1969 ± 8	1957 ± 8	0.61	42.15 ± 2.74	29.24 ± 2.11
Pulmón Exhalado	-487 ± 11	-488 ± 11	-0.20	10.00 ± 0.67	7.23 ± 0.55
SSIM					
Agua	0 ± 3	0 ± 3	–	21.98 ± 1.59	–
Hueso Trabecular	265 ± 11	257 ± 11	3.11	29.54 ± 1.89	5.84 ± 0.49
Tejido Adiposo	-57 ± 11	-65 ± 11	-12.31	21.76 ± 1.40	1.27 ± 0.27
Hueso Cortical	1968 ± 8	1957 ± 8	0.56	61.46 ± 3.13	43.27 ± 3.13
Pulmón Exhalado	-482 ± 11	-488 ± 11	-1.23	11.62 ± 0.77	10.61 ± 0.81
PSNR					
Agua	0 ± 3	0 ± 3	–	15.52 ± 1.11	–
Hueso Trabecular	259 ± 11	257 ± 11	0.78	23.86 ± 1.51	4.04 ± 0.34
Tejido Adiposo	-62 ± 11	-65 ± 11	-4.62	18.21 ± 1.11	0.97 ± 0.20
Hueso Cortical	1973 ± 9	1957 ± 8	0.82	43.09 ± 2.66	30.63 ± 2.18
Pulmón Exhalado	-489 ± 11	-488 ± 11	0.20	10.13 ± 0.67	7.59 ± 0.57

Tabla B.4: Comparación de los números CT, CNR y SNR en los insertos del HUP de acuerdo con la imagen reconstruida con proyecciones corregidas por radiación dispersada con una UNet entrenada con submuestreo.

Material	$\#CT_{UNet}$ [HU]	$\#CT_{ref}$ [HU]	Error [%]	SNR	CNR
Agua	0 ± 4	0 ± 3	–	14.48 ± 1.04	–
Hueso Trabecular	257 ± 11	257 ± 11	0.00	22.83 ± 1.50	3.73 ± 0.32
Tejido Adiposo	-64 ± 11	-65 ± 11	-1.54	17.58 ± 1.09	0.93 ± 0.18
Hueso Cortical	1967 ± 9	1957 ± 8	0.51	40.46 ± 2.65	28.48 ± 2.05
Pulmón Exhalado	-488 ± 11	-488 ± 11	0.00	9.85 ± 0.66	7.07 ± 0.54

Tabla B.5: Comparación de los números CT, CNR y SNR en los insertos del HUP de acuerdo con la imagen reconstruida con proyecciones corregidas por radiación dispersada con una UNet entrenada con *transfer learning* según los casos *B*, *C* y *D*, descritos en la sección 4.9.

Material	$\#CT_{UNet}$ [HU]	$\#CT_{ref}$ [HU]	Error [%]	SNR	CNR
Caso <i>B</i>					
Agua	0 ± 6	0 ± 3	–	14.16 ± 1.11	–
Hueso Trabecular	249 ± 12	257 ± 11	-3.11	22.43 ± 1.54	3.53 ± 0.34
Tejido Adiposo	-77 ± 10	-65 ± 11	18.46	17.25 ± 1.09	1.09 ± 0.20
Hueso Cortical	1988 ± 30	1957 ± 8	1.58	40.66 ± 2.93	28.17 ± 2.24
Pulmón Exhalado	-504 ± 10	-488 ± 11	3.28	9.57 ± 0.65	7.14 ± 0.58
Caso <i>C</i>					
Agua	0 ± 3	0 ± 3	–	10.16 ± 0.67	–
Hueso Trabecular	140 ± 10	138 ± 10	1.45	12.47 ± 0.85	0.17 ± 0.12
Tejido Adiposo	-16 ± 10	-17 ± 10	-5.88	14.22 ± 1.08	1.43 ± 0.15
Hueso Cortical	943 ± 8	939 ± 8	0.43	7.57 ± 0.55	4.43 ± 0.31
Pulmón Exhalado	-435 ± 10	-435 ± 10	0.00	22.10 ± 1.55	9.59 ± 0.64
Caso <i>D</i>					
Agua	0 ± 5	0 ± 3	–	14.85 ± 1.06	–
Hueso Trabecular	250 ± 11	257 ± 11	-2.72	23.17 ± 1.52	3.72 ± 0.33
Tejido Adiposo	-75 ± 11	-65 ± 11	15.38	17.70 ± 1.10	1.13 ± 0.20
Hueso Cortical	2014 ± 18	1957 ± 8	2.91	43.97 ± 2.96	29.91 ± 2.15
Pulmón Exhalado	-501 ± 10	-488 ± 11	2.66	9.79 ± 0.65	7.45 ± 0.56