Universidad de Cantabria

Facultad
de
Ciencias

Aplicación de Modelos Generativos de Inteligencia Artificial a la Reconstrucción de Imágenes Tomográficas

Application of Generative Artificial Intelligence Models to Tomographic Image Reconstruction

Trabajo de Fin de Grado
para acceder al

GRADO EN FÍSICA

Autor: Óscar Castro Remesal

Director: Pablo Martínez Ruiz del Árbol

12/06/2025

Agradecer a Pablo, por su tiempo y dedicación como director, por su orientación y conocimientos que me han ayudado a aprender todo lo necesario para realizar este trabajo y por su disposición a ayudarme en todo momento.

A mis padres y mi hermana, por apoyarme siempre incondicionalmente, y a mis amigos, con los que he pasado todos los buenos y malos momentos. Si he llegado hasta aquí, es en gran parte por ellos.

Gracias.

Resumen

Este trabajo está situado en el contexto de la tomografía muónica, una técnica no destructiva que permite obtener imágenes de objetos sin necesidad de dañarlos o de tener acceso directo a ellos, y de las redes neuronales, un modelo de aprendizaje automático que ha adquirido mucha popularidad en los últimos años y cuyo funcionamiento se inspira en el funcionamiento del sistema nervioso de los animales.

El proyecto que aquí se desarrolla tiene como objetivo principal comprobar la viabilidad de un modelo concreto de red neuronal, la U-Net, para la eliminación de ruido y de artefactos en las imágenes que se obtienen en tomografía muónica, en concreto de tuberías de gas. Para ello se realizaron simulaciones de las tuberías y se obtuvieron las imágenes partiendo de un algoritmo de reconstrucción clásico que luego se le proporcionaron a la U-Net para que las mejore. En base a los resultados obtenidos este modelo empleado puede ser útil para visualizar la tubería, pero no si se quieren obtener detalles de la misma tales como su grosor o el material del que está compuesta.

Palabras Clave: muografía, inteligencia artificial, redes neuronales convolucionales, U-Net

Abstract

This work is framed in the context of muon tomography, a non-destructive technique that enables obtaining images of objects without damaging them or requiring direct access, and neural networks, a machine learning model that has gained a lot of popularity in the past few years and whose functioning is inspired by the functioning of animal nervous systems.

The main objective of the project developed here is to test the feasibility of a specific neural network model, the U-Net, for noise and artifact elimination in images obtained from muon tomography, specifically of gas pipelines. To achieve this, simulations of the pipelines were performed, then a classical reconstruction algorithm was used to obtain the images that were later provided to the U-Net for enhancement. Based on the results obtained, this model can be useful for visualizing the pipeline, but not for obtaining specific details such as its thickness or the material it's made of.

Key Words: muography, artificial intelligence, convolutional neural networks, U-Net

Índice

1. Introducción **1**

2. Tomografía Muónica **2**

2.1. Muones y su Interacción con la Materia . . . 2

2.1.1. Rayos Cósmicos . . . 3

2.1.2. Interacción con la Materia . . . 3

2.2. Muografía . . . 5

2.3. Aplicaciones . . . 5

2.4. Detectores . . . 6

2.4.1. Detectores de Emulsión . . . 7

2.4.2. Detectores Centelleadores . . . 7

2.4.3. Detectores de Gas . . . 7

2.5. Algoritmos para Muografía de Dispersión . . . 8

2.5.1. Maximum Likelihood Expectation Maximization . . . 8

2.5.2. Point of Closest Approach . . . 8

2.6. Problema de la Tubería . . . 10

3. Inteligencia Artificial: Redes Convolucionales **12**

3.1. Machine Learning . . . 12

3.2. Redes Neuronales . . . 13

3.2.1. Aprendizaje . . . 14

3.2.2. Tipos de Redes Neuronales . . . 15

3.3. Redes Neuronales Convolucionales . . . 16

3.3.1. Arquitectura . . . 16

3.4. U-Net . . . 17

4. Entrenamiento y Ajuste del Modelo U-Net **19**

4.1. Sistema Hardware . . . 19

4.2. Muestras de Entrenamiento . . . 19

4.3. Entrenamiento de la U-Net . . . 21

5. Resultados y Validación del Modelo U-Net **25**

5.1. Análisis Cualitativo . . . 25

5.2. Análisis Cuantitativo . . . 27

6. Conclusiones **30**

6.1. Posibles Mejoras . . . 31

Referencias **32**

1 Introducción

Los muones son unas partículas elementales muy parecidas a los electrones, salvo porque son mucho más pesados. Gracias a su gran masa son capaces de atravesar la materia con facilidad, debido a que se ven menos afectados por pérdidas de energía. La tomografía muónica o muografía, es una técnica que emplea estas cualidades de los muones para la reconstrucción de imágenes de objetos a partir de muones cósmicos. Esta técnica utiliza muones provenientes de las capas altas de la atmósfera para mapear el interior de objetos muy grandes sin necesidad de destruirlos o de tener acceso físico a ellos, proporcionando una técnica no destructiva con la que estudiar objetos sin dañarlos.

Para llevar a cabo la reconstrucción de la imagen a partir de los muones cósmicos existen diversos algoritmos, uno de los cuáles, el que se va a emplear en este trabajo es el algoritmo POCA (point of closest approach). Este algoritmo sencillo permite de forma rápida obtener una imagen más o menos aproximada del objeto analizado, pero normalmente contienen mucho ruido y artefactos (errores o construcciones no reales).

Existen varios algoritmos para reconstruir las imágenes a partir de las trayectorias de los muones, aunque en en los últimos años, resultan bastante prometedoras aquellos basados en redes neuronales. En concreto las redes neuronales convolucionales son un tipo de redes neuronales muy centradas en el trabajo con imágenes. Son muy utilizadas para la caracterización de objetos o la reducción de ruido (denoising) en imágenes. La red convolucional concreta que se utilizará en este trabajo será una U-Net [1], la cuál es muy utilizada para denoising en diversos campos.

La finalidad de este trabajo es comprobar la viabilidad de un algoritmo de tipo U-Net para eliminar el ruido y los artefactos en imágenes generadas mediante tomografía muónica, siendo primero reconstruidas mediante el algoritmo POCA. En este trabajo se ha seleccionado como caso de uso un problema muy típico en la industria que consiste en la realización de muografías del interior de tuberías transportadoras de gases en fábricas de la industria pesada.

2 Tomografía Muónica

2.1. Muones y su Interacción con la Materia

El muón (μ^-) es una partícula elemental que forma parte del modelo estándar de la física de partículas. Este modelo explica tres de las cuatro interacciones fundamentales entre partículas (electromagnética, nuclear fuerte y nuclear débil) [2]. Actualmente es el modelo matemático más preciso y aceptado gracias no solo a la capacidad de explicar fenómenos observados con mucha precisión, si no también por la gran cantidad de predicciones que posteriormente han sido confirmadas, como el descubrimiento del quark top [3] o idel bosón de Higgs [4].

Según el modelo estándar, existen una serie de partículas elementales, representadas en la Fig. 2.1, a partir de las cuales se explica la formación todas las demás partículas y de sus interacciones entre ellas. Las partículas responsables de formar la materia son los fermiones (junto con sus anti-partículas que forman la anti-materia), y las mediadoras de las fuerzas son los bosones. Los fermiones son 12 partículas divididas en dos grupos de 6 según que interacciones fundamentales les afecten. Un grupo son los quarks, los cuales interaccionan fuerte y electromagnéticamente, y el otro los leptones, que se ven afectados por las tres interacciones. Son este segundo grupo al cuál pertenecen los muones, por lo que nos centraremos en él de ahora en adelante.

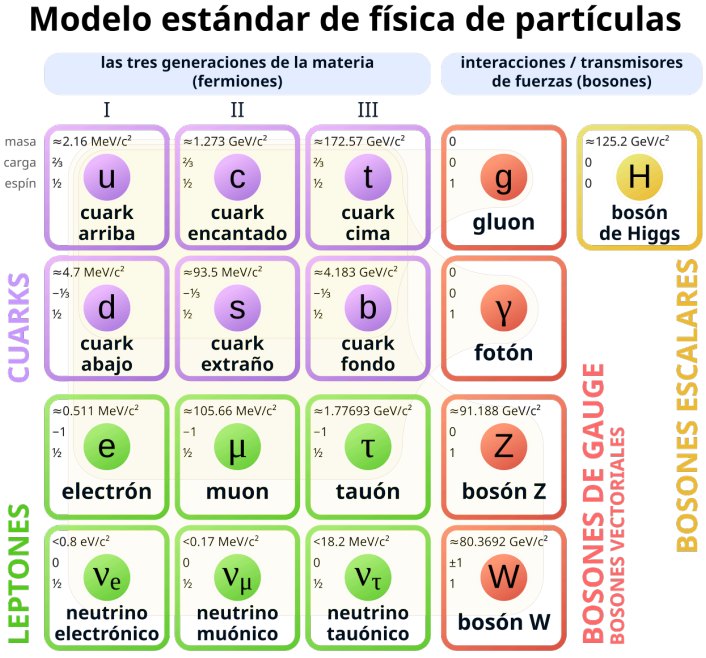

Figura 2.1: Tabla de todas las partículas pertenecientes al modelo estándar [5]. Están clasificadas por familias, leptones, quarks, bosones de gauge y bosones escalares.

Los leptones se dividen en tres familias, a la segunda de las cuales pertenece el muón. Los muones son muy parecidos en comportamiento a los electrones, que pertenecen a la primera familia, salvo porque tienen una masa aproximadamente 200 veces mayor, lo que significa que no son partículas estables. Tienen una vida media aproximada de 2,2 μs [6], y normalmente decaen a un electrón y un par de neutrinos. Aunque aparentemente sea un tiempo de vida pequeño, es mucho mayor que el de otras partículas fundamentales inestables, siendo capaz de recorrer unos 700 m antes de su decaimiento, lo que hace posible considerar a los muones estables en algunos procesos.

2.1.1. Rayos Cósmicos

La forma más sencilla de observar muones en la naturaleza es gracias a los rayos cósmicos, flujos de partículas muy energéticas (principalmente protones y núcleos atómicos) que vienen del espacio y al impactar con las moléculas de la atmósfera comienzan un decaimiento en cadena tal y como se muestra en la Fig. 2.2. Primero se producen varios piones neutros y cargados, decayendo estos últimos a muones y neutrinos. Gracias a este proceso una pequeña cantidad de los muones que se producen llegan a la superficie de la Tierra, lo que significa que existe una fuente natural de estas partículas que se puede aprovechar para distintas aplicaciones, entre ellas la que concierne a este trabajo, la muografía.

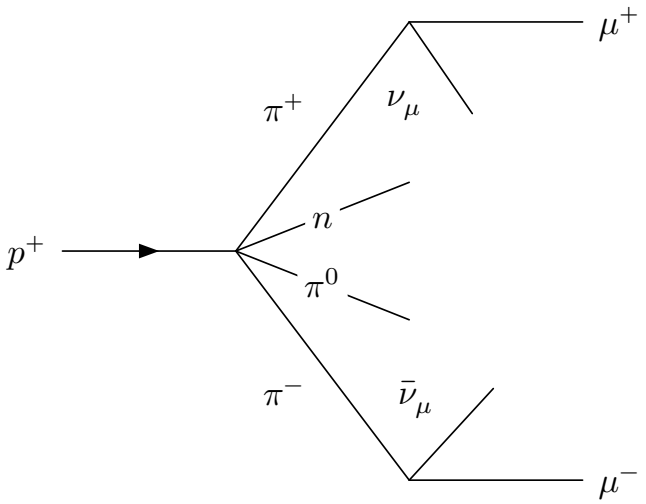

Figura 2.2: Diagrama mostrando una cadena típica de decaimiento inducido por un rayo cósmico muy energético al llegar a la atmósfera y colisionar con las partículas que la componen.

2.1.2. Interacción con la Materia

Los muones interactúan con la materia principalmente mediante dos procesos, ionización y dispersión múltiple (multiple scattering). La ionización se produce por colisiones del muón con los electrones del material, que absorben parte de su energía. Este comportamiento se describe mediante la ecuación de Bethe-Bloch [7], que relaciona la pérdida de energía por unidad de distancia $\left\langle \frac{\mathrm{d}E}{\mathrm{d}x} \right\rangle$ con distintos parámetros como la carga de la partícula incidente z ($z = 1$ en el caso de los muones), la carga y el número másico del material Z y A de la forma:

$$-\left\langle \frac{\mathrm{d}E}{\mathrm{d}x} \right\rangle = Kz^2 \frac{Z}{A} \frac{1}{\beta^2} \left[\frac{1}{2} \ln \left(\frac{2m_e c^2 \beta^2 \gamma^2 W_{\max}}{I^2} \right) - \beta^2 - \frac{\delta(\beta\gamma)}{2} \right] \tag{2.1}$$

donde β y γ son los factores relativistas, m_e es la masa del electrón, c la velocidad de la luz, K es una constante, I es la energía media de excitación y $W_{\max}$ es el máximo de energía transferible a un electrón debido a una sola colisión. Esta ecuación es válida en el rango $0{,}1 \lesssim \beta\gamma \lesssim 1000$ con una precisión alta.

Como se puede observar en la Fig. 2.3, con muones de energías entre varios cientos de MeV y algunos GeV, la ionización que sufren los muones es mínima. A las partículas que se encuentran en el mínimo de ionización se les denomina *partículas mínimamente ionizantes* (MIPs por sus siglas en inglés). Sabiendo que los muones cósmicos pertenecen a la categoría de MIPs, las pérdidas de energía que sufren por ionización son muy pequeñas, y por tanto esta interacción de los muones con la materia no se tendrá en cuenta en este trabajo.

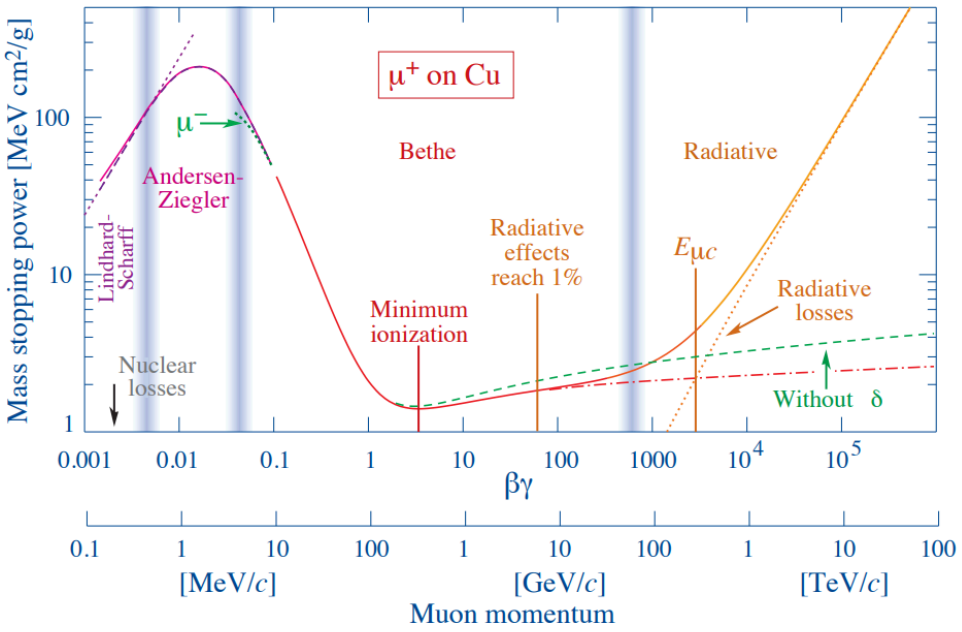

Figura 2.3: Pérdida de energía por un muón en el cobre debido al proceso de ionización, frente al momento del muón [7].

El otro tipo de interacción que sufren los muones con la materia es la dispersión múltiple. Debido a que los muones son partículas cargadas, interaccionan con los núcleos de los átomos del medio que atraviesan y sufren dispersión de Coulomb varias veces, lo que modifica la trayectoria final de los muones. El ángulo de desviación en cada dispersión es probabilístico, con una distribución que se aproxima a una gaussiana con los extremos más largos. El 98% central de la distribución angular proyectada se puede describir con una gaussiana, que tiene por anchura el parámetro θ_0 definido como [7]:

$$\theta_0 = \frac{13{,}6 \text{ MeV}}{\beta c p} \sqrt{\frac{x}{X_0}} \left[1 + 0{,}038 \ln\left(\frac{x}{X_0 \beta^2} \right) \right] \tag{2.2}$$

donde p es el momento del muón incidente y X_0 es la longitud de radiación, que se define como la cantidad de materia atravesada característica de la partícula incidente para una interacción en particular. De esta expresión se puede concluir que un muón con un momento mayor se verá menos desviado, mientras que un muón en un material más denso se verá más desviado.

2.2. Muografía

El objetivo principal de la muografía o tomografía muónica es determinar las propiedades del medio que se quiere observando el efecto que se produce en los muones que lo atraviesan. Dado que como se ha visto, la dispersión múltiple depende de distintas características de dicho medio, en vez de intentar calcular la desviación que sufren los muones, se puede medir e intentar realizar el proceso a la inversa, deducir a partir de estas medidas características sobre el material.

A este tipo de problemas donde se tienen las condiciones iniciales y el resultado final, y se quiere obtener el proceso que ha ocurrido, se les conoce como "problema inverso", y en general son muy difíciles de resolver y no suelen tener una única solución. Además, en muografía este proceso no es determinista, es probabilístico, lo cual lo convierte en un problema mucho más complicado.

Para la tomografía muónica se utilizan muones cósmicos, que como ya se ha detallado, provienen de rayos cósmicos y existe un flujo constante de ellos en la superficie terrestre, por lo que la muografía es una técnica denominada no destructiva (NDT por sus siglas en inglés), ya que no se está sometiendo al medio a ningún tipo de radiación que lo altere o dañe.

Utilizando este flujo de muones se puede generar un mapa de densidad del interior del objeto. Gracias a la poca interacción que tienen los muones con la materia, habitualmente cuentan con gran capacidad de penertración en el medio, y por tanto se pueden llegar a analizar objetos muy grandes o de materiales muy densos. Dentro de esta técnica existen dos categorías dependiendo de cómo se vaya a obtener exactamente el mapa de densidad: la muografía de absorción y la muografía de dispersión.

- **Muografía de absorción:** Esta estrategia simplemente consiste en conocer cuál es el flujo de muones cósmicos esperados en promedio y su energía, y compararlo con el flujo que se obtiene cuando los muones atraviesan el medio. Analizando la pérdida de energía sufrida por los muones debido a la ionización es como se obtiene el mapa de densidad del objeto. La gran ventaja de esta estrategia es que solamente se necesita un detector, lo que ayuda a reducir costes y hace que sea ideal para objetos muy grandes. La parte negativa es que este método requiere de tiempos de exposición muy largos (a veces meses) para obtener unos buenos resultados.
- **Muografía de dispersión:** Este método utiliza dos detectores, uno a cada lado del objeto a caracterizar, para así poder medir la dispersión que sufre la trayectoria de los muones en la dirección del flujo incidente de los mismos. Esta estrategia es útil para medir objetos pequeños y de forma rápida, puesto que el tiempo de exposición es corto. Como se ha comentado antes en la Ec. (2.2), cuanto más denso y grueso sea el objeto, mayor será la desviación producida y será más fácil generar el mapa. Es este segundo método en el que se ha utilizado para todo el trabajo.

2.3. Aplicaciones

La muografía es una técnica que se conoce desde hace tiempo. El primer intento de utilizarla fue en 1970 [8], cuando se intentaron encontrar cámaras secretas dentro de pirámides. Desde entonces la técnica ha evolucionado mucho y se han ido desarrollando distintas aplicaciones en campos muy diversos.

Uno de estos campos en los que han surgido varias aplicaciones es la geología, donde mayoritariamente se aplica la muografía de absorción. Los volcanes son un fenómeno natural que conlleva muchos peligros para los humanos, y el mayor problema es que es muy difícil conocer cuando van a entrar en erupción. La tomografía muónica se puede usar para monitorizar los volcanes y así ser capaces de estudiar o predecir erupciones [9]. En la Fig. 2.4 se puede ver una imagen tomada con muografía en la que se observa el interior del volcán, donde se encuentra una acumulación de magma.

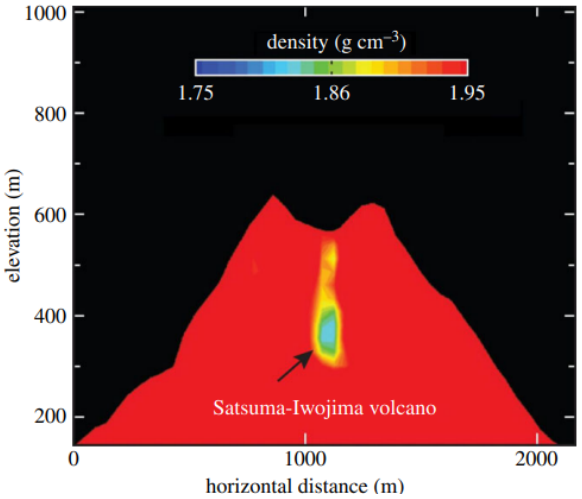

Figura 2.4: Imagen tomada con muografía del interior de un volcán [9].

En el campo de la geología también se aplica la muografía a la minería, como ayuda en la detección de cavidades oculas o la búsqueda de materiales, en este caso uranio o plomo [10].

La muografía de dispersión también tiene muchas aplicaciones, centradas en áreas más relacionadas con la industria. Desde recosntrucciones de altos hornos para observar si su estructura interna está o no dañada, hasta el análisis del contenido de camiones sin necesidad de inspeccionarlos, pudiendo añadir un sistema de seguridad adicional en lugares como fronteras o puertos [11]. También existen aplicaciones en entornos nucleares, por ejemplo en la caracterización de residuos nucleares almacenados en contenedores sellados [12].

El problema que se va a tratar en este trabajo también es una de sus aplicaciones comunes, el estudio de tuberías, en donde esta técnica puede ayudar a conocer el estado de las tuberías o de que material están hechas sin necesidad de acceder a ellas, ya que normalmente estas son inaccesibles.

2.4. Detectores

Los detectores de muones son un elemento clave dentro de la tomografía muónica, y como tal han de cumplir ciertos requisitos para poder emplearse en esta técnica. Primeramente los detectores han de tener una alta granularidad que aporte gran resolución espacial para las detecciones de muones y su trayectoria. También han de contar con gran resolución angular, cualidad que se puede conseguir situando diferentes planos del detector con la suficiente distancia como para estimar correctamente la dirección de los muones.

Desde un punto de vista más práctico, se requiere que los detectores tengan una gran captación para evitar tiempos de exposición muy largos, además de la capacidad de trabajar en condiciones extremas (altas temperaturas, entornos con gases o lluvia, ...). También es importante que su coste no sea desorbitado. Los tres tipos de detectores que mejor cumplen estas características, y por tanto son los más utilizados hoy en día son los detectores centelleadores, de emulsión y las cámaras proporcionales de multihilos.

2.4.1. Detectores de Emulsión

Este tipo de detectores fueron unos de los primeros en desarrollarse, siendo utilizados hoy en día en grandes experimentos con neutrinos. Están compuestos por placas fotográficas que gracias a sus granos muy uniformes y finos son capaces de recoger la trayectoria de partículas cargadas. La resolución de detección viene determinada por el tamaño de los granos, típicamente del orden del micrómetro, y la resolución angular puede conseguirse del orden de unos pocos miliradianes acumulando varias películas de emulsiones.

Como grandes ventajas de este tipo de detectores, cuentan con un coste relativamente bajo y no requieren de ninguna fuente de alimentación, por lo que son ideales para colocar en localizaciones de difícil acceso. Además son capaces de conseguir una resolución espacial bastante alta con capas muy finas ($\sim$ 1 mm), cuando otro tipo de detectores necesitan del orden de 1 m.

Por otro lado, tienen ciertas desventajas que los convierten en detectores no muy utilizados en general. El principal problema es que no son capaces de medir tiempos, ya que la película de emulsión empieza a registrar las trayectorias desde el principio y todas las trayectorias registradas durante el tiempo de exposición se superponen. Esto hace que este tipo de detectores no puedan utilizarse para muografía de dispersión, y limita mucho las aplicaciones en muografía de absorción, ya que no se puede controlar la evolución del sistema a lo largo del tiempo, por lo que no se pueden estudiar sistemas dinámicos. Otro gran problema es su sensibilidad a altas temperaturas, que también descarta muchos casos de uso.

2.4.2. Detectores Centelleadores

Construidos a base de un material centelleador, material que emite luz por fluorescencia cuando los átomos han sido excitados por el paso de una partícula cargada, estos detectores funcionan recolectando la luz que generan las partículas a su paso en un fotomultiplicador, que transforma la luz en una señal eléctrica que contiene información sobre la cantidad de energía liberada por la partícula mediante ionización.

Estos detectores pueden fabricarse de distintos materiales, pero es muy común en muografía utilizar plástico ya que es fácil de trabajar con él y es bastante resistente. Esta facilidad de trabajo permite construir detectores con formas muy diversas, una de las cuales muy utilizada en la muografía son los hilos, con los que se pueden formar distintas capas para obtener muy buena resolución en ambas direcciones del plano.

2.4.3. Detectores de Gas

Este tipo de detectores utilizan un gas como el material sensible al paso de las partículas. El gas se ioniza por el paso de la partícula, y aplicando un campo eléctrico se recolectan los electrones generados hacia unos electrodos que generan una señal eléctrica. Dependiendo de la diferencia de potencial entre los electrodos existen diferentes tipos de detectores gaseosos, entre los cuales se encuentran las cámaras proporcionales multihilo, que son los detectores que se usarán durante el resto del trabajo.

Como ventajas generales de los detectores de gases, utilizan menos canales electrónicos y son más simples en general que los centelleadores. Además se pueden construir detectores con alta resolución con menor coste, lo que los hace una mejor opción sobre los detectores centelleadores.

Las cámaras multihilo constan de distintas capas de hilos rotadas 90º para conseguir una resolución angular muy alta. Los hilos hacen las veces de ánodos, y las paredes que contienen el gas hacen las veces de cátodos, generando el campo eléctrico necesario para que el detector recoga los iones y electrones generados por las partículas mediante ionización. Las señales eléctricas que se generan tienen información sobre la energía y la trayectoria de las partículas.

2.5. Algoritmos para Muografía de Dispersión

Contando con todos los datos proporcionados por los detectores, se pueden resconstruir las trayectorias de los muones. En el caso de la muografía de dispersión, se tienen las trayectorias tanto antes como después de atravesar el objeto. Para reconstruir la imagen del objeto analizado, este se discretiza en vóxeles (el equivalente a los píxeles en 3D), y mediante algún algoritmo se extrapolan las trayectorias dentro del volumen ocupado por el objeto. Existen diferentes algoritmos para llevar a cabo esta reconstrucción, siendo los más comunes el POCA (point of closest approach) y el MLEM (maxium likelihood expectation maximization).

2.5.1. Maximum Likelihood Expectation Maximization

Este algoritmo consiste en asignar la dispersión a las regiones de la imagen donde realmente ocurren las desviaciones de los muones. Durante varias iteraciones se representa la distribución de densidad que es más probable que cause la dispersión observada de los muones, convergiendo finalmente a la solución correcta.

El problema principal de este algoritmo es que se necesita un alto poder de computación para converger a la solución final, ya que el número de vóxeles necesarios para tener una buen resolución de imagen es muy alto. En la práctica se consideran algunas aproximaciones que simplifican el problema y ayudan a disminuir la carga computacional, pero aún así sigue siendo un método muy costoso y con pocos beneficios de precisión frente al método POCA.

Otro problema que tiene el método es la alta sensibilidad a los datos, que hace que pequeñas fluctuaciones estadísticas dificulten mucho el proceso de reconstrucción. Esto puede solucionarse con altos tiempos de exposición, que hagan despreciables las fluctuaciones estadísticas, pero no siempre es posible tener tiempos de exposición altos.

2.5.2. Point of Closest Approach

Este algoritmo es mucho más simple debido a que considera una serie de aproximaciones a la hora de reconstruir la imagen, siendo a su vez muy efectivo y requiriendo mucho menor potencia computacional. Se parte de las dos trayectorias conocidas, las de entrada y salida del muón, dadas por los detectores situados antes y después del objeto a reconstruir. Se considera que toda la desviación del muón se ha producido en un solo punto del volumen entre los detectores, y se toma ese punto como parte del propio objeto.

Las trayectorias de entrada y salida, que normalmente se aproximan como líneas rectas, se extrapolan hasta su punto de mayor acercamiento, tomando este punto como el punto de dispersión, y el ángulo entre ellas como el ángulo de dispersión. Estas aproximaciones funcionan muy bien en situaciones en las que se tienen objetos pequeños, estando el resto del volumen entre los detectores vacío.

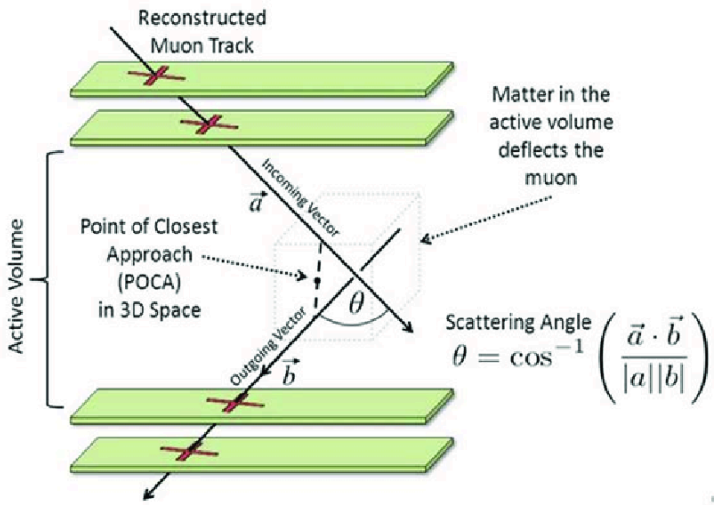

Figura 2.5: Representación de las rectas seguidas por los muones en la muografía, así como del punto POCA y del ángulo de dispersión [13].

En la Fig. 2.5 se puede ver un esquema de como sería la reconstruccción de las rectas seguidas por los muones, así como del punto POCA y del ángulo de dispersión. Debido a que este algoritmo es el que se va a utilizar durante el resto del trabajo, a continuación se tiene una explicación detallada de cómo obtener tanto el ángulo de dispersión, como el punto POCA. Primero hay que empezar definiendo las dos rectas:

$$\begin{aligned} r_1 &: \; x_1 = p_1 + \lambda \cdot \vec{v}_1 \\ r_2 &: \; x_2 = p_2 + \mu \cdot \vec{v}_2 \end{aligned} \tag{2.3}$$

Se puede identificar p_1 como el punto en el que se ha detectado el muón en la entrada, v_1 su velocidad al entrar, y análogamente p_2 el punto de detección en la salida y v_2 la velocidad a la salida. λ y μ son los parámetros de las rectas.

Conociendo las ecuaciones de las rectas, el ángulo de dispersión se puede obtener trivialmente como:

$$\theta = \arccos\left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1||\vec{v}_2|}\right) \tag{2.4}$$

Para encontrar el punto de mayor aproximación, lo primero es encontrar el segmento entre las dos rectas cuya longitud es mínima, y luego basta con obtener el punto medio de dicho segmento. Este segmento tiene un punto concreto en cada recta, el cual se puede denotar como:

$$\begin{aligned} x_{1_{\text{poca}}} &= p_1 + \lambda_{\text{poca}} \cdot \vec{v}_1 \\ x_{2_{\text{poca}}} &= p_2 + \mu_{\text{poca}} \cdot \vec{v}_2 \end{aligned} \tag{2.5}$$

Si se toma un vector $\vec{v}_3$, tal que $\vec{v}_3 \perp \vec{v}_1, \vec{v}_2$, este es el vector director del segmento que se busca, puesto que este ha de ser perpendicular a ambas rectas. Teniendo esto en cuenta, y sabiendo que tanto $x_{1_{\text{poca}}}$ como $x_{2_{\text{poca}}}$ son puntos pertenecientes al segmento, se puede escribir la relación:

$$x_{1_{\text{poca}}} + \rho_{\text{poca}} \cdot \vec{v}_3 = x_{2_{\text{poca}}} \tag{2.6}$$

En la que sustituyendo las equivalencias de $x_{1_{\text{poca}}}$ y $x_{2_{\text{poca}}}$ definidas en la Ec. (2.5) y despejando se llega a:

$$\lambda_{\text{poca}} \cdot \vec{v}_1 - \mu_{\text{poca}} \cdot \vec{v}_2 + \rho_{\text{poca}} \cdot \vec{v}_3 = p_2 - p_1 \tag{2.7}$$

que se puede escribir también de forma matricial para mayor claridad:

$$\begin{pmatrix} v_{1x} & -v_{2x} & v_{3x} \\ v_{1y} & -v_{2y} & v_{3y} \\ v_{1z} & -v_{2z} & v_{3z} \end{pmatrix} \begin{pmatrix} \lambda_{\text{poca}} \\ \mu_{\text{poca}} \\ \rho_{\text{poca}} \end{pmatrix} = \begin{pmatrix} p_{2x} - p_{1x} \\ p_{2y} - p_{1y} \\ p_{2z} - p_{1z} \end{pmatrix} \tag{2.8}$$

Solamente hace falta resolver el sistema para $\begin{pmatrix} \lambda_{\text{poca}} \\ \mu_{\text{poca}} \\ \rho_{\text{poca}} \end{pmatrix}$ y basta con sustituir los valores en la Ec. (2.5). Siguiendo cualquier método de resolución que se quiera, se llega a las expresiones finales para λ_{poca} y μ_{poca} (ρ_{poca} no es necesario obtenerlo), quedando finalemente:

$$\lambda_{\text{poca}} = \frac{\overrightarrow{(p_1 - p_2)} \cdot (\vec{v}_2 \times \vec{v}_3)}{\vec{v}_1 \cdot (\vec{v}_3 \times \vec{v}_2)} \qquad \mu_{\text{poca}} = \frac{\overrightarrow{(p_1 - p_2)} \cdot (\vec{v}_1 \times \vec{v}_3)}{\vec{v}_1 \cdot (\vec{v}_3 \times \vec{v}_2)} \tag{2.9}$$

Para comprobar la correcta implementación del algortimo, se utilizaron los datos de muografía que se tomaron para el "Variable Energy Cyclotron Centre" de Calcuta, donde se utilizaron como objetos a reconstruir las letras que conforman su acrónimo construidas en plomo. En la Fig. 2.6 se puede observar la imagen en 3D reconstruida a partir de las ecuaciones finales obtenidas.

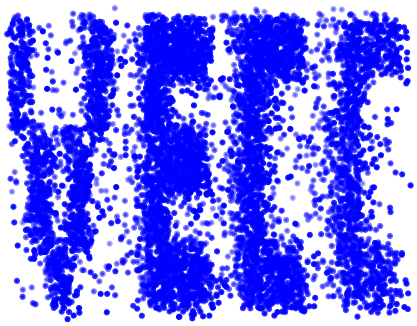

Figura 2.6: Reconstrucción de una imagen en 3D obtenida aplicando el algoritmo POCA a los datos de unas muestras tomadas para el "Variable Energy Cyclotron Centre" de Calcuta, donde el objeto a reconstruir eran las letras que forman su acrónimo construidas en plomo.

2.6. Problema de la Tubería

Como ya se ha comentado, la tomografía muónica puede aplicarse en muchos escenarios distintos. El caso concreto que se va a tratar de ahora en adelante es el de una tubería. Las tuberías son un caso muy común de objeto que puede requerir mantenimiento, pero que a su vez puede encontrarse en lugares de difícil acceso, por lo que una técnica que te permita reconstruir la estructura de una tubería sin necesidad de dañarla o tener acceso a ella, es muy útil para llevar a cabo revisiones periódicas, controlar posibles atascos u otras aplicaciones.

En la Fig. 2.7 se muestra un diagrama de cómo es la distribución de los detectores y la tubería en el caso de estudio. Se tiene un detector superior con dos planos, para poder medir con precisión la posición y dirección del muón al entrar, y en la parte inferior se tiene la misma distribución

para medir la posición y dirección a la salida. En el centro se encuentra el objeto a reconstruir, la tubería, que ocasiona la dispersión de los muones.

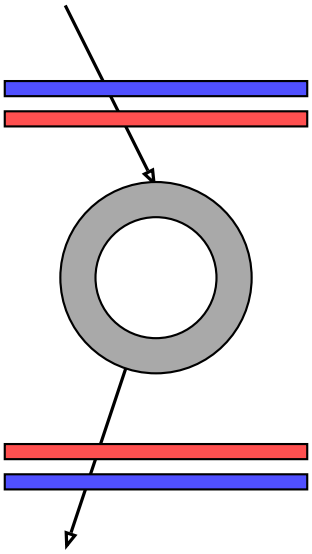

Figura 2.7: Representación de la distribución de detectores en el problema de la tubería. Se tiene un detector con dos planos en las partes superior e inferior para poder medir tanto a la entrada como a la salida la posición y dirección del muón.

Desde el punto de vista de la reconstrucción de la imagen, las tuberías pueden estar hechas de muchos materiales, así como tener grosores diferentes, ambos datos que se han tenido en cuenta en el estudio que se ha realizado. En la Fig. 2.8 se muestran dos ejemplos de tuberías reales típicas de este problema.

Figura 2.8: Ejemplo de dos tuberías reales con distintos tamaños y materiales a las que se les puede aplicar la muografía de dispersión.

Para conseguir las muografías de dispersión de muchas tuberías con distintos tamaños, grosores y materiales, en vez de medir datos reales, que sería un proceso muy largo y costoso, se ha optado por utilizar el programa de simulación Geant4 [14], que es capaz de generar simulaciones del paso de partículas por la materia de manera muy realista. Se han utilizado para las simulaciones los detectores “Proportional Multiwire chambers”, que cuentan con 206 hilos en cada plano de detección y la separación entre los mismos es de 4 mm.

3 Inteligencia Artificial: Redes Convolucionales

En el campo de la inteligencia artificial los desarrollos que van surgiendo con el tiempo no siempre vienen acompañados de publicaciones o artículos con explicaciones detalladas de los mismos. Es por esto que para la realización de este capítulo se ha utilizado como referencia la guía de PyTorch para deep learning (aprendizaje profundo) [15], un campo dentro de la inteligencia artificial y el machine learning, al cual pertenecen las redes neuronales como la que se tratará a lo largo de todo el trabajo.

3.1. Machine Learning

El machine learning (aprendizaje automático en español), es un campo de las ciencias computacionales que tiene como objetivo desarrollar técnicas que permitan a los ordenadores aprender. El término "aprender" en este caso se define como la mejora en el desempeño de la tarea con la experiencia y el uso de datos que se proporcionen. Es una rama de la inteligencia artificial, una disciplina cuyo propósito es conseguir imitar la inteligencia humana a través de algoritmos o sistemas informáticos.

Cuando se desarrolla un sistema de aprendizaje automático, los tipos de algoritmos que se pueden emplear se pueden clasificar según como es el aprendizaje que lleva a cabo el sistema. Los principales tipos de algoritmos son:

- **Aprendizaje supervisado:** En este tipo de algoritmos se cuenta con un conjunto de datos etiquetados, es decir, se tienen tanto los datos de entrada como los que se quiere obtener a la salida. Este tipo de aprendizaje engloba dos tipos de modelos principales, los algoritmos de regresión, que predicen valores reales, y los de clasificación, que predicen una categoría discreta.
- **Aprendizaje no supervisado:** En este aprendizaje, al contrario que en el supervisado, solamente se proporcionan los datos de entrada, sin etiquetar, por lo que el sistema no tiene de antemano los datos con los que puede aprender, tiene que ser capaz de reconocer patrones para poder clasificar las nuevas entradas.
- **Aprendizaje semisupervisado:** Este tipo de algoritmos aprovecha las virtudes de los dos anteriores, suministrando al sistema tanto datos etiquetados como sin etiquetar. Normalmente se utiliza un pequeño grupo de datos etiquetados y un gran grupo de datos sin etiquetar, de esta forma los datos etiquetados ayudan a guíar al sistema en el resto del entrenamiento.
- **Aprendizaje por refuerzo:** Este modelo de aprendizaje cuenta con un sistema de recompensas que le indica cuándo una acción es deseada o no, aprende mediante ensayo y error. Se proporcionan datos no etiquetados y las respuestas del sistema se evalúan siguiendo el sistema de recompensas, incentivando así las decisiones que lleven al objetivo final. Con el

sistema de recompensas el propio sistema aprende a aplicar las buenas decisiones y desechar las malas.

Como ya se ha mencionado, para llevar a cabo cualquier tipo de aprendizaje se necesita un conjunto de datos. Normalmente durante el aprendizaje de cualquier sistema, se tienen dos conjuntos de datos, los datos de entrenamiento y los datos de prueba. Ambos son parte del mismo conjunto de datos, pero este se ha separado en dos subgrupos, uno de ellos utilizado para que el sistema aprenda, y el otro utilizado para comprobar la efectividad del mismo.

A priori esta separación puede parecer que no tiene sentido, ya que sistema se puede comprobar también con los propios datos de entrenamiento, por lo que parece que no hay necesidad de separarlos. Realmente separar los conjuntos de entrenamiento y evaluación tiene diferentes ventajas. La primera de ellas es la generalización, el objetivo final del sistema es que sea capaz de predecir valores sobre datos que nunca antes ha visto, por lo tanto emplear los datos de entrenamiento para su evaluación no es de mucha utilidad para medir la precisión del sistema.

Otra ventaja es evitar el problema del sobreajuste, es decir, el problema de que el sistema "memorice" los datos de entrenamiento, de forma que con esos datos siempre daría un resultado muy bueno pero con unos nuevos no sería capaz de predecir nada. Es muy importante asegurarse de que el sistema no tiene este problema.

Desde sus inicios han existido muchos paradigmas diferentes de aprendizaje automático, como las máquinas de vectores de soporte o las redes bayesianas, pero en los últimos años ha adquirido mucha popularidad uno que inspira su diseño en como funciona el sistema nervioso de los animales, las redes neuronales. Es este el paradigma que se utilizará de aquí en adelante en el trabajo.

3.2. Redes Neuronales

Las redes neuronales, son un modelo dentro del aprendizaje automático que se inspira en el funcionamiento del sistema nervioso de los animales. Tienen como base unos nodos conectados entre sí, lo que se identificaría como las neuronas y sus conexiones. Cada una de estas neuronas artificiales recibe información, la procesa y envía el resultado a las siguientes neuronas que tenga conectadas.

La información que reciben las neuronas son números reales, que vienen de las distintas conexiones que tiene la neurona. El procesamiento consiste en sumar de forma ponderada todos los valores de entrada multiplicándolos por un **peso** y aplicar una función no lineal, llamada **función de activación**. Esta función es necesario que sea no lineal, ya que introducir no linealidad al modelo es lo que le permite a la red eventualmente aprender relaciones complejas de los datos. Este **peso** le da o quita importancia a distintas entradas y es un parámetro que se ajusta durante el entrenamiento. A la suma ponderada se le añade un término extra conocido como **bias** (sesgo en español), que hace las veces de término independiente de la función no lineal, permitiendo a la neurona "activarse" aunque todos los valores de entrada sean cero, o desplazando la función de activación.

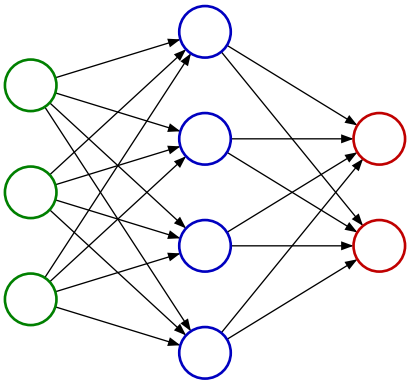

Figura 3.1: Esquema de una red neuronal en la que cada nodo representa una neurona y cada flecha una conexión. Tiene tres capas, una de entrada, una oculta y una de salida. Cada neurona tiene conexiones con todas las neuronas de la siguiente capa, y en el caso de la capa oculta, también tienen conexiones con todas las neuronas de la capa anterior.

Las redes neuronales están organizadas en capas, que no son más que conjuntos de neuronas. En la Fig. 3.1 se muestra un esquema de una red neuronal que en este caso contiene tres capas. Existen tres tipos de capas, la capa de entrada, que recibe los valores inciales y cuenta normalmente con una neurona por cada valor inicial, las capas ocultas, que son capas intermedias por las que van pasando los valores y modificándose. Puede haber una o varias de estas capas, y son donde ocurre la mayor parte del procesamiento de los datos. Por último está la capa de salida, que devuelve el resultado final, y donde el número de neuronas y sus funciones de activación dependen del tipo de problema que se esté resolviendo.

3.2.1. Aprendizaje

En una red neuronal, el aprendizaje consiste en encontrar los valores óptimos para los parámetros de todas las neuronas (pesos y bias), de forma que la red sea capaz de predecir los valores esperados. Para conseguir esto las redes neuronales típicamente se entrenan utilizando la minimización de riesgos empíricos, es decir, el objetivo es optimizar los parámetros de la red para minimizar la diferencia entre el resultado que se espera y el que predice la red. Debido a que hay una comparación entre el valor esperado y la predicción, este método implica el uso de datos etiquetados, por lo que como se ha visto en la sección anterior, este aprendizaje es supervisado.

El proceso de aprendizaje se puede dividir en diferentes pasos:

1. **Inicialización:** Los pesos y sesgos iniciales se inicializan aleatoriamente, normalmente con valores pequeños.
2. **Propagación hacia adelante:** Se hacen pasar los datos de entrenamiento por toda la red, obteniendo para cada uno la predicción.
3. **Cálculo de la pérdida:** En este punto se lleva a cabo la comparación entre el resultado esperado y la predicción de la red, para lo cual se utiliza una **función de coste** que cuantifique cuánto de equivocada está la red en su predicción.
4. **Propagación hacia atrás:** El algoritmo de propagación hacia atrás es el encargado de corregir a la red. Para ello propaga el error calculado en el paso anterior hacia atrás. A cada neurona se le asigna su contribución al error total obtenido calculando los gradientes de la función de coste con respecto a los pesos y sesgos de esa neurona.

5. **Optimización:** Una vez se ha calculado la contribución al error para cada neurona, se ajustan adecuadamente los parámetros de las mismas para reducir el error. El algoritmo más común que se utiliza es el **descenso del gradiente**. Este algoritmo permite encontrar mínimos locales en funciones diferenciables. Para ello lo que se hace es variar el valor de los parámetros una pequeña cantidad, que se ajusta con un parámetro conocido como **learning rate** (tasa de aprendizaje), en la dirección decreciente del gradiente, osea en dirección opuesta al gradiente, ya que este siempre apunta en la dirección de máximo crecimiento. De esta forma se avanza poco a poco hacia un punto donde el error es mínimo.

Una iteración de todos los pasos a partir del segundo se denomina "época", y estos se repiten muchas veces, con algún criterio de parada, por ejemplo se pueden fijar el número de épocas a iterar o se puede establecer un criterio en el que se considera que la red ha alcanzado el nivel de rendimiento necesario.

3.2.2. Tipos de Redes Neuronales

Dentro de las redes neuronales existen muchos tipos distintos según el tipo de problema que se pretende que resuelvan, según su estructura de capas o según su modelo de aprendizaje. Algunos de los modelos más utilizados son:

- **Redes neuronales feedforward:** Son el tipo más básico, con un conjunto de capas a través de las cuales va pasando la información hacia adelante, desde la capa de entrada hasta la de salida. Suelen aplicarse a problemas de clasificación o regresión, y tienen la ventaja de ser sencillas de construir. Como desventaja no son capaces de trabajar con datos que tengan dependencias espaciales o temporales.
- **Redes neuronales convolucionales:** Tienen una arquitectura inspirada en la corteza visual de los animales, estando diseñadas especialmente para procesar imágenes. Este modelo de red neuronal es el que se va a utilizar para este trabajo, por lo que se tratará más en profundidad en la siguiente sección.
- **Redes neuronales recurrentes:** Diseñadas especialmente para tratar con datos donde el orden de los mismos es importante. Cuentan con conexiones que permiten el uso de los valores de salida como entrada para la próxima iteración, guardando así una "memoria" de la información procesada hasta el momento. Suelen aplicarse entre otras a procesamiento del lenguaje o reconocimiento de voz. Un gran problema que puede aparecer en este tipo de redes es el desvanecimiento o explosión del gradiente, en secuencias muy largas de aprendizaje los gradientes pueden volverse muy pequeños (desvanecimiento) o muy grandes (explosión), dificultando mucho el entrenamiento.
- **Redes generativas adversativas:** Este modelo aprovecha el aprendizaje no supervisado creando dos redes neuronales que compiten entre sí. Una de ellas, el generador, intenta crear datos inventados que puedan pasar por datos reales como los utilizados para el entrenamiento, empieza generando ruido y aprende a convertirlo en datos realistas. La otra red es el discriminador, que tiene que ser capaz de distinguir entre los datos que son verdaderos y los falsos que el generador intenta crear. Esta competición del generador intentando engañar al discriminador, y este a su vez intentando no dejarse engañar, lleva a una generación de datos artificiales de muy buena calidad. La aplicación más común de este tipo de redes es la generación de imágenes.

3.3. Redes Neuronales Convolucionales

Como se ha comentado en la sección anterior, las redes convolucionales están inspiradas en la corteza visual de los animales, lo que les proporciona una arquitectura especializada para procesar datos en forma de cuadrícula, como por ejemplo imágenes. Su diseño inicial se desarrolló por primera vez en 1980 [16], y desde entonces se ha perfeccionado tanto su diseño como el método de entrenamiento.

El entrenamiento de este tipo de redes es muy similar al entrenamiento de una red neuronal. Normalmente se utiliza un aprendizaje supervisado, donde el conjunto de datos suelen ser pares de imágenes, la imagen que se le pasa a la red, y la que se espera obtener como salida. A partir de una función definida de pérdida, se indica a la red la diferencia entre la salida y la imagen que se quería obtener, para que pueda ajustar sus parámetros correctamente.

3.3.1. Arquitectura

Habitualmente las redes convolucionales constan de varias capas especializadas que realizan distintas funciones transformando los datos de entrada en las salidas esperadas. Primero se encuentra una capa de entrada, que recibe los datos iniciales, en caso de ser una imagen será una matriz de píxeles. Luego hay una sucesión de diferentes capas ocultas que sirven distintos propósitos, capas convolucionales, de agrupación y totalmente conectadas, y por último tienen una capa de salida, que produce la predicción de la red [17]. Dependiendo del problema que se esté tratando (clasificación o regresión), esta última capa tendrá unas características distintas.

Las capas convolucionales, que dan nombre al tipo de red, tienen como objetivo detectar características locales de los datos, para lo cual aplican una operación matemática llamada convolución. Estas capas cuentan con varios **filtros**, que no son más que matrices de pesos con tamaños menores que la de los datos iniciales, siendo los pesos variables durante el entrenamiento para ajustar los valores idóneos. Estos filtros sirven para resaltar ciertas características concretas en los datos, y para aplicarlos, se hace uso de la operación de convolución, la cual recorre los datos iniciales y hace una multiplicación elemento a elemento con los valores de la matriz de filtro y el subconjunto de datos correspondientes. Los resultados de las multiplicaciones se suman y se produce un valor en el *mapa de características*. En la Fig. 3.2 se muestra un diagrama en el que se aplica un filtro y se obtiene el mapa de características.

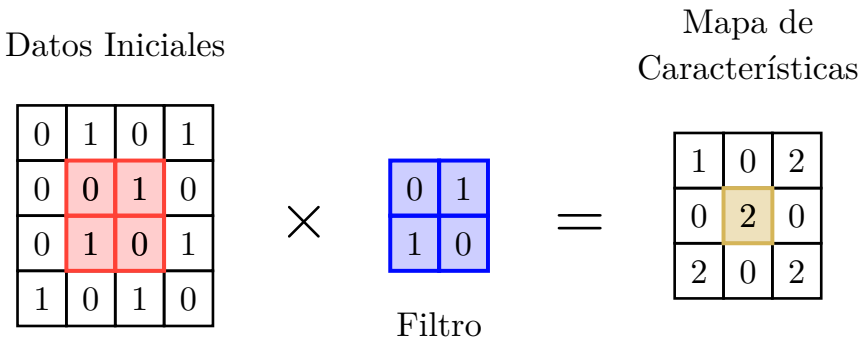

Figura 3.2: Representación de como actúa un filtro de una capa convolucional sobre los datos de entrada. Se multiplican elemento a elemento los valores del filtro y los de la ventana actual en los datos iniciales para obtener el mapa de características de ese filtro.

El mapa de características no es más que el conjunto de todos los valores obtenidos de la convolución de la imagen con uno de los filtros, por lo que la capa da como salida a la siguiente el conjunto de mapas de características formado por el mapa de cada filtro una vez estos hayan pasado por la función de activación de la capa, que normalmente es una función ReLU, la cual devuelve el máximo entre 0 y el valor que se le pasa. Estas capas convolucionales tienen otros parámetros como el **paso**, que indica cuánto avanza el filtro a la vez sobre los datos y el **relleno**, que añade datos vacíos a los bordes para que los filtros se puedan aplicar mejor en los bordes.

Las capas de agrupación tienen como objetivo reducir progresivamente la dimensionalidad de los mapas de características, disminuyendo así la cantidad de parámetros de la red y por tanto la carga computacional. Existen varias formas en las que aplicar esta reducción de dimensiones, pero las más comunes son el *max pooling* y el *average pooling*. En ambos se define una **ventana** (por ejemplo 2x2) y un **paso**, que se aplica a los datos, cogiendo como resultado el valor máximo o el valor medio de los datos que hay en la ventana en cada momento respectivamente.

Por último están las capas totalmente conectadas, que son muy similares a las de una red neuronal convencional. Estas recogen los datos aplanados y realizan la tarea de clasificación o regresión final, dependiendo del problema que se esté tratando.

Dentro de las redes neuronales convolucionales existen muchos tipos diferentes en cada uno de los cuales la distribución concreta de capas varía acorde con el problema que se quiere resolver con esa red en concreto. Una de las más utilizadas y la que se va a utilizar de ahora en adelante, es la U-Net.

3.4. U-Net

La U-Net es fundamentalmente una red neuronal convolucional, que tiene una arquitectura muy concreta desarrollada inicialmente para la segmentación de imágenes [1]. Fue diseñada con el propósito de ser capaz de entrenar con menos imágenes y que fuera muy precisa en la segmentación. La arquitectura general consiste primero en reducir el tamaño de la imagen a base de convoluciones y agrupaciones, extrayendo así sus características principales sin necesidad de calcular tantos parámetros, esta etapa se conoce como contracción.

Una vez se han extraído las características importantes, se pasa a la etapa de expansión, que aumenta de nuevo el tamaño de la imagen, típicamente hasta su tamaño original, pero manteniendo las características extraídas durante la etapa de contracción. Estas dos fases están directamente conectadas, como se puede ver en la Fig. 3.3, donde se representa una arquitectura típica de una U-Net.

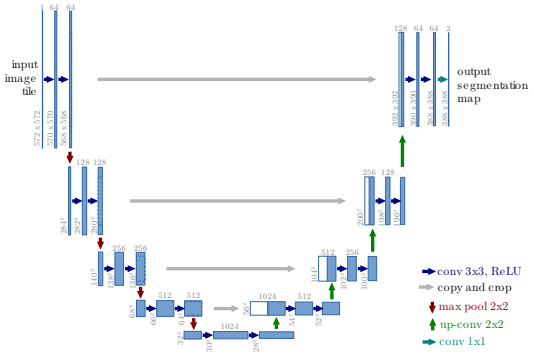

Figura 3.3: Esquema de la estructura de una red convolucional de tipo U-Net [1]. Cada rectángulo azul representa un conjunto de mapas de características.

Desde su primera aparición, este tipo de red convolucional se ha utilizado para diversas tareas relacionadas con el tratamiento de imágenes, principalmente la segmentación y el denoising (reducción de ruido). La segmentación consiste en identificar diferentes regiones de la imagen que sean un solo conjunto, como objetos o animales. Al entrenar una red para segmentación, lo que se obtiene como salida es una máscara, una imagen donde los píxeles están clasificados como parte del fondo, o parte de un objeto de la imagen que ha sido segmentado. En la Fig. 3.4 se muestra un ejemplo de segmentación, a la izquierda se tiene la imagen a segmentar y a la derecha la segmentación realizada por la red.

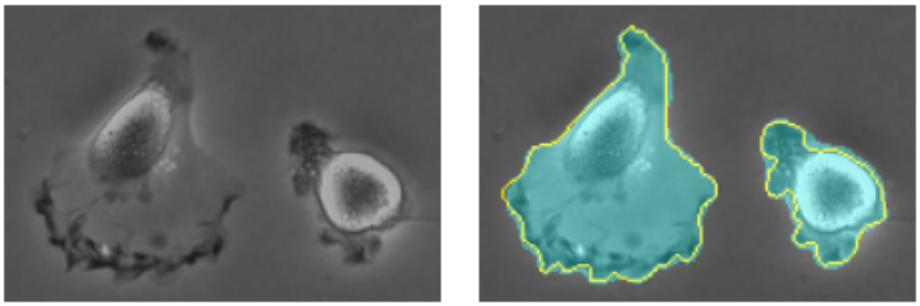

Figura 3.4: Ejemplo de una red U-Net realizando segmentación de imágenes. A la izquierda se tiene la imagen a segmentar, y a la derecha la salida segmentada de la red [1].

El denoising consiste en tomar imágenes con ruido y tratar de reducirlo o eliminarlo, intentando mantener la mayor cantidad de detalles de la imagen original posibles. Esta aplicación, que es la que se va a usar en este proyecto, se ha utilizado anteriormente en otros campos como la medicina [18]. En la Fig. 3.5 se observa una imagen con ruido y su posterior transformación al ser tratada con una U-Net.

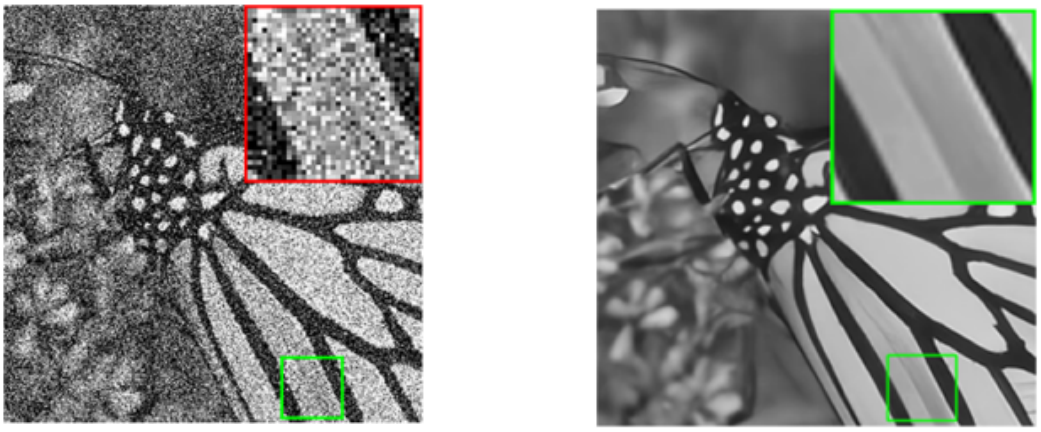

Figura 3.5: Ejemplo de reducción de ruido en imágenes con el uso de una U-Net [19].

Para este trabajo, se ha decididio que en lugar de diseñar una red neuronal convolucional desde cero, se ha buscado un diseño concreto de U-Net que estuviera diseñado para el tratamiento con imágenes, y así solamente tener que modificar pequeñas partes de la red si fuera necesario. El modelo elegido está sacado de un artículo [20], que incluye una explicación detallada sobre su construcción y diseño, así como una demostración de su uso.

4 Entrenamiento y Ajuste del Modelo U-Net

4.1. Sistema Hardware

Como ya se ha comentado, el objetivo del trabajo es entrenar una red neuronal, y estas requieren de un gran poder computacional para poder obtener resultados acertados en tiempos razonables. Es por esto que para la realización de este trabajo, se ha contado con acceso al servicio de computación del IFCA, el cual cuenta con más de 3000 núcleos con conexiones InfiniBand, que son conexiones de red de muy alta velocidad utilizadas para centros de computación de alto rendimiento. También cuenta con un almacenamiento de más de 3 petabytes administrados mediante la tecnología GPFS para almacenamiento de alto rendimiento y compartidos a través de las distintas particiones en las que está separado todo el poder de computación. Este sistema de almacenamiento es compartido entre todas las particiones para facilitar el acceso a los datos desde cualquiera de ellas.

Se puede acceder al sistema simplemente mediante una interfaz de usuario desde la que se pueden mandar a ejecutar jobs (tareas) a las distintas particiones existentes, cada una de las cuales está especializada en una tarea concreta. Para gestionar el sistema se utiliza "slurm" [21], un gestor de cargas de trabajo y planificador de tareas que permite acceder a las distintas particiones y planificar tareas a ejecutar, entre otras muchas funciones.

En el caso de este trabajo, se va a utilizar slurm para que proporcione una sesión interactiva con una partición que cuenta con 8 GPUs NVIDIA Ampere A30 de 24 GB de memoria, unidades de procesamiento gráfico diseñadas especialmente para la computación de alto rendimiento y para entrenar algoritmos de inteligencia artificial.

4.2. Muestras de Entrenamiento

Una vez se tiene el poder computacional necesario para entrenar la red, lo único que queda por hacer es preparar el conjunto de imágenes con el que se va a entrenar la red, y también el conjunto de prueba con el que se van a obtener los resultados finales. Como se ha comentado ya, se van a utilizar imágenes reconstruidas de tuberías de distintos tamaños y materiales, generadas por el software Geant4. Este programa proporciona los resultados de la simulación en archivos con formato `.root` que se transforman a archivos `.hd5` para mayor facilidad de manipulación. Este archivo contiene la información de las posiciones y direcciones de todos los muones simulados tanto antes de atravesar el objeto como después.

Junto con esta información también se tiene un archivo que contiene los datos de la tubería simulada. Para evitar el sobreajuste, además de dividir el conjunto de datos en conjunto de entrenamiento y de evaluación como se ha mencionado antes, también se pueden proporcionar datos que tengan distintos parámetros entre sí. En este caso las tuberías se han simulado

tomando un material aleatorio entre cinco materiales típicos de tuberías: plomo, cobre, hierro, acero y aluminio. Su posición en la imagen en ambos ejes se tomó con una distribución uniforme entre -2 y 2 y también se escogieron aleatoriamente los valores del radio interno, con una distribución uniforme entre 15,0 cm y 20,0 cm, y los valores del radio externo, con una distribución uniforme entre 16,0 cm y 21,0 cm, asegurando siempre que el radio interno es menor que el radio externo.

Con todos estos datos se desarrolló un script en el lenguaje de programación Python que a partir de los datos en ambos archivos proporcionase dos imágenes en 2D con 512 x 512 píxeles de tamaño y en escala de grises, una en la que se muestre la imagen reconstruida a partir del algoritmo POCA, y la otra que contenga una representación a escala de la tubería real, que será a partir de la cual se calcule la pérdida durante el entrenamiento. Dado que la imagen es en 2D, se ha escogido la visión frontal de la tubería, donde se observa el vacío central, y las dos circunferencias de la tubería, interior y exterior.

A la hora de crear la imagen reconstruida de la tubería, primero se calculan los puntos POCA, donde se descartan las situaciones donde las rectas de entrada y salida del muón son paralelas, puesto que no tienen sentido físico y su punto de máximo acercamiento es cualquier punto en el medio de ambas. También se descartan los puntos cuyo ángulo de dispersión sea menor que un cierto valor elegido, en este caso 0,06°, ya que una dispersión tan pequeña normalmente significa que el muón no ha sufrido dispersión con la tubería sino con el aire, lo que no es útil para la imagen. El valor se escogió observando cuál era un compromiso óptimo entre reducir ruido de la imagen y no perder información de dispersiones con la tubería.

Como última selección se descartaban los puntos que estuvieran muy separados del ancho y alto donde se encontrase la tubería, ya que no representan dispersiones reales con la tubería. Una vez se tienen los puntos finales que van a componer la imagen se representan en un histograma y se crea una imagen como la mostrada en la Fig. 4.1.

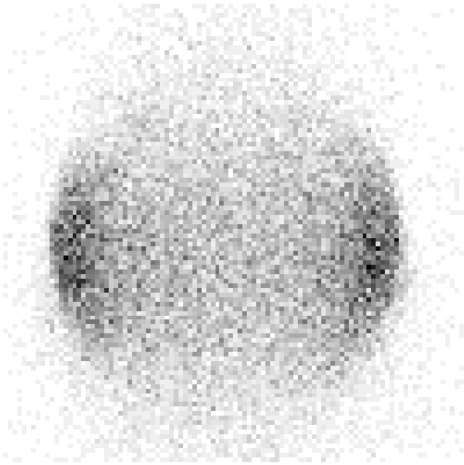

Figura 4.1: Ejemplo de imagen de una tubería reconstruida aplicando el algoritmo POCA.

Para construir la imagen de referencia de la tubería, primero se creó una matriz de píxeles blancos del tamaño de la imagen, para luego ajustar la escala de gris solamente en los correspondientes a la tubería. Para esto se calculó la distancia al centro de la tubería de cada píxel, teniendo en cuenta que la posición de la tubería puede estar desplazada con respecto al origen.

Una vez se tiene esa distancia se comprueba si el píxel está en el área comprendido entre los radios interior y exterior de la tubería, en cuyo caso ese píxel es un píxel de la tubería.

Como cada tubería puede ser de un material diferente, se estableció el valor de gris de los píxeles en función de la densidad del material, siendo un gris más oscuro cuanto más denso es el material. En la Tab. 4.1 se recogen las densidades de los distintos tipos de materiales que se consideraron.

Tabla 4.1: Materiales de los que están construidas las diferentes tuberías junto con su densidad.

Material	Plomo	Cobre	Hierro	Acero	Aluminio
Densidad / g cm^{-3}	11,34	8,96	7,87	7,85	2,70

Teniendo la matriz de píxeles completa, con los valores de los píxeles dentro de la tubería ajustados según la densidad concreta del material, solo queda transformar la matriz de píxeles a una imagen, como la que se muestra en la Fig. 4.2.

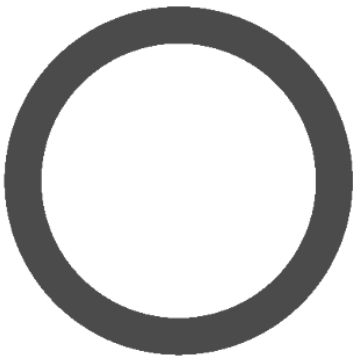

Figura 4.2: Ejemplo de imagen de referencia de la tubería, donde el material se indica a través de la escala de grises de los píxeles.

Como ya se ha explicado, es importante dividir los datos de los que se dispone en datos de entrenamiento y datos de evaluación de la red, por lo que en este caso, de los 1000 pares de imágenes que se han generado, 900 se utilizarán para el entrenamiento, y los otros 100 serán para realizar la evaluación de la U-Net.

4.3. Entrenamiento de la U-Net

Para entrenar la U-Net se va a aplicar un entrenamiento supervisado, utilizando 900 de los 1000 pares de imágenes generados. Como se comentó, se va a utilizar una U-Net ya diseñada, que solo se ha modificado para que la red devolviese las imágenes en escala de grises en vez de con los 3 canales rgb, ya que se va a trabajar con este tipo de imágenes.

El entrenamiento se ha llevado a cabo de nuevo a través de un script de Python, en el que se ha utilizado la librería de machine learning PyTorch para facilitar la implementación y poder hacer un buen uso de las GPUs disponibles para acelerar los entrenamientos. Se escogió como

función de pérdida la función MSELoss (mean squared error loss) proporcionada por PyTorch, que calcula el error cuadrático medio entre cada píxel de la imagen de salida de la red y la imagen de referencia de esa tubería concreta.

A nivel práctico el entrenamiento es bastante sencillo, primero se establece un número de épocas, que en este caso después de probar con distintos valores se determinó que 25 épocas eran adecuadas para no interrumpir el entrenamiento demasiado pronto pero tampoco alargarlo innecesariamente. Por cada época primero se obtienen las salidas del modelo para cada una de las 900 imágenes POCA, luego se calculan las pérdidas utilizando la función de coste mencionada para cada una de las salidas, se aplica la propagación hacia atrás con estas pérdidas, y por último se lleva a cabo la optimización.

Una vez se tiene preparado el entrenamiento, solamente queda la optimización de los hiperparámetros de la U-Net, el learning rate y el batch size. Estos se escogieron de manera cualitativa, poniendo a prueba distintas combinaciones de los mismos para determinar qué valores eran los óptimos para este problema concreto. El batch size es el número de muestras que se procesan en cada época de entrenamiento, es decir, cada cuántas muestras se actualizan los pesos de la red. Un batch size alto acelera los tiempos de entrenamiento pero hace que la red no aprenda tanto los detalles de las imágenes. En cambio un batch size pequeño, relentiza mucho el entrenamiento pero ayuda a que la red aprenda más detalles sobre las imágenes.

Para encontrar el batch size óptimo, se entrenó a la U-Net manteniendo el mismo learning rate pero con distintos valores del batch size, y se compararon visualmente las imágenes de salida obtenidas. Como se puede ver en la Fig. 4.3, se probó con un valor bajo, uno intermedio y uno alto, y la conclusión es que el batch size óptimo para este caso concreto es 1, ya que es con el que mejor resultados se obtienen. Esta misma comprobación se realizó para distintas tuberías y distintos learning rates, llegando a la misma conclusión.

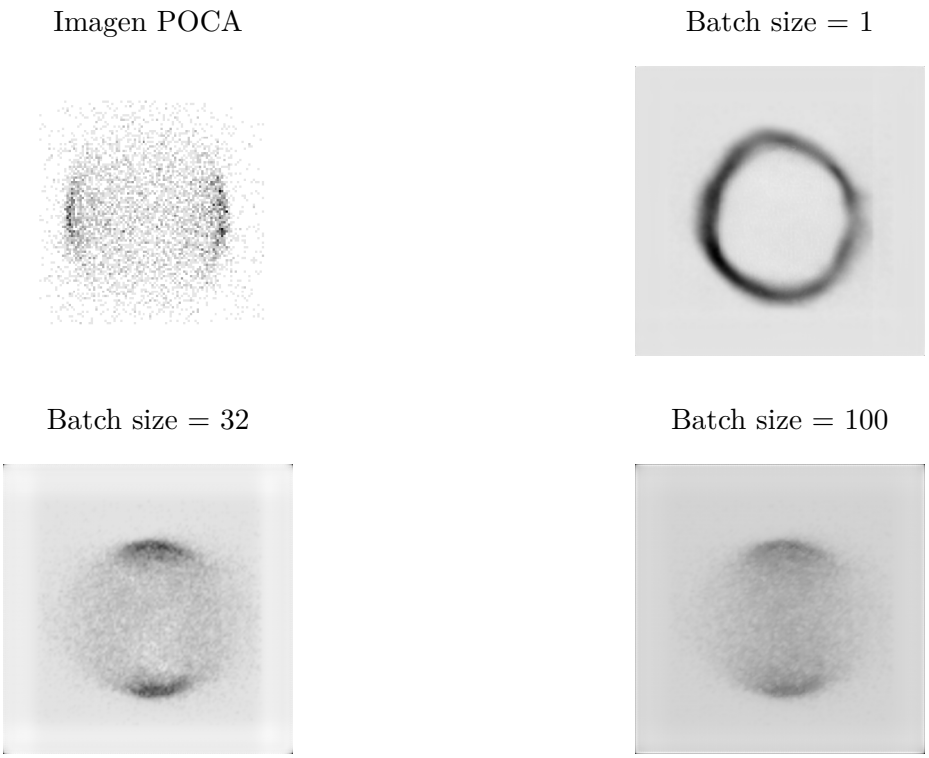

Figura 4.3: Ejemplo de una imagen obtenida mediante el algoritmo POCA y las imágenes de salida de la U-Net, todas con el mismo learning rate pero con distintos batch sizes.

El learning rate es la cantidad por la que se multiplica el valor de los parámetros de la red en la dirección decreciente del gradiente, para llegar finalmente al mínimo. También es muy sensible al igual que el batch size, un learning rate demasiado alto puede hacer que esos pequeños avances hacia el mínimo sean demasiado grandes y la red nunca sea capaz de alcanzarlo. Por otro lado si el learning rate es demasiado pequeño, existe la posibilidad de encontrar un pequeño mínimo local del que no se pueda salir con un paso tan pequeño, por lo que nunca se llegaría al mínimo deseado.

Actuando de forma similar que con el batch size, esta vez fijando este a 1 y utilizando varios learning rates, se obtuvieron diferentes resultados, como los que se ven en la Fig. 4.4, donde se muestran los resultados de evaluar la misma imagen POCA con entrenamientos con distintos learning rates, concluyendo que el valor óptimo para el mismo es $lr = 10^{-5}$. Al igual que con el batch size, se ha realizado esta comprobación con diferentes imágenes, obteniendo la misma conclusión.

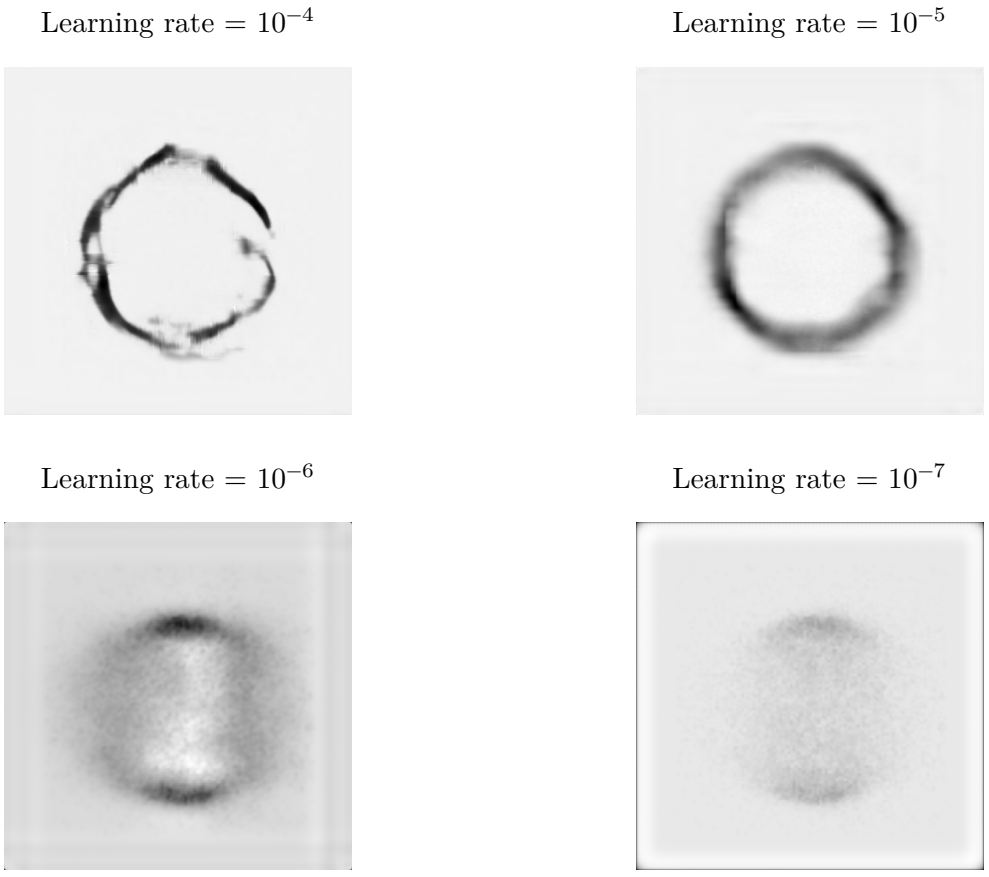

Figura 4.4: Ejemplos de imágenes de salida obtenidas con la U-Net para la misma imagen POCA con entrenamientos usando distintos learning rates.

Otra forma de encontrar el learning rate óptimo es observar el comportamiento de las pérdidas a lo largo del entrenamiento. En general se espera que la red comience con unas pérdidas relativamente altas, que luego irán bajando con cada iteración, llegando idealmente a cero. En el caso concreto que se está tratando, si se comparan por ejemplo las pérdidas con $lr = 10^{-7}$ con el valor óptimo, $lr = 10^{-5}$, se obtiene la representación recogida en la Fig. 4.5.

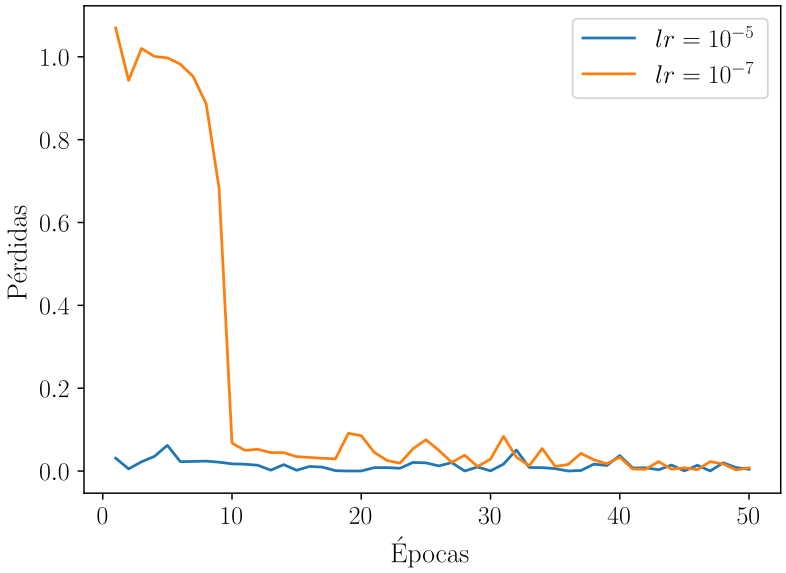

Figura 4.5: Representación de las pérdidas en el entrenamiento de la U-Net para dos learning rates diferentes.

Se puede ver como a pesar de que inicialmente se tiene un comportamiento más típico con $lr = 10^{-7}$, el valor final de las pérdidas es prácticamente el mismo, por lo que no existe una diferencia clara en este aspecto para distinguir que valor es mejor. Sin embargo observando también los resultados que se obtienen como se ha hecho anteriormente, se puede concluir que $lr = 10^{-5}$ es el valor óptimo para el learning rate.

5 Resultados y Validación del Modelo U-Net

Una vez se tiene todo el entrenamiento optimizado y la red entrenada, se evalúan las 100 imágenes que se habían separado para utilizarlas en la evaluación de la calidad de la U-Net. Con los resultados de las 100 imágenes, se han realizado dos tipos de análisis, uno cualitativo, que dado que se está tratando con imágenes es necesario, y uno cuantitativo, en el que se evalúa con detalle el rendimiento del modelo comparando directamente la salida con la imagen de referencia.

5.1. Análisis Cualitativo

Estudiando todas las salidas de la red de las imágenes de evaluación se pueden obtener diferentes métricas que evalúan el correcto funcionamiento de la U-Net. Su objetivo principal era eliminar el ruido de las imágenes POCA para así poder visualizar la tubería con claridad, por lo que una primera estadística que se puede sacar es el porcentaje de imágenes en las que se reconoce la tubería.

Entre las 100 imágenes de evaluación, se reconoce la tubería en 74 de ellas, por lo que se reconocen el 74% de las tuberías. Para poner un ejemplo del criterio que se ha tomado para determinar cuando se visualiza la tubería y cuando no, en la Fig. 5.1 se muestra un ejemplo de tubería que se reconcoce y otro que no.

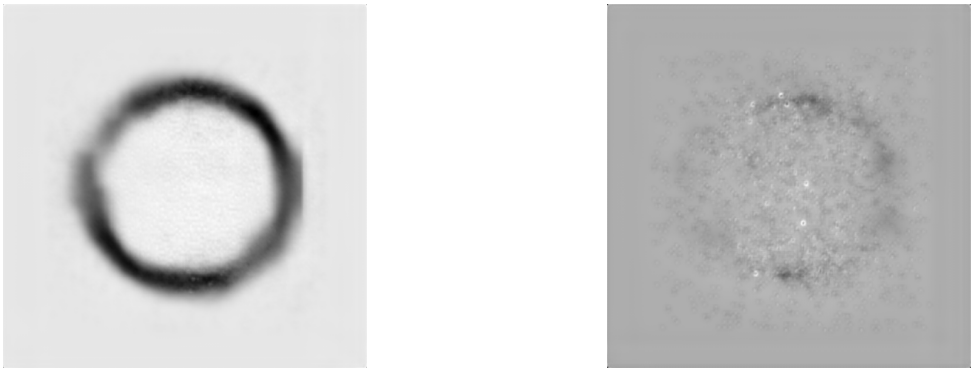

Figura 5.1: Ejemplo de una imagen generada por la U-Net en la que sí se ve la tubería (a la izquierda), y una imagen en la que no se distingue (a la derecha).

En base a las características que tenían las tuberías, grosor y material de construcción, también se pueden sacar datos estadísticos. En cuanto a lo que se refiere al material, es decir, a la densidad de la tubería, la U-Net parece no hacer ninguna distinción, puesto que no es posible en ninguna de las 100 imágenes reconocer el material en el que estaba fabricada la tubería. Al igual que antes, en la Fig. 5.2 se muestra una imagen cuya tubería originalmente era de cobre, y otra cuya tubería era de aluminio. Como se puede observar, ni siquiera con la gran diferencia

en densidades que hay entre ambos materiales se puede apreciar de qué materiales está hecha cada una.

Cobre ($\rho = 8{,}96\ \frac{g}{cm^3}$) Aluminio ($\rho = 2{,}70\ \frac{g}{cm^3}$)

Figura 5.2: Imágenes generadas por la U-Net que provienen de dos tuberías de materiales diferentes, a la izquierda cobre y a la derecha aluminio.

La última característica propia de las tuberías que es interesante ver si se observa en las imágenes dadas por la U-Net es el grosor de las mismas. Para comprobar si existe una correlación en el grosor, se clasificaron todas las imágenes cualitativamente en tres grupos, grosor bajo, medio y alto. Se hizo la misma divisón con las imágenes de referencia, aunque en este caso, dado que se conocían los grosores reales de las tuberías, se tomaron los grosores mínimo y máximo entre todas ellas y se dividió ese intervalo en tres grupos, para clasificar así las imágenes de referencia en los diferentes grosores.

Con esta información se obtuvo para cada grosor (bajo, medio, alto) la cantidad de imágenes de salida que se observaban con cada grosor, con respecto al total de imágenes de referencia con dicho grosor. Por ejemplo, de todas las imágenes de referencia que cuentan con grosor bajo, cuántas de ellas en la imagen de salida tienen grosor bajo, cuántas medio y cuántas alto. En la Tab. 5.1 se muestran en porcentaje todas las correlaciones entre los grosores de las imágenes de referencia y las imágenes de salida de la U-Net. Se han tenido en cuenta también las imágenes de salida en las que no se reconocía la tubería.

Tabla 5.1: Correlaciones entre los distintos grosores (bajo, medio, alto) de las imágenes de referencia y las imágenes de salida de la U-Net. También se han incluido las imágenes de salida donde no se reconoce la tubería.

Grosor		**Imágenes de Referencia**		
		bajo / %	medio / %	alto / %
Salida de la U-Net	bajo	6,25	0,00	4,76
	medio	41,67	41,94	33,33
	alto	31,25	22,58	38,10
	no se reconoce la tubería	20,83	35,48	23,81

Gracias a este análisis cualitativo de los resultados se puede observar como el modelo U-Net sí es capaz de reconstruir las tuberías la mayor parte de las veces, aunque se pierde la mayoría de la información sobre la tubería original, ya que no es posible identificar el material, y en muchos de los casos la U-Net no mantiene el grosor original con suficiente precisión.

5.2. Análisis Cuantitativo

Para llevar a cabo un análisis más cuantitativo de los resultados que se han obtendio, se han comparado una a una todas las imágenes de salida de la red en las que se puede reconocer la tubería con sus respectivas imágenes de referencia. Para poder comparar las imágenes, es necesario que ambas tengan los píxeles solamente en blanco y negro, y no tal y como están en escala de grises, por lo que hay que transformarlas antes de poder hacer la comparación.

Esto se puede conseguir aplicando una binarización a ambas imágenes, que consiste en situar todos los píxeles o en blanco o en negro, sin valores intermedios. Si se toman los valores típicos del color de los píxeles en escala de grises, donde el mínimo es 0 (negro) y el máximo es 255 (blanco), para binarizar una imagen solamente hace falta establecer un valor límite, por ejemplo 200, y darle a cada píxel de la imagen el valor 0 o 255 dependiendo de si el valor actual del píxel es menor o mayor respectivamente que este valor límite.

En el caso de las imágenes de referencia, encontrar este valor límite es sencillo, ya que cada una tiene un solo valor de los píxeles que dibujan la tubería, por lo que poniendo dicho valor como límite, todos los píxeles pertenecientes a la tubería se harán negros y el resto de la imagen quedará blanca.

Las imágenes de salida son un caso más complicado, ya que contienen píxeles con distintos valores de grises, por lo que se tomó un valor intermedio como valor límite, 150, manteniendo así un compromiso entre no convertir en blancos píxeles que realmente pertenecen a la reconstrucción de la tubería pero tampoco volver negros píxeles que pudieran haber sido parte del ruido que no se ha eliminado del todo.

Para mostrar un ejemplo de imagen de salida binarizada, se tiene la Fig. 5.3, donde se tiene tanto la imagen sin binarizar como binarizada, para que se observe como con este valor límite se alcanza un buen compromiso entre no contar el ruido como tubería y no contar como fondo partes de la tubería.

Figura 5.3: Ejemplo de una imagen de salida de la U-Net, a la izquierda sin binarizar, y a la derecha binarizada.

Una vez se tienen las imágenes binarizadas, se puede hacer la comparación entre sus píxeles. En concreto se han realizado dos comparaciones, la primera de ellas consiste en comparar cada píxel negro de la imagen de referencia con el correspondiente en la imagen de salida, si este también es negro (entendiendo que los píxeles negros son los que pertenecen a la tubería), ese píxel es un positivo verdadero, ya que según la imagen de referencia ahí debería haber un píxel negro y en la imagen de salida hay un píxel negro.

Teniendo el número de verdaderos positivos se puede definir una eficiencia, tomando este valor con respecto al número total de píxeles negros que hay en la imagen de referencia. Esta cantidad da una idea del porcentaje de acierto de la U-Net a la hora de reconstruir la tubería. Juntando los resultados para todas las imágenes, se ha construido el histograma de la Fig. 5.4, donde se representa el número de imágenes de salida en función del porcentaje de verdaderos positivos.

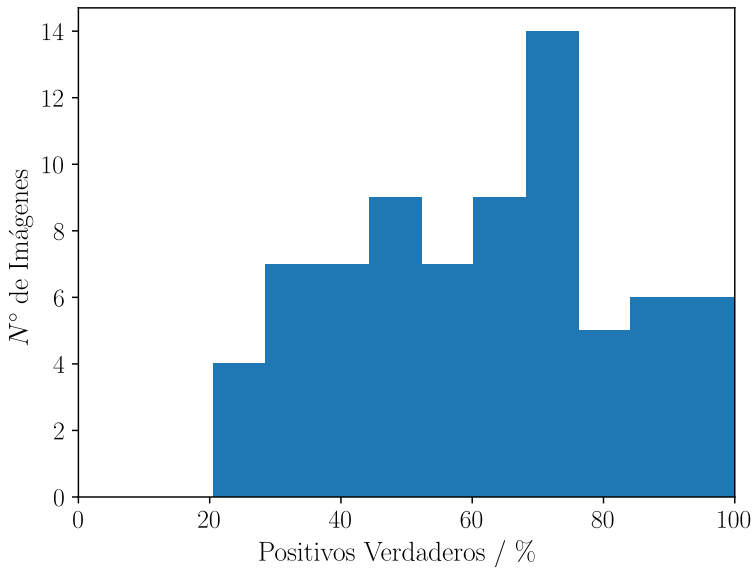

Figura 5.4: Histograma en el que se representa el número de imágenes de salida frente al porcentaje de verdaderos positivos al comparar con las imágenes de referencia.

Como se puede observar, la mayoría de las imágenes cuentan con un porcentaje de verdaderos positivos mayor al 60%, lo que indica que el modelo es capaz de representar la tubería con bastante acierto en la mayoría de ocasiones.

La otra comparación que se ha realizado consiste en observar esta vez los píxeles blancos de la imagen de referencia, si en la imagen de salida el píxel correspondiente es negro, significa que la red ha colocado un píxel negro en un lugar en el que no hay tubería, por lo que es un falso positivo.

Al igual que con los verdaderos positivos, se puede definir una eficiencia tomando el valor de los falsos positivos con respecto al número total de píxeles blancos que hay en la imagen de referencia. Agrupando los valores para todas las imágenes se obtiene el histograma de la Fig. 5.5, en el que se muestra el número de imágenes de salida en frente al porcentaje de falsos positivos.

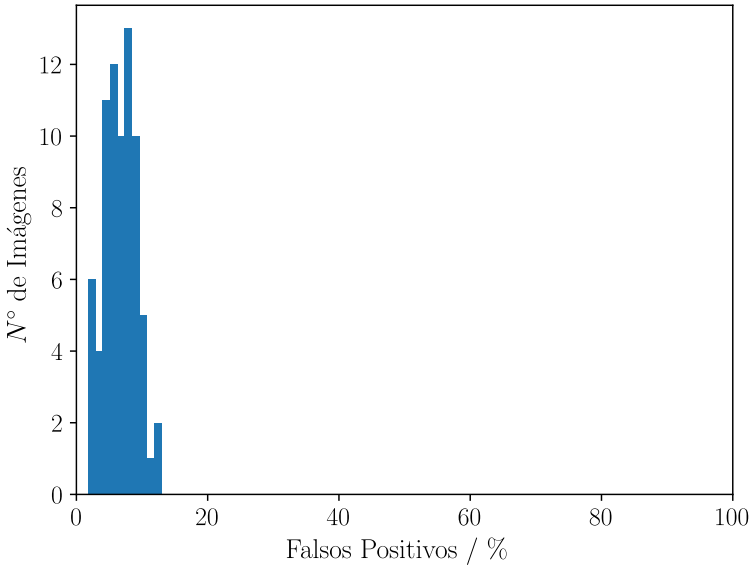

Figura 5.5: Histograma que muestra el número de imágenes de salida en función del porcentaje de falsos positivos al comparar con las imágenes de referencia.

Prácticamente todas las imágenes tienen un porcentaje de falsos positivos por debajo del 15%, indicador de que la U-Net a la hora de reconstruir las tuberías no comete muchos fallos en la interpretación de donde se encuentra la tubería, aunque como se ha visto por los resultados anteriores, no siempre sea capaz de reconstruirla con total exactitud.

6 Conclusiones

Como conclusión de este trabajo, el objetivo principal que se pretendía alcanzar era estudiar la viabilidad de un modelo de red neuronal convolucional, la U-Net, aplicado a la eliminación de ruido y artefactos presentes en imágenes generadas mediante tomografía muónica de dispersión, habiendo sido primero reconstruidas mediante el algoritmo POCA. En específico se ha estudiado el caso de imágenes de tuberías, un problema muy típico en la industria que puede ser de mucha utilidad dada la característica no destructiva de la técnica, así como no tener la necesidad de contar con acceso a la propia tubería para obtener la imagen.

Según los resultados que se han obtenido, este modelo puede ser un primer acercamiento hacia el estudio de otro tipo de redes neuronales que sean capaces de reconstruir mejor los detalles de las imágenes POCA. El modelo como se ha visto, es capaz de reconstruir en la mayoría de ocasiones una tubería reconocible, si bien no es capaz de mantener la información del material del que está compuesta la tubería, y no es del todo preciso a la hora de representar el grosor original de la misma.

Por otro lado se ha comprobado que aunque no se obtenga con precisión información sobre el grosor o el material, la reconstrucción que realiza el modelo es bastante similar a la tubería original, en el sentido de que comete muy pocos fallos a la hora de clasificar partes de la imagen como parte o no de la tubería.

En general, los resultados indican que entrenar una arquitectura U-Net directamente con imágenes POCA no es sencillo. El principal problema que puede haber es que estas imágenes POCA contienen demasiado ruido, y el modelo no es capaz de hacer una reconstrucción cuantitativa, manteniendo las características de la tubería, aunque sí es capaz de generar una reconstrucción cualitativa en la mayoría de ocasiones.

Otro problema que puede estar sucediendo es que las redes convolucionales U-Net suelen ser utilizadas para la eliminación de ruido o la eliminación de artefactos, pero nunca para ambas funciones a la vez, como se requiere en este caso. Quizá este modelo no esté capacitado para llevar a cabo las dos tareas simultáneamente sobre los mismos datos y por eso los resultados se acercan pero no llegan a los deseados.

La conclusión final es que el modelo de inteligencia artificial utilizado solamente puede llegar a ser viable si no se requieren detalles concretos de la tubería tales como el grosor o el material del que está compuesta, lo cuál es un escenario poco realista, ya que en general sí se van a necesitar dichas características. Por lo tanto para la mayoría de aplicaciones este modelo no es viable y se tendría que encontrar una alternativa que tuviese un mejor rendimiento.

6.1. Posibles Mejoras

Evidentemente este trabajo es solo un primer acercamiento al tratamiento de imágenes POCA con ciertos modelos de inteligencia artificial, y existen diferentes mejoras que pueden ayudar a obtener mejores resultados.

Sin necesidad de modificar la arquitectura ni de cambiar de tipo de red neuronal, existen dos mejoras que se pueden implementar. En este trabajo se han utilizado para el entrenamiento 900 pares de imágenes, lo cual para los modelos de inteligencia artificial de hoy en día es un conjunto de datos pequeño, por lo que algo que podría ayudar a mejorar los resultados que se han obtenido sería utilizar muchas más imágenes para el entrenamiento, mostrando así al modelo más variedad de tuberías, lo que podría facilitar su captación de los detalles.

Junto con esta mejora viene ligada la siguiente, y es mayor poder computacional. Esto de por sí no mejorará los resultados que se obtengan, pero añadiendo un número mucho mayor de imágenes, para evitar que los tiempos de entrenamiento no se alargen demasiado, sería útil contar con un mayor poder computacional del que se ha tenido durante la realización del trabajo.

La otra gran mejora que se puede aplicar en este caso es la de seguir un camino muy parecido al de este trabajo, pero en lugar de trabajar con una U-Net, utilizar otro modelo de red neuronal que también se aplique en el campo del tratamiento de imágenes. Este cambio puede ayudar a mejorar los resultados si verdaderamente la limitación en este caso es el modelo, y no la cantidad de datos de entrenamiento como se ha mencionado antes. Una recomendación de modelo podría ser una red neuronal convolucional con una estructura distinta a la U-Net, que tenga mayor capacidad para la eliminación tanto de ruido como de artefactos.

Referencias

[1] Olaf Ronneberger, Philipp Fischer, y Thomas Brox, «U-Net: Convolutional Networks for Biomedical Image Segmentation». doi: https://doi.org/10.48550/arXiv.1505.04597.

[2] G. Altarelli, «The Standard Model of Particle Physics». [En línea]. Disponible en: https://arxiv.org/abs/hep-ph/0510281

[3] S. Abachi et al, «Observation of the Top Quark», abril de 1995, *American Physical Society (APS)*. doi: 10.1103/physrevlett.74.2632.

[4] S. Chatrchyan et al, «Observation of a new boson at a mass of 125 GeV with the CMS experiment at the LHC», septiembre de 2012, *Elsevier BV*. doi: 10.1016/j.physletb.2012.08.021.

[5] Cush, *Modelo Estándar de Física de Partículas*. [En línea]. Disponible en: https://commons.wikimedia.org/w/index.php?curid=4286964

[6] M. tanabashi et al. (Particle Data Group), «Muon», vol. 98, n.º 30001, 2018.

[7] M. tanabashi et al. (Particle Data Group), «Passage of Particles Through Matter», vol. 98, n.º 30001, 2019.

[8] Alvarez *et al.*, «Search for Hidden Chambers in the Pyramids», marzo de 1970. doi: 10.1126/science.167.3919.83.

[9] H. K. M. Tanaka, «Japanese volcanoes visualized with muography». doi: 10.1098/rsta.2018.0142.

[10] D. Schouten, «Muon geotomography: selected case studies», 2019. doi: 10.1098/rsta.2018.0061.

[11] Vanini *et al.*, «Muography of different structures using muon scattering and absorption algorithms». doi: 10.1098/rsta.2018.0051.

[12] Mahon *et al.*, «First-of-a-kind muography for nuclear waste characterization», 2019. doi: 10.1098/rsta.2018.0048.

[13] Affum *et al.*, *Muon Imaging - Present Status and Emerging Applications*. 2022.

[14] S. Agostinelli *et al.*, «Geant4—a simulation toolkit», 2003. doi: https://doi.org/10.1016/S0168-9002(03)01368-8.

[15] Antiga, L.P.G. and Stevens, E. and Viehmann, y T., *Deep Learning with PyTorch*. Manning, 2020. [En línea]. Disponible en: https://books.google.es/books?id=BjszEAAAQBAJ

[16] K. Fukushima, «Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position», 1 de abril de 1980. doi: https://doi.org/10.1007/BF00344251.

[17] LeCun Yann, Bengio Yoshua, y Hinton Geoffrey, «Deep learning», mayo de 2015. doi: 10.1038/nature14539.

[18] Ju Zhang, Yan Niu, Zhibo Shangguan, Weiwei Gong, y Yun Cheng, «A novel denoising method for CT images based on U-net and multi-attention», 2023. doi: https://doi.org/10.1016/j.compbiomed.2022.106387.

[19] Zhang H, Lian Q, Zhao J, Wang Y, Yang Y, y Feng S., «RatUNet: residual U-Net based on attention mechanism for image denoising», 2022, *PeerJ Computer Science*. doi: https://doi.org/10.7717/peerj-cs.970.

[20] Mostafa Wael, «Cook your First U-Net in PyTorch». [En línea]. Disponible en: https://medium.com/data-science/cook-your-first-u-net-in-pytorch-b3297a844cf3

[21] M, Jette, y M. Grondona, «Slurm: Simple Linux Utility For Resource Management». [En línea]. Disponible en: https://slurm.schedmd.com/slurm_design.pdf