Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Segmentación de Anomalías en Imágenes Mamográficas con el Modelo LiteMedSAM

Tesis presentada al

Colegio de Física

como requisito parcial para la obtención del grado de

Licenciado en Física

por

Krystian Rodriguez Santos

Asesorado por

Dr. Benito De Celis Alonso
Dr. José Gerardo Suárez García

Puebla Pue.
noviembre 2025

Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

Segmentación de Anomalías en Imágenes Mamográficas con el Modelo LiteMedSAM

Tesis presentada al

Colegio de Física

como requisito parcial para la obtención del grado de

Licenciado en Física

por

Krystian Rodriguez Santos

Asesorado por

Dr. Benito De Celis Alonso
Dr. José Gerardo Suárez García

Puebla Pue.
noviembre 2025

Título: Segmentación de Anomalías en Imágenes Mamográficas con el Modelo LiteMedSAM
Estudiante: Krystian Rodriguez Santos

COMITÉ

Dr. Eduardo Moreno Barbosa Presidente	Dr. Juan Moisés Arredondo Velázquez Secretario

Mtr. Gerardo Uriel Pérez Rojas
Vocal

Dr. Benito De Celis Alonso
Asesor

Dr. José Gerardo Suárez García
Co-Asesor

A mis padres, por su amor incondicional, apoyo, enseñanzas y por creer en mí.

Agradecimientos

Quisiera expresar mi más sincero agradecimiento a todas las personas que han contribuido a la culminación de esta etapa de mi formación académica.

En primer lugar, a mis asesores de tesis, por su guía y paciencia. Gracias por ayudarme a realizar este trabajo de investigación.

A mis profesores, sus consejos y vivencias me inspiraron a continuar en esta etapa.

A amigos, Rafa y Zamudio por su constante motivación y las vivencias que marcaron mi paso por la facultad, y en general a mis compañeros que estuvieron en esta etapa.

A mi hermano, por las charlas sobre ciencia y por perseguir el sueño de ser científico.

Finalmente a mis padres, por el gran apoyo en mi vida, que sin ellos no estaría cumpliendo esta meta en mi vida. Gracias por enseñarme a no rendirme y a salir adelante.

Índice general

Índice de figuras

Capítulo 1

Introducción

El cáncer es una enfermedad que surge cuando las células del cuerpo comienzan a multiplicarse de manera descontrolada, invadiendo tejidos cercanos y propagándose a otras partes del organismo. Puede manifestarse en casi cualquier órgano o tejido, lo que da lugar a una amplia variedad de tipos de cáncer. Cada tipo recibe su nombre según el lugar del cuerpo donde se origina. Por ejemplo, el cáncer de seno, que comienza en el tejido mamario, conserva su denominación incluso si se extiende (metástasis) a otras áreas del cuerpo.

El cáncer de mama en mujeres ocupó el segundo lugar entre las principales causas de muerte a nivel mundial, representando el 11.6 % de los casos. En México, esta enfermedad se ha consolidado como una de las principales causas de mortalidad femenina. En 2023, según datos del Instituto Nacional de Estadística y Geografía (INEGI) [1] , se registró una tasa de 17.9 muertes por cada 100,000 mujeres mayores de 20 años. A nivel estatal, Sonora presentó la tasa de mortalidad más alta con 27.5 muertes por cada 100,000 mujeres, mientras que Campeche registró la más baja con 9.9. Por su parte, el estado de Puebla se ubicó con una tasa de 15.1 [1].

Cuando se sospecha de una patología mamaria maligna, ya sea por exploración clínica o mediante una mastografía de tamizaje, es necesario realizar una evaluación diagnóstica completa. Esta evaluación debe incluir una valoración clínica, estudios de imagen y, de ser necesario, una biopsia, la cual debe llevarse a cabo en un servicio especializado. Según estadísticas del INEGI, en México el 65 % de las mujeres entre 50 y 59 años practica la autoexploración, mientras que el 51.5 % se somete a mamografías.

La mamografía es el método más promovido en las campañas de detección temprana, especialmente por la Secretaría de Salud. Sin embargo, su interpretación presenta desafíos significativos. Los tejidos mamarios, como el conectivo, glandular y graso, tienen propiedades radiológicas similares, lo que dificulta la identificación de masas sospechosas, microcalcificaciones y otros indicadores de cáncer.

Actualmente, las mamografías digitales se implementan para las pruebas de tamizaje y diagnóstico, sin embargo, no todas las mamas son iguales, por lo que existen técnicas como la mamografía con contraste o CESM (*Contrast-Enhanced Spectral Mammography*, por su siglas en inglés), donde se utiliza un medio de contraste (generalmente yodo) que se inyecta en el torrente sanguíneo antes de la mamografía. El contraste ayuda a resaltar las áreas de la mama con mayor flujo sanguíneo, lo que puede ser indicativo de cáncer. Es útil en casos donde se necesita una evaluación más precisa, especialmente en mamas densas o cuando hay sospecha de cáncer.

1.1. Justificación

El cáncer de mama es una de las principales causas de mortalidad en mujeres a nivel mundial y la primera en México. Su detección temprana es fundamental para mejorar las tasas de supervivencia.

La mamografía es el método más utilizado para su diagnóstico; sin embargo, su interpretación depende en gran medida de la experiencia del radiólogo, lo que puede generar variabilidad en los resultados y errores en la detección de lesiones.

De acuerdo con cifras del INEGI, la población de mujeres entre 20 y 84 años en México ascendía a aproximadamente 36.7 millones en 2020 [2]. Por otro lado, el Sistema Nacional de Información en Salud reportó que, al 31 de diciembre de 2023, existen al menos 803 mastógrafos distribuidos en clínicas públicas a lo largo del país [3]. Aunque no existen cifras concretas sobre los módulos de mamografía móviles, es cierto que es insuficiente para el diagnóstico médico.

En este contexto, un modelo de aprendizaje profundo, que utiliza redes neuronales artificiales con múltiples capas para aprender y extraer características complejas de los datos, especialmente en áreas como la visión por computadora y el análisis de datos médicos, surge como una herramienta prometedora para reducir la carga de trabajo de los radiólogos, ofreciendo una alternativa que puede mejorar la precisión y eficiencia en la segmentación de anomalías mamarias. Actualmente, existen varios modelos que demuestran un alto desempeño en la segmentación de imágenes médicas. No obstante, la segmentación automática de estructuras mamarias representa un desafío debido a la baja relación de contraste, la variabilidad en la densidad del tejido y la presencia de artefactos en las imágenes.

Esta investigación tiene como objetivo evaluar un modelo de segmentación basado en Redes de *Vision Transformers* (un tipo de rede neuronal profunda), capaz de delimitar estructuras relevantes en mamografías con alta precisión mediante el uso de cuadros delimitadores. La relevancia de este estudio radica en un primer acercamiento a una nueva arquitectura de red neuronal artificial, ya que promete ser una gran alternativa para reducir la carga de trabajo de los radiólogos, facilitar la implementación de herramientas de apoyo clínico en hospitales y centros médicos. Además, la validación del modelo con bases de datos públicas y su comparación con metodologías convencionales permitirá evaluar su calidad de segmentación. En última instancia, este trabajo busca contribuir al desarrollo de herramientas de diagnóstico asistido por computadora, mejorando la accesibilidad y calidad del análisis mamográfico de una manera más eficiente y menos costosa.

1.2. Hipótesis

El uso de un modelo de segmentación basado en Redes de *Vision Transformers* permitirá mejorar la precisión en la detección y delimitación de anomalías en mamografías. Esto contribuirá a reducir la variabilidad en la interpretación de los radiólogos.

1.3. Objetivo principal

Evaluar la segmentación de anomalías en mamografías realizadas por el modelo *Segment Anything in Medical Images Lite* (LiteMedSAM) verificando su viabilidad operativa.

1.3.1. Objetivos Secundarios

1. Determinar la viabilidad operativa de LiteMedSAM: Analizar el desempeño y los requisitos computacionales de LiteMedSAM en una computadora portátil con recursos limitados y en una plataforma virtual.

2. Cuantificar la calidad de segmentación de LiteMedSAM: Evaluar la segmentación de anomalías después de afinar el modelo con mamografías digitales.

1.4. Resumen

La segmentación automática de imágenes médicas es un desafío clave en el diagnóstico asistido por computadora. Implementando una arquitectura de red neuronal profunda de transformador y haciendo uso de la transferencia de aprendizaje de un modelo ya entrenado, se pretende evaluar el modelo LiteMedSAM, para la segmentación de anomalías en mamografías.

Se utilizó un conjunto de datos compuesto por 643 imágenes del repositorio CDD-CESM (*Categorized Digital Database for Low energy and Subtracted Contrast Enhanced Spectral Mammography images*), en la modalidad de mamografía digital, con 80 % de entrenamiento y 10 % validación y 10 % para la prueba final. Se evaluó el desempeño del modelo mediante métricas como el Coeficiente de similitud de Dice (DSC) y Diferencia de superficie normalizada (NSD), además del análisis del tiempo de inferencia y los requisitos computacionales.

Los resultados obtenidos mostraron que LiteMedSAM supera en precisión a métodos de segmentación convencionales, alcanzando un DSC promedio del 83 % en la prueba. Sin embargo, se identificaron limitaciones en imágenes con alta densidad mamaria y bajo contraste, lo que sugiere la necesidad de una mejor optimización en el preprocesamiento de datos.

LiteMedSAM se comparó con otros modelos especializados en segmentación de imágenes, como U-Net [4], SegNet4, ResNet18 [5]. Cada uno de estos modelos especializados fue entrenado por igual con las mismas imágenes de diferentes modalidades, entre ellas la *mamografía*. Estas métricas sirven como una base sólida para compararlas con los resultados obtenidos en nuestra investigación, ya que durante la validación, estos modelos se utilizaron para segmentar imágenes de otras modalidades, demostrando así su desempeño.

Si bien LiteMedSAM tiene el potencial para integrarse en flujos de trabajo clínicos, su implementación requiere de validaciones adicionales con especialistas y una interfaz que facilite su uso por parte del personal médico, además de un análisis con conjunto de datos mayor. Se concluyó que este modelo representa una herramienta prometedora para la segmentación automática de anomalías en mamografías, ya que podría contribuir a la mejora del diagnóstico por imagen y la reducción de la carga de trabajo de los radiólogos.

Palabras clave: Segmentación de imágenes médicas, LiteMedSAM, red neuronal profunda, mamografía, diagnóstico asistido por computadora.

Capítulo 2

Revisión de literatura

La segmentación manual ha sido durante mucho tiempo la técnica empleada para delinear estructuras anatómicas y regiones patológicas, pero este proceso requiere de tiempo y alto grado de experiencia, lo que puede dificultar el diagnóstico y postergar el proceso de tratamiento.

La segmentación de imágenes por medio de inteligencia artificial con el empleo de redes neuronales, es una técnica, cuyo propósito es dividir una imagen digital en múltiples regiones o grupos discretos de píxeles, conocidos como segmentos. Cada uno de estos segmentos representa una parte significativa de la imagen, definida en función de ciertas características como la intensidad de los píxeles, el color, la textura o la continuidad espacial. Al identificar y agrupar los píxeles de interés, la segmentación facilita la identificación de objetos, la extracción de características y otras tareas relacionadas con el análisis visual [6]. Este enfoque es especialmente útil para analizar datos visuales complejos, ya que permite descomponer una imagen en componentes más manejables. Por ejemplo, en una imagen médica, los segmentos pueden corresponder a órganos, tejidos o tumores, mientras que en una fotografía natural, pueden representar diferentes objetos, como personas, vehículos o paisajes. Al dividir la imagen en estas partes significativas, la segmentación no solo simplifica la representación de la información visual, sino que también mejora la precisión y eficiencia de los algoritmos de procesamiento de imágenes.

Las técnicas de segmentación de imágenes abarcan un rango que va desde métodos heurísticos simples hasta sofisticadas implementaciones basadas en aprendizaje profundo. Los algoritmos tradicionales de segmentación analizan características visuales fundamentales de cada píxel, como el color o la intensidad, para delinear los contornos de los objetos y separar las regiones de fondo. Por otro lado, el aprendizaje automático utiliza conjuntos de datos anotados para entrenar modelos que permiten clasificar con alta precisión los distintos tipos de objetos y áreas presentes en una imagen, también conocido como aprendizaje supervisado.

Gracias a su flexibilidad, la segmentación de imágenes se aplica en una gran variedad de escenarios relacionados con la inteligencia artificial. Entre estos usos destacan su contribución al diagnóstico médico a través de imágenes. En el caso de las mamografías, esta técnica juega un papel esencial en la detección y caracterización temprana de lesiones sospechosas, tales como masas o microcalcificaciones, contribuyendo a la mejora en el diagnóstico del cáncer de mama.

2.1. Cáncer de Mama

El término *cáncer*, engloba un grupo de enfermedades caracterizadas por la formación de masas de células que crecen y se replican de manera descontrolada, con la capacidad de invadir y propagarse a otras partes del cuerpo distintas de su lugar de origen.

El cáncer de mama femenino ocupó el segundo lugar (2.3 millones de casos; 11.6 %) en las principales causas de muerte por cáncer a nivel mundial, después del cáncer de pulmón [7]. De acuerdo a la Organización Mundial de la Salud (OMS), se registran millones de nuevos casos anualmente.

En México también representa un problema de salud grave debido a la gran cantidad de fallecimientos asociados a éste, por lo que resalta la importancia de un diagnóstico temprano.

Para una detección adecuada del cáncer de mama, es esencial evaluar el riesgo de que puedan llegar a desarrollarlo durante su vida. Los principales factores de riesgo para el cáncer de mama incluyen: tener antecedentes familiares de esta enfermedad, haber recibido radiación en el área del pecho entre los 10 y los 30 años, antecedentes personales de ciertas afecciones mamarias como hiperplasia atípica o carcinoma *lobulillar in situ*, tener la primera menstruación antes de los 12 años, experimentar la menopausia después de los 55 años, no haber tenido hijos o haber tenido el primero después de los 30 años, utilizar terapia hormonal combinada de estrógenos y progesterona, consumir más de una bebida alcohólica al día, padecer obesidad y presentar tejido mamario denso en los estudios de mamografía[8]. Lo recomendable es que un profesional de la salud lleve a cabo una evaluación de riesgos en todas las mujeres que tengan entre 25 y 30 años; esto se realizaría durante un examen físico anual. Se estima que el riesgo de desarrollar cáncer de mama en la población general es del 12 %. Si se calcula que el riesgo de cáncer de mama de una paciente durante su vida es mayor al 20 %, se le consideraría de alto riesgo [8].

Existen técnicas de diagnóstico que pueden ayudar al médico a la evaluación del paciente de una manera más precisa, como la mamografía ya sea computarizada o estándar, la ecografía y la tomosíntesis digital de mama, que suelen ser técnicas no invasivas y de costó ligeramente accesible, y están las biopsias, que es una técnica invasiva y más comúnmente usada para identificar el cáncer en la paciente.

La detección temprana de anomalías a través de exámenes antes de la aparición de síntomas permite adelantar el diagnóstico, pero no modifica el curso natural de la enfermedad. No obstante, cuando el cáncer se identifica en sus primeras etapas, la tasa de supervivencia a cinco años aumenta considerablemente.

La mamografía de detección suele identificar cánceres de crecimiento lento, que generalmente tienen un mejor pronóstico en comparación con aquellos diagnosticados clínicamente, es decir, cuando el paciente ya presenta síntomas o acude al médico. Al incluir en las estadísticas tanto estos cánceres no progresivos como los potencialmente mortales, cuyo tratamiento temprano no altera el desenlace, la tasa de supervivencia a cinco años se incrementa. Sin embargo, esta mejora en las cifras no necesariamente se traduce en un mayor número de vidas salvadas mediante los exámenes de detección [8].

Los avances en las técnicas de diagnóstico por imagen, la evaluación de riesgos y las pruebas genéticas han resaltado la importancia de personalizar las recomendaciones para la detección del cáncer de mama. Sin embargo, tras una mamografía, a menudo se realiza una biopsia, que en algunos casos puede ser innecesaria, además de implicar un gasto de tiempo y recursos para las pacientes. Por ello, mejorar la precisión en el diagnóstico radiográfico es fundamental para aumentar el valor predictivo de la mamografía de detección y reducir la necesidad de procedimientos invasivos.

Diversos estudios han explorado técnicas de segmentación para imágenes mamográficas, desde métodos como el umbralizado y los modelos de contornos activos, hasta técnicas más avanzadas basadas en aprendizaje profundo, como redes convolucionales (CNNs) y arquitecturas especializadas en segmentación de imágenes (UNet, DeepLabV3+). Sin embargo, muchas de estas técnicas enfrentan desafíos en escenarios de imágenes médicas, especialmente en las mamografías [9].

2.2. Mamografía

2.2.1. Anatomía de la Mama

La mama es un órgano glandular de naturaleza predominantemente blanda, compuesto por una combinación de tejidos epiteliales, conectivos y adiposos. Estructuralmente, se organiza en lóbulos y lobulillos, que constituyen las unidades funcionales encargadas de la producción de leche durante la lactancia. Estos lóbulos están formados por células epiteliales especializadas y se conectan a través de una red de ductos que convergen hacia el pezón, permitiendo la secreción láctea.

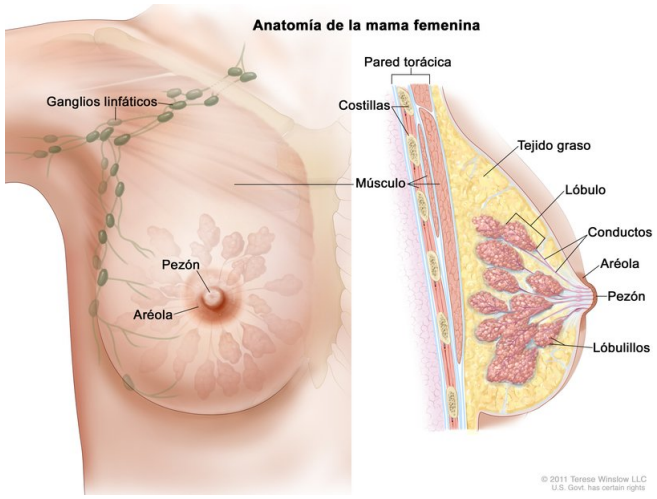

Figura 2.1: Anatomía de la mama femenina. A la izquierda, se muestra la vista frontal de la mama y a la derecha, se muestra la vista lateral de la mama. *Fuente: Terese Winslow (2011). National Cancer Institute(Ilustraciones Médicas en Español collection). Derechos de autor reservados. Reproducido bajo el principio de uso justo para fines académicos*

El tejido conectivo de la mama está compuesto principalmente por fibras de colágeno y elastina, que proporcionan soporte estructural y elasticidad al órgano. Este tejido se distribuye de manera heterogénea, formando septos que dividen los lóbulos y contribuyen a la firmeza de la mama. Además, el estroma mamario, que incluye tanto tejido conectivo como adiposo, varía en proporción según factores como la edad, el estado hormonal y la composición corporal. El tejido adiposo es otro componente fundamental de la mama, ya que rodea y protege las estructuras glandulares y ductales. Su distribución y volumen influyen en el tamaño y la forma del órgano, y su presencia es más abundante en mujeres posmenopáusicas debido a la regresión del tejido glandular (figura 2.1).

El seno de una mujer adulta está compuesto por estructuras glandulares y ductales, junto con un estroma formado por tejido fibroso que conecta los lóbulos individuales, además de tejido adiposo que se distribuye tanto dentro como entre estos lóbulos. Alrededor de 80 % a 85 % de la mama normal se compone de tejido adiposo [10].

Dado que la densidad másica y el número atómico de estos tejidos blandos son muy similares, las técnicas radiográficas convencionales no son efectivas para distinguirlas adecuadamente. En el rango de 70 - 100 kVp, la dispersión *Compton* es predominante en los tejidos blandos, lo que resulta en una absorción diferencial mínima entre ellos. Para mejorar la diferenciación de los tejidos, es necesario maximizar el efecto fotoeléctrico, lo cual se logra utilizando valores bajos de kVp. Este enfoque permite una mayor absorción diferencial y, por tanto, una mejor visualización de las estructuras mamarias [11].

2.2.2. Radiología de Tejidos Blandos

La evaluación radiográfica de tejidos blandos requiere métodos especializados que contrastan con los empleados en la radiografía convencional, donde se utilizan rayos X para obtener imágenes de los tejidos internos del cuerpo, también conocida como radiografía simple. Estas diferencias en las técnicas se deben a las características específicas de la anatomía que se examina. En la radiografía convencional, el contraste entre los tejidos es notable debido a las diferencias significativas en densidad másica y número atómico entre estructuras como el hueso, el músculo, el tejido adiposo y los pulmones.

En cambio, en la radiografía de tejidos blandos, el enfoque se centra en capturar imágenes de las estructuras musculares y adiposas. Estos tejidos presentan números atómicos efectivos y densidades másicas muy similares. Por ello, las técnicas radiográficas para tejidos blandos están diseñadas para maximizar la absorción diferencial entre estos tejidos, que poseen propiedades físicas muy parecidas. Un caso destacado de la aplicación de la radiografía en tejidos blandos es la mamografía, un estudio radiológico específicamente diseñado para examinar el tejido mamario.

La mamografía, sigue siendo la técnica estándar de oro en el diagnóstico de cáncer de mama. Este método se utiliza principalmente para detectar signos de cáncer en sus etapas iniciales, a veces hasta tres años antes de que sean palpables en una exploración física. Es una imagen de la mama obtenida mediante rayos X. Existen diversas técnicas para la captura de estas imágenes, que se clasifican en analógicas y digitales. Dentro de las digitales, se encuentran las mamografías bidimensionales (2D) y las tridimensionales (3D), estas últimas también conocidas como tomosíntesis digital de mama, tomosíntesis digital o simplemente tomosíntesis. Ambos tipos de mamografías, tanto la 2D como la 3D, se realizan utilizando el mismo principio físico.

Las mamografías involucran una cantidad muy pequeña de exposición a la radiación. De acuerdo a la Sociedad Americana contra el Cáncer (ACS, por sus siglas en inglés) [12], indica que la dosis de radiación que recibe una persona durante una mamografía de detección es aproximadamente la misma que recibe en su entorno natural también conocida como radiación de fondo, en un período de tres meses.

En promedio, la dosis total para un mamograma común a dos tomas para cada seno es de aproximadamente 0.4 milisieverts (mSv). Un mSv es una medida de la dosis de radiación. La dosis efectiva aproximada para una mamografía digital de detección es de 0.21 mSv, mientras que para la tomosíntesis digital de mama (mamografía 3D) es de 0.27 mSv. La mamografía es considerada una técnica altamente segura y efectiva [13]. Se estima que, por cada 1000 vidas salvadas, hay un solo caso de muerte asociada [11].

Existen dos tipos de exploración mamográfica. La mamografía de diagnóstico se realiza en pacientes que presentan síntomas o tienen factores de riesgo elevados. Sin embargo, es importante tener en cuenta que algunos de estos signos podrían estar asociados a problemas no cancerosos, conocidos como benignos. También es útil para analizar cambios detectados en una mamografía de cribado o en situaciones especiales donde el tejido es difícil de visualizar, como en el caso de personas con implantes mamarios.

Esta requiere más tiempo y utiliza una dosis total de radiación mayor, ya que se necesitan más radiografías tomadas desde diferentes ángulos de la mama. La mamografía de detección, que se realiza en mujeres asintomáticas mediante un protocolo de dos proyecciones, normalmente la oblicua lateral medial y la craneocaudal.

Para detectar un cáncer no sospechado, se suelen tomar dos o más imágenes de rayos X de cada mama. Comúnmente estas imágenes permiten detectar los tumores que no son posible palpar.

Con las mamografías de detección también se detectan microcalcificaciones (pequeños depósitos de calcio) que a veces indican que hay cáncer de mama [14].

Detectar el cáncer de mama en una etapa temprana a través de mamografías de detección permite iniciar el tratamiento al comienzo de la enfermedad, posiblemente antes de que se propague. También es importante considerar los beneficios de la mamografía de detección en conjunto con los posibles riesgos asociados, que incluyen lo siguiente:

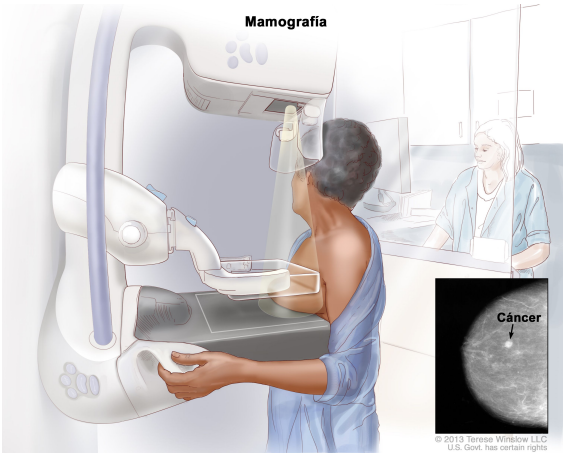

Figura 2.2: Mamografía Digital. La mama se presiona entre dos placas y se usan rayos-X para obtención de imágenes del tejido (mamograma). En un recuadro, se muestra una imagen radiográfica de la mama y se señala con una flecha un área circular de tejido anormal (cáncer). *Fuente: Terese Winslow (2011). National Cancer Institute(Ilustraciones Médicas en Español collection.). Derechos de autor reservados. Reproducido bajo el principio de uso justo para fines académicos*

Los **falsos positivos** en una mamografía ocurren cuando los radiólogos detectan algo anormal, pero realmente no existe cáncer. Cuando esto sucede, es necesario realizar pruebas adicionales, como una mamografía de diagnóstico, una ecografía o incluso una biopsia, para confirmar si se trata o no de cáncer. Estas pruebas complementarias pueden ser costosas, consumir tiempo y causar molestias físicas. Los falsos positivos son más comunes en mujeres jóvenes, con mamas densas, antecedentes de biopsias mamarias, historial familiar de cáncer de mama o aquellas que toman estrógenos, como en el caso de la terapia hormonal para la menopausia. La probabilidad de tener un falso positivo aumenta con la cantidad de mamografías realizadas.

Los **falsos negativos** ocurren cuando una mamografía no detecta un cáncer que realmente está presente. Se estima que las mamografías de detección no identifican alrededor del 20 % de los casos de cáncer de mama existentes en el momento del examen, por lo que pueden retrasar el inicio del tratamiento y generar una falsa sensación de seguridad. Uno de los factores principales que contribuyen a los falsos negativos es la alta densidad del tejido mamario. Las mamas están compuestas por tejido graso y tejido denso (como el tejido fibroglandular, formado por glándulas y tejido conectivo). En una mamografía, el tejido graso aparece oscuro, mientras que el tejido fibroglandular es blanco. Debido a que los tumores también se ven blancos, resulta más difícil identificarlos en mamas densas. Los falsos negativos son más frecuentes en mujeres jóvenes, ya que es más probable que tengan mamas densas. Con la edad, las mamas suelen contener más tejido graso, lo que reduce la probabilidad de falsos negativos en las mamografías. En algunos casos, el cáncer de mama puede avanzar rápidamente y desarrollarse pocos meses después de un resultado normal. En estas situaciones, no se considera que el resultado previo sea un falso negativo, ya que la evaluación inicial fue correcta. Sin embargo, esto puede generar una falsa sensación de seguridad.

Algunos cánceres que no se detectan en mamografías de detección podrían identificarse mediante exámenes clínicos de mama.

Se le conoce como **sobrediagnóstico** cuando las mamografías de detección detectan tumores

pequeños que, en algunos casos, no causan síntomas ni representan un riesgo para la vida de la persona, a diferencia de cuando se identifican casos de cáncer y de carcinoma ductal *in situ* (DCIS), que suelen ser agresivos. El problema es que tratar estos casos pequeños de cánceres, resulta innecesario, lo que se conoce como **sobretratamiento**.

La forma más efectiva de detectar el cáncer de mama es mediante exámenes clínicos realizados por profesionales de la salud y mamografías de alta calidad realizadas de forma regular.

El autoexamen de mama, es una práctica que las personas pueden realizar en casa. Sin embargo, no se recomienda como un método de detección confiable para el cáncer de mama, ya que los estudios clínicos no han demostrado que esta práctica reduzca las tasas de mortalidad asociadas con la enfermedad [14].

El Colegio Estadounidense de Radiología (ACR) desarrolló un sistema llamado *BI-RADS*, diseñado para estandarizar las descripciones utilizadas por los radiólogos al interpretar los resultados de una mamografía. Este sistema incluye siete categorías estándar, cada una de las cuales proporciona una guía clara para ayudar a los radiólogos y médicos a determinar el seguimiento más adecuado para cada caso. Se muestran en la tabla 2.1 [15].

Categoría	**Evaluación**	**Seguimiento**	**Probabilidad de cáncer**
0	Incompleto	Toma de imágenes adicionales y/o comparación con exámenes anteriores	N/A
1	Negativo	Mamografía de detección de rutina	0 % de probabilidad de malignidad
2	Benigno	Mamografía de detección de rutina	0 % de probabilidad de malignidad
3	Probablemente Benigno	Mamografía de seguimiento a corto plazo (6 meses) o de vigilancia continua	$> 0\,\%$ pero $\leq 2\,\%$ de probabilidad de malignidad
4	Sospechoso	Diagnóstico de tejidos	> 2 pero $< 95\,\%$ de probabilidad de malignidad
4A	Baja sospecha de malignidad		> 2 a $\leq 95\,\%$ de probabilidad de malignidad
4B	Sospecha moderada de malignidad		> 10 a $\leq 50\,\%$ de probabilidad de malignidad
4C	Alta sospecha de malignidad		> 50 a $< 95\,\%$ de probabilidad de malignidad
5	Altamente indicativo de malignidad	Diagnóstico de tejidos	$\geq 95\,\%$ de probabilidad de malignidad
6	Conocido y comprobado por biopsia	Escisión quirúrgica cuando sea clínicamente apropiado	N/A

Tabla 2.1: Categorías de evaluación de BI-RADS®.

A diferencia de las evaluaciones exigidas por la FDA, que no están directamente relacionadas con recomendaciones de tratamiento, las categorías de evaluación (BI-RADS®) han sido diseñadas para alinearse con recomendaciones terapéuticas específicas. Esta relación entre las categorías y las recomendaciones fortalece la práctica médica al proporcionar una guía más clara. Todas las evaluaciones finales (BI-RADS®) deben basarse en un análisis detallado de las características mamográficas relevantes o en la determinación de que el examen no muestra anomalías o presenta hallazgos benignos.

2.3. Sistema de Imágenes Mamográficas

Un profesional clínico que realiza diagnósticos a partir de imágenes médicas evalúa diversos tipos de evidencias. Estas pueden incluir alteraciones en la forma, como el aumento o disminución de una estructura específica, variaciones en la intensidad de la imagen en comparación con el tejido sano circundante y la presencia de características inusuales, como lesiones que normalmente no estarían presentes.

Un diagnóstico integral puede apoyarse en información proveniente de múltiples modalidades de imagen, las cuales pueden complementar o enriquecer el contenido informativo. Cada año, los avances en ingeniería contribuyen a mejoras significativas en la tecnología de las modalidades de imagen médica [16].

La radiografía planar es una técnica ampliamente utilizada para evaluar diversas condiciones, como el grado de fractura ósea en lesiones agudas, la presencia de masas en casos de cáncer de pulmón o enfisema, y las microcalcificaciones en tejido mamario. Los huesos y pequeñas calcificaciones tienen una mayor capacidad de absorción de rayos X en comparación con los tejidos blandos. En este procedimiento, una fuente de rayos X emite radiación hacia el paciente (figura 2.3a), y los rayos que logran atravesar el cuerpo son captados por un detector plano de estado sólido ubicado debajo del paciente. La energía detectada se transforma primero en luz, luego en voltaje y, finalmente, se digitaliza para generar una imagen digital, la cual es una proyección bidimensional de los tejidos situados entre la fuente y el detector.

Al atravesar el cuerpo, parte de los rayos X no solo son absorbidos, sino que también pueden dispersarse, generando un ruido de fondo que disminuye el contraste de la imagen. Para mitigar este efecto, se emplea una rejilla anti dispersión (figura 2.3a), que permite registrar únicamente los rayos X que pasan directamente desde la fuente hasta el detector.

En la figura 2.3b, se muestra un ejemplo de radiografía plana bidimensional, donde se aprecia un marcado contraste entre los huesos (que aparecen en blanco al absorber más rayos X) y el tejido pulmonar (que se ve oscuro debido a su baja absorción de rayos X).

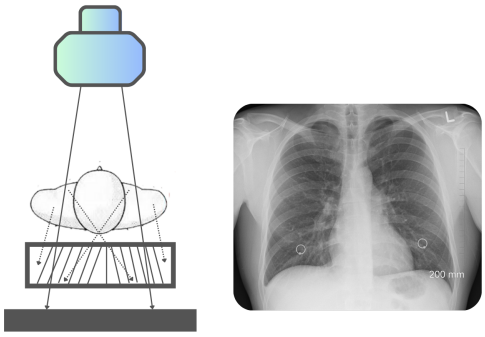

Figura 2.3: Rayos X en el cuerpo humano. a) Imagen radiográfica bidimensional. Se utiliza una rejilla antidifusión colocada justo delante del detector con el fin de minimizar la interferencia de los rayos X dispersados y mejorar la calidad del contraste en la imagen. b) Ejemplo de una radiografía bidimensional de la región torácica. Las estructuras óseas absorben los rayos X de manera más significativa en comparación con los tejidos blandos de los pulmones, lo que hace que aparezcan con mayor brillo en la imagen obtenida.

En 1966, la mamografía con rayos X se consolidó como una técnica clínicamente válida tras la adopción del molibdeno como material para el blanco y el filtro, junto con la implementación

de receptores de imagen de pantalla-película de emulsión única en 1972. Para la década de 1990, avances como la técnica de la rejilla, la mayor atención en la compresión, los generadores de alta frecuencia y el control automático de exposición (AEC, por sus siglas en inglés) elevaron la mamografía a un nivel de excelencia en la obtención de imágenes mamarias [11].

Los equipos tradicionales de rayos X no son adecuados para la mamografía, ya que esta requiere dispositivos específicamente desarrollados. Por lo que, la mayoría de los fabricantes de equipos de rayos X elaboran sistemas especializados para este fin.

Estos sistemas mamográficos están diseñados para proporcionar una alta adaptabilidad en el posicionamiento de la paciente, incluyen un mecanismo de compresión integrado, una rejilla de baja relación, control automático de exposición y un tubo de rayos X con un foco extremadamente pequeño.

Al igual que en la radiografía convencional, la mamografía con pantalla-película captura las imágenes del tejido mamario en una película. No obstante, se distingue de las radiografías tradicionales por el uso de compresión sobre el tejido mamario. Dado que este tejido está compuesto por capas densas, la compresión mejora tanto la resolución espacial como la de contraste, lo que permite obtener imágenes más claras y, al mismo tiempo, reduce la dosis de radiación recibida por la paciente. Una vez que se ha obtenido una exposición adecuada, las imágenes se procesan mediante un casete que contiene una pantalla y una película. Una capa de fósforo dentro del casete se ilumina al ser expuesta a los rayos X, lo que revela la película de manera similar a como ocurre en la fotografía tradicional.

La mamografía con película tiene una sensibilidad limitada para la detección del cáncer de mama en mujeres con mamas radiográficamente densas. Por lo que existe otra técnica para evaluar este tipo de casos.

La mamografía digital utiliza rayos X con niveles de energía considerablemente menores en comparación con las radiografías tradicionales, lo que permite obtener imágenes con una resolución significativamente superior. La información obtenida mediante una mamografía digital se transforma en señales eléctricas que pueden ser interpretadas y procesadas por una computadora. Este proceso se realiza en cuestión de segundos.

Complementario a esto, existen las proyecciones o vistas: craneocaudal (CC) y oblicua mediolateral (MLO), que se utilizan en mamografía porque ofrecen una visualización complementaria del tejido mamario, permitiendo una mejor detección y caracterización de anomalías.

En la proyección Craneocaudal (CC) la mama se comprime horizontalmente desde arriba (craneal) hacia abajo (caudal). Muestra el tejido desde la parte superior hasta la inferior, brindando una vista clara de la parte central y medial (interna). Permite identificar si una lesión está más cerca de la pared torácica o del pezón aunque puede ser difícil visualizar el tejido cercano a la axila.

En la proyección oblicua Mediolateral (MLO) la mama se comprime en un ángulo de aproximadamente 45°, desde el centro del cuerpo (medial) hacia el lateral. Muestra mejor el tejido axilar y la parte superior de la mama, donde suelen aparecer adenopatías y lesiones ocultas en la proyección CC. Permite evaluar la extensión de las lesiones en sentido vertical y su relación con los ganglios linfáticos aunque puede haber más superposición de tejido si la colocación no es óptima.

Una lesión puede ser visible en una proyección pero no en la otra. Al comparar ambas vistas, los radiólogos pueden determinar si la anomalía está más cerca de la pared torácica o del pezón, y si se encuentra en el cuadrante superior o inferior de la mama. Al reducir la superposición de tejidos, aumenta la sensibilidad para identificar microcalcificaciones y masas pequeñas.

La tomosíntesis mamaria, también conocida como mamografía 3D, es una técnica avanzada de imagenología que mejora la detección temprana del cáncer de mama en comparación con la mamografía convencional (2D), que está limitada por la superposición del parénquima mamario y la densidad mamaria.

En la tomosíntesis digital de mama, se generan múltiples imágenes de proyección que permiten examinar visualmente secciones finas del tejido mamario, con cortes que pueden alcanzar un grosor de hasta 0.5 mm. Esto facilita una mejor identificación y análisis de lesiones que no presentan calcificaciones.

Para obtener estas imágenes, se utiliza un detector digital junto con una fuente de rayos X que se desplaza en un ángulo arco limitado mientras se mantiene la mama comprimida. Posteriormente, los datos recopilados se procesan mediante algoritmos especializados, produciendo una secuencia de imágenes que abarcan toda la mama para su evaluación clínica [17].

En 2011, la FDA (Administración de Alimentos y Medicamentos de Estados Unidos) autorizó el uso de la tomosíntesis junto con la mamografía digital convencional para la detección del cáncer de mama. Aunque la combinación de ambas técnicas implica una exposición a la radiación cercana al doble de la que se recibe en una mamografía digital individual, esta cantidad sigue estando dentro de los márgenes de seguridad establecidos por la FDA [18].

El surgimiento de la mamografía digital y los avances en los algoritmos de reconstrucción computarizada han impulsado el desarrollo de nuevas tecnologías, como la tomosíntesis.

En la mamografía digital convencional, se aplica radiación ionizante a una mama comprimida. La energía que atraviesa el tejido mamario se convierte en una señal eléctrica mediante un detector, generando así la imagen médica. Durante este proceso, el tubo de rayos X, la mama y el detector permanecen fijos. La imagen resultante, ya sea en una vista CC o MLO, es una representación bidimensional de una estructura tridimensional. Esto implica que cada píxel refleja un promedio de la información recogida a lo largo de todo el grosor de la mama. Una representación tridimensional del tejido mamario sería más beneficiosa, similar a las imágenes tridimensionales que ofrecen técnicas como la tomografía computarizada (TC), la resonancia magnética (RM) o la ecografía.

2.3.1. Rayos X

Los rayos X se generan cuando los electrones que se desplazan a alta velocidad impactan contra un material metálico. Durante este proceso, la energía cinética de los electrones se convierte en energía electromagnética.

El objetivo de un sistema de imágenes por rayos X es regular el flujo de electrones con la intensidad necesaria para crear un haz de rayos X adecuado para la obtención de imágenes. Los elementos fundamentales de este sistema incluyen el tubo de rayos X, la consola de control y el generador de alto voltaje.

El dispositivo que genera rayos X es un equipo especializado denominado tubo de rayos X. Todos los elementos que componen este sistema se encuentran encapsulados en un recipiente al vacío, el cual está sumergido en aceite para garantizar su refrigeración y proporcionar aislamiento eléctrico.

El conjunto completo está envuelto en una cubierta de plomo que incluye una ventana de cristal, por donde se libera el haz de rayos X. Estos rayos se generan cuando un flujo de electrones de alta energía impacta contra un objetivo metálico. Un cátodo, cargado negativamente, funciona como la fuente de estos electrones y está compuesto por un filamento delgado de tungsteno en forma de espiral, por el cual circula una corriente eléctrica, se puede observar en la figura 2.4. Cuando el filamento alcanza una temperatura aproximada de 2200 °C, los electrones adquieren la energía suficiente para liberarse de la superficie del metal. Para concentrar el haz de electrones, se utiliza una copa de enfoque con carga negativa que rodea al cátodo.

Por otro lado, un ánodo, que es un blanco metálico, se carga positivamente con un alto voltaje. Se establece una diferencia de potencial entre el ánodo y el cátodo, que varía entre 25 y 140 kV (según el tipo de estudio clínico), lo que hace que los electrones emitidos por el cátodo sean atraídos hacia el ánodo a gran velocidad, impactando con fuerza. Esta diferencia de voltaje se conoce como voltaje de aceleración o kVp.

Cuando los electrones de alta energía impactan contra la superficie del ánodo, una porción de su energía cinética se transforma en rayos X principalmente por dos mecanismos, la radiación de frenado (*Bremsstrahlung*) o la radiación característica.

El material del ánodo debe ser eficiente en la generación de rayos X, y al mismo tiempo, resistir las temperaturas extremadamente altas que se producen. La eficiencia aumenta con el número atómico del metal utilizado en el ánodo, ya que los elementos con mayor número atómico generan rayos X con mayor efectividad.

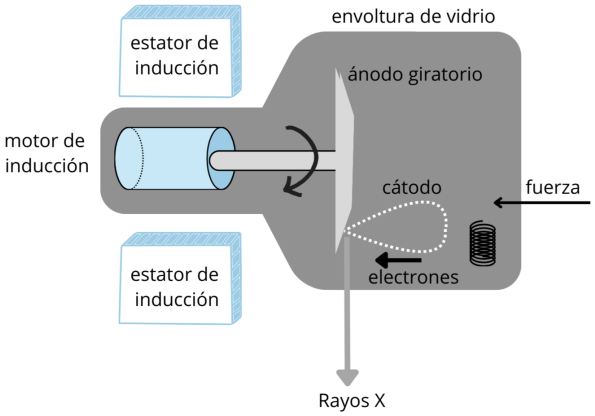

Figura 2.4: Componentes individuales de un tubo de rayos X.

El tungsteno es el metal más empleado debido a su alto número atómico (74) y su elevado punto de fusión (3,422 °C), como se muestra en la tabla 2.2. Además, posee una excelente conductividad térmica y una baja presión de vapor, lo que facilita mantener un vacío óptimo dentro del tubo de rayos X, permitiendo que los electrones se desplacen sin obstáculos entre el cátodo y el ánodo. A pesar de la eficiencia del tungsteno, solo alrededor del 1 % de la energía de los electrones se convierte en rayos X, mientras que el resto se disipa en forma de calor.

Elemento y Símbolo Químico	**Número Atómico**	**Energía K de los rayos X (keV)**	**Temperatura de Fusión (°C)**
Tungsteno (W)	74	69.5	3422
Molibdeno (Mo)	42	17.5	2623
Rodio (Rh)	45	23.2	1964

Tabla 2.2: Características de los blancos de rayos X

En la aplicación clínica, se emplea una aleación de tungsteno y renio (con un contenido de renio del 2 % al 10 %) para mejorar la estabilidad mecánica del ánodo.

En el caso de la mamografía digital, que requiere rayos X de muy baja energía, el metal utilizado en el ánodo es el molibdeno en lugar del tungsteno.

La intensidad del haz de rayos X es mayor en la zona cercana al cátodo que en la cercana al ánodo, un fenómeno denominado efecto talón, que se puede observar en la práctica. Esto ocurre debido a las variaciones en la distancia que los rayos X deben atravesar dentro del material del ánodo, lo que afecta su intensidad, aunque se pueden utilizar algoritmos de procesamiento de imágenes para corregir este fenómeno, pero no es significativamente perjudicial para la calidad diagnóstica de las imágenes.

Los tubos de rayos X generan rayos con un amplio espectro de energías, alcanzando un valor máximo determinado por el kVp aplicado. Existen dos mecanismos principales que explican la

producción de estos rayos X: uno que genera un rango amplio y continuo de energías, y otro que produce líneas espectrales definidas y específicas.

El primer mecanismo ocurre cuando un electrón se acerca al núcleo de un átomo en el ánodo y es desviado de su trayectoria original debido a la atracción electrostática del núcleo, que está cargado positivamente. Esta desviación provoca que el electrón pierda parte de su energía cinética, la cual se transforma en un fotón de rayos X. La energía máxima que puede tener este fotón corresponde a la energía cinética total del electrón, es decir, el valor del kVp aplicado.

Sin embargo, debido al tamaño reducido del núcleo en comparación con el átomo completo, es más probable que el electrón pierda solo una fracción de su energía, lo que da lugar a un espectro continuo de energías de rayos X, conocido como radiación de frenado o *bremsstrahlung*. Aunque este espectro disminuye de manera aproximadamente lineal con la energía, muchos de los rayos X de muy baja energía son absorbidos por la estructura del propio tubo de rayos X.

Además del espectro continuo, se observan picos agudos en la distribución de energía de los rayos X. Estos picos corresponden a energías específicas que son características del material utilizado en el ánodo, de ahí el nombre de *radiación característica*. Este fenómeno ocurre cuando un electrón acelerado desde el cátodo colisiona con un electrón fuertemente ligado en la capa K del ánodo, expulsándolo. El vacío resultante en la capa K es ocupado por un electrón de una capa externa (L o M), y la diferencia de energía entre estas capas se libera en forma de un fotón de rayos X con una energía bien definida.

En resumen, la producción de rayos X combina un espectro continuo (*bremsstrahlung*) y líneas espectrales características, dependiendo de las interacciones entre los electrones acelerados y los átomos del ánodo, como se observa en la figura 2.5.

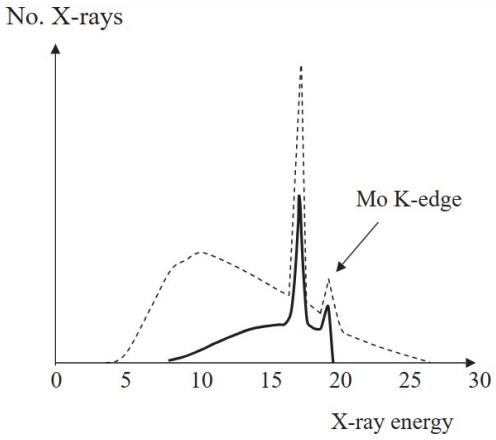

Figura 2.5: El espectro de rayos X utilizado para la mamografía digital. El kVp del tubo de rayos X es de 26 keV. El ánodo está hecho de molibdeno y produce una cantidad sustancial de rayos X de baja energía (línea discontinua). Se utiliza un filtro de molibdeno de 30 μm de espesor para reducir la contribución de los rayos X de muy baja energía (línea continua). *Fuente: N. Smith and A. Webb. (2011). Introduction to Medical Imaging: Physics, Engineering and Clinical Applications(p. 63). Cambridge University Press. Derechos de autor reservados. Reproducido bajo el principio de uso justo para fines académicos.*

En la radiografía convencional, el contraste entre tejidos como huesos, músculos, tejido adiposo y pulmones es notable por sus diferencias en densidad y número atómico. Sin embargo, en la radiografía de tejidos blandos, el desafío es capturar imágenes de músculos y tejido adiposo, que tienen densidades y números atómicos muy similares. Por ello, las técnicas empleadas buscan maximizar la absorción diferencial entre estos tejidos, que presentan propiedades físicas casi idénticas.

2.3.2. Interacción de los Rayos X con la Materia

Cuando un haz de rayos X atraviesa un material, puede ocurrir una interacción entre los fotones y la materia, lo que resulta en la transferencia de energía al medio. El primer paso en este proceso implica la expulsión de electrones de los átomos del material que absorbe la radiación. Estos electrones, que se mueven a alta velocidad, transfieren su energía al causar ionización y excitación en los átomos que encuentran en su camino. Si el material absorbente es tejido biológico, es posible que se deposite suficiente energía dentro de las células como para afectar su capacidad de reproducción. No obstante, la mayor parte de la energía absorbida se transforma en calor y no genera efectos biológicos significativos.

Para lograr imágenes con una relación señal-ruido (SNR) y una relación contraste-ruido (CNR) óptimas, es necesario cumplir tres condiciones fundamentales: (1) se debe permitir que una cantidad suficiente de rayos X atraviese el cuerpo para garantizar una SNR alta, (2) la absorción de los rayos X debe variar significativamente entre los diferentes tejidos para generar un contraste adecuado, y (3) es indispensable contar con un mecanismo que elimine los rayos X que se dispersan en direcciones impredecibles al pasar a través del cuerpo [16].

En el ámbito de las energías utilizadas en diagnóstico por imágenes de rayos X, existen dos procesos principales mediante los cuales los rayos X interactúan con los tejidos. El primero es la interacción fotoeléctrica, que produce una atenuación diferencial, especialmente notable entre el hueso y el tejido blando. El segundo es el efecto *Compton*, donde los rayos X se desvían de su trayectoria original.

Aunque su energía disminuye, los rayos X dispersados aún pueden llegar al detector. Sin embargo, estos rayos X dispersados generan un ruido de fondo aleatorio, por lo que es crucial minimizar su impacto para mejorar la CNR.

Además de estos mecanismos, existen otras interacciones, como la dispersión coherente, pero en el rango de energías utilizadas en radiografía clínica, su contribución es mínima y no se toma en cuenta en este contexto.

Efecto fotoeléctrico

Cuando un fotón es absorbido por un átomo y como resultado uno de sus electrones orbitales es expulsado, se le denomina efecto fotoeléctrico. En este proceso, toda la energía ($h\nu$) del fotón es absorbida primero por el átomo y después, esencialmente toda, es transferida al electrón atómico. La energía cinética del electrón expulsado (llamado fotoelectrón) es igual a $h\nu - E_B$, donde E_B es la energía de enlace del electrón. Las interacciones de este tipo pueden tener lugar con electrones de las capas K, L, M o N [19].

Cuando un electrón es expulsado de un átomo, se genera un espacio vacante en su capa electrónica, dejando al átomo en un estado de excitación. Este vacío puede ser ocupado por un electrón de una capa externa, lo que resulta en la emisión de un rayo X característico. Alternativamente, se puede dar la emisión de electrones *Auger*, un proceso en el que la energía liberada al llenarse el espacio vacante se transfiere a otro electrón de una capa superior, provocando su expulsión. En los tejidos blandos, la energía de enlace de la capa K es de aproximadamente 0.5 keV, lo que significa que los fotones característicos generados en estos tejidos tienen una energía muy baja. Como consecuencia, estos fotones suelen absorberse localmente, impidiendo que lleguen al detector. Esto implica que el rayo X incidente es absorbido por completo en el tejido.

Nótese que las configuraciones electrónicas de los elementos más importantes en el tejido, en términos de interacciones fotoeléctricas, son: carbono (K2, L4), oxígeno (K2, L6) y calcio (K2, L8, M8, N2).

Para los átomos con número atómico reducido, como los que se encuentran en los tejidos blandos, la energía de unión de los electrones de la capa K es baja (aprox. 0.3 keV para el carbono). Por

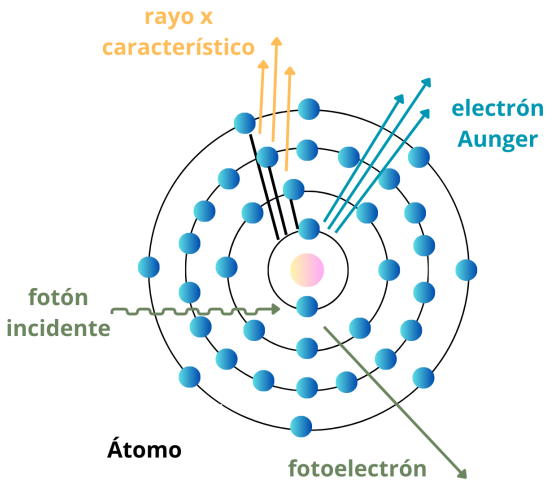

Figura 2.6: Efecto fotoeléctrico.

este motivo, el fotoelectrón es liberado con una energía cinética casi igual a la energía del rayo X incidente.

La probabilidad de absorción fotoeléctrica (P) depende de la energía (E) del fotón incidente, del número atómico efectivo (Z_{eff}) del tejido y de la densidad del tejido (ρ):

$$P \propto \rho \frac{Z_{\text{eff}}^3}{E^3} \tag{2.1}$$

El número atómico efectivo del tejido blando es ∼7.5, el de la mama ∼6.8 y el del hueso ∼13.8 (el alto valor se debe principalmente a la presencia de calcio, tabla 2.3). Las densidades relativas son 1: 0.9: 1.85.

La ecuación (2.1) indica que, a bajas energías de rayos X, el efecto fotoeléctrico produce un alto contraste entre el hueso (alta atenuación) y el tejido blando (baja atenuación), pero que el contraste disminuye al aumentar la energía de los rayos X.

Tipo de material	**Número atómico efectivo**
Tejidos humanos	
Grasa	5.9 - 6.5
Mama	6.5 - 7.0
Tejidos blandos	7.4 - 7.6
Hueso	11.6 - 13.8
Otros	
Molibdeno	42
Bario	56
Tungsteno	74
Plomo	82

Tabla 2.3: Números atómicos efectivos de materiales importantes en ciencia radiológica.

La probabilidad de interacción que varía en proporción a la tercera potencia experimenta cambios rápidos. En el contexto del efecto fotoeléctrico, esto implica que incluso una ligera modificación en el número atómico del tejido o en la energía de los rayos X genera un cambio significativo en la probabilidad de que ocurra una interacción fotoeléctrica. Este comportamiento contrasta con el caso de la interacción *Compton*, donde la situación es distinta.

Efecto Compton

Los rayos X dentro del rango diagnóstico pueden interactuar con los electrones de las capas más externas de los átomos. Esta interacción no solo provoca la dispersión del rayo X, sino que también reduce su energía y, al mismo tiempo, ioniza el átomo involucrado.

En el proceso *Compton*, una pequeña fracción de la energía del rayo X incidente se transfiere a este electrón débilmente ligado. Con la energía adicional, el electrón es expulsado y el rayo X se desvía de su trayectoria original en un ángulo θ, como se observa en la figura 2.7.

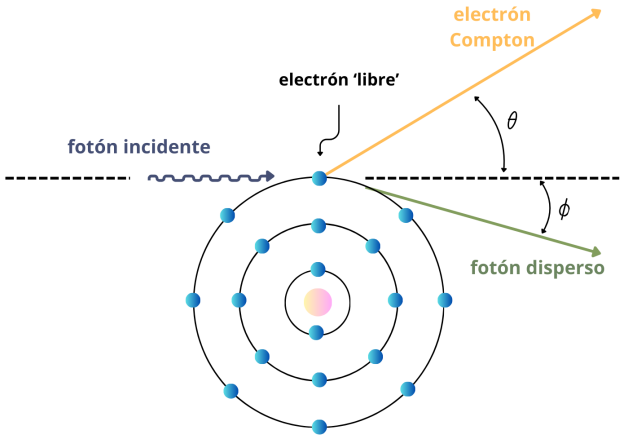

Figura 2.7: Efecto compton.

El electrón expulsado se denomina electrón *Compton*. El rayo X continúa en una dirección diferente y con una energía menor. La energía de los rayos X dispersados se puede calcular aplicando las leyes de conservación del momento y la energía.

La probabilidad de que un rayo X sufra dispersión *Compton* es independiente del número atómico, proporcional a la densidad electrónica del tejido y sólo depende débilmente de la energía del rayo X incidente, cualquier rayo X es probable que desarrolle el efecto *Compton* así con un átomo de tejido blando, como con un átomo de hueso, véase en la figura 2.8. Dado que el Z_{eff} del hueso es aproximadamente el doble que el del tejido, la atenuación es mucho mayor en el hueso que en el tejido y el contraste tisular es excelente. Sin embargo, a energías de rayos X más altas, la contribución de la dispersión *Compton* se vuelve más importante y, por lo tanto, el contraste disminuye.

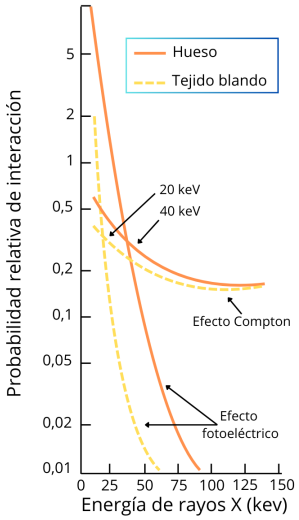

Figura 2.8: Probabilidades de las interacciones fotoeléctricas y de *Compton* en tejidos blandos y huesos. Las interacciones de estas curvas indican las energías de los rayos X a las que el cambio de la absorción fotoeléctrica iguala la probabilidad de la dispersión *Compton*.

2.4. Segmentación de Imágenes Médicas

La segmentación de imágenes es el proceso de dividir una imagen en regiones homogéneas y/o heterogéneas que representan estructuras relevantes y siendo esenciales para extraer información significativa de las imágenes[6]. Las distintas técnicas de segmentación se pueden clasificar ampliamente en métodos basados en Umbralización Global, en Regiones o Agrupamiento, y enfoques de aprendizaje automático como: Redes Convolucionales Clásicas (CNNs), modelos basados en *Vision Transformers* (ViTs) o modelos específicos para segmentación médica.

2.4.1. Imagen Digital

Una imagen puede definirse como una función bidimensional, $f(x, y)$, donde x e y representan coordenadas espaciales en un plano, y la amplitud de f en cualquier par de coordenadas (x, y) se conoce como la intensidad o nivel de gris de la imagen en ese punto. En el caso de una imagen, el nivel de color se puede descomponer en tres canales, cada uno representando un componente de color rojo, verde y azul o *Red, Blue* y *Green*. Cuando x, y y los valores de intensidad de f son cantidades finitas y discretas, se denomina *imagen digital* que está compuesta de elementos individuales denominados píxeles [20].

La mayoría de las imágenes generadas por rayos X son monocromáticas, es decir, se derivan de un solo color base que es extendido mediante el uso de tonalidades claras y oscuras del mismo color. Comúnmente las radiografías se percibe como una degradación que va desde el negro, pasando por los tonos grises, hasta el blanco. La gama de valores que abarca este intervalo se conoce como escala de grises. En esta escala, cada píxel tiene un valor que representa la intensidad del nivel de gris; por ejemplo, 0 corresponde al negro y 255 al blanco [20].

En la radiografía digital, las imágenes se generan mediante dos métodos: digitalizando películas de rayos X o utilizando dispositivos, como una pantalla de fósforo, que convierten los rayos X en luz. Ésta es captada por un sistema de digitalización sensible a la luz, que la transforma en una imagen digital [20].

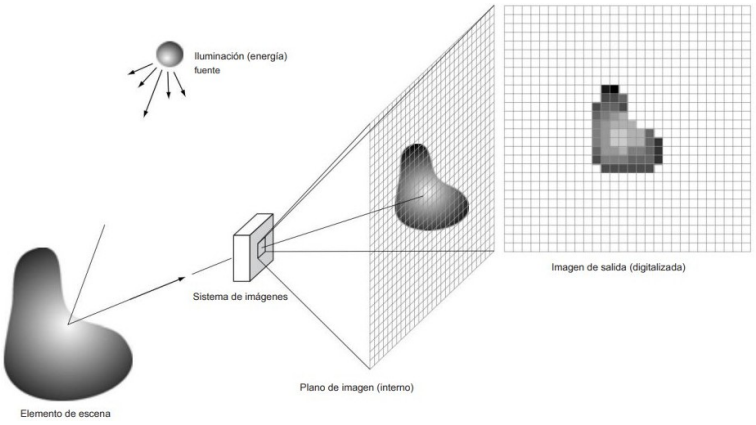

Figura 2.9: Ejemplo del proceso de adquisición de imágenes digitales.
[21]

2.4.2. Aprendizaje automático

El término Inteligencia Artificial engloba a los campos de aprendizaje de máquinas (*Machine Learning*) y aprendizaje profundo (*Deep Learning*).

El **aprendizaje automático** es un conjunto de técnicas informáticas que utilizan datos (como palabras, imágenes, audio, entre otros) para dotar a las computadoras de la capacidad de aprender sin necesidad de ser programadas de manera explícita.

El término *aprendizaje* se utiliza porque el proceso se asemeja a un entrenamiento en el que los datos se emplean para resolver el problema de encontrar un modelo. Por esta razón, los datos que se utilizan durante este proceso se denominan datos de entrenamiento. A partir de estos datos, se construye un modelo, que es el resultado final del aprendizaje automático. En esencia, el modelo es la representación aprendida que permite a la computadora realizar predicciones o tomar decisiones basadas en la información proporcionada [21].

Una vez que el proceso de aprendizaje automático encuentra el modelo a partir de los datos de entrenamiento, se aplica a los datos de campo reales.

En la figura 2.10 se muestra el proceso de aprendizaje. El flujo vertical indica el proceso de aprendizaje, y el modelo entrenado se describe como el flujo horizontal, que se denomina inferencia.

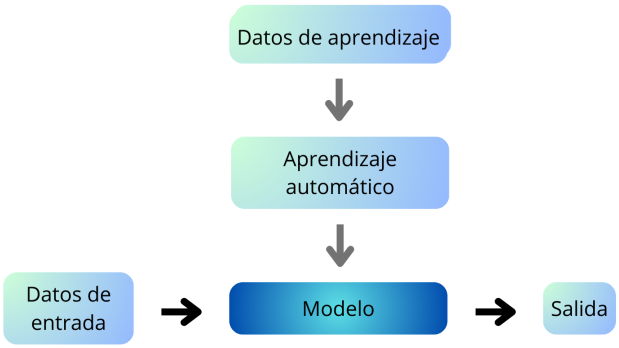

Figura 2.10: Aplicación de un modelo basado en datos.
[21]

Los datos utilizados para el modelado en aprendizaje automático y los datos suministrados en la aplicación son distintos, esta distinción entre los datos es el desafío estructural que enfrenta el

aprendizaje automático.

En términos generales, ningún enfoque de *Machine Learning* puede alcanzar los resultados deseados si los datos de entrenamiento no son adecuados. Por lo que los modelos de aprendizaje automático deben basarse en datos de entrenamiento que sean imparciales y que representen de manera fiel las características del problema que se busca resolver.

Un aspecto clave en este proceso es la generalización, que se refiere a la capacidad del modelo para mantener un rendimiento consistente, independientemente de los datos de entrenamiento o de entrada que se le proporcionen. La generalización es fundamental porque garantiza que el modelo no solo funcione bien con los datos utilizados durante el entrenamiento, sino que también sea capaz de adaptarse a nuevos datos no vistos previamente.

Una de las principales causas que afectan negativamente el proceso de generalización es el sobreajuste. Esto ocurre porque los algoritmos de aprendizaje automático tienden a considerar todos los datos, incluyendo el ruido, lo que puede llevar a la creación de un modelo incorrecto [21].

Una forma de contrarrestar el sobreajuste es la validación, que es un proceso que reserva una parte de los datos de entrenamiento y los utiliza para controlar el rendimiento.

Se han desarrollado diversos tipos de técnicas de aprendizaje automático para resolver problemas en diversos campos, entre los que se encuentran el supervisado, no supervisado, semi supervisado y por refuerzo [22].

En el aprendizaje supervisado, el conjunto de datos utilizado ha sido previamente etiquetado y clasificado por usuarios. Esto permite que el algoritmo evalúe su precisión y mejore su rendimiento. Por otro lado, en el aprendizaje no supervisado, el conjunto de datos no está etiquetado, y el algoritmo debe identificar patrones y relaciones dentro de los datos sin ninguna guía externa. En el aprendizaje semi supervisado, el conjunto de datos incluye tanto información estructurada (etiquetada) como no estructurada (sin etiquetar). Esta combinación guía al algoritmo para que pueda sacar conclusiones por sí mismo, aprendiendo a etiquetar datos no etiquetados a partir de los datos estructurados. Finalmente, en el aprendizaje por refuerzo, el conjunto de datos utiliza un sistema de recompensas y penalizaciones. Este enfoque proporciona retroalimentación al algoritmo, permitiéndole aprender de sus propias experiencias a través de prueba y error.

2.4.3. Redes Neuronales Artificiales

Los modelos de redes neuronales se inspiraron originalmente en estudios sobre el procesamiento de la información en el cerebro de los seres humanos y otros mamíferos. Las unidades fundamentales de este procesamiento son células llamadas neuronas, que son capaces de recibir información en forma de señales eléctricas y químicas, para luego transformarla y retransmitirla a otras células nerviosas [23].

Cuando una neurona se activa eléctricamente, envía un impulso eléctrico por el axón hasta llegar a unas uniones, llamadas sinapsis, que forman conexiones con otras neuronas. En las sinapsis se liberan señales químicas denominadas neurotransmisores, que pueden estimular o inhibir la activación de las neuronas siguientes (figura 2.11).

El grado en que una neurona puede provocar la activación de otra depende de la fuerza de la sinapsis, y son los cambios en estas fuerzas los que representan un mecanismo clave por el que el cerebro puede almacenar información y aprender de la experiencia [23].

La neurona en sí no tiene capacidad de almacenamiento; su función es transmitir señales entre neuronas. El cerebro, por su parte, es una red gigantesca de estas células, y es la interconexión entre ellas lo que genera información específica. Así como el cerebro se compone de innumerables conexiones neuronales, una red artificial se construye mediante nodos interconectados, que equivalen a las neuronas biológicas. De esta forma, la red neuronal reproduce el principio más importante del cerebro: la asociación de neuronas a través de valores de peso.

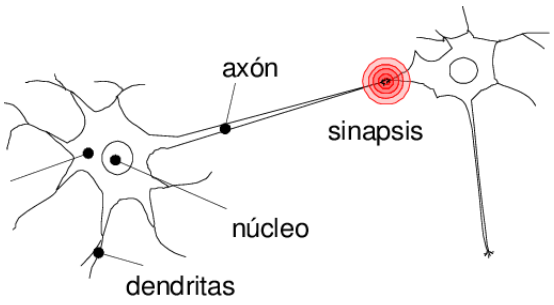

Figura 2.11: Conexión entre dos neuronas del cerebro humano.
[23]

Estas propiedades de las neuronas se han plasmado en modelos matemáticos, conocidos como redes neuronales artificiales, que luego constituyen la base de los enfoques computacionales del aprendizaje [23].

Una forma simple de una neurona artificial o nodo se puede expresar matemáticamente como se muestra en la ecuación 2.2 y 2.3:

$$a = \sum_{i=0}^{n} w_i x_i + b \tag{2.2}$$

$$y = f(a) \tag{2.3}$$

Los datos de entrada son $x_0, x_1, x_2, ..., x_n$, los cuales también pueden ser producto de la salida de otras neuronas que envían conexiones a esta neurona. Los pesos $w_0, w_1, w_2, ..., w_n$, son valores numéricos que representan la importancia de cada entrada, no son fijos ya que se ajustan durante el proceso de entrenamiento de la red. La unidad de sesgo (b), es un valor numérico adicional que se suma a la suma ponderada de las entradas de una neurona o nodo, antes de que se le aplique la función de activación, proporciona una flexibilidad adicional a la neurona, permitiéndole generar una salida no nula incluso cuando las entradas son bajas, o ajustar su umbral de activación independientemente de las entradas. La cantidad (a) se denomina preactivación, la función no lineal $f(\cdot)$ se denomina función de activación, esta le permiten a la red aprender y modelar relaciones complejas en los datos, en redes de una sola capa usualmente se asume que la función de activación de cada nodo es lineal, y la salida (y) se denomina activación.

El funcionamiento general es: la señal de entrada, que proviene del exterior, se multiplica por un peso antes de llegar al nodo. Una vez que las señales ponderadas llegan al nodo, se suman para obtener la suma ponderada. Este proceso se muestra en la ecuación 2.4:

$$a = (w_1 x_1) + (w_2 x_2) + (w_3 x_3) + b \tag{2.4}$$

El nodo introduce la suma ponderada en la función de activación y genera su salida. Si el valor que produce un nodo supera su valor límite, este se activa y envía información a la siguiente capa. Si no lo supera, no pasa información, por lo que la función de activación determina el comportamiento del nodo.

$$y = f(a) = f(w_n x_n + b) \tag{2.5}$$

Existen muchos tipos de funciones de activación en una red neuronal, deben ser derivables para permitir el cálculo del gradiente en el entrenamiento con *backpropagation*[23][21], esto se detallara más adelante .

La suma ponderada de la ecuación 2.2 se puede escribir con matrices, donde w_n y x_n se definen como:

$$w = \begin{bmatrix} w_0 & w_1 & w_2 & ... & w_n \end{bmatrix} \quad x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ ... \\ x_n \end{bmatrix}$$

La formulación matemática dada por las ecuaciones (2.2) y (2.5) ha constituido la base de los modelos de redes neuronales desde los años 60 hasta la actualidad, y puede representarse en forma de diagrama como se muestra en la figura 2.12.

Uno de los modelos más importantes en la historia de la computación neuronal es el *perceptrón (Rosenblatt, 1962)*, en el que la función de activación $f(\cdot)$ es una función escalonada como se muestra en la ecuación 2.6. Puede considerarse como un modelo simplificado de activación neuronal en el que una neurona se activa si, y sólo si, la entrada ponderada total supera un umbral de 0.

El perceptrón fue creado por *Rosenblatt* en 1962, desarrolló un algoritmo de entrenamiento específico que tiene la propiedad de que si existe un conjunto de valores de peso para los que el perceptrón puede lograr una clasificación perfecta de sus datos de entrenamiento, entonces se garantiza que el algoritmo encuentre la solución en un número finito de pasos [23].

$$f(a) = \begin{cases} 1, & \text{si } a \leq 0 \\ 0, & \text{si } a > 0 \end{cases} \tag{2.6}$$

Una configuración típica de perceptrón tenía varias capas de procesamiento, pero sólo una de ellas podía aprender de los datos, por lo que el perceptrón se considera una red neuronal *monocapa*. Las propiedades de los perceptrones fueron analizadas por Minsky y Papert en 1969, aportaron pruebas formales de las capacidades limitadas de las redes monocapa. Aunque hace tiempo que los perceptrones desaparecieron de la práctica del aprendizaje automático, su nombre sigue vivo, ya que existe una red neuronal denomina a veces como perceptrón multicapa o MLP.

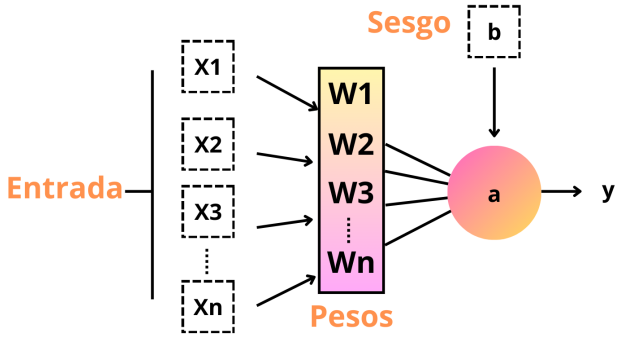

Figura 2.12: Diagrama del funcionamiento simple de neuronas artificiales, con un nodo que recibe n entradas.

Como se mención al inicio de está supsección, el cerebro es una gigantesca red de neuronas, y la asociación de las neuronas forma una información específica, la red neuronal es una red de nodos. Se pueden crear diversas redes neuronales según cómo estén conectados los nodos. Uno de los tipos de redes neuronales más utilizados como las *Redes Neuronales Convolucionales* emplea una estructura de nodos en capas, como se muestra en la figura 2.13.

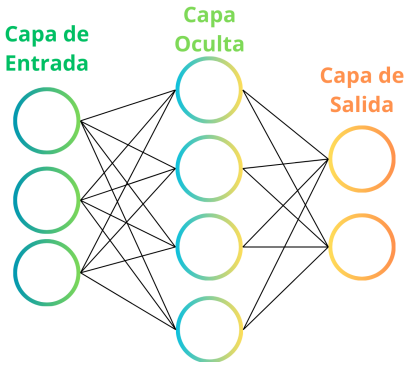

Figura 2.13: Diagrama general de una Red Neuronal Artificial en capas de nodos.

Al añadir capas ocultas a una red neuronal de una sola capa, se produce una red neuronal multicapa. La red neuronal multicapa consta de una capa de entrada, una o varias capas ocultas y una capa de salida. Una red neuronal multicapa que contiene dos o más capas ocultas se denomina red neuronal profunda. En la red neuronal en capas, la señal entra en la capa de entrada, atraviesa las capas ocultas y sale por la capa de salida. Durante este proceso, la señal avanza capa por capa. En otras palabras, los nodos de una capa reciben la señal simultáneamente y envían la señal procesada a la siguiente capa simultáneamente [21].

Se puede observar cómo se procesan los datos de entrada a medida que pasan por las capas en la figura 2.14.

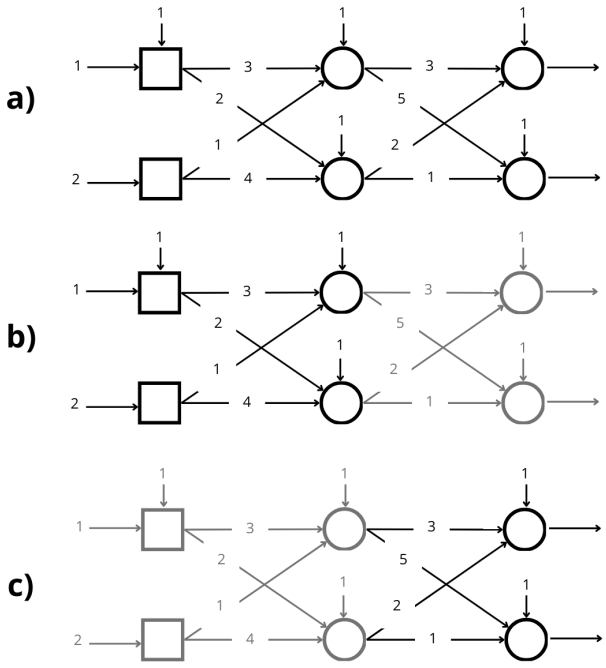

Figura 2.14: a) Red neuronal con una sola capa oculta. b) Cálculo de la salida de la capa oculta. c) Determinación de las salidas de la capa de salida.
[23]

Considerando la red neuronal con una sola capa oculta y asumiendo que la función de activación de cada nodo es lineal , podemos calcular la salida de la capa oculta de la figura 2.14.

Los cálculos de la suma ponderada se pueden expresar en una ecuación matricial de la siguiente manera:

$$a = \begin{bmatrix} 3 & 1 \\ 2 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 2 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 6 \\ 11 \end{bmatrix} \tag{2.7}$$

Los pesos del primer nodo de la capa oculta se encuentran en la primera fila, y los del segundo nodo, en la segunda. Este resultado se puede generalizar mediante la ecuación 2.8:

$$a = Wx + b \tag{2.8}$$

Donde x es el vector de la señal de entrada y b es el vector de sesgos de una capa, donde cada elemento del vector corresponde al sesgo de una neurona específica en una capa oculta o de salida. La matriz W contiene los pesos de los nodos de la capa oculta en las filas correspondientes. Dado que tenemos todas las salidas de los nodos de la capa oculta, podemos determinar las salidas de la siguiente capa, que es la capa de salida como se observa en la figura 2.14**b)**. Ahora la señal de entrada proviene de la capa oculta, y tomando los pesos de la nueva capa tendríamos un calculo similar al anterior (figura 2.14**c)**).

Usando la forma matricial de la ecuación 2.7 para calcular la salida:

$$a = \begin{bmatrix} 3 & 2 \\ 5 & 1 \end{bmatrix} \begin{bmatrix} 6 \\ 11 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 41 \\ 42 \end{bmatrix} \tag{2.9}$$

Así la salida y de acuerdo a la ecuación 2.4 es:

$$y = f(a) = \begin{bmatrix} 41 \\ 42 \end{bmatrix} \tag{2.10}$$

Este ejemplo se limita a la una función lineal para la activación de los nodos ocultos. Sin embargo, el uso de una función lineal para los nodos niega el efecto de añadir una capa, por lo que el modelo es matemáticamente idéntico a una red neuronal de una sola capa, que no tiene capas ocultas.

La red neuronal almacena información en forma de pesos. Por lo tanto, para entrenarla con nueva información, los pesos deben modificarse. El enfoque sistemático para modificar los pesos según la información proporcionada se denomina regla de aprendizaje, que es el algoritmo o conjunto de procedimientos que la red utiliza para ajustar los pesos y sesgos. Dado que el entrenamiento es la única forma en que la red neuronal almacena la información sistemáticamente, la regla de aprendizaje es un componente vital en la investigación de redes neuronales [21].

La regla delta, es la regla de aprendizaje representativa de la red neuronal de una sola capa. Si bien no permite el entrenamiento de redes neuronales multicapa, resulta muy útil para estudiar los conceptos importantes de la regla de aprendizaje de la red neuronal. Esta regla se expresa con la ecuación 2.11.

$$w_{ij} \leftarrow w_{ij} + \alpha e_i x_j \tag{2.11}$$

Donde x_j la salida del nodo de entrada j. El error del nodo de salida i está dado por e_i. El peso w_{ij} entre el nodo de salida i y el nodo de entrada j. α es la tasa de aprendizaje ($0 < \alpha \leq 1$).

La tasa de aprendizaje α, determina cuánto cambia el peso. Si este valor es demasiado alto, la salida se desvía de la solución y no converge. Por el contrario, si es demasiado bajo, el cálculo llega a la solución lentamente.

El proceso de entrenamiento de una red neuronal de una sola capa utilizando la regla delta puede resumirse de la siguiente manera:

1) Se comienza asignando valores adecuados a los pesos de la red, los cuales dependen del objetivo específico que se busca alcanzar, no se asignan de forma arbitraria o predeterminada a un valor fijo como cero o uno. En cambio, se utilizan estrategias específicas para la inicialización aleatoria.

2) Se toma un conjunto de datos de entrenamiento (compuesto por una entrada y su correcta salida) y se introduce en la red neuronal. Luego, se calcula el error entre la salida generada por la red, y_i, y la salida, d_i, utilizando la ecuación 2.12.

$$e_i = d_i - y_i \tag{2.12}$$

3) Con base en el error calculado, se determinan las actualizaciones de los pesos aplicando la regla delta, como se describe en la ecuación 2.13.

$$\Delta w_{ij} = \alpha e_i x_j \tag{2.13}$$

4)Los pesos de la red se actualizan utilizando la ecuación 2.14.

$$w_{ij} \leftarrow w_{ij} + \Delta w_{ij} \tag{2.14}$$

5) De los pasos 2 a 4 se repiten para todos los pares de datos de entrenamiento disponibles. Y finalmente los pasos 2 a 5 se ejecutan de manera iterativa hasta que el error global de la red alcance un nivel de tolerancia aceptable.

Una vez completado el paso 5, el modelo se ha entrenado con todos los datos. Entonces, ¿por qué lo entrenamos utilizando los mismos datos de entrenamiento? Esto se debe a que la regla delta busca la solución al repetir el proceso, en lugar de resolverlo todo de una vez. Todo el proceso se repite, ya que volver a entrenar el modelo con los mismos datos puede mejorarlo. El número de iteraciones de entrenamiento, en cada una de las cuales todos los datos de entrenamiento pasan por los pasos 2 a 5 una vez, se denomina época. Por ejemplo, época = 10 significa que la red neuronal realiza 10 procesos de entrenamiento repetidos con el mismo conjunto de datos [21].

2.4.4. Descenso de Gradiente

El descenso de gradiente comienza desde el valor inicial y avanza hasta la solución, *su nombre se debe a su comportamiento de buscar la solución como si una pelota rodara cuesta abajo por la trayectoria más empinada.* En esta analogía, la posición de la pelota es la salida ocasional del modelo, y el fondo es la solución. Cabe destacar que el método de descenso de gradiente no puede dejar caer la pelota hasta el fondo con un solo lanzamiento La regla delta es un tipo de método numérico llamado descenso de gradiente [21].

Sin embargo, existe una forma más generalizada de la regla delta. Para una función de activación arbitraria, la regla delta se expresa mediante la siguiente ecuación.

$$w_{ij} \leftarrow w_{ij} + \alpha \delta_i x_j \tag{2.15}$$

Es lo mismo que la regla delta de la ecuación 2.11, excepto que e_i se reemplaza por δ_i. En la ecuación 2.15, δ_i se define como:

$$\delta_i = \varphi'(\nu_i) e_i \tag{2.16}$$

Donde e_i es el error del nodo de salida i. La suma ponderada del nodo de salida i está dada por ν_i. La derivada de la función de activación φ del nodo de salida i es φ_i [21].

Aunque la fórmula de actualización de peso es complicada, mantiene el concepto donde el peso se determina en proporción al error del nodo de salida, e_i y el valor del nodo de entrada, x_j.

2.4.5. Descenso de Gradiente Estocástico

El Descenso de Gradiente Estocástico (SGD) calcula el error de cada dato de entrenamiento y ajusta las ponderaciones inmediatamente. Si tenemos 100 puntos de datos de entrenamiento, el SGD ajusta las ponderaciones 100 veces [21].

A medida que el descenso de gradiente estocástico ajusta el peso de cada punto de datos, el rendimiento de la red neuronal se ve afectado durante el proceso de entrenamiento. El término *estocástico* implica un comportamiento aleatorio durante el proceso de entrenamiento. El SGD calcula las actualizaciones de peso con la ecuación 2.17

$$\Delta w_{ij} = \alpha \delta_i x_j \tag{2.17}$$

La solución al problema del entrenamiento de redes neuronales multicapa de parámetros de aprendizaje, provino del uso del cálculo diferencial y la aplicación de métodos de optimización basados en gradientes. Un cambio importante fue reemplazar la función escalonada de la ecuación 2.6 por funciones de activación diferenciables continuas con un gradiente distinto de cero. Otra modificación clave fue la introducción de funciones de error diferenciables que definen la precisión con la que una elección dada de valores de parámetros predice las variables objetivo en el conjunto de entrenamiento.

Con estos cambios, ahora disponemos de una función de error cuyas derivadas con respecto a cada uno de los parámetros de la red pueden evaluarse. Ahora podemos considerar redes con más de una capa de parámetros. La figura 2.13 muestra una red simple con dos capas de procesamiento. Los nodos de la capa intermedia se denominan unidades ocultas porque sus valores no aparecen en el conjunto de entrenamiento, solo proporcionan valores de entrada y salida. Cada una de las unidades ocultas y de salida de la figura 2.13 calcula una función con la forma dada por la ecuación 2.2 y 2.5. Para un conjunto dado de valores de entrada, los estados de todas las unidades ocultas y de salida pueden evaluarse mediante la aplicación repetida de las ecuaciones 2.2 y 2.5, donde la información fluye hacia adelante a través de la red en la dirección de las líneas. Por esta razón, estos modelos a veces también se denominan redes neuronales de propagación hacia adelante.

Para entrenar una red de este tipo, los parámetros se utilizan estrategias específicas para la inicialización aleatoria como la inicialización de Xavier o la inicialización de He que mantienen la varianza de las activaciones de las neuronas en un rango razonable a lo largo de las capas, evitando que los valores se vuelvan demasiado grandes o demasiado pequeños para luego actualizarse iterativamente mediante técnicas de optimización basadas en gradientes. Esto implica evaluar las derivadas de la función de error, lo cual se puede realizar mediante un proceso conocido como retropropagación (*backpropagation*) de errores. En la *backpropagation*, la información fluye en sentido inverso a través de la red desde las salidas hacia las entradas. Existen numerosos algoritmos de optimización que utilizan gradientes de la función a optimizar, pero el más común en el aprendizaje automático se conoce como *descenso de gradiente estocástico*.

Una idea clave es que el aprendizaje a partir de datos implica suposiciones previas, a veces denominadas conocimiento previo. Estos podrían incorporarse explícitamente, por ejemplo, al diseñar la estructura de una red neuronal de modo que la clasificación de una lesión cutánea no dependa de su ubicación dentro de la imagen, o podrían adoptar la forma de suposiciones implícitas que surgen de la forma matemática del modelo o de su forma de entrenamiento.

El desarrollo de la *backpropagation* y la optimización basada en gradientes aumentó drásticamente la capacidad de las redes neuronales para resolver problemas prácticos. Sin embargo, en redes con muchas capas, solo los pesos de las dos últimas capas aprendían valores útiles. Con algunas excepciones, en particular los modelos utilizados para el análisis de imágenes, conocidos como redes neuronales convolucionales (LeCun et al., 1998), hubo muy pocas aplicaciones exitosas de redes con más de dos capas [23]. Nuevamente, esto limitó la complejidad de los problemas que podían abordarse eficazmente con este tipo de redes. Para lograr un rendimiento razonable

en muchas aplicaciones, fue necesario utilizar un preprocesamiento manual para transformar las variables de entrada en un nuevo espacio donde, se esperaba, el problema de aprendizaje automático fuera más fácil de resolver. Esta etapa de preprocesamiento a veces también se denomina extracción de características. Aunque este enfoque a veces era efectivo, sería claramente mucho mejor si las características pudieran aprenderse de los datos en lugar de crearse manualmente [23].

Estas redes aprenden con los datos de entrenamiento, y mejoran su precisión y generalización con datos de validación y prueba. Una vez que están bien ajustadas, es decir, que ha alcanzado un equilibrio óptimo entre su capacidad para aprender patrones en los datos de entrenamiento y su capacidad para generalizar esos patrones a datos nuevos y no vistos, se convierten en herramientas muy potentes en informática e inteligencia artificial, permitiendo clasificar y organizar datos de manera rápida y eficiente. Por ejemplo, tareas como el reconocimiento de voz o de imágenes, que podrían llevar horas si las hicieran expertos humanos, pueden realizarse en minutos con una red neuronal [24].

2.4.6. Aprendizaje Profundo

El aprendizaje profundo es un subcampo del aprendizaje automático que se basa en redes neuronales con múltiples capas. Estas redes son especialmente eficaces para manejar tareas complejas que involucran grandes volúmenes de datos y requieren la identificación de patrones intrincados, como el reconocimiento de imágenes o el procesamiento del lenguaje natural como clasificación de texto, traducción automática, generación de texto y más [25].

Como una red neuronal aprende automáticamente características directamente de los datos, se necesitan grandes cantidades de datos en bruto, es decir, datos que no han sido previamente analizados o interpretados. Cuantos más datos se procesen, mejor será el rendimiento del modelo predictivo, ya que este mejora constantemente con la información que recibe.

En el aprendizaje automático, la literatura distingue entre aprendizaje supervisado, no supervisado y de refuerzo, refiriéndose a un entorno donde la variable de resultado es conocida (supervisada) o desconocida (no supervisada).

El aprendizaje supervisado se corresponde con tareas de regresión o clasificación. En el caso de la regresión, la variable de salida es de tipo numérico, mientras que en la clasificación, la salida es categórica. Por ejemplo, en una tarea de clasificación, el objetivo podría ser asignar entradas, como imágenes, a etiquetas específicas, como "gato". Para lograr esto, el modelo se entrena observando una gran cantidad de ejemplos de pares de entradas y salidas, lo que le permite aprender a asociar correctamente las entradas con las salidas deseadas.

El aprendizaje no supervisado trabaja con datos no etiquetados y se centra principalmente en la generación de una nueva variable objetivo mediante algoritmos de agrupamiento adecuados o modelos de factores latentes. El aprendizaje supervisado se centra principalmente en la predicción, mientras que el aprendizaje no supervisado abarca muchos tipos de los denominados modelos generativos; modelos en los que se genera un resultado a partir de datos brutos [26].

En el caso del *Deep Learning*, este mapeo de entrada a salida se realiza a través de una red neuronal artificial compuesta por múltiples capas organizadas de forma jerárquica.

La red comienza aprendiendo algo simple en la primera capa y luego envía esa información a la siguiente. La segunda capa toma esa información básica, la combina para formar algo un poco más complejo, y la pasa a la tercera capa. Este proceso continúa en cada capa, donde cada una construye algo más complejo a partir de la información recibida de la capa anterior (figura 2.15). De esta manera, la red aprende gradualmente a través de la exposición a los datos de ejemplo.

La forma en que cada capa transforma la información que recibe se almacena en los pesos de la capa (que son números). Es decir, la transformación de datos en cada capa está definida por

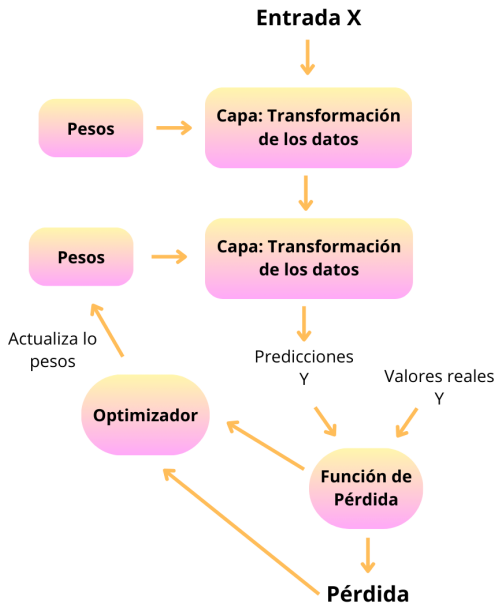

Figura 2.15: Diagrama de flujo del funcionamiento de Deep Learnig.

sus pesos. Para que la red aprenda correctamente, se deben ajustar los pesos de todas las capas de manera que la red realice un mapeo preciso entre las entradas y las salidas deseadas.

Para el rendimiento de la red neuronal, se necesita medir qué tan lejos está la salida generada por la red de la salida esperada. Esto es responsabilidad de la función de pérdida, que compara las predicciones del modelo con los valores objetivos (lo que se quiere que la red produzca) y calcula la diferencia entre ambos. Esta diferencia nos indica qué tan bien está funcionando el modelo para un ejemplo específico.

El principio clave del *Deep Learning* es utilizar el valor de la función de pérdida para retroalimentar la red y ajustar los pesos en la dirección que reduzca la pérdida. Este proceso de ajuste es llevado a cabo por el optimizador, que implementa el algoritmo de propagación hacia atrás (*backpropagation*). A través de este mecanismo, la red mejora gradualmente su capacidad para realizar predicciones precisas [22].

La propagación hacia atrás o *backpropagation* es un algoritmo de optimización que funciona mediante la determinación de la pérdida (o error) en la salida y luego propagándose de nuevo hacia atrás en la red. De esta forma los pesos se van actualizando para minimizar el error resultante de cada neurona, como el descenso de gradiente, que ajusta los pesos iterativamente para mejorar el rendimiento de la red. Este algoritmo es lo que les permite a las redes neuronales aprender.

En el ambiente médico, el aprendizaje automático ayuda en el diagnóstico de enfermedades ya que pueden analizar imágenes médicas, como rayos X y resonancias magnéticas, para detectar y diagnosticar enfermedades como el cáncer, mejorando los resultados de diagnóstico y así promover mejores tratamientos en los pacientes [27].

2.4.7. Transferencia de Aprendizaje (Transfer Learning)

El aprendizaje por transferencia es un subcampo de la inteligencia artificial que busca desarrollar habilidades similares en problemas de aprendizaje automático, con el objetivo de

mejorar la velocidad de aprendizaje, la eficiencia y reducir los requisitos de datos. A diferencia de otros algoritmos de aprendizaje automático, que suelen enfocarse en una sola tarea, el aprendizaje por transferencia se basa en aprovechar el conocimiento adquirido en una o varias tareas de origen para mejorar el rendimiento en una tarea de destino relacionada [28]. En lugar de entrenar un modelo desde cero, se utiliza un modelo ya entrenado como punto de partida, lo que permite ahorrar tiempo, recursos computacionales y mejorar el rendimiento.

Existen diferentes métodos que son utilizados en el aprendizaje por transferencia para lograr una mayor precisión en las CNNs (figura 2.16), como la técnica de *Fine-tuning* (ajuste fino), la cual consiste en utilizar modelos previamente entrenados, retirando la última capa completamente conectada y reemplazando esta por una nueva, que tiene el mismo número de clases (etiquetas) en la tarea de clasificación. Este enfoque es particularmente efectivo en diversos dominios, incluida la verificación de palabras, el reconocimiento facial y las imágenes médicas. El ajuste fino puede mejorar el rendimiento de los modelos y reducir la necesidad de una amplia formación y anotación de datos, y se ha utilizado ampliamente en la clasificación de imágenes médicas, donde los datos suelen ser escasos, variables y complejos [29].

Un modelo de aprendizaje profundo se entrena en un conjunto de datos grande y diverso, como **Segment Anything Model (SAM)** que es un modelo de segmentación de imágenes que tiene la capacidad de adaptarse de forma rápida y fácil a diferentes situaciones en las tareas de análisis de imágenes debido a su entrenamiento con un gran repositorio de imágenes. Luego, en lugar de entrenar un modelo desde cero para una nueva tarea, se aprovechan los conocimientos aprendidos en las primeras capas del modelo y solo se ajustan las capas finales para la tarea específica.

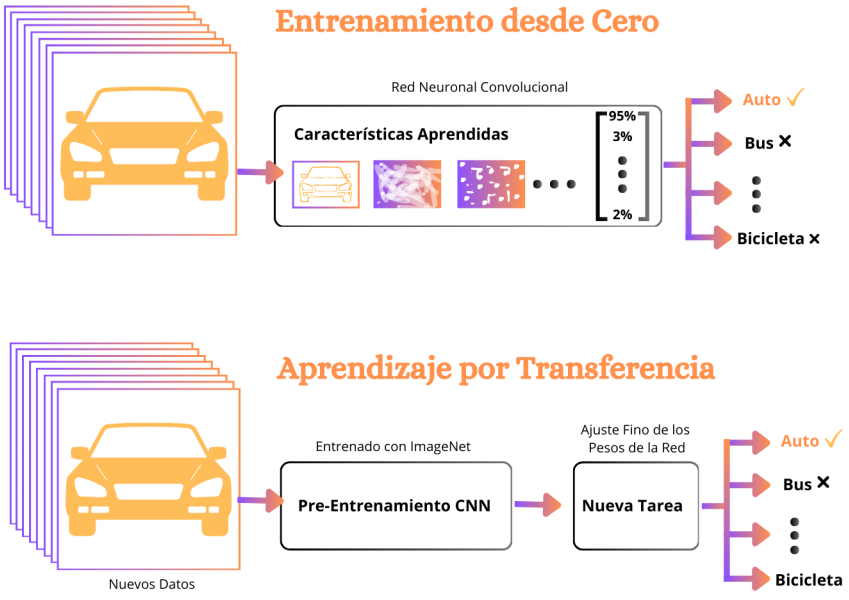

Figura 2.16: Ejemplo de aprendizaje por transferencia para la clasificación de imágenes de una CNN.

Por ejemplo, el modelo *SAM* fue preentrenado y adaptado, dando como resultado un nuevo modelo: **Medical Segment Anything Model (MedSAM)** que hace uso del *Fine-tuning*, mejora significativamente el rendimiento de segmentación de SAM para imágenes médicas [30].

2.4.8. Transformers en Visión por Computadora (ViTs)

Los modelos **Transformer** fueron introducidos por *Vaswani et al.* en 2017. Originalmente diseñados para tareas de procesamiento de lenguaje natural (PLN), que se enfoca en la interacción entre las computadoras y el lenguaje humano. Su objetivo es permitir que las máquinas entiendan, interpreten, generen y manipulen el lenguaje humano de manera útil y significativa. Los *Transformers* han demostrado ser altamente efectivos en diversas aplicaciones, incluidas la visión por computadora y la segmentación médica. Son basados en el mecanismo de autoatención (*self-attention*), que permite modelar relaciones entre diferentes partes de una entrada sin la necesidad de operaciones secuenciales como en redes convolucionales o recurrentes [31].

Dosovitskiy et al. propusieron el *Vision Transformer* (ViT) en 2020, una adaptación de la arquitectura *Transformer* para el análisis de imágenes. ViT desafía el uso tradicional de Redes Convolucionales (CNNs), que han dominado la visión por computadora durante años, al eliminar por completo el uso de convoluciones y reemplazarlas con mecanismos de atención [32].

Un ViT procesa imágenes dividiéndolas en parches y aplicando mecanismos de autoatención (figura 2.17), similar a cómo los *Transformers* procesan palabras en PLN. El funcionamiento puede verse, de la siguiente manera; sea una imagen de tamaño $H \times W$ se divide en pequeños parches de tamaño $P \times P$, generando un conjunto de N parches, cada parche se aplana y se convierte en un vector de características mediante una proyección lineal. Como los *Transformers* no tienen una estructura espacial inherente (a diferencia de las CNN), se añade un vector de posición a cada parche para retener información espacial. Cada parche es procesado a través de múltiples capas de *self-attention* (autoatención), un mecanismo que permite a un modelo *prestar atención* a diferentes partes de una secuencia de entrada para entender cómo se modelan las relaciones globales entre los elementos de la imagen. Se agrega un *token* especial o *class token*, que resume la información global y se pasa a una red completamente conectada para generar una predicción. La representación final del *class token* se utiliza para clasificar la imagen (por ejemplo, predecir si es un gato o un perro) [32].

Los ViTs han demostrado ser eficaces en tareas de segmentación médica, como la segmentación de órganos en imágenes de resonancia magnética o la detección de anomalías en mamografías. Modelos como MedSAM combinan la flexibilidad de los *Transformers* con estructuras tipo U-Net para mejorar la segmentación en imágenes médicas.

2.4.9. Segment Anything Model (SAM): Fundamentos

SAM representa un gran paso en el campo de la segmentación de imágenes, destacándose por su versatilidad en una amplia gama de aplicaciones. Éstas incluyen desde el procesamiento de campos de luz e imágenes médicas hasta la mejora de la robustez en escenarios adversariales, es decir, la capacidad de los modelos de aprendizaje automático para resistir diversos tipos de ataques diseñados para engañarlos [23]. Su arquitectura se fundamenta en el *Vision Transformer (ViT)*, que aprovecha mecanismos de atención para procesar y comprender imágenes, permitiéndole capturar relaciones espaciales complejas y segmentar objetos con alta eficiencia [33].

SAM se entrena con un extenso conjunto de datos que incluye más de mil millones de máscaras en 11 millones de imágenes. Esto le permite desarrollar capacidades robustas de *zero-shot* (cero disparos) y *few-shot* (pocos disparos), es decir, la habilidad de realizar tareas de segmentación de imágenes sin necesidad de entrenamiento específico para esas tareas (*zero-shot*) o con muy pocos ejemplos (*few-shot*). Estas capacidades son especialmente valiosas, ya que permiten al modelo generalizar a nuevas tareas o dominios con mínima o ninguna supervisión adicional.

El modelo es capaz de generar máscaras de manera rápida y precisa para diversos tipos de indicaciones, ofreciendo interactividad en tiempo real y manejando eficazmente situaciones ambiguas. Su principal objetivo es servir como un modelo fundamental para la segmentación de imágenes basada en indicaciones como puntos o cuadros delimitadores para especificar los

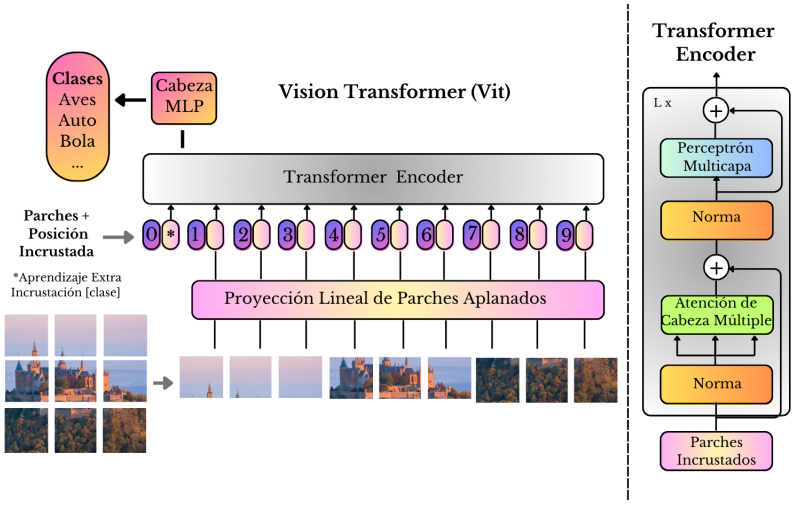

Figura 2.17: Descripción general del modelo de *Vision Transformer*. La imagen se divide en parches de tamaño fijo, cada uno de los cuales se transforma en una representación vectorial mediante una proyección lineal. A estas representaciones se les añaden *embeddings* (representaciones de vectores de números reales) de posición para preservar la información espacial. La secuencia resultante de vectores se introduce en un codificador *Transformer* convencional. Para la tarea de clasificación, se emplea el enfoque estándar de agregar un *class token* adicional, cuyo *embedding* se aprende durante el entrenamiento.

[32]

objetivos de segmentación, permitiendo una generalización sólida en una amplia variedad de distribuciones de datos [34].

La arquitectura de SAM se puede ver en la figura 2.18 y se compone de un codificador de imágenes que utiliza un enfoque basado en MAE (*Masked Autoencoder*), que reduce un modelo ViT-H/16 de 1024×1024 a 64×64 mediante convoluciones para disminuir la cantidad de canales, es decir, la operación de convolución recibe como entrada o *input* (imagen) y luego aplica sobre ella un filtro que nos devuelve un mapa de las características de la imagen original, de esta forma logramos reducir el tamaño de los parámetros [22].

Un codificador de indicaciones ya que SAM acepta dos conjuntos de indicaciones: dispersas (puntos, cuadros, texto) y densas (máscaras), que guían las tareas de segmentación. Los puntos y cuadros se representan mediante codificaciones posicionales sumadas con incrustaciones aprendidas para cada tipo de indicación, y las indicaciones densas se incrustan utilizando convoluciones y se suman elemento por elemento con la incrustación de imagen. Esta flexibilidad facilita su aplicación en una amplia gama de tareas de segmentación sin necesidad de un entrenamiento específico para cada una [22].

Y un decodificador de máscaras, este componente asocia las indicaciones con las incrustaciones de imágenes, empleando mecanismos de autoatención de indicaciones y atención cruzada en dos direcciones (incrustación de indicaciones a imagen y viceversa) para actualizar todas las incrustaciones. Después de ejecutar dos bloques, realiza un muestreo ascendente de la incrustación de imagen y un *perceptrón multicapa* (MLP) que es un tipo de red neuronal artificial compuesta por múltiples capas de neuronas, lo que le permite aprender funciones complejas [35], esté asigna el *token* de salida a un clasificador lineal dinámico, para generar la máscara final de salida.

Aunque SAM no está diseñado específicamente para segmentación médica, ha sido adaptado a ciertos contextos médicos debido a su capacidad para segmentar cualquier tipo de objeto o región de interés (ROI).

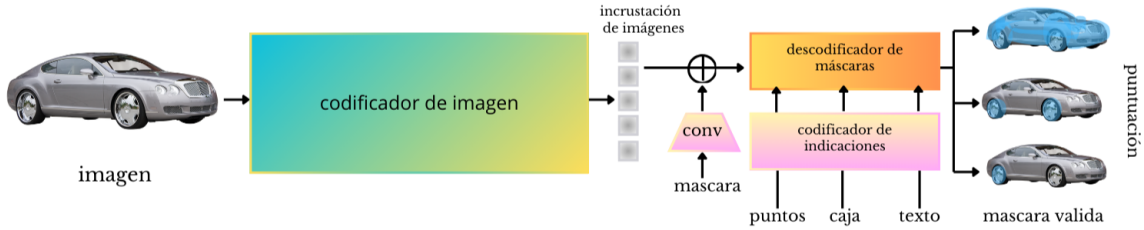

Figura 2.18: Descripción general del modelo SAM. Para solicitudes ambiguas correspondientes a más de un objeto, SAM puede generar múltiples máscaras válidas y puntajes de confianza asociados, que indica qué tan seguro está el modelo sobre la validez de una segmentación generada. [33]

2.4.10. Medical Segment Anything Model (MedSAM)

MedSAM es una adaptación especializada del modelo SAM, diseñada específicamente para la segmentación de imágenes médicas. Este modelo ha sido ajustado utilizando un conjunto de datos sin precedentes que incluye más de un millón de pares de imágenes médicas y sus correspondientes máscaras, lo que le permite abordar las particularidades únicas de este dominio [30].

MedSAM está optimizado para manejar las características distintivas de las imágenes médicas, como la diversidad de modalidades de imagen, la complejidad de las estructuras anatómicas (partes del cuerpo) y la variabilidad en los límites de los objetos. Su objetivo es servir como una herramienta universal para la segmentación de diversas estructuras anatómicas y anomalías en múltiples modalidades de imágenes médicas, ofreciendo una solución versátil y precisa.

Al igual que SAM, MedSAM se basa en una arquitectura de *transformers*. Sin embargo, mientras SAM se entrena con el conjunto de datos *SA-1B*, MedSAM ha sido entrenado con un extenso conjunto de datos médicos que incluye 1,570,263 pares de imágenes-máscaras, abarcando 10 modalidades de imagen (entre ellas la mamografía), más de 30 tipos de cáncer y una variedad de protocolos de adquisición de imágenes. Este entrenamiento exhaustivo le permite abordar una amplia gama de tareas de segmentación en el ámbito médico sin necesidad de ajustes adicionales específicos para cada tarea.

Además, MedSAM facilita la segmentación interactiva de imágenes médicas en 2D, permitiendo una delineación precisa de las estructuras de interés con intervenciones mínimas por parte del usuario como lo son las cajas delimitadoras. Esto lo convierte en una herramienta eficiente y accesible para profesionales de la salud que requieren segmentaciones rápidas en su trabajo diario [30].

Al estar basado en SAM, la arquitectura suele ser la misma, y consta de estos tres componentes clave (figura 2.19):

Codificador de imágenes: genera incrustaciones de imágenes. Una incrustación de imágenes es una representación de características de una imagen que codifica la semántica de su contenido, es decir una imagen que ha sido analizada y comprendida en términos de su significado, objetos, escenas, conocimiento y emociones, a través de la extracción de características y razonamiento de alto nivel en el procesamiento de imágenes y el reconocimiento de patrones. El codificador de imágenes analiza la imagen y genera múltiples características. Todas estas características juntas conforman la incrustación de imágenes [36].

Codificador de indicaciones: transforma las indicaciones proporcionadas por el usuario (como los cuadros delimitadores, puntos o texto) en representaciones de características. Básicamente, codifica la información espacial de las indicaciones, lo que permite que MedSAM comprenda el contexto del objeto que desea segmentar y guíe su proceso de segmentación.

Descodificador de máscara: genera máscaras de segmentación utilizando incrustaciones de imágenes e incrustaciones de indicaciones.

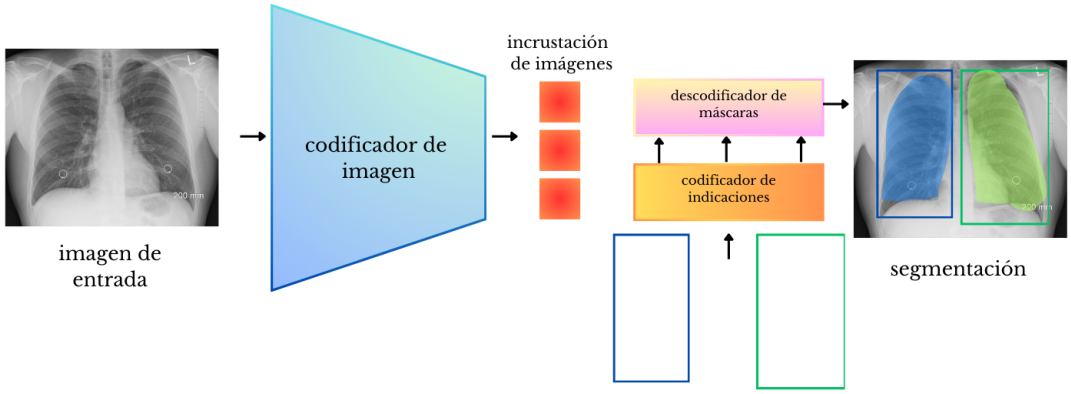

Figura 2.19: MedSAM es un método de segmentación programable en el que los usuarios pueden utilizar cajas delimitadoras para especificar los objetivos de segmentación.

El modelo es capaz de procesar tanto imágenes 2D como 3D, tratando las imágenes 3D como una secuencia de cortes 2D. Se observa que las cajas delimitadoras ofrecen un contexto espacial más claro para la región de interés, lo que permite al algoritmo identificar con mayor precisión el área objetivo.

MedSAM fue evaluado mediante validación interna y externa. En particular, se comparó con el modelo de segmentación SAM, así como con los modelos especializados en segmentación de imágenes médicas *U-Net* y *DeepLabV3+*(figura 2.20). Los resultados mostraron que MedSAM superó a SAM por un amplio margen como se muestra en la figura 2.20. También demostró una distribución más estrecha en los puntajes del **Coeficiente de Similitud de Dice (DSC)** en las tareas de validación interna, que incluyeron la segmentación de órganos y estructuras anatómicas, como el hígado, los riñones, el corazón y los pulmones en imágenes de tomografía computarizada (TC) y resonancia magnética (RM). Además, se evaluó en la detección de lesiones y anomalías, como tumores, quistes y otras patologías en diversas modalidades de imagen, así como en la segmentación de tejidos específicos, por ejemplo, el tejido cerebral en imágenes de RM. El modelo también se aplicó a diferentes modalidades de imagen, como ultrasonido, mamografías y tomografías por emisión de positrones (PET), abarcando una amplia variedad de estructuras anatómicas, condiciones patológicas y técnicas de imagen médica.

En comparación con los modelos, MedSAM mostró una mayor robustez en diversas tareas de segmentación, es decir, tiene la capacidad de soportar incertidumbres y funcionar con precisión en diferentes contextos [30].

Además, MedSAM tiene el mejor desempeño de segmentación en la mayoría de las tareas, superando el rendimiento de los modelos especializados *U-Net* y *DeepLabV3+*, comparándolos con las mismas bases de datos [30]. MedSAM es capaz de segmentar con precisión una amplia variedad de objetivos en diferentes condiciones de imagen, logrando un rendimiento comparable o incluso superior al de los modelos especializados *U-Net* y *DeepLabV3+*. Esto lo posiciona como una herramienta altamente efectiva y versátil para la segmentación en el ámbito médico.

El programa o concurso *CVPR 2024: Segment Anything In Medical Images On Laptop* tiene como objetivo desarrollar modelos universales de segmentación de imágenes médicas que puedan ejecutarse en computadoras portátiles u otros dispositivos periféricos, sin depender de *GPUs* de alto rendimiento [37].

Por lo que, **LiteMedSAM** surgío como una versión ligera del modelo original. Según lo mencionado en el *CVPR 2024*, esta nueva versión podría ejecutarse directamente desde una laptop, lo

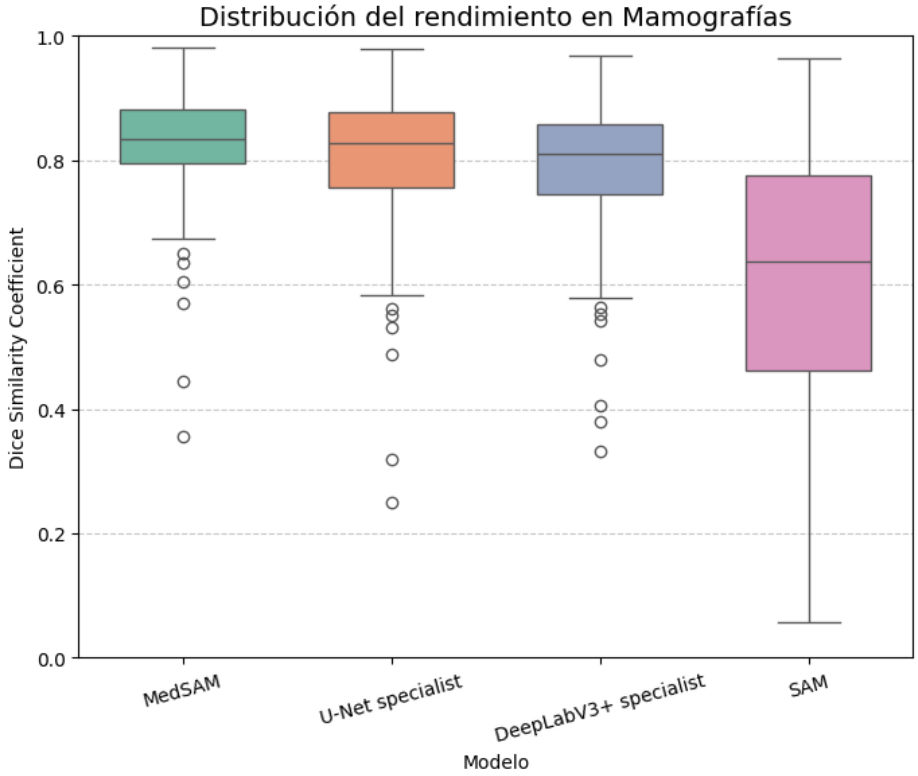

Figura 2.20: Distribución del desempeño de las tareas de validación interna en términos de la puntuación del coeficiente de similitud dice (DSC). La línea central dentro del cuadro representa la mediana, con los límites inferior y superior del cuadro delineando los percentiles 25 y 75, respectivamente.

que la haría mucho más práctica y accesible para su uso en entornos clínicos y otros escenarios donde la disponibilidad de hardware avanzado es limitada como en las unidades móviles. Esta mejora representa un avance significativo hacia la democratización de herramientas avanzadas de segmentación médica.

2.4.11. LiteMedSAM: Arquitectura de la Red

LiteMedSAM es una versión optimizada de MedSAM, un modelo basado en Segment Anything Model (SAM) pero especializado para la segmentación de imágenes médicas. Está diseñado para lograr una segmentación precisa en imágenes médicas con menores requerimientos computacionales en comparación con la versión estándar de MedSAM. Por lo tanto, se puede considerar que LiteMedSAM mantiene la misma funcionalidad que MedSAM, excepto por la incorporación de un codificador de imágenes más pequeño y eficiente.

Inspirado en las características de MobileSAM [38], donde se reemplaza el codificador por uno más ligero, LiteMedSAM adopta un enfoque similar. El modelo implementa un *Encoder* Ligero basado en una versión más eficiente del *Vision Transformer* (ViT), conocida como TinyViT. TinyViT es una familia de transformadores de visión compactos y altamente eficientes. Para optimizar aún más el rendimiento, los logits de modelos más grandes se precalculan, dispersan y almacenan en el disco, lo que reduce los costos de memoria y la sobrecarga computacional [39].

La reducción del número de parámetros en el modelo es una estrategia clave para disminuir

la carga computacional sin comprometer significativamente la precisión. Esto representa una alternativa viable para adaptar modelos robustos a entornos con recursos limitados.

LiteMedSAM introduce optimizaciones que lo hacen más eficiente y adecuado para su implementación en entornos con recursos limitados, como dispositivos móviles o clínicas con hardware restringido. El modelo pretende mantener un alto nivel de precisión en la segmentación de imágenes médicas. Esto puede permitir lograr un primer diagnóstico más rápido, lo que es especialmente crucial en casos donde se detectan indicios de cáncer, evitando así la postergación innecesaria de los tratamientos y facilitando una intervención temprana y efectiva.

2.5. Métricas de Evaluación

La validación del análisis de imágenes consiste en un conjunto de técnicas y metodologías diseñadas para evaluar el desempeño de los algoritmos que procesan y analizan imágenes. Este proceso es fundamental para asegurar que los modelos sean confiables, precisos y capaces de generalizar a nuevos datos. La validación adquiere especial relevancia en aplicaciones críticas, como el diagnóstico médico, la conducción autónoma o la vigilancia.

Aunque existen diversas métricas de evaluación, en el contexto de la segmentación de imágenes, las recomendaciones de Metrics Reloaded [40], junto con las implementadas en MedSAM, destacan dos métricas principales: el *Coeficiente de Similitud de Dice (DSC)* y la *Diferencia de Superficie Normalizada (NSD)*. Estas métricas son ampliamente empleadas para evaluar la precisión y la robustez de los modelos de segmentación, ya que no solo deben demostrar un rendimiento óptimo en condiciones ideales, sino que también conserven su precisión incluso en escenarios adversos o imprevistos.

En la práctica, la aceptabilidad de un valor en DSC o NSD depende del contexto clínico y de los requisitos específicos de cada aplicación. En casos donde se requiere una precisión alta, como en procedimientos quirúrgicos asistidos por imágenes, se buscarían valores más elevados. Un valor de 0.7 o superior en DSC y NSD, suele considerarse indicativo de una buena segmentación, ya que es un indicador de que el modelo captura características más completas [41].

2.5.1. Coeficiente de Similitud Dice (DCS)

El *Coeficiente de Similitud de Dice* (DSC), también llamado *Dice Score* o Índice de Dice, es una métrica utilizada para evaluar la precisión de modelos de segmentación de imágenes, especialmente en aplicaciones médicas [40].

El DSC mide la similitud entre dos conjuntos, típicamente una máscara de segmentación predicha y la máscara real (*ground truth*). Matemáticamente, se define como:

$$\mathrm{DSC}(G, S) = \frac{2|G \cap S|}{|G| + |S|} \tag{2.18}$$

Donde G es el conjunto de píxeles de la máscara de referencia (*ground truth*), S es el conjunto de píxeles de la máscara de segmentación predicha. $|G \cap S|$ es el número de píxeles comunes entre la intersección de la predicción y el *ground truth* y $|G| + |S|$ es la suma de los píxeles en cada máscara. El valor del *Coeficiente de Similitud Dice* (DSC) varía entre 0 y 1. Un valor de 1 indica una coincidencia perfecta entre la predicción y el *ground truth*, mientras que un valor de 0 significa que no hay coincidencia alguna.

Sin embargo, aunque el DSC es fácil de implementar e interpretar, tiene una limitación ya que solo mide la superposición de áreas entre las máscaras, sin tener en cuenta la forma o la distribución espacial de las regiones segmentadas [40].

2.5.2. Diferencia de Superficie Normalizada (NSD)

Es una métrica basada en límites. Se enfoca en la discrepancia entre los contornos de la segmentación predicha y la máscara de referencia (*ground truth*) en lugar de comparar únicamente la superposición de áreas o volúmenes como el DSC, lo que la hace especialmente útil en aplicaciones donde los bordes son importantes como en la segmentación de órganos en imágenes médicas [42]. Los resultados de la segmentación se evalúan dentro de un margen de tolerancia de acuerdo a la distancia máxima permitida por el usuario entre la segmentación automática y la segmentación de referencia para que un voxel o píxel sea considerado correctamente segmentado, el cual se establece mediante la ecuación 2.19.

$$\text{NSD} = \frac{|\partial G \cap \partial B_{\partial S}^{(\tau)}| + |\partial S \cap \partial B_{\partial G}^{(\tau)}|}{|\partial G| + |\partial S|} \tag{2.19}$$

Donde G es el conjunto de píxeles de la máscara de referencia (ground truth), S es el conjunto de píxeles de la máscara de segmentación predicha. ∂G representa el contorno de la máscara de referencia, ∂S representa el contorno de la segmentación predicha, $\partial B_{\partial S}^{(\tau)}$ es una región τ de espesor alrededor de la superficie ∂S y $\partial B_{\partial G}^{(\tau)}$ es una región τ de espesor alrededor de la superficie ∂G.

Si el valor de NSD (Diferencia de Superficie Normalizada) es igual a 1, significa que la segmentación predicha coincide completamente con la segmentación de referencia dentro del umbral τ. Por el contrario, si el NSD es 0, no hay coincidencia entre la segmentación predicha y la de referencia dentro del mismo umbral τ.

El NSD es una métrica más informativa que el DSC en escenarios donde la forma y los contornos de las regiones segmentadas son críticos. Por esta razón, se utiliza frecuentemente en la segmentación de órganos, tumores y estructuras pequeñas, donde los errores en los límites pueden tener implicaciones clínicamente significativas.

Capítulo 3

Implementación del Modelo

3.1. Hardware y Software

El modelo se implementó en una computadora portátil Dell, equipada con un procesador *Intel Core* i5-1135G7 de 11 generación (2.40 GHz, 4 núcleos y 8 procesadores lógicos), 16 GB de memoria *RAM* y una tarjeta gráfica integrada *Intel Iris Xe Graphics*, sin contar con una GPU profesional.

En este entorno, se llevó a cabo la caracterización de las imágenes, tanto para el ajuste del modelo como para la inferencia externa, asegurando que el proceso fuera eficiente y adecuado para las capacidades del *hardware* disponible.

El modelo está desarrollado en *Python*, por lo que se utilizó la versión 3.12.7 para su ejecución y evitar errores de compatibilidad. Para el análisis de datos y la visualización de resultados, se emplearon varios paquetes de *Python*, entre los que destacan: *PyTorch* (2.5.1), *SimpleITK* (2.4.0), *nibabel* (5.3.2), *torchvision* (0.20.1), *numpy* (1.26.4), *scikit-image* (0.24.0), *scipy* (1.13.1), *Pandas* (2.2.2), *matplotlib* (3.9.2), *OpenCV* (4.10.0) y *plotly* (5.24.1).

Adicionalmente, se utilizó la plataforma *Google Colab* como entorno alternativo para la implementación del modelo. Se configuró un entorno de ejecución de *Python* 3 con una GPU T4, lo que permitió aprovechar los recursos de una tarjeta gráfica donde algunas librerías como *PyTorch*, *OpenCV* y *torchvision*, se pueden desempeñar de una forma adecuada.

3.2. Conjunto de Datos

La mamografía digital (MD) es la modalidad de imagen de referencia para la detección temprana del cáncer de mama. Sin embargo, presenta limitaciones en pacientes con mamas densas, ya que su especificidad global disminuye en estos casos. La mamografía espectral con contraste (*CESM*, por sus siglas en inglés) es una técnica avanzada basada en el uso de contraste, aprobada por la *Food and Drug Administration* (FDA) en 2011. Esta técnica se utiliza como complemento de la MD y los exámenes ecográficos para la localización y caracterización de lesiones ocultas o no concluyentes [43].

Se ha demostrado que la CESM (Mamografía Espectral con Contraste) es una técnica factible y detecta cánceres de mama con una tasa similar a la resonancia magnética (MRI), lo que la posiciona como una alternativa prometedora a la MRI [44]. Entre las ventajas de la CESM sobre la MRI incluyen: la limitación del movimiento gracias al uso de compresión, una reducción en los costos, un menor tiempo de examen y la posibilidad de ser utilizada en pacientes que no pueden someterse a una MRI debido a incompatibilidades, como la presencia de un marcapasos o expansores de tejido, por lo que la calidad de las imágenes es considerablemente mejor.

El repositorio en línea de imágenes médicas enfocadas para la investigación del cáncer *The Cancer Imaging Archive* (TCIA), proporciona acceso gratuito a estudios de imágenes anónimas y así proteger la privacidad de los pacientes, además incluye la información clínica y patológica relevante [45]. De este repositorio se selecciona un conjunto de imágenes considerando la técnica de obtención de imagen CESM.

Se verificó que contengan las proyecciones o vistas, craneocaudal (CC) y oblicua mediolateral (MLO). Estas proyecciones se utilizan porque ofrecen una visualización complementaria del tejido mamario, permitiendo una mejor detección y caracterización de anomalías. Cada proyección proporciona una perspectiva diferente de la mama, lo que ayuda a reducir la superposición de tejidos y a localizar con mayor precisión las lesiones.

El conjunto de datos seleccionado es **CDD-CESM** (*Categorized Digital Database for Low energy and Subtracted Contrast Enhanced Spectral Mammography images*). Es una colección de 2006 imágenes (1003 de baja energía y 1003 imágenes sustraídas) de mamografía espectral mejorada con contraste (CESM) de alta resolución acompañadas de anotaciones e informes médicos. Estas imágenes tienen una resolución en lo general de 2355×1315 píxeles por pulgada [46]. Se utilizaron dos equipos diferentes para la adquisición de imágenes para la generación de la base de datos; *GE Healthcare Senographe DS* y *Hologic Selenia Dimensions Mammography Systems*. Cada vista consta de dos exposiciones, una con baja energía (valores de kilovoltaje pico que varían de 26 a 31 kVp) y otra con alta energía (45 a 49 kVp). Luego, las imágenes de baja y alta energía se recombinan y se sustraen mediante un procesamiento de imágenes para suprimir el parénquima mamario de fondo, con un software que realiza la adquisición de imágenes de doble energía.

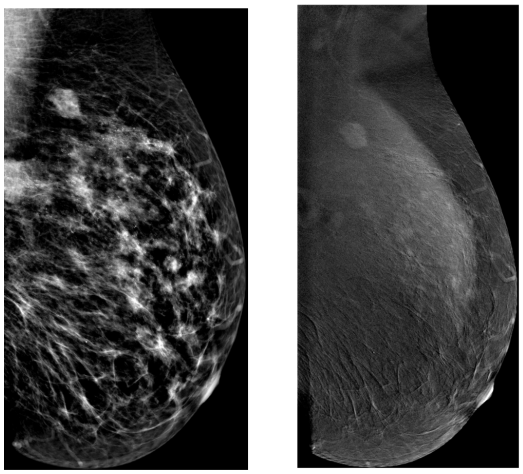

Figura 3.1: Comparación de Métodos de Imagen Mamográfica para el Diagnóstico de Patologías Mamarias: a) Imagen de baja energía y b) Imagen sustraída, ambas imágenes se extraen con la técnica CESM.

Este repositorio fue seleccionado debido a la alta calidad de sus imágenes y la técnica empleada, que permite una mejor distinción de las anomalías. Estas características contribuyen a un aprendizaje más efectivo del modelo y a una segmentación más precisa.

3.2.1. Selección y División de Datos

Dado que el objetivo principal de este estudio es evaluar la segmentación y el desempeño del modelo, no se limitó el análisis a una única patología o tipo de anomalía como microcalcificaciones, masas benignas o malignas. En su lugar, se optó por generalizar el enfoque para el afinamiento del modelo.

Para el conjunto de datos utilizado en el afinamiento del modelo, se llevó a cabo una selección manual de mamografías digitales de bajo contraste. Estas imágenes incluyen cualquier tipo de clasificación según el sistema BI-RADS (tabla 2.1), abarcando un rango de edad de 21 a 84 años. Se prioriza que la mayoría de los casos contarán con ambas proyecciones estándar: cráneo-caudal (CC) y mediolateral oblicua (MLO). Tras este proceso, se obtuvo un total de 643 imágenes, con la división de datos de 80 % o 515 imágenes para el entrenamiento y 10 % o 64 imágenes para la validación y 10 % para Prueba, cada una acompañada de su respectiva máscara de referencia.

3.2.2. Preprocesamiento y ensamblaje de datos

El archivo *npy* es el formato binario estándar de *NumPy* para almacenar arreglos de una sola matriz. Permite lectura y escritura eficientes, sin pérdida de información, mientras que el archivo *npz* es una versión extendida del archivos npy que permite almacenar múltiples arreglos en un solo archivo comprimido. Los arreglos se almacenan con nombres clave (*keys*), lo que permite una fácil recuperación.

Ambos están optimizados para su uso en *Python*, lo que permite una carga rápida de los datos en memoria. Además del almacenamiento sin pérdida de precisión, ya que puede guardar datos en punto flotante sin compresión destructiva.

Todas las imágenes se convertirán en archivos de tipo *npy* como parte del preprocesamiento[47]. Esta elección se debe principalmente a que este formato permite una carga rápida de datos, lo cual es esencial para optimizar el rendimiento. Además, los archivos *numpy* actúan como una interfaz de datos universal, lo que facilita la unificación de diferentes formatos de datos. Para simplificar la depuración y la inferencia, también se almacenan las imágenes y las etiquetas originales en archivos *npz*.

Se aplicó una normalización máximo-mínima para reescalar los valores de intensidad al rango [0, 255].

Dado que las imágenes no pueden procesarse directamente en el modelo debido a su alta resolución, se dividen en parches de tamaño uniforme de $1024 \times 1024 \times 3$ píxeles después de la etapa de división de datos, aunque 3 canales es para una imagen de color (*Red, Green, Blue*), para escala de grises se convierte en una representación (RGB) para mejor la compatibilidad, es decir, replican los valores de intensidad en los tres canales de color (RGB).

Esto significa que si la imagen en escala de grises tiene un valor de intensidad $I(x, y)$ en un píxel, la conversión a (RGB) se haría como se muestra en la ecuación 3.1.

$$I(x, y) = R(x, y) = G(x, y) = B(x, y) \tag{3.1}$$

Se utilizó la interpolación bicúbica para redimensionar las imágenes, mientras que la interpolación del vecino más cercano se aplicó para redimensionar las máscaras para preservar sus límites precisos y evitar la introducción de artefactos no deseados. Estos procedimientos de estandarización garantizaron uniformidad y compatibilidad entre todas las imágenes.

Para ejecutar este proceso, se utiliza el siguiente *script* en la terminal de *Anaconda Power Shell*[48]:

```
pre_grey_rgb.py
```

Se verifica que las dimensiones de las imágenes y sus máscaras de referencia cumplan con los requisitos previamente establecidos. Las máscaras de referencia corresponden a la segmentación de la imagen, creadas manualmente por médicos expertos. Están dadas por el conjunto de imágenes, por lo que el tipo de archivo tiene características diferentes y puede ser un problema en el momento de realizar la evaluación de las segmentaciones.

Los cuadro delimitadores son generados a partir de las máscaras de referencia, ya que estos van en un mismo archivo *npz* para la validación del modelo.

Se realiza una verificación del tipo de etiquetado que tiene cada conjunto de imágenes. El etiquetado corresponde a cada anomalía que existe en la imagen, por ejemplo: si una imagen contiene 3 anomalías, cada una de éstas tiene asignado un valor en la escala de grises (figura 3.2) y este valor tendrá la etiqueta, 1, 2, y 3. Dependerá del número de anomalías de cada caso el número de etiquetas que contengan.

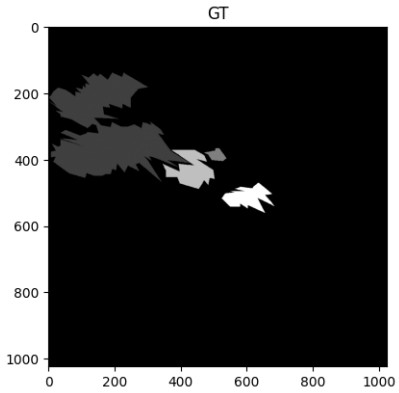

Figura 3.2: Máscara de referencia con anomalías con diferente escala de gris.

Dado que estas funcionalidades no estaban disponibles en el repositorio base, se desarrolló un *script* específico para llevar a cabo estas caracterizaciones.

Las limitaciones de *hardware* y las librerías implementadas como *pytorch* que es una biblioteca de código abierto para aprendizaje profundo, es ampliamente utilizada en visión por computadora y trabaja mejor con una tarjeta gráfica. Debido a esto, el proceso se vio interrumpido por el exceso de trabajo, por lo que se decidió hacer uso de la plataforma de *Google Colap* para las tareas de afinación y validación, ya que son las tareas que tienen mayor carga de trabajo.

3.3. Afinamiento (Fine-tuning)

Al igual que MedSAM, su versión ligera está construido sobre una arquitectura basada en transformadores, que integra tres componentes principales: un codificador de imágenes basado en un Transformador de Visión (ViT), responsable de extraer características de la imagen; un codificador de indicaciones para procesar las interacciones del usuario, como los cuadros delimitadores; y un decodificador de máscaras que genera la segmentación final y las puntuaciones de confianza utilizando las incrustaciones de la imagen, las indicaciones y un *token* de salida.

El modelo ViT base utilizado consta de 12 capas de transformador, donde cada bloque incluye un mecanismo de auto-atención de múltiples cabezales y un *Perceptrón Multicapa* (MLP) con

normalización de capas. La imagen de entrada, con un tamaño de $1024 \times 1024 \times 3$, en lugar de analizar la imagen completa, se divide en una secuencia de parches pequeños 2D aplanados de $16 \times 16 \times 3$, aplanar una imagen se refiere a la transformación de un parche 2D en una representación 1D. Es decir, se toma una matriz bidimensional y se convierte en un vector unidimensional. La imagen original de 1024×1024 se convierte en una representación de 64×64 *tokens* que contienen la información relevante para el modelo, esto significa que la resolución espacial se ha reducido 16 veces en comparación con la imagen original.

Los codificadores de indicaciones transforman las coordenadas de los puntos de las esquinas de los cuadros delimitadores en incrustaciones vectoriales de 256 dimensiones. Cada cuadro delimitador se representa mediante un par de incrustaciones correspondientes a las coordenadas de la esquina superior izquierda y la esquina inferior derecha.

Para garantizar la interactividad en tiempo real, se utiliza un decodificador de máscaras ligero. Este consta de dos capas de transformador que fusionan la incrustación de la imagen con la codificación de las indicaciones, seguido de dos capas convolucionales transpuestas que aumentan la resolución de la incrustación a 256×256. Una capa convolucional transpuesta es una operación que aumenta la resolución espacial de un mapa de características, ya que a diferencia de una convolución tradicional, que reduce el tamaño de la entrada. Finalmente, la incrustación pasa por una función de activación *sigmoidea*, que es una función matemática utilizada para introducir no linealidad y permitir que el modelo aprenda relaciones complejas entre los datos de entrada y salida, y se aplica una interpolación bilineal para ajustar el tamaño de salida al de la imagen de entrada, que utiliza el promedio ponderado por la distancia de los cuatro valores de píxeles más cercanos para estimar un nuevo valor de píxel. Este diseño permite una segmentación precisa y eficiente, adaptada a las necesidades de interacción del usuario.

Se propone realizar un afinamiento del modelo LiteMedSAM utilizando *transfer learning*, partiendo del *checkpoint* preentrenado de LiteMedSAM, que representa el último punto guardado durante su entrenamiento.

Según los autores del modelo, recomiendan que la configuración inicial del entrenamiento se establezca en 10 épocas. Cada época permite que la red neuronal ajuste sus pesos y sesgos en función del error cometido en la iteración anterior. Un mayor número de épocas puede mejorar la capacidad de la red para optimizar sus parámetros y aprender patrones más complejos. No obstante, si el número de épocas es excesivamente alto, existe el riesgo de que el modelo sobreajuste (*overfitting*) los datos de entrenamiento, memorizando patrones específicos de las imágenes en lugar de aprender características generalizables. Esto reduciría su efectividad al enfrentarse a datos nuevos, como los conjuntos de validación o prueba.

Teniendo en cuenta estos factores, se realizaron pruebas iniciales para observar el comportamiento del modelo y determinar un equilibrio adecuado entre el número de épocas y el rendimiento.

El proceso de afinamiento se llevó a cabo en la plataforma de *Google Colab*, configurado con una GPU para acelerar los cálculos. Se contemplaron 15 épocas tras evaluar los resultados previos de algunos entrenamientos y observando la gráficas de validación, dando prioridad a las siguientes métricas de validación: *Coeficiente de Similitud de Dice*, *Recall* y *Presicion* [49] [50], para analizar que tan bien el modelo generaliza los datos que no ha visto durante el entrenamiento.

El último punto de control generado durante este proceso se seleccionó como el modelo final, ya que representaba el mejor equilibrio entre precisión y generalización.

Para ejecutar este proceso, se utiliza el siguiente *script* en una celda de *Google Colab*[48] con las correspondientes direcciones a los datos de entrenamiento:

```
train_one_gpu1.py
```

3.4. Segmentación y Evaluación

En primer lugar, se extraen los pesos del modelo afinado, los cuales se almacenan en una carpeta específica para su posterior uso durante la fase de Prueba. Este proceso se lleva a cabo ejecutando el siguiente *script*:

```
extract_weights.py
```

Los pesos del modelo son los valores aprendidos durante el entrenamiento que definen cómo la red neuronal procesa la información y genera predicciones. Una vez que el modelo está entrenado, se pueden usar los pesos guardados para hacer predicciones sin necesidad de volver a realizar el entrenamiento.

El modelo guarda dos puntos de control (*checkpoint*), con los sufijos *latest*, que guarda el estado más reciente del modelo al final del último ciclo de entrenamiento y basándose en la menor pérdida registrada durante el entrenamiento se guarda el *checkpoint best*.

Se utilizó el punto de control *medsam_lite_best_1.pth*, el cual contiene los pesos obtenidos durante el proceso de afinamiento. Estos pesos representan la configuración óptima del modelo después de haber sido ajustado para la tarea específica de segmentación en mamografías.

Para generar las segmentaciones a partir de las imágenes de entrada y sus respectivas cajas delimitadoras, se implementó un *script* en un entorno de *Google Colab*. El *script* permite cargar el modelo preentrenado, procesar las imágenes junto con las coordenadas de las cajas delimitadoras y producir las máscaras de segmentación correspondientes. A continuación, se muestra la línea de código utilizado para llevar a cabo este proceso:

```
CVPR24_LiteMedSAM_infer.py
```

Una vez obtenidos los resultados de la prueba, se desarrolló un *script* para realizar una verificación de las segmentaciones generadas. Este *script* se encarga de comprobar las dimensiones de las segmentaciones y las compara con las máscaras de referencia (gts), asegurando que ambas coincidan en tamaño y resolución. Esta validación es crucial para garantizar la coherencia de los datos antes de proceder con la evaluación del modelo. En otro *script*, se realizó un lector de etiquetas, el cual verifica que la asignación de parámetros en escala de grises en las segmentaciones coincida con las máscaras de referencia. Esto es para confirmar que las etiquetas utilizadas en el proceso de inferencia sean consistentes con las anotaciones originales, evitando discrepancias que podrían afectar la precisión de la evaluación.

Finalmente, para evaluar las segmentaciones realizadas por el modelo, se emplearon métricas ampliamente reconocidas en la segmentación del ámbito médico: (DSC) y (NSD). Estas métricas están integradas en un único *script* de evaluación, el cual calcula automáticamente los coeficientes DSC y NSD para cada imagen procesada. Los resultados se almacenan en un archivo CSV, que contiene los valores de DSC y NSD correspondientes a cada imagen. Este archivo permite un análisis posterior detallado, facilitando la verificación de la eficiencia del modelo.

La evaluación se ejecuta mediante el siguiente comando o proceso:

```
evaluation/compute_metrics.py
```

3.4.1. Análisis Estadístico

Para analizar y comparar estadísticamente el rendimiento del modelo de la segmentación antes y después de la realización del *Fini-Tuning*. Se realiza una verificación de la normalidad de las métricas (DSC y NSD), debido a la no normalidad de estos se emplea la prueba de rangos con signo de *Wilcoxon*. Esta prueba no paramétrica es ideal para comparar muestras pareadas (la misma

imagen antes y después). La prueba de *Wilcoxon* se basa en los rangos de las diferencias entre las observaciones pareadas, esto es que se calcula la diferencia entre cada par de observaciones. Por ejemplo, *DSC después* y *DSC antes* para cada imagen, la resta es: (*DSC después - DSC antes*).

La clave para interpretar la prueba de *Wilcoxon*, al igual que muchas otras pruebas estadísticas, es el *valor p*. Si el valor $p < \alpha$, se rechaza la hipótesis nula (H_0), es decir que existe una diferencia estadísticamente significativa entre las dos condiciones, si el valor $p \geq \alpha$, no rechaza la hipótesis nula (H_0), significa que no hay suficiente evidencia estadística para concluir que existe una diferencia significativa entre las dos condiciones. Podrías decir que el fine-tuning no produjo un cambio estadísticamente significativo en la métrica. Además al ser una muestra grande se verifica que la suma de rangos de diferencia sea mayormente positiva incluyendo también el estadístico de efecto r para las muestras de pares coincidentes sea mayor a 0.5.

Esta evaluación se realiza utilizando el software *Matlab*, con el modulo *signrank* que esta enfocado a estas prueba de comparación.

3.5. Disponibilidad de códigos

El código utilizado para el preprocesamiento de datos, entrenamiento, inferencia y evaluación de segmentaciones con LiteMedSAM están disponibles públicamente en el siguiente repositorio en: github LiteMedSAM.

Los script de inferencia y evaluación de métricas se pueden visualizar en el apéndice A.1 y A.2.

El sript completo de implementación del modelo se acoplo en archivo de Google Colab, y esta disponible en: Google Colab.

Capítulo 4

Resultados

4.1. Afinamiento

Se observa en la figura 4.1 las 35 épocas del *Fine-tuning*. Durante el afinamiento con un conjunto de 514 imágenes, las métricas de validación que se emplearon para verificar el rendimiento del modelo se pueden observar en la tabla 4.1. El modelo es estable, no hay sobreajuste fuerte ni caída brusca en las métricas.

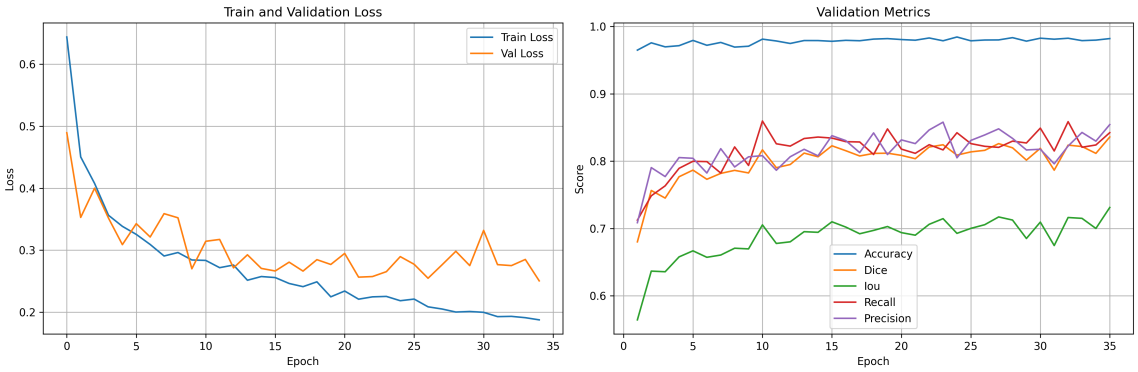

Figura 4.1: Pérdida de entrenamiento y validación (gráfico izquierdo) y Métricas de validación (gráfico derecho) durante el afinamiento del modelo a lo largo de 35 Épocas.

Métrica de Validación	**Rango**
Accuracy	$\sim$ 0.95 - 0.98
Dice	$\sim$ 0.78 - 0.83
IoU	$\sim$ 0.64 - 0.72
Recall	$\sim$ 0.80 - 0.87
Presicion	$\sim$ 0.80 - 0.86

Tabla 4.1: Rango de las métricas de validación durante el afinamiento.

4.2. Desempeño del Modelo

4.2.1. Inferencia

Para evaluar el rendimiento del modelo en imágenes mamográficas, se llevó a cabo una prueba con las imágenes del conjunto de datos seleccionadas antes de proceder con el afinamiento del modelo.

Los resultados de la prueba después de afinar el modelo se puede observar en la figura 4.2.

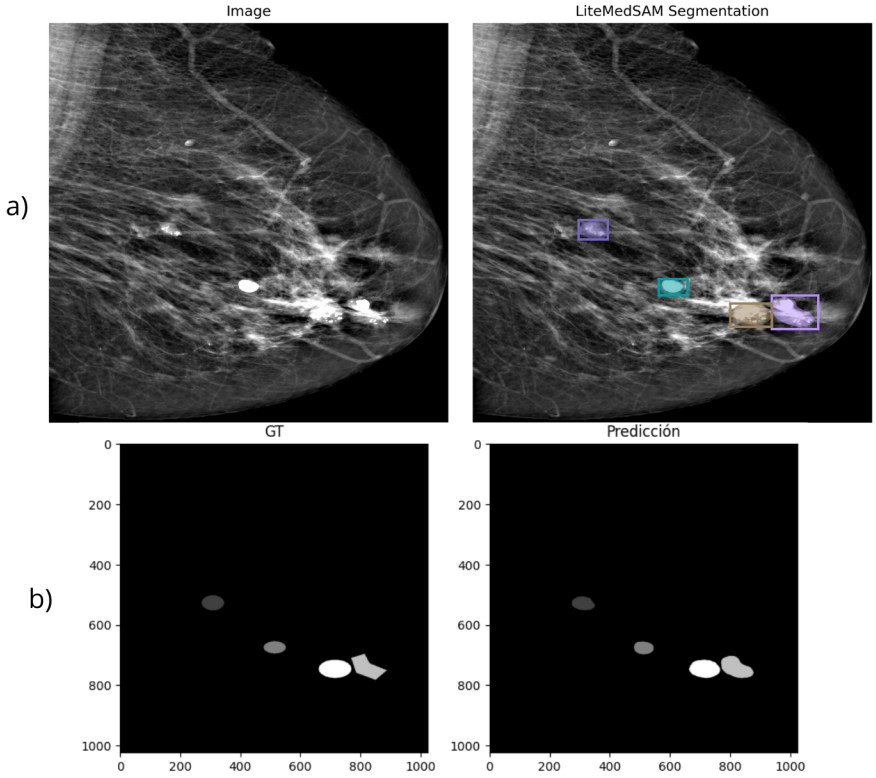

Figura 4.2: Comparación de segmentación con la mascara de referencia de caso (P149_L_DM_MLO) del conjunto de imágenes CDD-CESM. a) Imagen original e imagen segmentada, b) Mascara de referencia y mascara de la segmentación.

Se midió el tiempo que tarda el modelo en segmentar cada una de las imágenes de acuerdo al conjunto de prueba. La tabla 4.2 muestra el tiempo promedio para antes y después del afinamiento del modelo.

Tiempo en PC Promedio	**Tiempo en G. Colab Promedio**
3.15 s	5.73 s

Tabla 4.2: Comparación promedio de tiempo de segmentación del conjunto de prueba sin afinamiento y después del afinamiento del modelo.

La arquitectura al usar una configuración transformer más ligera, permitió que el tiempo de segmentación sea poco debido al número de pesos reducido evitando el uso excesivo de recursos computacionales.

4.3. Métricas de comparación

El modelo de segmentación LiteMedSAM fue evaluado utilizando las métricas DSC y NSD, las cuales se emplean de manera conjunta para obtener una evaluación más integral de la calidad de la segmentación. El DSC es más adecuado para evaluar el volumen de la segmentación, mientras que el NSD resulta más apropiado para analizar la forma y la precisión de los bordes.

En la tabla 4.3 se muestra la comparación de las medianas de las métricas de antes y después de que se afina el modelo, al igual que se muestran los rangos intercuartiles.

	Antes		**Después**	
Métrica	**DSC**	**NSD**	**DSC**	**NSD**
Mediana	0.79	0.84	0.85	0.90

Tabla 4.3: Comparación de medianas de ambas métricas antes y después del Fine-tuning con rangos cuartiles.

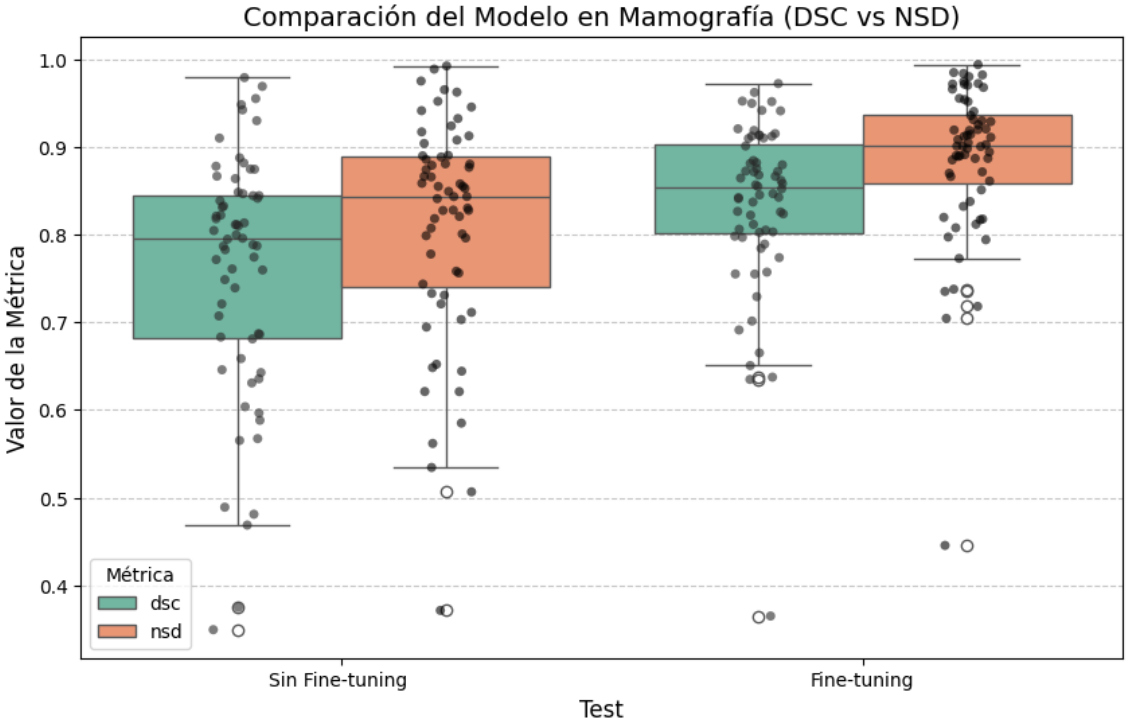

Figura 4.3: Distribución de las métricas de comparación a) DSC y b) NSD, para andes y después de la afinación del modelo

La comparación estadística de los resultados de estos coeficientes para antes y después de la afinación del modelo se pueden ver en la tabla 4.4, en donde se muestra la prueba no paramétrica de rangos con signo *Wilcoxon*.

En complemento a la comparación estadística de los resultados se presentan la figura 4.3, donde se muestran las inferencias realizadas. Se usa un *boxplot* (o diagrama de cajas y bigotes) ya que es una representación gráfica de la distribución de los conjunto de datos. Se puede visualizar la mediana, los cuartiles y los valores atípicos.

Métrica	Mediana de las diferencias	p-valor	$\sum R+$	$\sum R-$	Estadístico de efecto r
DSC	0.0419	$2.3492e-08$	1875	205	0.69
NSD	0.0406	$8.1838e-09$	1902	178	0.72

Tabla 4.4: Prueba de rangos con signo de *Wilcoxon* para las métricas DSC y NSD, con un p-valor < 0.05.

Posteriormente, el modelo LiteMedSAM se incorporó a la comparación con otros modelos usados para la segmentación de lesiones en mamarias [30][4][5], tal y como se puede observar en la la tabla 4.5

	Método	Conjunto de Datos	DSC Promedio
1	**Unet**[4]	INbreast	0.62
2	**Red de fusión**[4]	INbreast	0.62
3	**AUnet**[4]	INbreast	0.64
4	**SegNet**[5]	CBIS-DDSM	0.89
5	**ResNet**[5]	INbreast	0.87
6	**MedSAM**	CDD-CESM	0.86
7	**LiteMedSAM**	CDD-CESM	0.83

Tabla 4.5: Resultados de segmentación para diferentes modelos en la modalidad de mamografía.

Si se desea usar el modelo afinado de esta investigación pude consultar el siguiente repositorio en GitHub.

4.4. Discusión

LiteMedSAM es un modelo basado en aprendizaje profundo diseñado para la segmentación de una amplia variedad de estructuras anatómicas y lesiones en diversas modalidades de imágenes médicas. A diferencia de su predecesor, MedSAM, este modelo utiliza recursos más ligeros, lo que lo hace más eficiente. Gracias a su arquitectura de transferencia de aprendizaje, LiteMedSAM puede adaptarse a una modalidad específica sin requerir un conjunto de datos excesivamente grande. Al igual que su versión original, su configuración programable logra un equilibrio óptimo entre automatización y personalización[30], convirtiéndolo en una herramienta prometedora para la segmentación en mamografías.

La segmentación de imágenes mamográficas es una tarea compleja debido a diversas razones inherentes a la naturaleza de las imágenes y a las características del tejido mamario. Además, las lesiones mamarias suelen presentar bordes irregulares o poco definidos, lo que dificulta su segmentación precisa [51]. Esto resaltó la importancia de realizar el afinamiento, ya que mejoró significativamente la calidad de la segmentación.

Al realizar la prueba inicial de segmentación, se observa que la calidad de las imágenes y la fisiología de la mama representan un desafío para la segmentación. Es bien sabido que la obtención de mamografías de alta calidad depende de instrumentación actualizada, lo cual no siempre es posible debido a limitaciones en la infraestructura de los centros médicos.

El tiempo de segmentación puede variar dependiendo del tipo de imagen, además de la capacidad computacional en la que se realice la segmentación.

En la evaluación estadística se puede observar en la tabla 4.4, donde la prueba de rangos con signo de Wilcoxon ($T = 205$; $n = 64$; $p < 0.05$) indicó que el afinamiento produce una mejora

significativa. Además, la suma de los rangos de diferencia positiva fue mayor que la suma negativa, lo que demuestra un impacto positivo del afinamiento. También, (r) para las muestras de pares coincidentes fue de 0.69. Por lo tanto, el análisis proporciona evidencia de que el afinamiento de modelo está proporcionando mejoras en la segmentación de anomalías.

Como se observa una diferencia entre el modelo base y afinado, evidencia que el desempeño mejora cuando el modelo se especializa en un modalidad específica. Esto también demuestra que el modelo puede adaptarse a tareas particulares, ofreciendo así una mayor flexibilidad y adaptabilidad. Además, se puede apreciar que los cuadros delimitadores brindan un contexto espacial más claro y preciso para la región de interés, lo que permite al algoritmo identificar con mayor exactitud el área objetivo. Esto destaca frente a otros modelos de segmentación, como los mostrados en la tabla 4.5, que utilizan diferentes tipos de arquitecturas y procesos de segmentación.

Por otro lado, el uso de cuadros delimitadores resulta eficiente, especialmente en escenarios que involucran la segmentación de múltiples objetos, lo que agiliza el proceso en comparación con métodos más complejos.

Capítulo 5

Conclusiones

5.1. Conclusiones

La evaluación de LiteMedSAM ha demostrado su eficacia en la segmentación de una modalidad compleja como la mamografía. Su rendimiento refuerza la validez del modelo original, MedSAM, ya que, a pesar de ser una versión más ligera, no compromete sus capacidades. Los resultados obtenidos muestran que el modelo alcanza una buena precisión en el Coeficiente de Similitud de Dice (DSC) y la Diferencia de Superficie Normalizada (NSD), compitiendo contra otros modelos de segmentación, que utilizan diferentes tipos de arquitecturas y procesos de segmentación. Esto se debe a su capacidad para proporcionar una delineación precisa de estructuras anatómicas, anomalías mamarias, la implementación de cajas delimitadoras y esencialmente a la transferencia de aprendizaje.

En análisis de los resultados se observó que se adaptó a las imágenes obtenidas digitalmente con contraste dando una buena segmentación.

Dado que LiteMedSAM ha aprendido características de imágenes de buena calidad y representativas del conjunto de entrenamiento a gran escala del modelo original como imágenes de tomografía computarizada (TC), resonancia magnética (RM) y endoscopia, y se ha optimizado en una arquitectura más ligera, los requerimientos computacionales se han reducido significativamente. Esto permite su implementación en plataformas virtuales como *Google Colab* y en una computadora portátil.

El modelo cuenta con transferencia de aprendizaje, que parte desde el modelo base SAM, seguido de MedSAM [30] y posteriormente LiteMedSAM. El afinamiento resultó favorable para mejorar el modelo para el área de mamografías, las métricas de comparación mostraron que las segmentaciones de las lesiones mejoraron después del entrenamiento.

Esta investigación resalta la viabilidad de un modelo más ligero capaz de manejar segmentaciones en la modalidad de mamografías. LiteMedSAM, como modelo inicial en la segmentación de imágenes médicas, tiene un gran potencial para acelerar el desarrollo de nuevas herramientas diagnósticas. Aunque aún existen aspectos por mejorar, es una opción prometedora, dada su fácil adaptación y utilidad para considerarlo para entornos clínicos.

5.2. Limitaciones

A pesar de sus sólidas capacidades, LiteMedSAM presenta ciertas limitaciones. Una de ellas son las cajas delimitadoras, ya que estas se tiene que generar por parte del usuario, y puede ser difícil para aquéllos que no son médicos, por esa razón es importante que los conjuntos de datos contengan las máscara de referencia, ya que por medio de programación se pueden generar.

La anatomía mamaria genera otra limitación en la segmentación, aunque esto podría mitigarse con una mejor técnica de obtención de las imágenes médicas, como el caso de las imágenes generadas por contraste.

Aunque el modelo generó buenas estadísticas de evaluación, puede llegar a limitarse por el tipo de imagen que se segmenta. Para esta investigación, las etiquetas asignadas a las anomalías en escala de grises generó algunas complicaciones, si el valor de las etiquetas de la máscara de referencia varía con respecto a la segmentación generada, la validación puede resultar afectada, por lo que limita un correcto análisis del modelo.

5.3. Trabajo futuro

Si bien, este modelo tiene una intervención por parte del usuario como lo son los cuadros delimitadores, el poder adaptar otro software complementario sería de gran ayuda para el médico especialista, ya que reduciría la carga de trabajo significativamente, además de ser útil como instrumento de apoyo en capacitaciones para nuevos médicos en el área diagnóstica.

Considerar un ajuste usando conjuntos de datos más grandes aumentaría su efectividad, al igual si estos datos están enriquecidos con más casos de mamas malignas, benignas y normales, para que la red neuronal identifique con mayor claridad cada tipo de caso.

Finalmente, esta investigación reafirma la importancia del uso de la inteligencia artificial en el apoyo al diagnóstico mamográfico y destaca la necesidad de continuar con investigaciones en este campo para lograr una integración efectiva de estos modelos en entornos clínicos reales.

Apéndice A

Apéndice

A.1. Código de Inferencia CVPR24

```
from os import listdir, makedirs
from os.path import join, isfile, basename
from glob import glob
from tqdm import tqdm
from time import time
import numpy as np
import torch
import torch.nn as nn
import torch.nn.functional as F
from segment_anything.modeling import MaskDecoder, PromptEncoder,
    ↪ TwoWayTransformer
from tiny_vit_sam import TinyViT
from matplotlib import pyplot as plt
import cv2
import argparse
from collections import OrderedDict
import pandas as pd
from datetime import datetime
```

```
#%% set seeds
torch.set_float32_matmul_precision('high')
torch.manual_seed(2024)
torch.cuda.manual_seed(2024)
np.random.seed(2024)

parser = argparse.ArgumentParser()

parser.add_argument(
    '-i',
    '--input_dir',
    type=str,
    default='test_demo/imgs/',
    # required=True,
    help='root directory of the data', )
parser.add_argument(
    '-o',
    '--output_dir',
```

```
    type=str,
    default='test_demo/segs/',
    help='directory to save the prediction', )
parser.add_argument(
    '-lite_medsam_checkpoint_path',
    type=str,
    default="/content/drive/MyDrive/liteMedSAM/work_dir/medsam_lite_best_1.
        ↪ pth" ,
    help='path to the checkpoint of MedSAM-Lite', )
parser.add_argument(
    '-device',
    type=str,
    default="cpu",
    help='device to run the inference', )
parser.add_argument(
    '-num_workers',
    type=int,
    default=4,
    help='number of workers for inference with multiprocessing', )
parser.add_argument(
    '--save_overlay',
    default=True,
    action='store_true',
    help='whether to save the overlay image' )
parser.add_argument(
    '-png_save_dir',
    type=str,
    default="/content/drive/MyDrive/liteMedSAM/data/inf_CBIS/overlay",
    help='directory to save the overlay image' )

args = parser.parse_args()

data_root = args.input_dir
pred_save_dir = args.output_dir
save_overlay = args.save_overlay
num_workers = args.num_workers
if save_overlay:
    assert args.png_save_dir is not None, "Please specify the directory to
        ↪ save the overlay image"
    png_save_dir = args.png_save_dir
    makedirs(png_save_dir, exist_ok=True)

lite_medsam_checkpoint_path = args.lite_medsam_checkpoint_path
makedirs(pred_save_dir, exist_ok=True)
device = torch.device(args.device)
image_size = 256

def resize_longest_side(image, target_length=256):
    """
    Resize image to target_length while keeping the aspect ratio
    Expects a numpy array with shape HxWxC in uint8 format.
    """
    oldh, oldw = image.shape[0], image.shape[1]
    scale = target_length * 1.0 / max(oldh, oldw)
    newh, neww = oldh * scale, oldw * scale
    neww, newh = int(neww + 0.5), int(newh + 0.5)
```

```
    target_size = (neww, newh)

    return cv2.resize(image, target_size, interpolation=cv2.INTER_AREA)

def pad_image(image, target_size=256):
    """
    Pad image to target_size
    Expects a numpy array with shape HxWxC in uint8 format.
    """
    # Pad
    h, w = image.shape[0], image.shape[1]
    padh = target_size - h
    padw = target_size - w
    if len(image.shape) == 3: ## Pad image
        image_padded = np.pad(image, ((0, padh), (0, padw), (0, 0)))
    else: ## Pad gt mask
        image_padded = np.pad(image, ((0, padh), (0, padw)))

    return image_padded

class MedSAM_Lite(nn.Module):
    def __init__(
            self,
            image_encoder,
            mask_decoder,
            prompt_encoder
        ):
        super().__init__()
        self.image_encoder = image_encoder
        self.mask_decoder = mask_decoder
        self.prompt_encoder = prompt_encoder

    def forward(self, image, box_np):
        image_embedding = self.image_encoder(image) # (B, 256, 64, 64)
        # do not compute gradients for prompt encoder
        with torch.no_grad():
            box_torch = torch.as_tensor(box_np, dtype=torch.float32, device
                ↪ =image.device)
            if len(box_torch.shape) == 2:
                box_torch = box_torch[:, None, :] # (B, 1, 4)

        sparse_embeddings, dense_embeddings = self.prompt_encoder(
            points=None,
            boxes=box_np,
            masks=None,
        )
        low_res_masks, iou_predictions = self.mask_decoder(
            image_embeddings=image_embedding, # (B, 256, 64, 64)
            image_pe=self.prompt_encoder.get_dense_pe(), # (1, 256, 64, 64)
            sparse_prompt_embeddings=sparse_embeddings, # (B, 2, 256)
            dense_prompt_embeddings=dense_embeddings, # (B, 256, 64, 64)
            multimask_output=False,
          ) # (B, 1, 256, 256)

        return low_res_masks
```

```
    @torch.no_grad()
    def postprocess_masks(self, masks, new_size, original_size):
        """
        Do cropping and resizing

        Parameters
        ----------
        masks : torch.Tensor
            masks predicted by the model
        new_size : tuple
            the shape of the image after resizing to the longest side of
                ↪ 256
        original_size : tuple
            the original shape of the image

        Returns
        -------
        torch.Tensor
            the upsampled mask to the original size
        """
        # Crop
        masks = masks[..., :new_size[0], :new_size[1]]
        # Resize
        masks = F.interpolate(
            masks,
            size=(original_size[0], original_size[1]),
            mode="bilinear",
            align_corners=False,
        )

        return masks


def show_mask(mask, ax, mask_color=None, alpha=0.5):
    """
    show mask on the image

    Parameters
    ----------
    mask : numpy.ndarray
        mask of the image
    ax : matplotlib.axes.Axes
        axes to plot the mask
    mask_color : numpy.ndarray
        color of the mask
    alpha : float
        transparency of the mask
    """
    if mask_color is not None:
        color = np.concatenate([mask_color, np.array([alpha])], axis=0)
    else:
        color = np.array([251/255, 252/255, 30/255, alpha])
    h, w = mask.shape[-2:]
    mask_image = mask.reshape(h, w, 1) * color.reshape(1, 1, -1)
    ax.imshow(mask_image)
```

```
def show_box(box, ax, edgecolor='blue'):
    """
    show bounding box on the image

    Parameters
    ----------
    box : numpy.ndarray
        bounding box coordinates in the original image
    ax : matplotlib.axes.Axes
        axes to plot the bounding box
    edgecolor : str
        color of the bounding box
    """
    x0, y0 = box[0], box[1]
    w, h = box[2] - box[0], box[3] - box[1]
    ax.add_patch(plt.Rectangle((x0, y0), w, h, edgecolor=edgecolor,
        ↪ facecolor=(0,0,0,0), lw=2))


def get_bbox256(mask_256, bbox_shift=3):
    """
    Get the bounding box coordinates from the mask (256x256)

    Parameters
    ----------
    mask_256 : numpy.ndarray
        the mask of the resized image

    bbox_shift : int
        Add perturbation to the bounding box coordinates

    Returns
    -------
    numpy.ndarray
        bounding box coordinates in the resized image
    """
    y_indices, x_indices = np.where(mask_256 > 0)
    x_min, x_max = np.min(x_indices), np.max(x_indices)
    y_min, y_max = np.min(y_indices), np.max(y_indices)
    # add perturbation to bounding box coordinates and test the robustness
    # this can be removed if you do not want to test the robustness
    H, W = mask_256.shape
    x_min = max(0, x_min - bbox_shift)
    x_max = min(W, x_max + bbox_shift)
    y_min = max(0, y_min - bbox_shift)
    y_max = min(H, y_max + bbox_shift)

    bboxes256 = np.array([x_min, y_min, x_max, y_max])

    return bboxes256

def resize_box_to_256(box, original_size):
    """
    the input bounding box is obtained from the original image
    here, we rescale it to the coordinates of the resized image
```

```
    Parameters
    ----------
    box : numpy.ndarray
        bounding box coordinates in the original image
    original_size : tuple
        the original size of the image

    Returns
    -------
    numpy.ndarray
        bounding box coordinates in the resized image
    """
    new_box = np.zeros_like(box)
    ratio = 256 / max(original_size)
    for i in range(len(box)):
        new_box[i] = int(box[i] * ratio)

    return new_box


@torch.no_grad()
def medsam_inference(medsam_model, img_embed, box_256, new_size,
    ↪ original_size):
    """
    Perform inference using the LiteMedSAM model.

    Args:
        medsam_model (MedSAMModel): The MedSAM model.
        img_embed (torch.Tensor): The image embeddings.
        box_256 (numpy.ndarray): The bounding box coordinates.
        new_size (tuple): The new size of the image.
        original_size (tuple): The original size of the image.
    Returns:
        tuple: A tuple containing the segmented image and the intersection
            ↪ over union (IoU) score.
    """
    box_torch = torch.as_tensor(box_256[None, None, ...], dtype=torch.float
        ↪ , device=img_embed.device)

    sparse_embeddings, dense_embeddings = medsam_model.prompt_encoder(
        points = None,
        boxes = box_torch,
        masks = None,
    )
    low_res_logits, iou = medsam_model.mask_decoder(
        image_embeddings=img_embed, # (B, 256, 64, 64)
        image_pe=medsam_model.prompt_encoder.get_dense_pe(), # (1, 256, 64,
            ↪  64)
        sparse_prompt_embeddings=sparse_embeddings, # (B, 2, 256)
        dense_prompt_embeddings=dense_embeddings, # (B, 256, 64, 64)
        multimask_output=False
    )

    low_res_pred = medsam_model.postprocess_masks(low_res_logits, new_size,
        ↪  original_size)
```

```
    low_res_pred = torch.sigmoid(low_res_pred)
    low_res_pred = low_res_pred.squeeze().cpu().numpy()
    medsam_seg = (low_res_pred > 0.5).astype(np.uint8)

    return medsam_seg, iou

medsam_lite_image_encoder = TinyViT(
    img_size=256,
    in_chans=3,
    embed_dims=[
        64, ## (64, 256, 256)
        128, ## (128, 128, 128)
        160, ## (160, 64, 64)
        320 ## (320, 64, 64)
    ],
    depths=[2, 2, 6, 2],
    num_heads=[2, 4, 5, 10],
    window_sizes=[7, 7, 14, 7],
    mlp_ratio=4.,
    drop_rate=0.,
    drop_path_rate=0.0,
    use_checkpoint=False,
    mbconv_expand_ratio=4.0,
    local_conv_size=3,
    layer_lr_decay=0.8
)

medsam_lite_prompt_encoder = PromptEncoder(
    embed_dim=256,
    image_embedding_size=(64, 64),
    input_image_size=(256, 256),
    mask_in_chans=16
)

medsam_lite_mask_decoder = MaskDecoder(
    num_multimask_outputs=3,
        transformer=TwoWayTransformer(
            depth=2,
            embedding_dim=256,
            mlp_dim=2048,
            num_heads=8,
        ),
        transformer_dim=256,
        iou_head_depth=3,
        iou_head_hidden_dim=256,
)

medsam_lite_model = MedSAM_Lite(
    image_encoder = medsam_lite_image_encoder,
    mask_decoder = medsam_lite_mask_decoder,
    prompt_encoder = medsam_lite_prompt_encoder
)

lite_medsam_checkpoint = torch.load(lite_medsam_checkpoint_path,
    ↪ map_location='cuda',weights_only=True) #------cambios
medsam_lite_model.load_state_dict(lite_medsam_checkpoint)
```

```
medsam_lite_model.to(device)
medsam_lite_model.eval()

def MedSAM_infer_npz_2D(img_npz_file):
    npz_name = basename(img_npz_file)
    npz_data = np.load(img_npz_file, 'r', allow_pickle=True) # (H, W, 3)
    img_3c = npz_data['imgs'] # (H, W, 3)
    assert np.max(img_3c)<256, f'input data should be in range [0, 255],
        ↪ but got {np.unique(img_3c)}'
    H, W = img_3c.shape[:2]
    boxes = npz_data['boxes']
    segs = np.zeros(img_3c.shape[:2], dtype=np.uint8)

    ## preprocessing
    img_256 = resize_longest_side(img_3c, 256)
    newh, neww = img_256.shape[:2]
    img_256_norm = (img_256 - img_256.min()) / np.clip(
        img_256.max() - img_256.min(), a_min=1e-8, a_max=None
    )
    img_256_padded = pad_image(img_256_norm, 256)
    img_256_tensor = torch.tensor(img_256_padded).float().permute(2, 0, 1).
        ↪ unsqueeze(0).to(device)
    with torch.no_grad():
        image_embedding = medsam_lite_model.image_encoder(img_256_tensor)

    for idx, box in enumerate(boxes, start=1):
        box256 = resize_box_to_256(box, original_size=(H, W))
        box256 = box256[None, ...] # (1, 4)
        medsam_mask, iou_pred = medsam_inference(medsam_lite_model,
            ↪ image_embedding, box256, (newh, neww), (H, W))
        segs[medsam_mask>0] = idx
        # print(f'{npz_name}, box: {box}, predicted iou: {np.round(iou_pred
            ↪ .item(), 4)}')

    np.savez_compressed(
        join(pred_save_dir, npz_name),
        segs=segs,
    )

    # visualize image, mask and bounding box
    if save_overlay:
        fig, ax = plt.subplots(1, 2, figsize=(10, 5))
        ax[0].imshow(img_3c)
        ax[1].imshow(img_3c)
        ax[0].set_title("Image")
        ax[1].set_title("LiteMedSAM Segmentation")
        ax[0].axis('off')
        ax[1].axis('off')

        for i, box in enumerate(boxes):
            color = np.random.rand(3)
            box_viz = box
            show_box(box_viz, ax[1], edgecolor=color)
            show_mask((segs == i+1).astype(np.uint8), ax[1], mask_color=
                ↪ color)
```

```
        plt.tight_layout()
        plt.savefig(join(png_save_dir, npz_name.split(".")[0] + '.png'),
            ↪ dpi=300)
        plt.close()


def MedSAM_infer_npz_3D(img_npz_file):
    npz_name = basename(img_npz_file)
    npz_data = np.load(img_npz_file, 'r', allow_pickle=True)
    img_3D = npz_data['imgs'] # (D, H, W)
    spacing = npz_data['spacing'] # not used in this demo because it treats
        ↪  each slice independently
    segs = np.zeros_like(img_3D, dtype=np.uint8)
    boxes_3D = npz_data['boxes'] # [[x_min, y_min, z_min, x_max, y_max,
        ↪ z_max]]

    for idx, box3D in enumerate(boxes_3D, start=1):
        segs_3d_temp = np.zeros_like(img_3D, dtype=np.uint8)
        x_min, y_min, z_min, x_max, y_max, z_max = box3D
        assert z_min < z_max, f"z_min should be smaller than z_max, but got
            ↪  {z_min=} and {z_max=}"
        mid_slice_bbox_2d = np.array([x_min, y_min, x_max, y_max])
        z_middle = int((z_max - z_min)/2 + z_min)

        # infer from middle slice to the z_max
        # print(npz_name, 'infer from middle slice to the z_max')
        z_max = min(z_max+1, img_3D.shape[0])
        for z in range(z_middle, z_max):
            img_2d = img_3D[z, :, :]
            if len(img_2d.shape) == 2:
                img_3c = np.repeat(img_2d[:, :, None], 3, axis=-1)
            else:
                img_3c = img_2d
            H, W, _ = img_3c.shape

            img_256 = resize_longest_side(img_3c, 256)
            new_H, new_W = img_256.shape[:2]

            img_256 = (img_256 - img_256.min()) / np.clip(
                img_256.max() - img_256.min(), a_min=1e-8, a_max=None
            )  # normalize to [0, 1], (H, W, 3)
            ## Pad image to 256x256
            img_256 = pad_image(img_256)

            # convert the shape to (3, H, W)
            img_256_tensor = torch.tensor(img_256).float().permute(2, 0, 1)
                ↪ .unsqueeze(0).to(device)
            # get the image embedding
            with torch.no_grad():
                image_embedding = medsam_lite_model.image_encoder(
                    ↪ img_256_tensor) # (1, 256, 64, 64)
            if z == z_middle:
                box_256 = resize_box_to_256(mid_slice_bbox_2d,
                    ↪ original_size=(H, W))
            else:
                pre_seg = segs_3d_temp[z-1, :, :]
```

```
                pre_seg256 = resize_longest_side(pre_seg)
                if np.max(pre_seg256) > 0:
                    pre_seg256 = pad_image(pre_seg256)
                    box_256 = get_bbox256(pre_seg256)
                else:
                    box_256 = resize_box_to_256(mid_slice_bbox_2d,
                        ↪ original_size=(H, W))
            img_2d_seg, iou_pred = medsam_inference(medsam_lite_model,
                ↪ image_embedding, box_256, [new_H, new_W], [H, W])
            segs_3d_temp[z, img_2d_seg>0] = idx

        # infer from middle slice to the z_max
        # print(npz_name, 'infer from middle slice to the z_min')
        z_min = max(-1, z_min-1)
        for z in range(z_middle-1, z_min, -1):
            img_2d = img_3D[z, :, :]
            if len(img_2d.shape) == 2:
                img_3c = np.repeat(img_2d[:, :, None], 3, axis=-1)
            else:
                img_3c = img_2d
            H, W, _ = img_3c.shape

            img_256 = resize_longest_side(img_3c)
            new_H, new_W = img_256.shape[:2]

            img_256 = (img_256 - img_256.min()) / np.clip(
                img_256.max() - img_256.min(), a_min=1e-8, a_max=None
            )  # normalize to [0, 1], (H, W, 3)
            ## Pad image to 256x256
            img_256 = pad_image(img_256)

            img_256_tensor = torch.tensor(img_256).float().permute(2, 0, 1)
                ↪ .unsqueeze(0).to(device)
            # get the image embedding
            with torch.no_grad():
                image_embedding = medsam_lite_model.image_encoder(
                    ↪ img_256_tensor) # (1, 256, 64, 64)

            pre_seg = segs_3d_temp[z+1, :, :]
            pre_seg256 = resize_longest_side(pre_seg)
            if np.max(pre_seg256) > 0:
                pre_seg256 = pad_image(pre_seg256)
                box_256 = get_bbox256(pre_seg256)
            else:
                box_256 = resize_box_to_256(mid_slice_bbox_2d,
                    ↪ original_size=(H, W))
            img_2d_seg, iou_pred = medsam_inference(medsam_lite_model,
                ↪ image_embedding, box_256, [new_H, new_W], [H, W])
            segs_3d_temp[z, img_2d_seg>0] = idx
        segs[segs_3d_temp>0] = idx
    np.savez_compressed(
        join(pred_save_dir, npz_name),
        segs=segs,
    )

    # visualize image, mask and bounding box
```

```
    if save_overlay:
        idx = int(segs.shape[0] / 2)
        fig, ax = plt.subplots(1, 2, figsize=(10, 5))
        ax[0].imshow(img_3D[idx], cmap='gray')
        ax[1].imshow(img_3D[idx], cmap='gray')
        ax[0].set_title("Image")
        ax[1].set_title("LiteMedSAM Segmentation")
        ax[0].axis('off')
        ax[1].axis('off')

        for i, box3D in enumerate(boxes_3D, start=1):
            if np.sum(segs[idx]==i) > 0:
                color = np.random.rand(3)
                x_min, y_min, z_min, x_max, y_max, z_max = box3D
                box_viz = np.array([x_min, y_min, x_max, y_max])
                show_box(box_viz, ax[1], edgecolor=color)
                show_mask(segs[idx]==i, ax[1], mask_color=color)

        plt.tight_layout()
        plt.savefig(join(png_save_dir, npz_name.split(".")[0] + '.png'),
            ↪ dpi=300)
        plt.close()


if __name__ == '__main__':
    img_npz_files = sorted(glob(join(data_root, '*.npz'), recursive=True))
    efficiency = OrderedDict()
    efficiency['case'] = []
    efficiency['time'] = []
    for img_npz_file in tqdm(img_npz_files):
        start_time = time()
        if basename(img_npz_file).startswith('3D'):
            MedSAM_infer_npz_3D(img_npz_file)
        else:
            MedSAM_infer_npz_2D(img_npz_file)
        end_time = time()
        efficiency['case'].append(basename(img_npz_file))
        efficiency['time'].append(end_time - start_time)
        current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
        print(current_time, 'file name:', basename(img_npz_file), 'time
            ↪ cost:', np.round(end_time - start_time, 4))
    efficiency_df = pd.DataFrame(efficiency)
    efficiency_df.to_csv(join(pred_save_dir, 'efficiency.csv'), index=False
        ↪ )
```

A.2. Código Evaluación de Métricas

```
# %% move files based on csv file
import numpy as np
import nibabel as nb
import os
from os import listdir, makedirs
from os.path import basename, join, dirname, isfile, isdir
from collections import OrderedDict
import pandas as pd
from SurfaceDice import compute_surface_distances,
    ↪ compute_surface_dice_at_tolerance, compute_dice_coefficient
from tqdm import tqdm
import multiprocessing as mp
import argparse
```

```
def compute_multi_class_dsc(gt, seg):
    dsc = []
    for i in range(1, gt.max()+1):
        gt_i = gt == i
        seg_i = seg == i
        dsc.append(compute_dice_coefficient(gt_i, seg_i))
    return np.mean(dsc)

def compute_multi_class_nsd(gt, seg, spacing, tolerance=2.0):
    nsd = []
    for i in range(1, gt.max()+1):
        gt_i = gt == i
        seg_i = seg == i
        surface_distance = compute_surface_distances(
            gt_i, seg_i, spacing_mm=spacing
        )
        nsd.append(compute_surface_dice_at_tolerance(surface_distance,
            ↪ tolerance))
    return np.mean(nsd)


parser = argparse.ArgumentParser()
parser.add_argument('-s', '--seg_dir', default='test_demo/segs', type=str)
parser.add_argument('-g', '--gt_dir', default='test_demo/gts', type=str)
parser.add_argument('-csv_dir', default='test_demo/metrics.csv', type=str)
parser.add_argument('-num_workers', type=int, default=5)
parser.add_argument('-nsd', default=True, type=bool, help='set it to False
    ↪ to disable NSD computation and save time')
args = parser.parse_args()

seg_dir = args.seg_dir
gt_dir = args.gt_dir
csv_dir = args.csv_dir
num_workers = args.num_workers
compute_NSD = args.nsd

def compute_metrics(npz_name):
    metric_dict = {'dsc': -1.}
    if compute_NSD:
        metric_dict['nsd'] = -1.
```

```
    npz_seg = np.load(join(seg_dir, npz_name), allow_pickle=True, mmap_mode
        ↪ ='r')
    npz_gt = np.load(join(gt_dir, npz_name), allow_pickle=True, mmap_mode='
        ↪ r')
    gts = npz_gt['gts']
    segs = npz_seg['segs']
    if npz_name.startswith('3D'):
        spacing = npz_gt['spacing']

    dsc = compute_multi_class_dsc(gts, segs)
    # comupute nsd
    if compute_NSD:
        if dsc > 0.2:
        # only compute nsd when dice > 0.2 because NSD is also low when
            ↪ dice is too low
            if npz_name.startswith('3D'):
                nsd = compute_multi_class_nsd(gts, segs, spacing)
            else:
                spacing = [1.0, 1.0, 1.0]
                nsd = compute_multi_class_nsd(np.expand_dims(gts, -1), np.
                    ↪ expand_dims(segs, -1), spacing)
        else:
            nsd = 0.0
    return npz_name, dsc, nsd

if __name__ == '__main__':
    seg_metrics = OrderedDict()
    seg_metrics['case'] = []
    seg_metrics['dsc'] = []
    if compute_NSD:
        seg_metrics['nsd'] = []

    npz_names = listdir(gt_dir)
    npz_names = [npz_name for npz_name in npz_names if npz_name.endswith('.
        ↪ npz')]
    with mp.Pool(num_workers) as pool:
        with tqdm(total=len(npz_names)) as pbar:
            for i, (npz_name, dsc, nsd) in enumerate(pool.imap_unordered(
                ↪ compute_metrics, npz_names)):
                seg_metrics['case'].append(npz_name)
                seg_metrics['dsc'].append(np.round(dsc, 4))
                if compute_NSD:
                    seg_metrics['nsd'].append(np.round(nsd, 4))
                pbar.update()
    df = pd.DataFrame(seg_metrics)
    # rank based on case column
    df = df.sort_values(by=['case'])
    df.to_csv(csv_dir, index=False)
```

Bibliografía

[1] Instituto Nacional de Estadística y Geografía (INEGI). Estadísticas a propósito del día internacional de la lucha contra el cáncer de mama (19 de octubre). https://www.inegi.org.mx/contenidos/saladeprensa/aproposito/2024/EAP_LuchaCMama24.pdf, 2024. Accedido: 4 de enero de 2025.

[2] Sistema Nacional de Información en Salud. Banco de indicadores. https://www.inegi.org.mx/app/indicadores/?t=123&ag=00#D123, 2020. Accedido: 10 de enero de 2025.

[3] Instituto Nacional de Estadística y Geografía (INEGI). Recursos públicos disponibles para la atención en salud. https://sinaiscap.salud.gob.mx/DGIS/#, 2023. Accedido: 10 de enero de 2025.

[4] R. Oza P. Oza, U. Oza. Digital mammography dataset for breast cancer diagnosis research (dmid) with breast mass segmentation analysis. *Biomedical Engineering Letters*, 14:317–330, 2024. https://doi.org/10.1007/s13534-023-00339-y.

[5] A. Lalas D. Manolakis, P. Bizopoulos. A two-stage lightweight deep learning framework for mass detection and segmentation in mammograms using yolov5 and depthwise segnet. *Journal of Imaging Informatics in Medicine*, 2025. https://doi.org/10.1007/s10278-025-01471-0.

[6] Richard Szeliski. *Computer Vision Algorithms and Applications*. Springer Cham, 2 edition, 2022.

[7] Freddie Bray, Mathieu Laversanne, Hyuna Sung, Jacques Ferlay, Rebecca L. Siegel, Isabelle Soerjomataram, and Ahmedin Jemal. Global cancer statistics 2022: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer J Clinic*, 74(3):229–263, 2024. doi:10.3322/caac.21834 ,https://doi.org/10.3322/caac.21834.

[8] K. Farrell, D. L. Bennett, and T. L. Schwartz. Screening for breast cancer: What you need to know. *Missouri medicine*, 117(2):133–135, 2020. PMC7144719 PMID: 32308238, https://pmc.ncbi.nlm.nih.gov/articles/PMC7144719/.

[9] Timothy de Moor, Alejandro Rodriguez-Ruiz, Albert Gubern Mérida, Ritse Mann, and Jonas Teuwen. Automated soft tissue lesion detection and segmentation in digital mammography using a u-net deep learning network. *CoRR*, abs/1802.06865, 2018. https://arxiv.org/abs/1802.06865.

[10] Karam A. *Diagnóstico y tratamiento ginecoobstétricos, Cap 'Mama'*. McGraw Hill Education, New York, NY, 12 edition, 2021.

[11] Steward C. Bushong. *Manual de Radiología para Técnicos: Física, Biología y Protección Radiológica*. Elsevier España, S.L.U., Barcelona, España, 9 edition, 2010.

[12] American Cancer Society. Conceptos básicos del mamograma. https://www.cancer.org/es/cancer/tipos/cancer-de-seno/pruebas-de-deteccion-y-deteccion-temprana-del-cancer-de-seno/mamogramas/conceptos-basicos-del-mamograma.html#:~:text=La%20dosis%20de%20radiaci%C3%B3n%20que,de%20alrededor%20de%207%20semanas., 2020. Accedido: 21 de enero de 2025.

[13] Radiological Society of North America Inc. (RSNA). Dosis de radiación. https://www.radiologyinfo.org/es/info/safety-xray, 2011. Accedido: 28 de Febrero de 2025.

[14] National Cancer Institute. Mamografías. https://www.cancer.gov/espanol/tipos/seno/hoja-informativa-mamografias, 2025. Accedido: 20 de enero de 2025.

[15] Keppke AL Magny SJ, Shikhman R. Breast imaging reporting and data system. https://www.ncbi.nlm.nih.gov/books/NBK459169/, 2023. Accedido: 21 de enero de 2025.

[16] N. Smith and A. Webb. *Introduction to Medical Imaging: Physics, Engineering and Clinical Applications*. Cambridge University Press, Cambridge CB2 8RU, UK, 1 edition, 2011.

[17] Helvie MA. Digital mammography imaging: breast tomosynthesis and advanced applications. *Radiol Clin North Am*, 48(5):917–929, 2010. PMID: 20868894; PMCID: PMC3118307. 10.1016/j.rcl.2010.06.009.

[18] C. Parker C & Damodaran S & Bland K.I. & Hunt K.K. *Schwartz's Principles of Surgery, Cap 'The breasts'*. McGraw Hill Education, New York, NY, 11 edition, 2019.

[19] Faiz M. Khan. *The Physics of Radiation Therapy*. WOLTERS KLUWER, Philadelphia, 5 edition, 2014.

[20] R. Gonzalez and R Woods. *Digital Image Processing*. Pearson Education Limited, 330 Hudson Street, New York, 4 edition, 2018.

[21] Phil Kim. *MATLAB Deep Learning: With Machine Learning, Neural Networks and Artificial Intelligence*. Adress, Seoul, Soul-t'ukpyolsi, Korea, 2017.

[22] Raúl E. López Briega. *Introducción al Deep Learning*. IAARBook, 2017.

[23] Christopher M. Bishop and Hugh Bishop. *Deep Learning Foundations and Concepts*. Springer, 2024.

[24] International Business Machines Corporation (IBM). ¿qué es una red neuronal? https://www.ibm.com/mx-es/topics/neural-networks, 2020. Accedido: 28 de Febrero de 2025.

[25] Giovanni Cerulli. *Deep Learning, In: Fundamentals of Supervised Machine Learning. Statistics and Computing*. Springer International Publishing, Cham, 2023.

[26] Giovanni Cerulli. *Fundamentals of Supervised Machine Learning With Applications in Python, R, and Stata*. Springe, 2023.

[27] P. Sharma, R. Begum, P. Shrivastava, and N. Sharma. Machine learning. *Artificial Intelligence and their Applications*, (3):pp. 138–150, 2024. e-ISBN: 9789362527172 https://doi.org/10.58532/nbennurch61.

[28] K. Weiss, T.M. Khoshgoftaar, and D Wang. A survey of transfer learning. *Journal of Big Data*, 3:9, 2016. https://doi.org/10.1186/s40537-016-0043-6.

[29] Ana Davila, Jacinto Colan, and Yasuhisa Hasegawa. Comparison of fine-tuning strategies for transfer learning in medical image classification. *Image and Vision Computing*, 146, 2024. https://doi.org/10.1016/j.imavis.2024.105012.

[30] J. Ma, Y. He, and F. et al Li. Segment anything in medical images. *Nature Communications*, 15 (654), 2024. ISSN 2041-1723 (online) https://doi.org/10.1038/s41467-024-44824-z.

[31] A. Vaswani, N. Shazeer, J. Parmar, N. anda Uszkoreit, L. Jones, A. N Gomez, L. Kaiser, and I. Polosukhin. Attention is all you need. *CoRR*, abs/1706.03762, 2017. http://arxiv.org/abs/1706.03762.

[32] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, and N. et al. Unterthiner, T. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *CoRR*, abs/2010.11929, 2020. https://arxiv.org/abs/2010.11929.

[33] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C Berg, W. Lo, P. Dollár, and R.B. Girshick. Segment anything. *arXiv.cs.CV*, arXiv:2304.02643, 2023. https://arxiv.org/abs/2304.02643.

[34] Ana Davila, Jacinto Colan, and Yasuhisa Hasegawa. Interactive medical image segmentation using deep learning with image-specific fine-tuning. *IEEE Transactions on Medical Imaging*, 74, 2018. http://dx.doi.org/10.1109/TMI.2018.2791721.

[35] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016.

[36] Zhongzhi Shi. *Intelligence Science Leading the Age of Intelligence*. Elsevier, 2021.

[37] Jun Ma. Cvpr 2024: Segment anything in medical images on laptop. https://www.codabench.org/competitions/1847/, 2024. Accedido: 5 de febreo de 2025.

[38] Z. Chaoning, H. Dongshen, Q. Yu, K. Jung, B. Sung-Ho, L. Seungkyu, and H.C. Seon. Faster segment anything: Towards lightweight sam for mobile applications. *arXiv.cs.CV*, 2023. https://arxiv.org/pdf/2207.10666.

[39] K. Wu, J. Zhang, H. Peng, M. Liu, B. Xiao, J. Fu, and L. Yuan. Tinyvit: Fast pretraining distillation for small vision transformers. *arXiv.cs.CV*, arXiv:2207.10666v1, 2022. https://arxiv.org/pdf/2207.10666.

[40] L. Maier-Hein, A. Reinke, and P. et a. Godau. Metrics reloaded: recommendations for image analysis validation. *Nature Methods*, 21, 2024. http://dx.doi.org/10.1038/s41592-023-02151-z.

[41] Yizhi Pan, Junyi Xin, Tianhua Yang, Siqi Li, Le-Minh Nguyen, Teeradaj Racharak, Kai Li, and Guanqun Sun. A mutual inclusion mechanism for precise boundary segmentation in medical images. *Frontiers in Bioengineering and Biotechnology*, 12, 2024. https://www.frontiersin.org/journals/bioengineering-and-biotechnology/articles/10.3389/fbioe.2024.1504249.

[42] Deepmind surface-distance. https://github.com/google-deepmind/surface-distance, 2018. Accedido: 20 de enero de 2025.

[43] U. C. Lalji, C. R. Jeukens, I. Houben, P. J. Nelemans, R. E. van Engen, E. van Wylick, R. G. Beets-Tan, J. E. Wildberger, L. E. Paulis, and M. B. Lobbes. Evaluation of low-energy contrast-enhanced spectral mammography images by comparing them to full-field digital mammography using euref image quality criteria. *European radiology*, 25(10), 2015. PMCID: PMC4562003 PMID: 25813015, https://doi.org/10.1007/s00330-015-3695-2.

[44] G. S. Henríquez and E. E. Sánchez. Informe de seminario de investigación mamografía espectral : alternativa viable de la resonancia mamaria en la detección del cáncer de mama, 2024. https://repositorio.unab.cl/handle/ria/63653.

[45] National Cancer Institute. The cancer imaging archive (tcia). https://www.cancerimagingarchive.net/, 2015. Accedido: 15 de junio de 2024.

[46] R. Khaled, M. Helal, O. Alfarghaly, O. Mokhtar, A. Elkorany, H. Kassas, and A. Fahmy. Categorized digital database for low energy and subtracted contrast enhanced spectral mammography images [dataset]. https://www.cancerimagingarchive.net/collection/cdd-cesm/, 2021. The Cancer Imaging Archive, DOI: 10.7937/29kw-ae92, Accedido: 20 de junio de 2024.

[47] F. Isensee, P. F. Jaeger, S. A. Kohl, J. Petersen, and K. H. Maier-Hein. nnu-net: a self-configuring method for deep learning-based biomedical image segmentation. https://github.com/MIC-DKFZ/nnUNet/blob/master/readme.md, 2021. Accedido: 25 de junio de 2024.

[48] Jun. Ma, Yuting. He, Feifei. Li, Lin. Han, Chenyu. You, and Bo. Wang. Medsam (branches: Litemedsam). https://github.com/bowang-lab/MedSAM/tree/LiteMedSAM, 2024. Accedido: 4 de junio de 2024.

[49] I. Soto-Rey D. Müller and F. Kramer. Towards a guideline for evaluation metrics in medical image segmentation. *BMC Research Notes*, 15:210, 2022. https://doi.org/10.1186/s13104-022-06096-y.

[50] Yuxiao Gao, Yang Jiang, Yanhong Peng, Fujiang Yuan, Xinyue Zhang, and Jianfeng Wang. Medical image segmentation: A comprehensive review of deep learning-based methods. *Tomography*, 11, 2025. https://doi.org/10.3390/tomography11050052.

[51] Jaime Simarro, Zohaib Salahuddin, Ahmed Gouda, and Anindo Saha. Leveraging slic superpixel segmentation and cascaded ensemble svm for fully automated mass detection in mammograms. *CoRR*, abs/2010.10340, 2020. https://arxiv.org/abs/2010.10340.