UNIVERSITAT DE BARCELONA

Trabajo Final de Grado

GRADO EN INFORMÁTICA

Facultad de Matemáticas e Informática
Universidad de Barcelona

Clasificación de densidad mamaria utilizando Aprendizaje Profundo

---

Autor: Alejandro Cantero Lopez

Directora: Laura Igual Muñoz
Departamento: Matemáticas e Informática

Barcelona, 10 de junio de 2025

Abstract

Breast density is a clinically relevant factor in the detection of breast cancer, as it influences both the difficulty of identifying lesions in mammograms and the risk associated with the development of the disease. Higher breast density increases not only the complexity of image-based diagnosis but also the likelihood of developing breast cancer.

In recent years, technological advances have enabled the integration of artificial intelligence tools into the medical field, aiming to improve diagnostic accuracy and support the work of healthcare professionals. In this context, machine learning—particularly convolutional neural networks—has proven especially effective in medical image analysis tasks.

This project aims to develop a machine learning model capable of classifying mammograms according to breast density. To achieve this, a convolutional neural network based on the DenseNet-121 architecture has been implemented using the Python programming language. The model has been trained and evaluated using two public mammography datasets, in order to analyze its performance in a multiclass classification task characterized by technical variability and clinical subjectivity.

Resumen

La densidad mamaria es un factor clínico relevante en la detección del cáncer de mama, ya que influye tanto en la dificultad de identificar lesiones en las mamografías como en el riesgo asociado al desarrollo de esta enfermedad. Una mayor densidad implica un aumento tanto en la complejidad del diagnóstico por imagen como en la probabilidad de padecer cáncer mamario.

En los últimos años, el avance tecnológico ha permitido integrar herramientas de inteligencia artificial en el ámbito médico, con el objetivo de mejorar la precisión diagnóstica y apoyar la labor de los profesionales de la salud. En este contexto, el aprendizaje automático, y en particular las redes neuronales convolucionales, han demostrado ser especialmente eficaces en tareas de análisis de imágenes médicas.

Este trabajo tiene como objetivo desarrollar un modelo de aprendizaje automático capaz de clasificar mamografías según su densidad mamaria. Para ello, se ha implementado una red neuronal convolucional basada en la arquitectura DenseNet-121, utilizando el lenguaje de programación Python. El modelo ha sido entrenado y evaluado empleando dos bases de datos públicas de mamografías, con el fin de analizar el comportamiento en una tarea de clasificación multiclase que presenta variabilidades técnicas y subjetividad clínica.

Agradecimientos

En primer lugar quiero agradecer a la Dra. Laura Igual Muñoz por el asesoramiento durante todo el proceso de desarrollo del proyecto. Asimismo, agradezco la cesión del servidor dotado con equipamiento de alto rendimiento, indispensable para la ejecución de las pruebas computacionalmente intensivas que requiere este trabajo.

También deseo agradecer a Lídia Garrucho, por su amabilidad al compartir su conocimiento, así como por los valiosos consejos y explicaciones teóricas que han contribuido significativamente a la comprensión y desarrollo del presente trabajo.

Finalmente, hago extensivo mi agradecimiento a todas las personas que, de una u otra forma, han ofrecido su apoyo y motivación durante la realización de este proyecto académico. Sin olvidar a aquellos que me acompañaron durante el grado.

Índice

Introducción

La densidad mamaria está asociada con un incremento en el riesgo de padecer cáncer de mama. A su vez, una mayor densidad mamaria dificulta la detección de tumores en radiografías. Es por ello que hay líneas de investigación y desarrollo activas que tratan de desarrollar modelos de aprendizaje automático que clasifiquen la densidad mamaria en radiografías, sirviendo como apoyo a los profesionales de la salud. Sumándose este modelo a otros ya existentes que ayudan a médicos entre otras profesiones a realizar su trabajo correctamente, ayudando a minimizar el riesgo de fallo.

1.1. Objetivos

En este proyecto se desarrollará un modelo de aprendizaje automático realizando los procesos de entrenamiento e inferencia en dos conjuntos de datos distintos. El objetivo es estudiar el comportamiento de los modelos utilizando ambos conjuntos de datos, con la finalidad de extraer conclusiones sobre la importancia de los datos y la configuración del modelo frente a un problema de clasificación que presenta un factor subjetivo.

Planificación

A continuación se muestra en un diagrama de Gantt cual ha sido la planificación del proyecto durante los cuatro meses de desarrollo. El diagrama simplifica el proceso de desarrollo en sus cuatro partes más importantes.

Figura 2.1: Diagrama de Gantt.

Fundamentos Teóricos

Este capítulo tiene como objetivo establecer los fundamentos teóricos que sustentan el desarrollo de este proyecto. Para abordar adecuadamente el problema de la clasificación automática de la densidad mamaria en imágenes de mamografía mediante redes neuronales convolucionales, es necesario comprender los principios clínicos y técnicos que intervienen en dicho proceso.

3.1. Mamografías

Una mamografía es una imagen basada en rayos X utilizada para examinar el tejido mamario. Por lo general, se obtienen una o más imágenes por cada mama con el objetivo principal de detectar tumores, incluso aquellos que no pueden palparse físicamente [1].

Para cada pecho, se obtienen radiografías desde dos ángulos distintos: la proyección cráneo-caudal (CC) y la proyección oblicua mediolateral (MLO). La proyección CC proporciona una vista de arriba hacia abajo, mientras que la MLO se realiza desde un ángulo lateral, permitiendo observar el tejido desde un costado del tórax. Ambas vistas son complementarias y ofrecen información relevante para una evaluación completa de la anatomía mamaria. La Figura 3.1 muestra un ejemplo de mamografía en proyecciones CC y MLO.

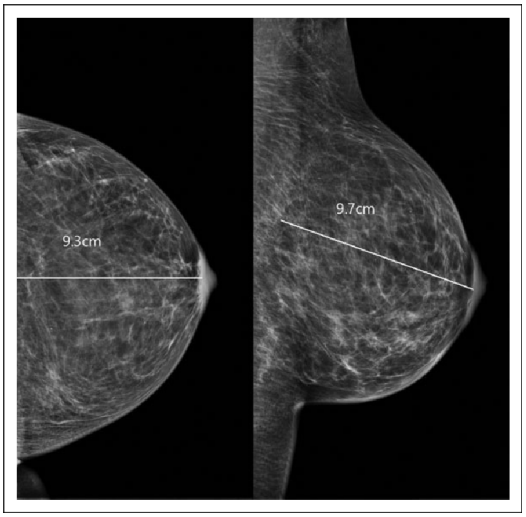


Figura 3.1: Mamografía con proyección CC (izquierda) y MLO (derecha)

Gracias a las mamografías es posible identificar diversas anomalías, siendo las más comunes las masas y las calcificaciones [17, 7]:

- **Masas:** Son áreas de tejido con una estructura diferente a la del tejido circundante. Pueden ser benignas o malignas, por lo que su detección y caracterización son fundamentales.
- **Calcificaciones:** Consisten en depósitos de calcio que aparecen como pequeñas manchas blancas en las imágenes. Aunque la mayoría son benignas, algunas pueden representar un signo temprano de cáncer.

3.2. ¿Qué es la densidad mamaria?

La densidad mamaria hace referencia a la proporción de tejido fibroglandular (compuesto por glándulas y conductos) en relación con el tejido adiposo (graso) en la mama. En las mamografías, el tejido graso aparece en tonos oscuros, mientras que el tejido fibroglandular se visualiza en blanco, lo que puede dificultar la detección de anomalías que también presentan tonalidades blancas [8].

El sistema *BI-RADS* (Breast Imaging Reporting and Data System), desarrollado por el Colegio Americano de Radiología, proporciona una clasifica-

ción estandarizada para describir la densidad mamaria. Esta clasificación se divide en cuatro categorías:

1. **Tipo A – Predominantemente graso:** La mama está compuesta casi por completo por tejido adiposo. Representa un 10 % de la población mundial [1].

2. **Tipo B – Tejido fibroglandular disperso:** Hay algunas áreas de tejido fibroglandular, pero predomina el tejido graso. Representa un 40 % de la población mundial.

3. **Tipo C – Heterogéneamente denso:** La cantidad de tejido denso es significativa, lo cual puede dificultar la visualización de pequeñas masas. Representa un 40 % de la población mundial.

4. **Tipo D – Extremadamente denso:** La mayoría del tejido mamario es denso (fibroglandular), lo cual dificulta la visualización de masas, además, aumenta el riesgo de tumores. Representa un 10 % de la población mundial.

3.3. ¿Por qué queremos clasificarla?

La clasificación de la densidad mamaria es de gran relevancia tanto clínica como técnicamente. Desde el punto de vista clínico, una densidad mamaria elevada (categorías C y D del sistema `BI-RADS`) se asocia con un incremento del riesgo de desarrollar cáncer de mama. Además, como se ha mencionado en la sección 3.2, el tejido fibroglandular y muchas anomalías, como tumores o calcificaciones, se visualizan en blanco en la mamografía (mayor intensidad). Esta similitud en la apariencia puede dificultar la detección de anomalías, al ser difícilmente diferenciables del tejido denso [9, 3].

En la Figura 3.2 se puede apreciar el contraste visual entre los distintos tipos de densidad mamaria. La clasificación de densidad mamaria resulta esencial para mejorar la precisión del diagnóstico y personalizar los métodos de detección.

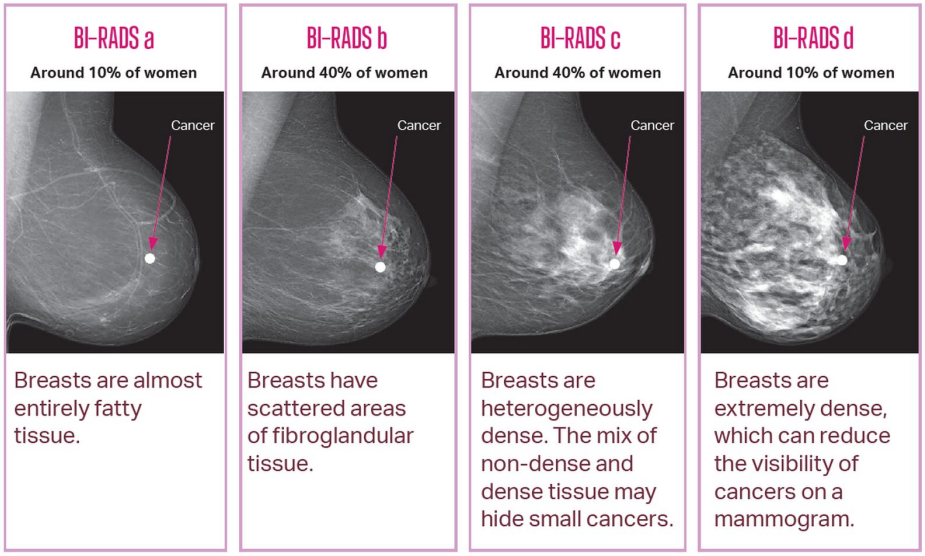


Figura 3.2: Ejemplos de las distintas categorías de densidad mamaria en clasificación BI-RADS.

Fuente: BreastScreen SA. *Breast Density and Cancer Risk* [3]

3.4. Redes Neuronales

Una red neuronal es un modelo computacional de aprendizaje automático inspirado en el funcionamiento del cerebro humano. En este modelo, se define el concepto de neurona artificial, un componente que se organiza en capas. Estas capas están interconectadas, emulando el comportamiento de las neuronas biológicas para reconocer patrones y extraer conclusiones a partir de los datos.

3.4.1. Neurona

La neurona artificial, también conocida como `perceptrón`, es la unidad básica de las redes neuronales. Este componente simula el comportamiento de una neurona biológica: recibe múltiples señales de entrada, las procesa y genera una señal de salida.

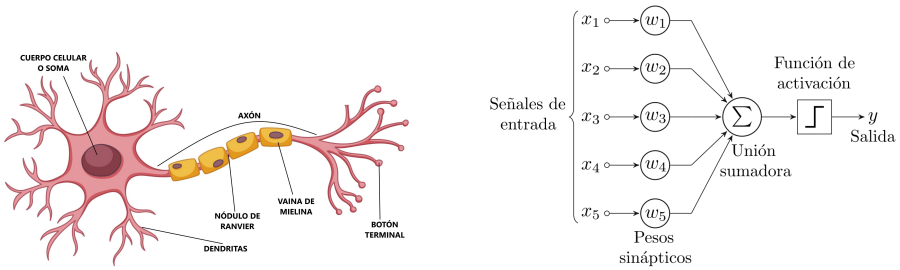
El funcionamiento de una neurona artificial puede describirse mediante la siguiente expresión matemática:

$$f(X) = \varphi\left(w_0 \cdot b + \sum_{i=1}^{n} w_i \cdot x_i\right)$$

Donde:

- $X = (x_1, ..., x_n)$ representa el vector de entradas.
- $W = (w_1, ..., w_n)$ es el vector de pesos asociados a cada entrada.
- b es el sesgo, o término independiente.
- φ es la función de activación.

Este cálculo genera una salida que se propaga hacia la siguiente capa. En la Figura 3.3 se ilustra la analogía entre una neurona biológica y un perceptrón.



(a) Neurona biológica (b) Perceptrón

Figura 3.3: Comparación entre una neurona biológica y una neurona artificial (perceptrón).

3.4.2. Funciones de activación

Las funciones de activación son elementos fundamentales en las redes neuronales, ya que introducen no linealidad al modelo, permitiéndole aprender relaciones complejas entre los datos. Además, como su propio nombre indica, esta función decide también si el perceptrón se activa (tiene una salida mayor a cero) o no.

Las funciones de activación son componentes esenciales en las redes neuronales, ya que introducen no linealidad al modelo, permitiendo que las redes aprendan relaciones complejas entre las entradas y las salidas. A

continuación, se describen las funciones de activación más comunes, junto con sus representaciones gráficas.

Función ReLU (Rectified Linear Unit)

La función ReLU es ampliamente utilizada en redes neuronales modernas debido a su simplicidad y eficiencia computacional. Se define como:

$$\varphi(x) = \text{máx}(0, x)$$

Esta función activa la neurona solo si la entrada es positiva, devolviendo su valor; en caso contrario, devuelve cero. ReLU ayuda a mitigar el problema del desvanecimiento del gradiente, ya que su derivada es constante para valores positivos, facilitando la propagación del error durante el entrenamiento. [10].

Además, esta es la función de activación por defecto de la red neuronal convolucional utilizada en este proyecto[1].

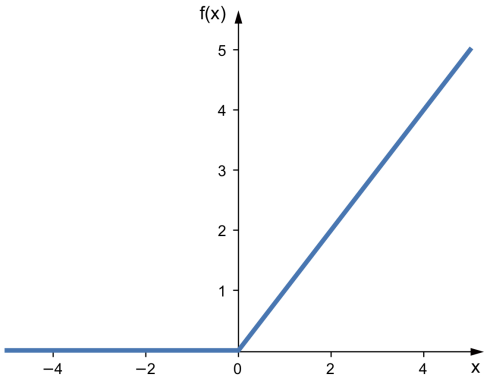


Figura 3.4: Función de activación ReLU

Fuente: *Datacamp - Introduction to Activation Functions in Neural Networks* [10]

[1]Documentación de MONAI. DenseNet. `https://docs.monai.io/en/stable/networks.html#monai.networks.nets.DenseNet`

Función Sigmoide

La función sigmoide transforma cualquier valor real en un rango entre 0 y 1, lo que la hace adecuada para tareas de clasificación binaria. Su fórmula es:

$$\varphi(x) = \frac{1}{1 + e^{-x}}$$

Aunque fue popular en las primeras redes neuronales, la función sigmoide presenta el problema del desvanecimiento del gradiente. A medida que los valores de entrada se alejan de cero, la derivada de la función se aproxima a cero, lo que dificulta el ajuste de los pesos en las capas profundas durante el entrenamiento [10].

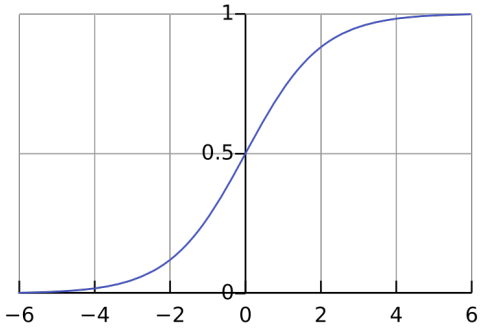


Figura 3.5: Función de activación sigmoide

Fuente: *Datacamp - Introduction to Activation Functions in Neural Networks* [10]

Función Softmax

La función Softmax se emplea en la capa de salida de modelos de clasificación multiclase. Convierte un vector de valores en probabilidades normalizadas que suman uno, facilitando la interpretación de las salidas del modelo. Su fórmula es:

$$\varphi(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{n} e^{x_j}}$$

Esta función es especialmente útil cuando se necesita asignar una probabilidad a cada clase en problemas de clasificación multiclase. Por este motivo, esta función se ha utilizado en la salida de la red neuronal desarrollada.

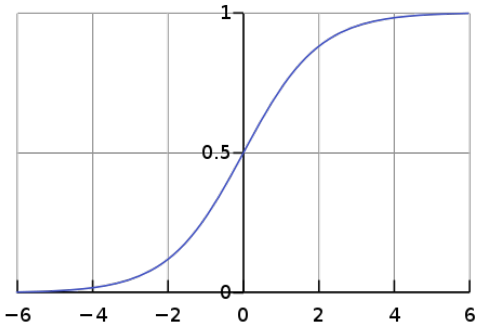


Figura 3.6: Función de activación Softmax

Fuente: *Datacamp - Introduction to Activation Functions in Neural Networks* [10]

3.5. Redes Neuronales Convolucionales

Una red neuronal convolucional (CNN, por sus siglas en inglés) es un tipo de algoritmo de aprendizaje profundo fundamentado en las redes neuronales que se especializa en el procesamiento y análisis de datos en dos dimensiones, especialmente cuando la posición relativa de los elementos es importante, como es el caso de las imágenes.

Las CNN hacen uso de capas convolucionales que extraen diferentes características de la imagen, esto lo hacen aplicando filtros (o kernels) que recorren la imagen de entrada, esta operación se llama convolución y es la principal característica que diferencia las CNN de otros algoritmos. Además, es una parte fundamental para el rendimiento de estas.

3.5.1. Capas

Las redes neuronales están organizadas en capas. Existen diferentes tipos de capas y cada una desempeña un papel específico en el procesamien-

to de la información. El orden y la combinación de estas capas es esencial para obtener un modelo eficaz y eficiente. A continuación, se describen las principales capas utilizadas en redes neuronales convolucionales.

Capas Convolucionales

Las capas convolucionales son la base de las redes neuronales convolucionales (CNN). Su función principal es extraer características espaciales locales mediante la aplicación de filtros (también llamados *kernels*) sobre la entrada. Estos filtros se desplazan por la imagen o el mapa de características realizando una operación de convolución, generando así un nuevo mapa de activación que resalta la presencia de patrones específicos como bordes, esquinas o texturas.

El resultado de esta operación permite a la red aprender representaciones jerárquicas de la información visual, donde las primeras capas detectan características simples y las últimas capas capturan estructuras más complejas. Estas capas reducen la necesidad de extracción manual de características y aumentan la capacidad de generalización del modelo.

Capas de Pooling

Las capas de *pooling*, o submuestreo, se utilizan para reducir la dimensionalidad espacial de los mapas de activación generados por las capas convolucionales. Esta reducción no solo disminuye la cantidad de parámetros del modelo, sino que también mejora la eficiencia computacional y reduce el riesgo de sobreajuste. Normalmente, estas capas están situadas entre dos capas convolucionales.

Además, estas capas introducen una cierta invariancia a traslaciones pequeñas de los datos de entrada, lo cual puede hacer el modelo más robusto. Existen varios tipos de pooling:

- **Max Pooling:** selecciona el valor máximo dentro de una región definida (por ejemplo, una ventana 2×2). Es el método más común y permite conservar las características más destacadas.
- **Average Pooling:** calcula el valor promedio dentro de la ventana, proporcionando una salida más suavizada.

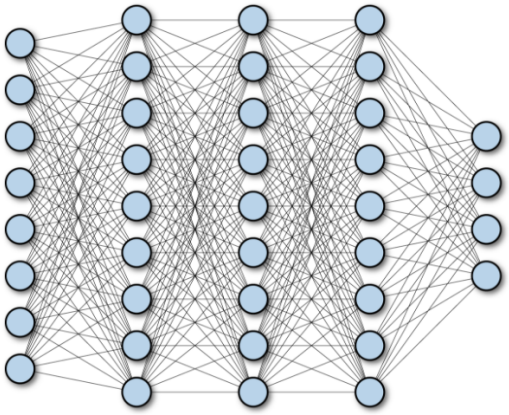


Figura 3.7: Diagrama de capas *Fully Connected*

- **Global Pooling:** realiza la operación de pooling (máximo o promedio) sobre toda la dimensión espacial del mapa de activación, reduciendo cada mapa a un solo valor.

Capas Totalmente Conectadas (Fully Connected Layers)

Las capas totalmente conectadas, también conocidas como capas densas, son componentes fundamentales en las redes neuronales profundas. En estas capas, cada neurona está conectada a todas las neuronas de la capa anterior, lo que permite integrar y procesar la información extraída por las capas anteriores para generar una salida final.

En el caso de las redes neuronales convolucionales, estas capas transforman los mapas de características multidimensionales en un vector unidimensional, que luego se procesa mediante funciones de activación, como la función *softmax* para clasificación multiclase, para producir la predicción final del modelo.

Estas capas totalmente conectadas pueden aumentar significativamente el número de parámetros del modelo, de todas formas, su inclusión es esencial para combinar y evaluar las características extraídas, permitiendo al modelo interpretar rasgos más precisos y complejos.

3.5.2. Densenet

DenseNet (Densely Connected Convolutional Network) es una arquitectura de red neuronal convolucional propuesta por Huang et al. en 2017. El principal aspecto innovador de esta arquitectura es que implementa el concepto de *Dense Block* o bloque denso. Esta estrategia de conectividad densa implica que, en lugar de tener únicamente conexiones entre capas adyacentes, cada capa recibe como entrada los mapas de características de todas las capas precedentes y transmite sus propias salidas a todas las capas subsiguientes. En la Figura 3.8 vemos una representación visual de la arquitectura de un bloque denso. Comparando con la Figura 3.7 vemos la diferencia en las conexiones. Como podemos ver, una capa L solo tiene conexión con la capa $L-1$ y $L+1$ mientras que en la arquitectura DenseNet una capa L se conecta con las capas posteriores, teniendo $\frac{L(L+1)}{2}$ conexiones por cada capa [2].

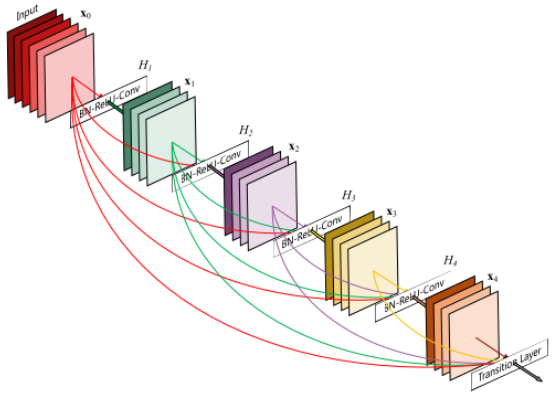


Figura 3.8: Diagrama de un bloque denso (*DenseBlock*).

Fuente: Huang, G. *Densely Connected Convolutional Networks* [18]

3.5.3. Ventajas de DenseNet

- **Mejora en la propagación de la información y los gradientes:** La conectividad densa facilita el flujo de información y de gradientes a

[2]Huang, G., Liu, Z., Van Der Maaten, L., y Weinberger, K. Q. (2016). *Densely Connected Convolutional Networks*. `https://arxiv.org/pdf/1608.06993`.

través de la red, lo que mitiga el problema del desvanecimiento del gradiente y permite el entrenamiento de redes más profundas.

- **Reutilización de características:** Al concatenar los mapas de características de todas las capas anteriores, DenseNet promueve la reutilización de características, lo que conduce a una representación más eficiente y rica de los datos.
- **Reducción del número de parámetros:** A pesar de la conectividad aumentada, DenseNet requiere menos parámetros que arquitecturas tradicionales, ya que evita la necesidad de aprender características redundantes.
- **Regularización implícita:** La estructura de DenseNet actúa como una forma de regularización, reduciendo el sobreajuste en conjuntos de datos con un número limitado de muestras.

3.6. Hiperpárametros

Los hiperparámetros son variables de configuración que se definen manualmente antes del entrenamiento y que permiten definir algunos aspectos sobre el entrenamiento del modelo. Son esenciales para el rendimiento y la generalización de las redes neuronales convolucionales. En este proyecto, se han considerado los siguientes hiperparámetros clave:

3.6.1. Epochs

Las redes neuronales se entrenan iterando un conjunto de datos múltiples veces, cada una de estas pasadas es nombrado *epoch* o época. Determinar el número adecuado de *epochs* es crucial: un número insuficiente puede llevar a un modelo subentrenado (*underfitting*), mientras que un número excesivo puede causar sobreajuste (*overfitting*).

3.6.2. Batch size

El *batch size* define la cantidad de muestras procesadas simultáneamente antes de actualizar los pesos del modelo. Este parámetro influye directamente en el uso de memoria de la GPU y en la estabilidad del entrenamiento. Un *batch size* mayor puede aprovechar mejor la capacidad de

paralelización de la GPU y proporcionar estimaciones más estables del gradiente, pero requiere más memoria. Por otro lado, un tamaño menor introduce más ruido en la estimación del gradiente, ya que hay menos generalización, lo que puede ayudar a escapar de mínimos locales, pero puede ralentizar la convergencia [3].

Además, el *batch size* influye no solo en la eficiencia computacional, sino también en la calidad del modelo entrenado. Un tamaño de lote más grande puede conducir a una convergencia más rápida, aunque puede comprometer la capacidad de generalización si no se acompaña de ajustes adecuados en la tasa de aprendizaje y otros hiperparámetros[4].

3.6.3. Learning Rate

La tasa de aprendizaje (*learning rate*) controla la magnitud de las actualizaciones de los pesos del modelo durante el entrenamiento. Es uno de los hiperparámetros más críticos, ya que, como su propio nombre indica, determina la velocidad y estabilidad del aprendizaje. Un *learning rate* demasiado alto puede causar oscilaciones o divergencias en la función de pérdida, mientras que una tasa demasiado baja puede resultar en una convergencia lenta y potencialmente quedarse atrapado en mínimos locales[5].

Es importante mencionar que este es un hiperparámetro estrictamente relacionado con el optimizador (explicado en la subsección 3.8) y, algunos optimizadores como el Adam regulan automáticamente el *learning rate* según factores como la pérdida (loss) durante el proceso de entrenamiento. De todas formas, hay que proporcionar un valor inicial que condicionará el resto del entrenamiento.

[3]SabrePC. "Epochs vs Batch Size vs Iterations: Differences in Deep Learning." *SabrePC*, 2021. Disponible en: `https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations`

[4]Masters, D., & Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. *Pattern Recognition Letters*, 131, 244–250. Disponible en: `https://www.sciencedirect.com/science/article/pii/S2405959519303455`

[5]Mishra, M. (2023). The Learning Rate: A Hyperparameter That Matters. *Medium*. Disponible en: `https://mohitmishra786687.medium.com/the-learning-rate-a-hyperparameter-that-matters-b2f3b68324ab`

3.7. Función de pérdida Cross Entropy

La función de pérdida es una función matemática que cuantifica el error cometido por la red neuronal. Esta función se evalúa constantemente durante la fase de entrenamiento para dar un valor numérico al rendimiento dado del modelo. Después, con el uso del *optimizador* explicado en la subsección 3.8 se buscará el mínimo de esta función, equivalente a minimizar el error del modelo.

Para el caso de clasificación, la opción preferente en la gran mayoría de los casos es la función de pérdida `Cross Entropy`. Además, esta función de pérdida es particularmente útil para clasificación multiclase y permite el uso de pesos para tratar el desbalance del conjunto de datos.

Pytorch tiene esta función implementada en su módulo *torch.nn.optim* de forma que solo necesitamos crear una instancia de la clase y pasarle por parámetro los pesos correspondientes al desbalance de clases de nuestro conjunto de datos.

3.8. Optimizador

El optimizador es un elemento fundamental de las redes neuronales, este se encarga de modificar los parámetros de las neuronas que forman red, los pesos y los sesgos (véase la sección 3.3, durante el proceso de entrenamiento. Estos optimizadores implementan algoritmos de optimización con el objetivo de minimizar la función de pérdida, permitiendo reducir el error cometido por la red, mejorando así su capacidad de generalización y rendimiento. Uno de los algoritmos más utilizados en este contexto es el descenso de gradiente estocástico (Stochastic Gradient Descent, SGD), que sirve de base para muchos optimizadores modernos. Estos, a su vez, introducen variaciones sobre el SGD que pueden tener un impacto significativo durante la fase de entrenamiento, ya sea reduciendo el tiempo de cómputo, facilitando una convergencia más estable o mejorando la calidad del mínimo alcanzado en la función de pérdida. Como consecuencia, estas mejoras contribuyen a un aumento en la precisión del modelo y a una representación más eficaz de los datos.

Descenso de gradiente y descenso de gradiente estocástico

Este algoritmo trata de encontrar mínimos locales en una función, en el caso de las redes neuronales, la función de pérdida previamente definida. Para esto, la función debe ser diferenciable y convexa. El algoritmo consiste en tomar pasos de manera iterativa en dirección contraria al gradiente, es decir, disminuyendo el valor de la función cuando se evalua en este nuevo punto. Véase la Figura 3.9.

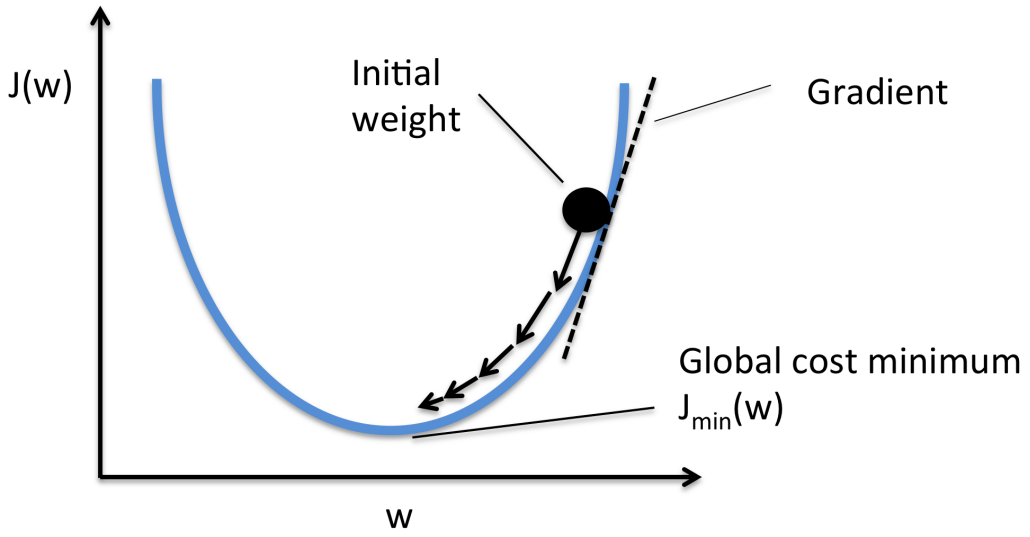
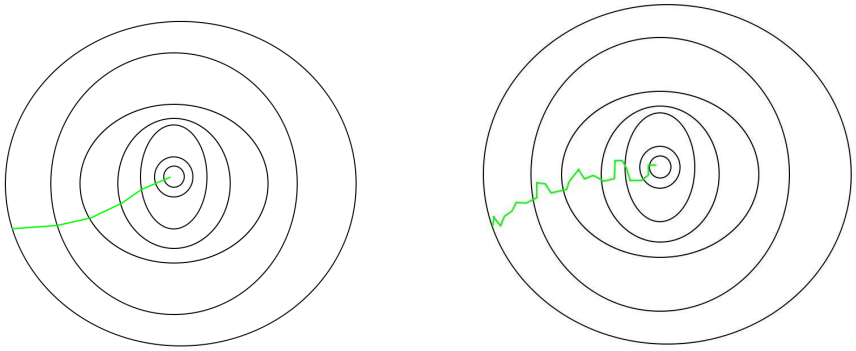


Figura 3.9: Interpretación del algoritmo SGD.

Fuente: Raschka Sebasitan. *What are gradient descent and stochastic gradient descent?* [14]

Es importante mencionar que la función mostrada en la Figura 3.9 es de dos dimensiones. Pero en el ámbito de las redes neuronales nos enfrentamos a funciones de decenas de miles de dimensiones.

Debido a las altas dimensiones con las que trabajan las redes neuronales y, en especial, las CNN, es inviable aplicar este método con todos los parámetros extraídos del conjunto de datos. Es por esto que se utiliza el descenso de gradiente estocástico.



(a) Descenso de gradiente

(b) Descenso de gradiente estocastico

Figura 3.10: Comparacion entre GD y SGD

Ahora el gradiente es calculado con una pequeña muestra del conjunto de datos, lo que llamamos *batch*. Esto mejora drásticamente el uso de recursos requeridos para computar el descenso de gradiente, pero introduce estocasticidad ya que la selección de la muestra de datos es aleatoria.

$$p_{n+1} = p_n - \lambda f'(p_n)$$

Adam

El optimizador *Adam* (del inglés *Adaptive Moment Estimation*) es uno de los algoritmos de optimización más usados en el caso de clasificación en redes neuronales convolucionales debido a su eficiencia y capacidad de adaptación. Adam combina las ventajas de dos algoritmos: el descenso de gradiente estocástico con momento y el algoritmo RMSProp[6].

A diferencia del descenso de gradiente tradicional, que utiliza una tasa de aprendizaje fija para todos los parámetros, Adam ajusta durante el en-

[6]Analytics Vidhya. (2021). A Comprehensive Guide on Deep Learning Optimizers. Disponible en: `https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/`

trenamiento la tasa de aprendizaje de forma individual por parámetro siguiendo una regla de actualización [20]. Esto se logra mediante el cálculo de estimaciones de primer y segundo momento (la media y la varianza) de los gradientes, lo que permite al algoritmo adaptarse a diferentes características de la forma de la función de pérdida[7].

Este optimizador implementa también una técnica conocida como momento. El momento consiste en acumular un valor de dirección del descenso del gradiente para suavizar las actualizaciones. Esta estrategia ayuda a reducir la variabilidad en la dirección del descenso del gradiente y acelera la convergencia, especialmente en regiones donde los gradientes son inconsistentes. Aquí, Adam toma inspiración del algoritmo RMSProp, aunque con una ligera diferencia: Adam añade la segunda derivada para calcular el momento, de forma que mantiene la estabilidad del entrenamiento al inicio, cuando las estimaciones aún no se han estabilizado.

[7]Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980*. Disponible en: `https://arxiv.org/abs/1412.6980`

Recursos

El desarrollo de una CNN puede ser un trabajo computacionalmente muy costoso. Por lo tanto, es importante tener el equipo adecuado para ello, tanto Hardware como Software.

4.1. Hardware

En el entrenamiento de redes neuronales, el hardware juega un papel fundamental en el rendimiento y la eficiencia del proceso. Si bien todos los componentes del sistema influyen en el resultado final, la unidad de procesamiento gráfico (GPU) es la más importante en nuestro caso, sobre todo debido a que la entrada de nuestro modelo es un elemento bidimensional, una imagen. Es por esto que una GPU, especializada en procesamiento paralelo, es el componente de hardware principal para realizar los cálculos.

Para este proyecto se empleó un servidor equipado con dos GPUs NVIDIA GeForce RTX 3090, cada una con 24 GB de memoria RAM dedicada. Estas tarjetas gráficas destacan por sus 10496 núcleos CUDA y su gran cantidad de memoria, lo que permite realizar entrenamientos más rápidos y eficientes.

Una de las principales ventajas de contar con GPUs con gran cantidad de memoria es la posibilidad de utilizar un mayor *batch size*. Como se explica en detalle en la subsección 3.6.2

De todas formas, debido a la disponibilidad del servidor, solo se ha utilizado una de las GPUs para entrenar el modelo.
Por lo tanto, disponer de GPUs con gran capacidad de memoria, como las RTX 3090, permite explorar configuraciones de entrenamiento más exigentes y optimizar tanto el rendimiento como los resultados del modelo.

4.2. Software

El software actúa como el intermediario entre el investigador y el hardware. A través de librerías y *Frameworks* nos permite definir una serie de instrucciones que configuran el entrenamiento y definen el qué y cómo se entrenará. Estas herramientas permiten abstraer muchos detalles de bajo nivel, ofreciendo interfaces accesibles para tareas como el preprocesamiento de datos, la construcción de arquitecturas neuronales, el entrenamiento de modelos y la evaluación de resultados.

En la siguiente lista se muestra el software y la versión del mismo utilizados para el desarrollo del modelo.

- **NVIDIA Drivers:** Versión 550.90.07
- **CUDA:** Versión 12.4, esta versión es importanteya que PyTorch no está disponible para todas las versiones de CUDA.
- **MONAI:** Se instaló *Monai-Weekly* esto instala la última versión en desarrollo de Monai, en este caso tenemos la versión 1.5.dev2513.
- **PyTorch:** Versión 2.6.0. Además, Monai utiliza pytorch-ignite, que nos proporciona muchas herramientas para conseguir métricas, o interactuar con el modelo. Esta librería se ha utilizado en su versión 0.4.11.

4.3. Herramientas utilizadas

En el siguiente apartado se describen las herramientas utilizadas para la creación, entrenamiento e inferencia del modelo.

4.3.1. Torch y PyTorch

`Torch` [1] es una librería de código abierto programada en Lua usada para crear redes neuronales. Esta es una de las librerías preferidas para hacer

[1] Torch. *Torch Documentation*. Disponible en: `https://pytorch.org/docs/stable/torch.html`. Accedido el 3 de junio de 2025.

investigación en aprendizaje profundo. *Torch* hace un buen uso de la GPU, es fácil de usar y muy eficiente. Al estar escrito en un lenguaje muy eficiente como es Lua y tener implementaciones directamente con CUDA.

`PyTorch` [2] es un framework de código abierto que permite un uso fácil de la librería Torch en Python. Este framework proporciona muchos objetos configurables como, por ejemplo, arquitecturas completas de redes neuronales, optimizadores, funciones de pérdida, etc.

4.3.2. MONAI

MONAI (Medical Open Network for AI)[3] es un *framework* de código abierto basado en *PyTorch*, diseñado específicamente para el desarrollo de modelos de inteligencia artificial aplicados a imágenes médicas. Su arquitectura modular y orientada a tareas clínicas permite una implementación más accesible, reproducible y eficiente de modelos de aprendizaje profundo en el ámbito sanitario.

MONAI se estructura en varios subframeworks, siendo MONAI Core el núcleo funcional principal. En este trabajo, se ha utilizado exclusivamente MONAI Core, ya que proporciona herramientas fundamentales para la carga, preprocesamiento, entrenamiento y evaluación de modelos de redes neuronales convolucionales en imágenes médicas, facilitando significativamente el desarrollo en entornos de investigación biomédica.

Este proyecto fue fundado originalmente por NVIDIA y el King's College London. Actualmente, MONAI está mantenido por una comunidad activa de investigadores, profesionales clínicos y colaboradores de la industria, con el objetivo común de acelerar la adopción de la inteligencia artificial en el análisis de imágenes médicas.

[2] PyTorch. *PyTorch Official Site*. Disponible en: `https://pytorch.org`. Accedido el 3 de junio de 2025.

[3] MONAI. *Medical Open Network for AI*. Disponible en: `https://monai.io`. Accedido el 3 de junio de 2025.

Implementación

Este capítulo describe en detalle el proceso de implementación llevado a cabo para el desarrollo de un modelo de aprendizaje profundo orientado a la clasificación de la densidad mamaria en mamografías.

En primer lugar, se presentan ambos conjuntos de datos con sus respectivos análisis exploratorios de los datos, donde se expone su composición enfocándonos en el aspecto de la densidad mamaria según la escala BI-RADS y otros datos relevantes para el entrenamiento del modelo de clasificación. Como la calidad, tamaño y formato de las imágenes.

Posteriormente, se describen las técnicas de preprocesado aplicadas sobre los datos, esenciales para garantizar una correcta estandarización de las muestras y facilitar la generalización del modelo. Entre estos procedimientos se incluyen la eliminación de etiquetas textuales (*delabeling*), el recorte automático del tejido mamario (*cropping*), la normalización de la orientación del pecho y la homogeneización de las dimensiones de entrada.

Finalmente, se introduce la estructura general de la implementación del modelo. En esta se abordará la arquitectura de la red neuronal, las estrategias de entrenamiento y evaluación, y el análisis de resultados obtenidos.

5.1. CBIS-DDSM

CBIS-DDSM [1] (Curated Breast Imaging Subset of DDSM), es una versión actualizada y estandarizada de la *Digital Database for Screening Mammography (DDSM)*. Contiene 2620 mamografías de 1566 pacientes, con vis-

[1] Sawyer-Lee, R., Gimenez, F., Hoogi, A. y Rubin, D. (2016). *Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM)* [Data set]. The Cancer Imaging Archive. doi: `https://doi.org/10.7937/K9/TCIA.2016.7O02S9CY`.

tas tanto CC como MLO por ambos pechos idealmente.

Las imágenes provienen de mamografías filmográficas que han sido escaneadas a formato digital, descomprimidas y convertidas a formato DICOM. Además, la base de datos contiene imágenes de segmentación *ROI* (Region of Interest) y *Bounding boxes* (delimitadores de contorno) delimitando el área donde se presenta la anomalía. Por último, CBIS-DDSM contiene un diagnóstico patológico realizado por un radiólogo experto en mamografías que se encuentra en un archivo formato csv.

Análisis exploratorio

Se ha realizado un análisis exploratorio de los datos. A continuación se muestra este análisis presentando estadísticas y gráficos que muestren diferentes características de los datos que serán especialmente útiles más adelante para la configuración de la red neuronal convolucional.

El conjunto de datos presenta una jerarquía de directorios algo compleja y que aporta información sobre la mamografía almacenada en los directorios. El nombre del primer directorio está compuesto por el tipo de anomalía presente (calcificación o masa), si forma parte del conjunto de entrenamiento o de test, el identificador del paciente, si es el pecho izquierdo o derecho y, por último, si es proyección craneocaudal (CC) u oblicua mediolateral (MLO). Antes de llegar a la imagen en formato DICOM, tenemos dos directorios cuyo nombre representa el *Media Storage SOP Instance UID*.

```
.
├── Mass-Training_P_00001_LEFT_CC
│   └── 1.3.6.1.4.1.9590.100.1.2.422112722213189649807611434612228974994
│       └── 1.3.6.1.4.1.9590.100.1.2.342386194811267636608694132590482924515
│           ├── 1-1.dcm
│           └── md5hashes.csv
```

Figura 5.1: Jerarquía de directorios CBIS-DDSM.

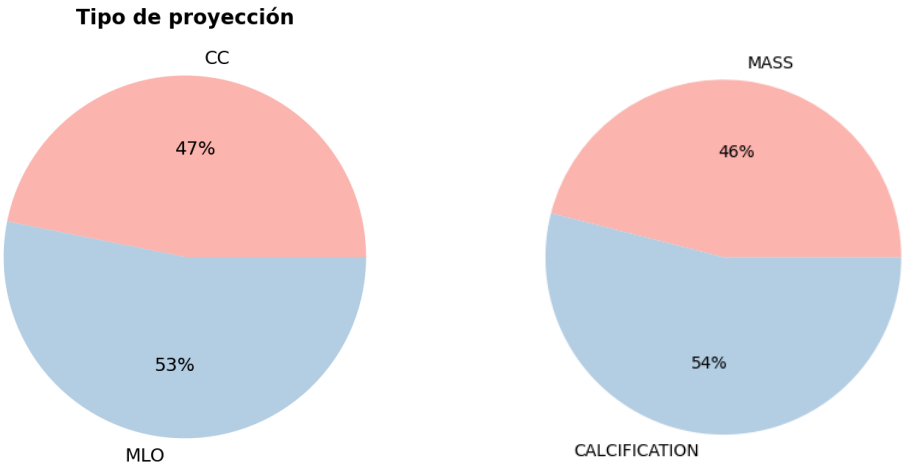
Este conjunto de datos viene separado en un conjunto de *Train* y uno de *Test*. En la fase de entrenamiento, se ha separado una parte de validación que representará el diez por ciento de los datos de entrenamiento. En el Cuadro 5.1 están detallados tanto las imágenes totales por cada conjun-

to de datos, como el número de imágenes por clasificación `BI-RADS` y su porcentaje por cada conjunto de datos.

		Clasificación BI-RADS			
Dataset	**Imágenes totales**	**1**	**2**	**3**	**4**
Train	2519	415 (16.5 %)	946 (37.5 %)	757 (30.0 %)	401 (15.9 %)
Validation	280	38 (13.6 %)	103 (36.8 %)	100 (35.7 %)	39 (13.9 %)
Test	689	74 (10.7 %)	282 (40.9 %)	217 (31.5 %)	114 (16.5 %)

Cuadro 5.1: Distribución absoluta y relativa (%) de imágenes por conjunto de datos y clase de densidad mamaria según la escala BI-RADS en el conjunto de datos CBIS-DDSM.

Hay dos grandes clasificaciones en las imágenes que tenemos que tener en cuenta, la primera es el tipo de anomalía que se presenta, masa o calcificación, y la segunda es la proyección que se ha utilizado al tomar la imagen (CC o MLO). En la Figura 5.2 se observa la proporción de imágenes con las características mencionadas. Estas propiedades están suficientemente balanceadas para no considerarlas una preocupación.



(a) Distribución de proyecciones CC y MLO.

(b) Distribucion anomalias de masas y calcificaciones.

Figura 5.2: Porcentaje de proyecciones y tipos de anomalias en conjunto CBIS-DDSM.

Idealmente, si se dispusiera de una cantidad infinita de datos de entrenamiento, un modelo podría alcanzar un rendimiento perfecto, ya que tendría acceso a todos los casos posibles y aprendería a generalizarlos de forma precisa. Sin embargo, en la práctica, tanto los datos como el tiempo de entrenamiento son limitados. Por este motivo, cuanto mayor sea el número de muestras disponibles para entrenar un modelo de clasificación de densidad mamaria, mayor será su capacidad de aprendizaje y, en consecuencia, su desempeño general.

Además, se debe tener en cuenta que, según el sistema BI-RADS, únicamente alrededor del 10 % de las mujeres presentan una densidad mamaria catalogada como extremadamente densa (categoría D)[2]. Esto implica que, si un modelo logra predecir correctamente las categorías A, B y C pero falla en la categoría D, podría alcanzar una precisión global del 90 %. Sin embargo, esta métrica sería engañosa, ya que ocultaría una deficiencia crítica en la detección de los casos más densos, precisamente aquellos con mayor dificultad diagnóstica.

Es por tanto esencial estudiar la distribución de las categorías de densidad mamaria dentro del conjunto de datos utilizado. Como se muestra en la Figura 5.3, el dataset presenta un desbalance significativo entre clases, con un porcentaje de imágenes correspondiente a cada categoría de densidad de aproximadamente 16 % (A), 37 % (B), 30 % (C) y 16 % (D). Aunque este desbalance es similar con la distribución reflejada en la población general, representa un desafío importante para el entrenamiento del modelo, especialmente en la clasificación correcta de mamografías con densidad A o D, debido a que tienen mucha menos presencia en el entrenamiento.

Otra etiqueta que vemos reflejada en el archivo csv es si la mamografía pertenece al pecho izquierdo o derecho. Esta información no es muy importante para el objetivo de clasificación del modelo. Además, como podemos ver en la Figura 5.4 esta propiedad no está casi desbalanceada, esto es debido a que a cada paciente le corresponde al menos dos mamografías por cada pecho, como se ha explicado anteriormente.

[2]American College of Radiology. Breast Imaging Reporting and Data System (BI-RADS). 5th Edition.

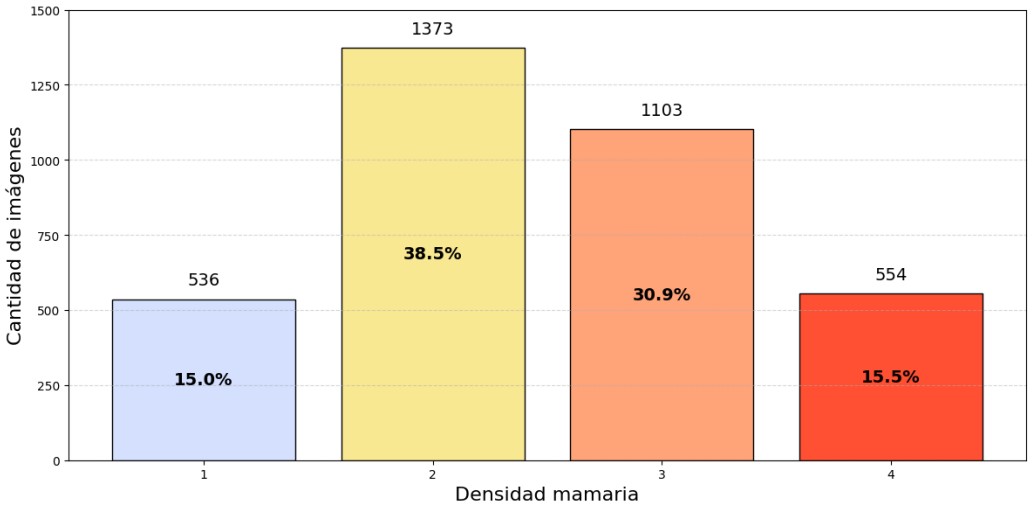


Figura 5.3: Distribución de la densidad mamaria en el conjunto CBIS-DDSM.

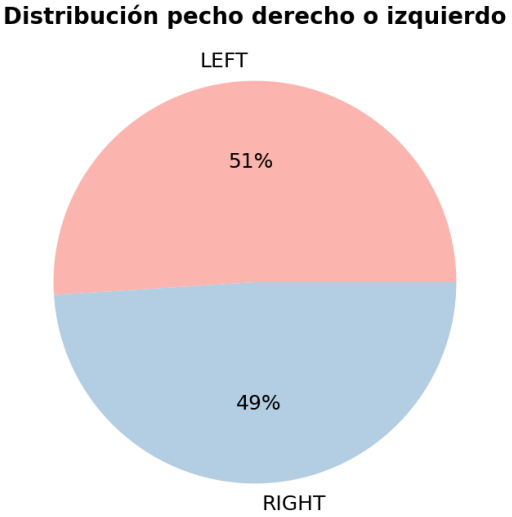


Figura 5.4: Distribución de a que pecho pertenece la mamografía

No todas las mamografías tienen la misma orientación del pecho. Como se puede ver en la Figura 5.5 los pechos pueden estar orientados generalmente para la izquierda o para la derecha. Si bien es cierto que, como veremos posteriormente en el apartado de data augmentation 5.4.1, tener diferentes orientaciones de pecho puede ser beneficioso para el modelo, ya que lo hace más robusto, se ha decidodo aplicar un algoritmo de estandarización para tener el control sobre la dirección del mismo.

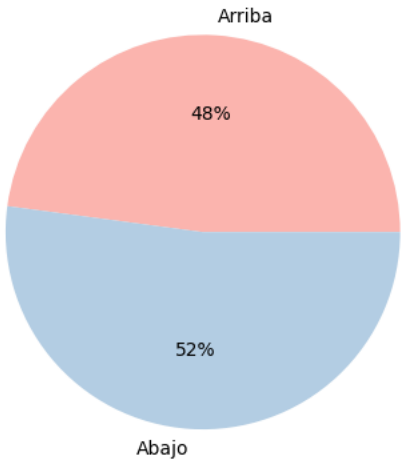


Figura 5.5: Distribución de la orientación de los pechos.

Otra característica que debe tenerse en cuenta al preparar los datos para el entrenamiento de una red neuronal convolucional es la resolución de las imágenes, así como su tamaño en disco. El formato DICOM permite conservar una gran cantidad de información y una resolución muy elevada, pero esto también conlleva un coste computacional considerable. Una mayor resolución implica que la red podrá captar más detalles en las imágenes, lo que puede ser beneficioso para identificar patrones sutiles en tejidos mamarios. Sin embargo, también incrementa notablemente el uso de memoria y el tiempo de procesamiento, dificultando el entrenamiento si no se dispone de los recursos computacionales adecuados. Por otro lado, reducir en exceso la resolución puede provocar pérdida de información clave, especialmente en un contexto médico donde pequeños detalles pueden ser determinantes. Es por ello que se debe encontrar un equilibrio que permita mantener la mayor cantidad de información relevante posi-

ble, sin comprometer la eficiencia del entrenamiento.

La Figura 5.6 muestra dos curvas de distribución normal (campanas de Gauss) superpuestas, correspondientes a la anchura (en color verde) y a la altura (en color azul). Estas curvas permiten observar la tendencia central y la dispersión de cada dimensión. En este análisis se detectó una fuerte heterogeneidad en el tamaño de las imágenes. La anchura presenta una media de 4919.6 píxeles y una desviación estándar de 635.7, mientras que la altura tiene una media de 2176.9 píxeles con una desviación estándar de 624.9. Estas desviaciones son elevadas en relación con sus respectivas medias, lo cual indica que las imágenes presentan una gran variabilidad de resolución y no siguen un formato homogéneo.

Estas curvas se han obtenido calculando la media (μ) de la anchura y la altura y su desviación estándar (σ). Finalmente, se crea una curva normal en correspondencia con estos parámetros:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

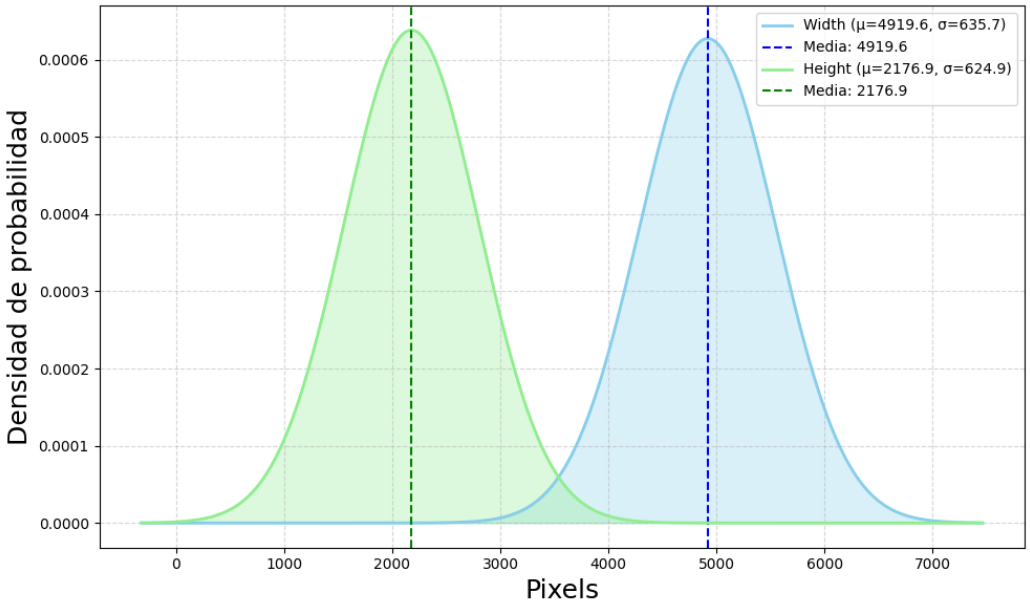


Figura 5.6: Distribuciones normales de anchura y altura de las imágenes en el conjunto CBIS-DDSM.

La segunda figura corresponde a un diagrama de dispersión (scatter plot), donde cada punto representa las dimensiones de una imagen. Sobre este gráfico se ha añadido un mapa de calor (heatmap) de fondo que resalta las zonas con mayor densidad de puntos, es decir, las resoluciones más frecuentes dentro del conjunto. Esta visualización refuerza la conclusión obtenida con las curvas de Gauss: existen múltiples resoluciones distintas, aunque se puede identificar una región de concentración en torno a ciertos tamaños específicos. No obstante, el conjunto global carece de uniformidad en términos de resolución.

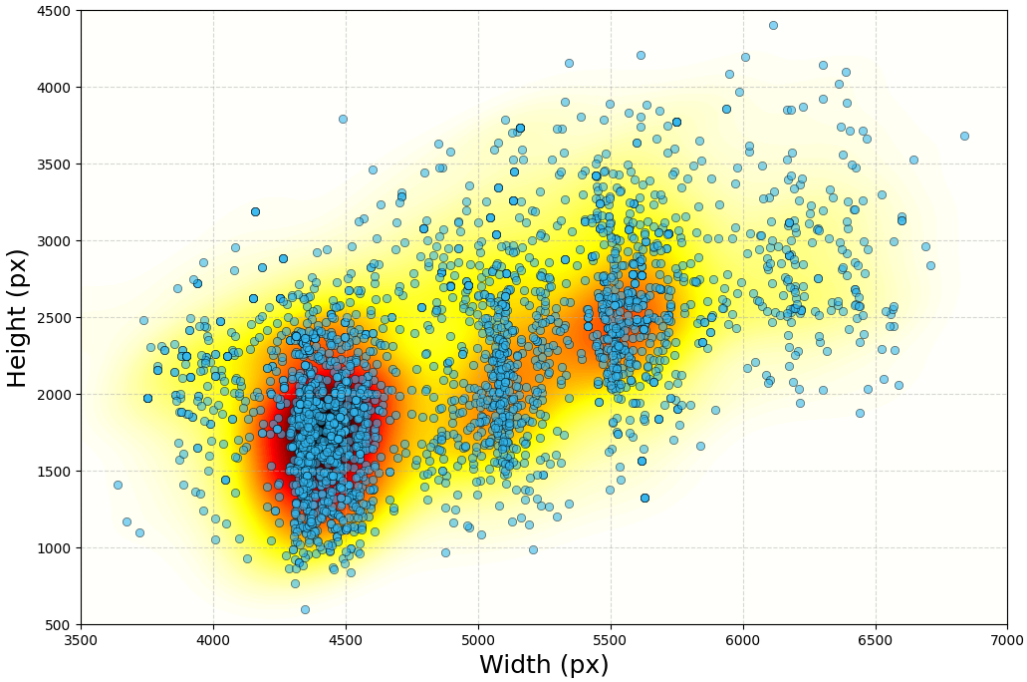


Figura 5.7: Distribución de las dimensiones de las imágenes en el conjunto CBIS-DDSM.

El hecho de que las imágenes presenten tanta dispersión en su resolución puede perjudicar al modelo. Ya que las imágenes tendrán que tener todas la misma resolución para ser una entrada válida del modelo. Por lo tanto, al hacer modificaciones de resolución tan exageradas, se pueden perder detalles o se pueden producir deformaciones en las imágenes debido a que no le corresponde el *aspect ratio*. En este caso, el aspect ratio es de 2.26. Siendo las imágenes, de media, 2.26 veces más anchas que altas.

5.2. RSNA

La segunda base de datos empleada en este trabajo es la versión procesada del conjunto de datos **RSNA Breast Cancer Detection**. Este repositorio ofrece mamografías obtenidas del desafío organizado por la Sociedad Radiológica de Norteamérica [3] (RSNA), el cual recoge estudios de cribado de cáncer de mama provenientes de Estados Unidos y Australia. El con-

[3]Sociedad Radiológica de Norteamérica. Disponible en: `https://www.rsna.org/`

junto original de RSNA consta de más de 54000 mamografías, con imágenes etiquetadas por radiólogos, incluyendo evaluaciones y resultados patológicos para casos sospechosos.

Este dataset ha sido creado por Theo Viel y está disponible a través de Kaggle [4]. Este usuario procesó las imágenes de formato DICOM a formato PNG aplicando un recorte para estandarizar todas las imágenes a tres resoluciones diferentes (256×256, 512×512 y 1024×1024 píxeles), se ha escogido la resolución 512×512 debido a que es una resolución modesta que balancea la carga computacional, con esta gran cantidad de imágenes especialmente, a la vez que permite detectar la densidad mamaria.

Análisis exploratorio

En esta base de datos viene incluido un archivo en formato csv que contiene información de cada mamografía. El archivo consiste en 14 columnas de las cuales únicamente vamos a utilizar 6. En el Cuadro 5.2 se detalla cada columna y una breve descripción de la misma.

Columna	**Descripción**
site_id	Identificador del hospital del cual proviene la imagen.
patient_id	Identificador del paciente.
image_id	Identificador de la imagen.
machine_id	Identificador de la máquina que hizo la radiografía.
density	Densidad mamaria en escala BI-RADS.
implant	Valor binario que indica la presencia de implante.

Cuadro 5.2: Descripción del significado de cada columna utilizada. Archivo csv con el dataset RSNA.

Si bien es cierto que la base de datos contiene una gran cantidad de imágenes, no todas ellas poseen un valor en la columna correspondiente a la densidad. En concreto, de las 54706 imágenes, tan solo 29470 imágenes

[4]Repositorio de Kaggle: `https://www.kaggle.com/datasets/theoviel/rsna-breast-cancer-512-pngs`.

poseen este atributo, lo que representa aproximadamente un 53.87 % de las imágenes totales. Por lo tanto, descartamos el resto de imágenes que no van a poder ser utilizables.

Este subconjunto de imágenes que sí contienen un valor de densidad será nuestro conjunto de datos a explorar. Este subconjunto son muestras de 5809 pacientes diferentes, donde el 46.7 % de ellos tiene asociadas 4 mamografías, dos por cada pecho, una con proyección CC y otra con proyección MLO (Véase la sección 3.1). El 88 % de los pacientes tiene un máximo de seis mamografías en el dataset. El valor más alto consta de catorce mamografías por paciente; este número solo lo tiene un paciente, por lo que no supone una preocupación en el sesgo de los datos.

La columna *site_id* contiene dos identificadores diferentes en todo el dataset, lo cual quiere decir que todas las imágenes provienen de únicamente dos centros. En cambio, la columna *machine_id* contiene diez identificadores distintos. Estos valores son importantes para el entrenamiento de redes neuronales debido al fenómeno conocido como *domain shift* [29], donde algunos factores como diferencias en protocolos de imagen, configuraciones de equipo, cuestiones demográficas, calibración de la máquina y modelo pueden sesgar el entrenamiento del modelo. Incluso, diferencias en máquinas del mismo modelo y mismo proveedor pueden tener ligeras diferencias que pueden sesgar al modelo.

No obstante, analizando los datos, vemos que todas las imágenes que contienen un valor asociado a la densidad mamaria provienen del mismo centro médico. En este caso con identificador 1. Por lo que esta columna no será un problema.

Analizando la columna *machine_id* no tenemos la misma suerte. De las diez máquinas presentes en todo el conjunto de datos, siete aún forman parte del subconjunto de datos que presentan valores en la columna de densidad. Como se ha explicado anteriormente, diferentes máquinas pueden dar imágenes diferentes. En la Figura 5.8 se observa una muestra de una mamografía por cada máquina, donde podemos ver que el estilo de imágenes se diferencia claramente en dos tipos que, en la figura, están manualmente separados en dos filas. Cada mamografía tiene como título el

identificador de la máquina.

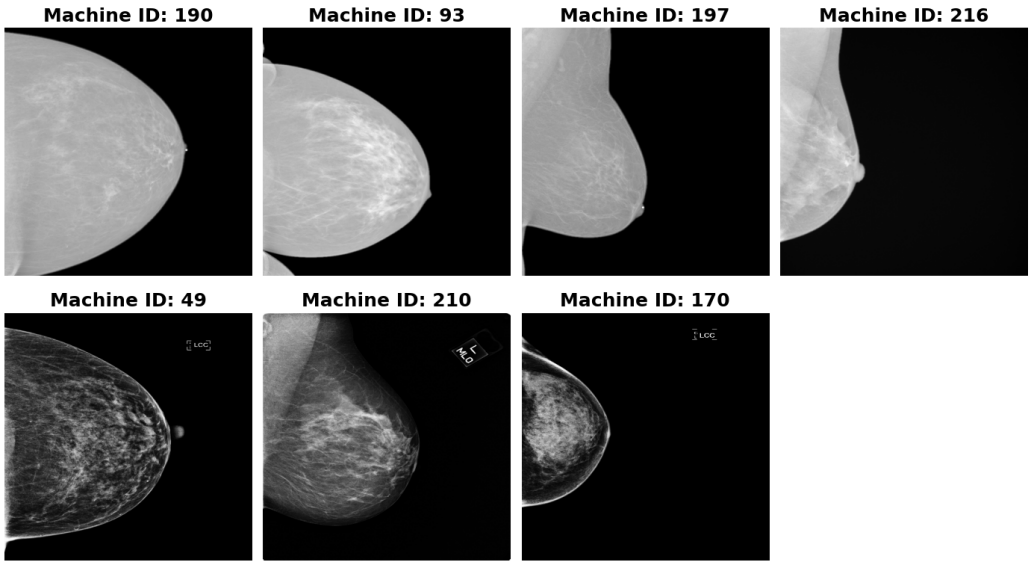


Figura 5.8: Ejemplo de una mamografía por cada máquina del subconjunto de RSNA.

En la Figura 5.8 se aprecian dos tipos de mamografías. Las mamografías de la fila superior tienen una intensidad media mayor que las de la fila inferior. Esta diferencia es notablemente apreciable en el histograma de las imágenes. Por este motivo, se ha calculado el histograma medio de las máquinas de la fila inferior y de la fila superior. En la Figura 5.9 se aprecia la diferencia de intensidades en las imágenes. Esto es una gran diferencia a nivel de intensidades; en este caso, las imágenes solo tienen un canal, por lo tanto, la intensidad es la única fuente de información que contiene la imagen, además de la relatividad posicional de los píxeles.

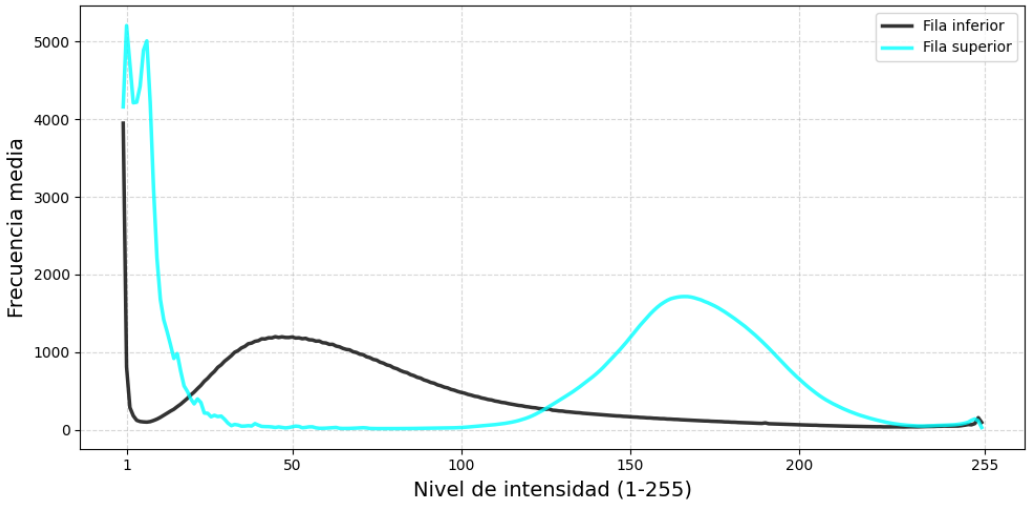


Figura 5.9: Frecuencia media por intensidad de las imágenes del subconjunto de RSNA.
Nota: *Como se aprecia en la gráfica, los valores de intensidad descartan el 0, esto es debido a que las mamografías tienen este valor en abundancia (fondo negro), dejando el resto de frecuencias imperceptibles en la gráfica.*

De todas formas, es interesante ver cuántas imágenes les corresponde a cada máquina. Ya que, en caso de querer incluir todas en el dataset de entrenamiento, o solo las de una de las dos filas, en caso de no tener importancia la fila que elegir, la cantidad de imágenes siempre tiene que ser la mayor posible. En la Figura 5.10 se muestra la distribución de imágenes por máquina. De las 29470 imágenes, 23488 provienen de la máquina 49 (un 79.7 %), en comparación, son diez veces más imágenes que el conjunto CBIS-DDSM completo.

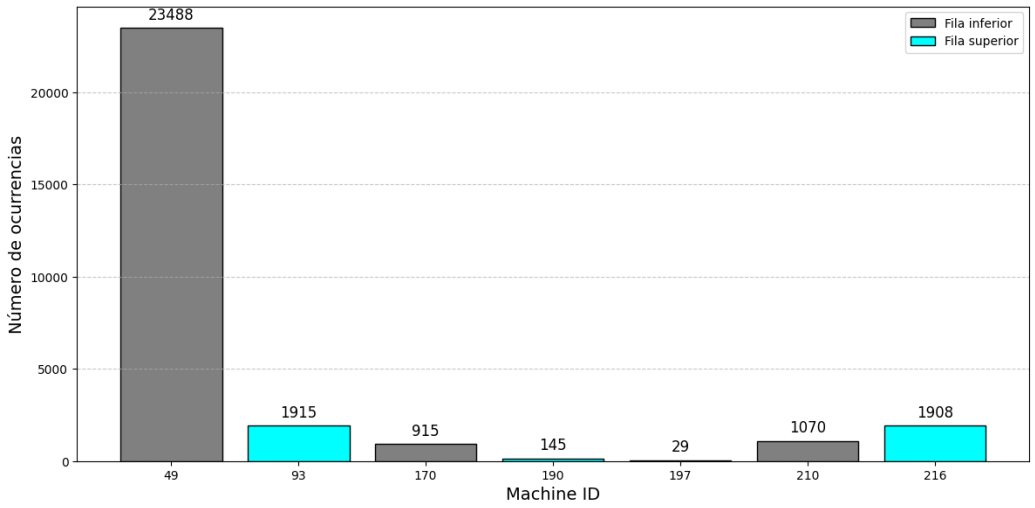


Figura 5.10: Cantidad de imágenes por máquina del subconjunto de RSNA.

Por último, existe una propiedad de valor binario que nos indica si el paciente tiene un implante mamario. Los implantes pueden no solo dificultar la visualización del tejido durante el cribado mamográfico, sino también estar asociados con cambios en la textura y la apariencia de la imagen, lo cual complica aún más su análisis automatizado. Según Rocky Mountain Cancer Centers, aunque no existe una relación directa entre los implantes y el cáncer de mama, estos sí pueden ocultar tejido mamario durante la mamografía y requerir técnicas especiales para su adecuada evaluación radiológica [30]. Por tanto, para evitar sesgos y variaciones innecesarias en la intensidad de las imágenes, se han excluido estas imágenes del conjunto de datos. En la Figura 5.11 se observa una mamografía con implante. En esta imagen podemos diferenciar claramente el implante y se aprecia cómo el tejido fibroglandular queda comprimido, de forma que se pierde resolución del tejido mamario y se sustituye con valores de intensidad altos que no aportan información.

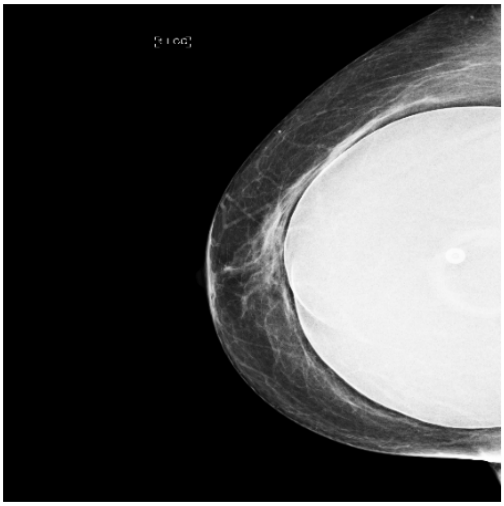


Figura 5.11: Mamografía con implante mamario.

En el subconjunto de datos donde el identificador de la máquina es el **49**, existen un total de 1340 imágenes con implante. Después de aplicar todos estos filtros para quedarnos con este subconjunto concreto por los motivos mencionados, finalizamos el filtraje de datos con un total de 22148 imágenes. Este será nuestro dataset RSNA final.

De la misma forma que con el conjunto de datos CBIS-DDSM, tenemos que comprobar la distribución de la densidad mamaria del dataset. Ya que es la característica que vamos a predecir, es especialmente importante mirar la distribución de los datos. En la Figura 5.12 observamos la distribución de esta característica en el subconjunto seleccionado. El dataset está aún más desbalanceado que el conjunto CBIS-DDSM, además, tampoco representa muy bien la distribución mundial, mencionada en la sección 3.2. De todas formas, hay más cantidad de imágenes en las clases minoritarias que en todo el conjunto CBIS-DDSM.

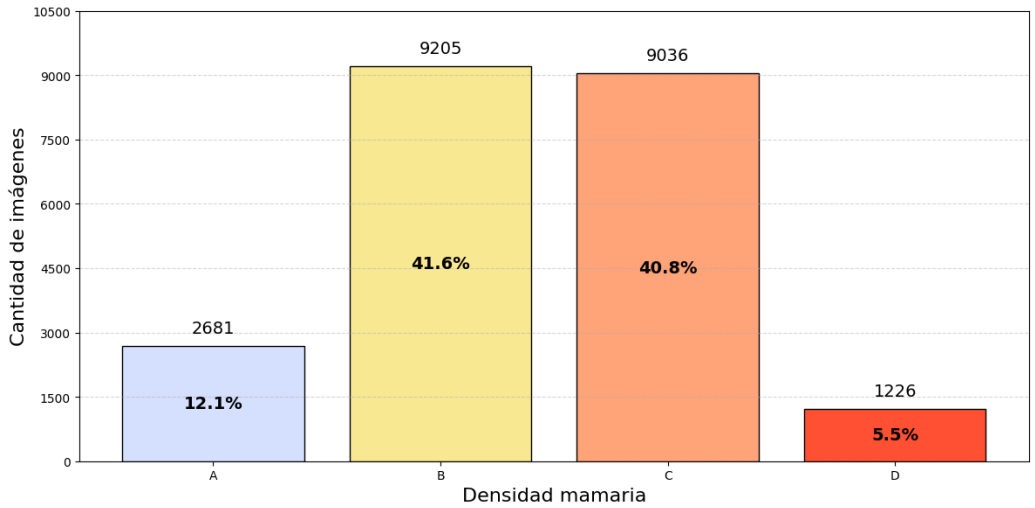


Figura 5.12: Distribución de densidad mamaria en el dataset RSNA.

En el Cuadro 5.3 están especificadas la cantidad de imágenes que pertenecen a cada escala BI-RADS según el conjunto de datos que pertenezcan (entrenamiento, validación o inferencia). Se observa que la distribución de las densidades en los conjuntos es similar entre los conjuntos de datos. Los conjuntos de datos han sido separados de la siguiente forma: 20 % de test, del 80 % restante, 85 % de train y 15 % de validación.

		Clasificación BI-RADS			
Dataset	**Imágenes totales**	**1**	**2**	**3**	**4**
Train	15060	1837 (12.2 %)	6280 (41.7 %)	6135 (40.74 %)	808 (5.36 %)
Validation	2658	332 (12.5 %)	1084 (40.78 %)	1078 (40.55 %)	164 (6.17 %)
Test	4430	512 (11.56 %)	1841 (41.56 %)	1823 (41.15 %)	254 (5.73 %)

Cuadro 5.3: Distribución absoluta y relativa (%) de imágenes por conjunto de datos y clase de densidad mamaria según la escala BI-RADS en el conjunto de datos RSNA.

5.3. Preprocesado

Antes de comenzar el proceso de entrenamiento del modelo, se aplican una serie de preprocesamientos a los datos. Esto se realiza por dos moti-

vos principales: estandarizar los datos y adecuarlos al formato más idóneo para el correcto funcionamiento del modelo.

5.3.1. Delabel

Las mamografías suelen incluir etiquetas que indican el tipo de proyección, como CC (Craneocaudal) o MLO (Medio-Lateral Oblicua). Aunque esta información es útil para los radiólogos, puede inducir sesgos no deseados en el modelo de aprendizaje profundo, ya que podría aprender a asociar ciertas características de las etiquetas con la densidad mamaria, en lugar de centrarse en las características relevantes del tejido mamario.

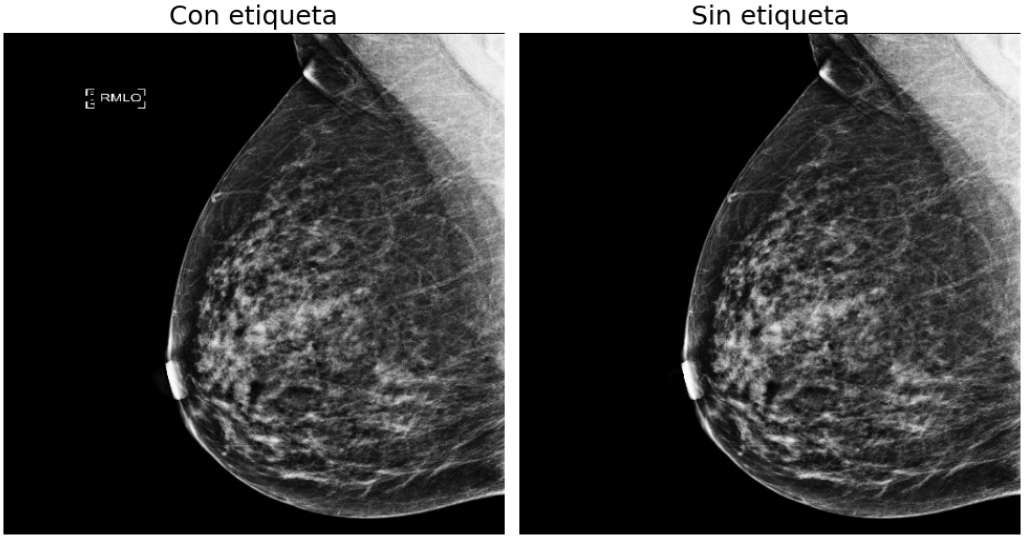


Figura 5.13: Comparación entre mamografía con etiquetas (izquierda) y sin etiquetas (derecha).

Para eliminar estas etiquetas se desarrolló un algoritmo basado en segmentación por umbral y análisis de contornos.

Primero, se binariza la imagen para detectar las regiones de interés. Luego, se identifican los contornos externos y se ordenan según su área. Generalmente, las mamografías contienen dos regiones blancas destacadas: una correspondiente al tejido mamario (más grande) y otra a la etiqueta (más pequeña). El algoritmo elimina las regiones pequeñas, hasta que solo quede la bounding box más grande, el pecho.

5.3.2. Cropping

Las imágenes originales contienen grandes áreas negras alrededor del pecho que no aportan información útil al modelo y aumentan el tamaño de entrada de manera innecesaria. Para solucionar esto, se ha aplicado un recorte automático que detecta y ajusta la imagen a la región anatómica relevante, eliminando así el máximo posible de área negra. Este proceso, conocido como *cropping*, tiene como objetivo reducir la carga computacional, acelerar el entrenamiento y mejorar la concentración del modelo en las regiones que contienen información diagnóstica, en este caso, el pecho.

En la Figura 5.14 se presenta una comparación entre una mamografía original y la misma imagen después de aplicar el algoritmo de recorte.

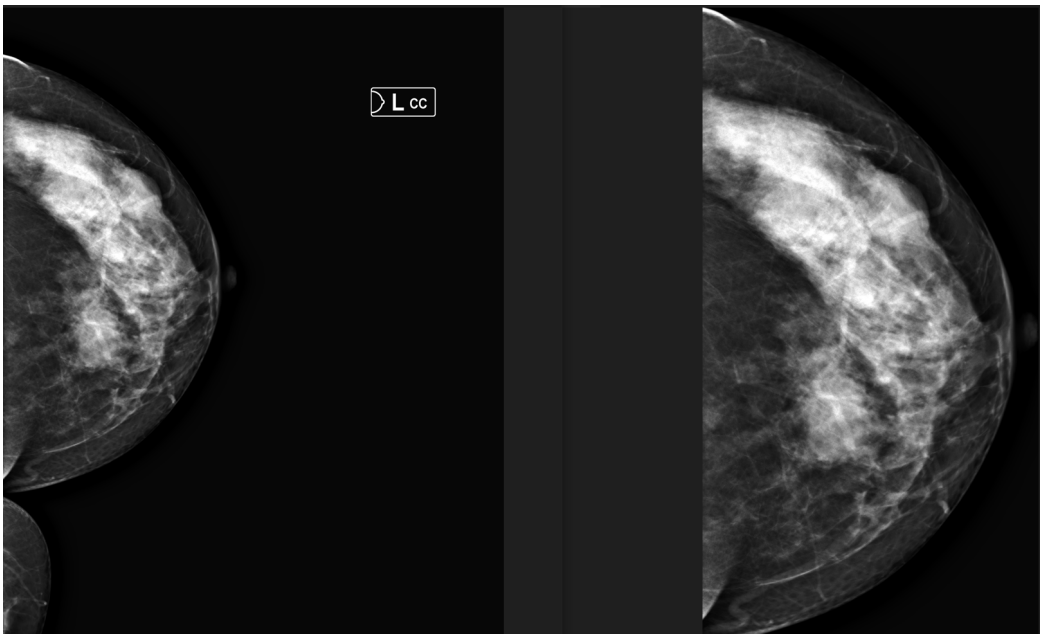


Figura 5.14: Comparación entre mamografía original (izquierda) y recortada (derecha).

El algoritmo empleado presenta una estructura similar a la descrita en la subsección anterior para la eliminación de etiquetas. En primer lugar, se aplica un filtro gaussiano con el objetivo de suavizar los bordes de la imagen. A continuación, se realiza un proceso de binarización para segmentar las regiones de interés, seguido de la detección de contornos. Finalmente,

se recorta la imagen utilizando el contorno de mayor área, asumiendo que corresponde a la región mamaria principal.

5.3.3. Orientación del pecho

Dado que las mamografías pueden estar orientadas de distintas formas, se ha normalizado la orientación de todas las imágenes para que el pecho aparezca siempre desde la misma dirección. En el caso del conjunto CBIS-DDSM el pecho proviene desde arriba, mientras que para el conjunto RSNA proviene desde la izquierda. Esta homogenización permite que el modelo aprenda patrones espaciales de manera más efectiva, ya que reduce la variabilidad espacial innecesaria entre muestras. Como se verá más adelante en la subsección 5.4.1, esta normalización parece ir en contra del propósito del data augmentation, pero el objetivo es controlar la propiedad de la orientación.

Este preprocesamiento, en particular, debe aplicarse también al conjunto de test, en caso de entrenarse con la orientación estandarizada, ya que una diferencia de orientación respecto al conjunto de entrenamiento podría afectar negativamente al rendimiento del modelo, reduciendo significativamente su capacidad de generalización.

El algoritmo utilizado para determinar la orientación correcta se basa en una observación sencilla: el lado desde el cual se origina el tejido mamario suele presentar una media de intensidad más alta. Esto se debe a la forma cóncava del pecho, que provoca una mayor concentración de píxeles blancos en esa región, incluso después del recorte. El procedimiento consiste en calcular la media de intensidad de los píxeles en ambos extremos laterales de la imagen. Aquel lado con mayor valor medio se asume como el punto de origen del pecho. En función de esta detección, se rota o no la imagen para asegurar una orientación homogénea en todo el conjunto de datos.

El motivo por el que los dos conjuntos de datos no están direccionados de la misma forma es debido a las dimensiones y orientaciones de los pechos en ambos datasets. A diferencia del dataset RSNA, CBIS-DDSM tiene una gran variación en los tamaños de las imágenes (véase la Figura 5.7). Por

este motivo se ha escogido la dimensión con mayor tamaño para orientar el pecho. En la Figura 5.16 visualizamos el algoritmo de orientación con el dataset CBIS-DDSM. Mientras que en la Figura 5.15 vemos un ejemplo del conjunto RSNA.

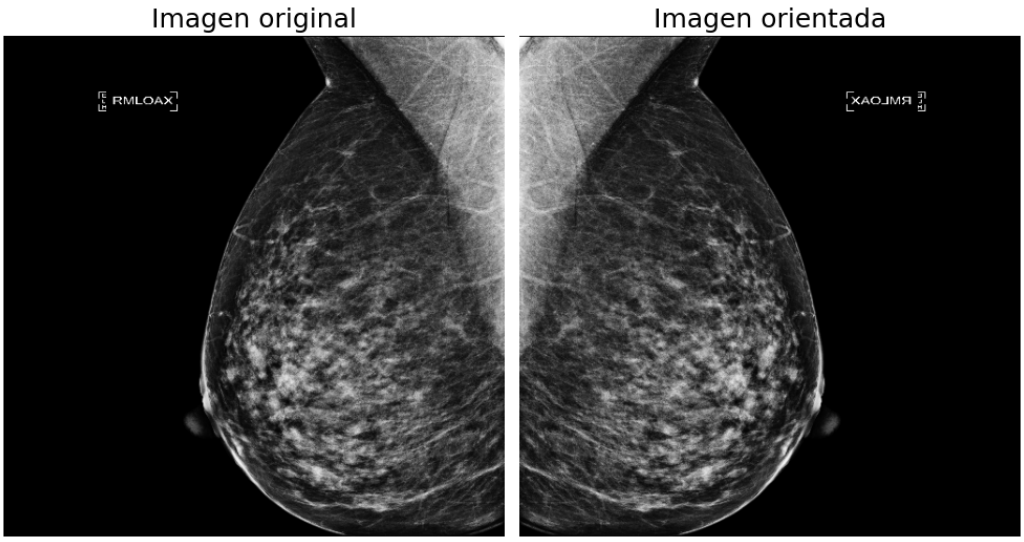


Figura 5.15: Comparación imagen original (izquierda) contra imagen orientada (derecha) conjunto RSNA.

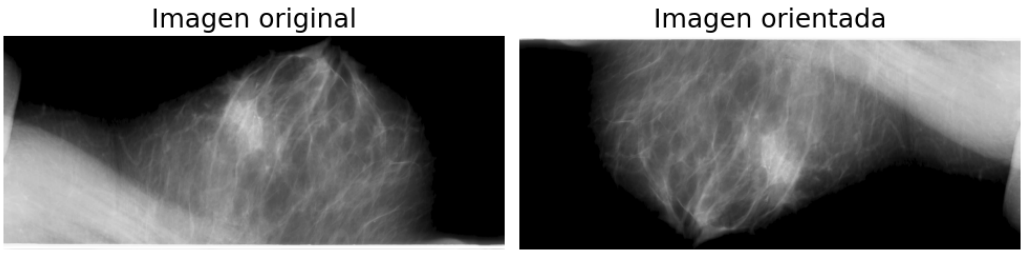


Figura 5.16: Comparación imagen original (izquierda) contra imagen orientada (derecha) conjunto CBIS-DDSM.

5.3.4. Resize

Ya sea debido al propio formato de los datos o debido a los anteriores preprocesos mencionados, las imágenes, generalmente, tienen un tamaño distinto debido a la fase de cropping. Dado que el modelo empleado re-

quiere que todas las imágenes de entrada sean del mismo tamaño, es necesario hacer un resize. Por lo tanto, después de aplicar todas estas transformaciones, se aplica un resize a la imagen. En el caso del conjunto RSNA, se aplica un resize a 224×224, ya que esta es la resolución de entrada óptima del modelo. En cambio, para el conjunto CBIS-DDSM se introduce este proceso en la fase de entrenamiento, donde el propio sistema de obtención y manipulación de imágenes hace un resize a las dimensiones deseadas.

Se ha empleado la interpolación bicúbica para redimensionar las imágenes mamográficas al tamaño de entrada requerido por la red neuronal convolucional. Esta decisión se fundamenta en la necesidad de preservar la mayor cantidad posible de información estructural y de textura del tejido mamario durante el proceso de redimensionado.

De acuerdo con Triwijoyo et al. [31], la interpolación bicúbica presenta un rendimiento superior frente a otros métodos comunes como el *Nearest Neighbour* o la interpolación bilineal. Las imágenes a las que se les aplicó un resize bicúbico presentan menor pixelación y una representación más suave de los bordes. Además, el método de interpolación bicúbica es rápido y eficiente.

Este tipo de interpolación también ha sido utilizado en estudios recientes aplicados a imágenes médicas para tareas de clasificación con redes neuronales profundas. Por ejemplo, Ahmed et al. [32] aplican esta técnica como parte del preprocesado estándar en imágenes de resonancia magnética, destacando su efectividad en la preservación de información diagnóstica. Asimismo, otros trabajos como el de Wulansari et al. [33] recomiendan la interpolación bicúbica en contextos de deep learning, argumentando que su precisión espacial ayuda a evitar distorsiones que podrían afectar negativamente el aprendizaje automático.

Todas las técnicas mencionadas se aplican a todas las imágenes del dataset, tanto imágenes de entrenamiento como de prueba. Finalmente, se obtienen los siguientes resultados notables en las figuras 5.17 y 5.18.

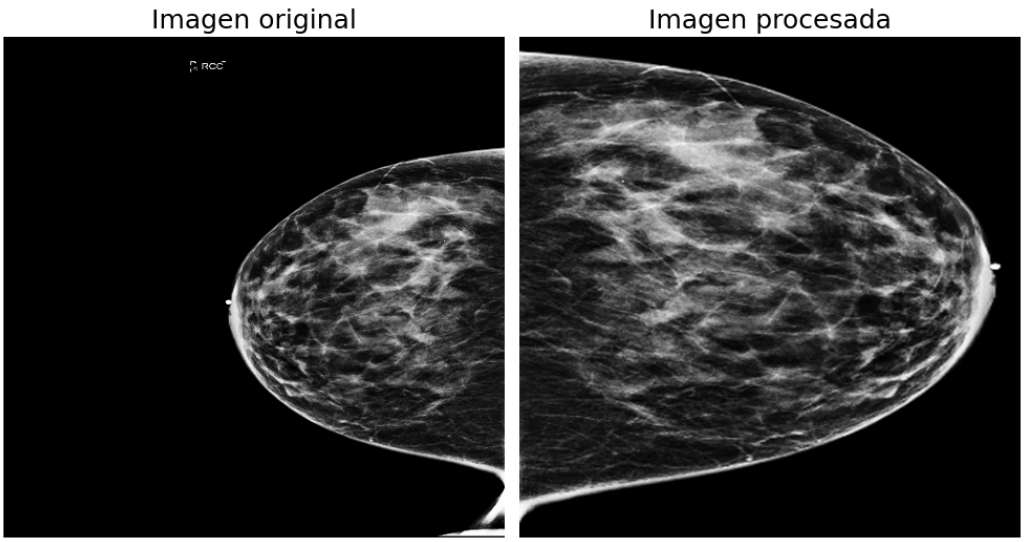


Figura 5.17: Comparación imagen original (izquierda) contra imagen procesada (derecha) conjunto RSNA.

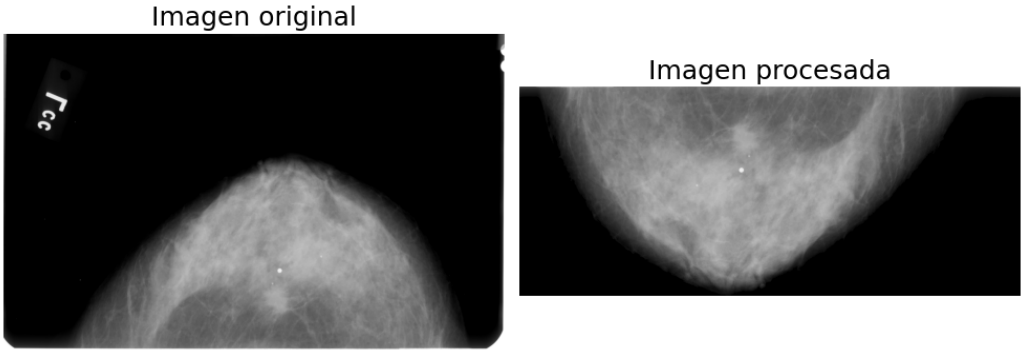


Figura 5.18: Comparación imagen original (izquierda) contra imagen procesada (derecha) conjunto CBIS-DDSM.

5.4. Definición del entrenamiento

En esta sección se expondrá la definición principal de los entrenamientos realizados. Pueden existir ligeras variaciones en algunos modelos expuestos que serán clarificadas en su descripción.

5.4.1. Data augmentation

Uno de los principales retos en el entrenamiento de redes neuronales profundas es la necesidad de grandes volúmenes de datos para que el modelo generalice correctamente y no sobre entrene. En este proyecto se dispone de dos conjuntos de datos completamente diferentes en cuanto a número de datos. En concreto, el conjunto con menos cantidad de datos dispone únicamente de 2.600 mamografías, una cantidad modesta para las necesidades de una red convolucional. Esta limitación puede derivar en problemas como el sobreajuste, en el que el modelo aprende características específicas del conjunto de entrenamiento y pierde capacidad de generalización sobre datos nuevos.

Para ayudar a combatir esta carencia de datos, se recurre a la técnica conocida como *Data Augmentation*, que consiste en aplicar transformaciones controladas sobre las imágenes originales para generar nuevas variantes que mantengan la clase original. Estas transformaciones simulan variaciones que podrían encontrarse en escenarios reales, como cambios de orientación, desplazamientos, ruido o alteraciones en el contraste. De esta forma, se incrementa la diversidad del conjunto de entrenamiento sin necesidad de recolectar nuevos datos, lo cual resulta especialmente útil en contextos médicos, donde la obtención de imágenes está restringida por cuestiones éticas, legales y logísticas.

En este trabajo se han empleado diversas técnicas de data augmentation utilizando la clase `Compose` del framework MONAI, que permite definir de forma modular y encadenada los diferentes pasos de transformación aplicados a cada imagen.

Las transformaciones aplicadas son las siguientes:

- **Transformación espejo:** Se producen transformaciones espejo en ambos ejes, verticalmente y horizontalmente.
- **Rotaciones:** Se aplica una rotación de 20º con una probabilidad del 30 %.
- **Zoom:** Se aplica un zoom, ya sea zoom out o zoom in de un 20 % con una probabilidad del 30 %.

- **Ruido Gaussiano:** Se aplica un riudo gausiano con media 0.2 y desviación estándard de 0.15 con una probabilidad del 30 %.

Es importante señalar que existen muchas otras técnicas de data augmentation que podrían haberse implementado en este trabajo. Entre ellas variabilidad de intensidad aleatoria, recortes de imagen o incluso deformaciones.

5.4.2. DenseNet121 en PyTorch

Para abordar la tarea de clasificación de densidad mamaria según la escala BI-RADS, se ha empleado la arquitectura `DenseNet121` proporcionada por la biblioteca `torchvision` de PyTorch. Esta implementación se basa en la arquitectura propuesta por Huang et al. en 2017 [5], y ha sido adaptada para facilitar su uso en diversas aplicaciones de visión por computador.

Estructura de la arquitectura

`DenseNet121` se caracteriza por una estructura compuesta por cuatro bloques densos (*dense blocks*) intercalados con capas de transición (*transition layers*). Cada bloque denso contiene un número específico de capas: 6, 12, 24 y 16, respectivamente. Estas capas están conectadas de manera densa, como se explica en la subsección 3.5.2. Las capas de transición, por su parte, se encargan de reducir las dimensiones espaciales de los mapas de características, constan de una capa de convolución y una capa de *Batch normalization* y una 1×1 capa convolucional seguida de una capa 2×2 (*average pooling*) [21]. Véase la Figura 5.19.

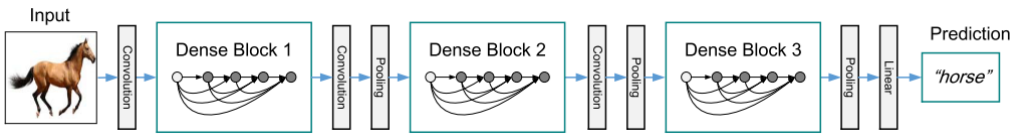


Figura 5.19: Diagrama de la arquitectura Densenet

[5]Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). *Densely Connected Convolutional Networks*. `https://arxiv.org/pdf/1608.06993`

Por último, por defecto, la resolución de entrada es de 224×224 píxeles. La implementación de Pytorch que se ha utilizado, acepta aumentar la resolución de entrada, pero esto conlleva un drástico aumento del consumo de memoria RAM, lo cual limita el tamaño del batch size. Por este motivo, se ha decidido aplicar esta resolución en el caso del conjunto de datos RSNA. En cambio, el conjunto CBIS-DDSM se ha modificado la resolución de entrada a 800×350 píxeles para conservar el ratio de aspecto de las imágenes.

La implementación en PyTorch de `DenseNet121` se encuentra disponible en la biblioteca `torchvision.models`. De forma que podemos instanciar el objeto sin y utilizarlo pasando los parámetros correspondientes.

En este proyecto, se ha optado por utilizar pesos preentrenados en el conjunto de datos ImageNet, lo que permite aprovechar características previamente aprendidas y facilita el proceso de entrenamiento en conjuntos de datos más pequeños, como es el caso de las mamografías disponibles en el conjunto CBIS-DDSM.

Ventajas de la implementación

Además, el número 121 del nombre de la arquitectura proviene del número de capas con parámetros entrenables que la componen. Torch dispone de variantes con diferente cantidad de capas, como `DenseNet161`, `DenseNet169` y `DenseNet201`, que incrementan progresivamente el número de capas y parámetros. Estas arquitecturas permiten al modelo refinar más el aprendizaje sobre los datos, permitiéndole capturar patrones más complejos y sutiles. Sin embargo, este aumento en profundidad también conlleva un mayor coste computacional y un mayor riesgo de sobreajuste, especialmente en conjuntos de datos reducidos. Por ello, se ha elegido `DenseNet121`, ya que es suficiente para nuestro caso de uso y evitamos coste computacional y un sobreajuste innecesario.

Experimentación y Resultados

En este capítulo se hará una explicación de las métricas utilizadas para la evaluación de los diferentes modelos entrenados para posteriormente mostrar los resultados obtenidos en las tres pruebas distintas realizadas. Finalmente, se hará una comparación entre los resultados obtenidos y se presentarán hipótesis y conclusiones.

6.1. Métricas de evaluación

Para evaluar el rendimiento del modelo se han utilizado varias métricas estándar en el ámbito del aprendizaje automático. Estas métricas permiten cuantificar la capacidad del modelo para realizar predicciones correctas, especialmente relevantes en contextos con clases desbalanceadas como el que presenta el conjunto de datos. Además, las diferentes métricas nos aportan diferente información que nos será útil para entender mejor cómo se comporta el modelo.

Estas métricas han sido seleccionadas siguiendo las recomendaciones habituales en tareas de clasificación multiclase con datos desbalanceados [2].

6.1.1. Accuracy

La *Accuracy* mide la proporción de predicciones correctas respecto al total de predicciones realizadas. Es una métrica sencilla que proporciona una visión general del rendimiento del modelo:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Donde TP son los verdaderos positivos, TN los verdaderos negativos, FP los falsos positivos y FN los falsos negativos. Sin embargo, cuando las clases están desbalanceadas, la exactitud puede inducir a error. Por ejemplo, un modelo que clasifique todas las imágenes como clase mayoritaria podría obtener una alta exactitud sin realizar una clasificación efectiva.

6.1.2. Precision

La precisión indica la proporción de verdaderos positivos entre todas las predicciones positivas del modelo. Evalúa cuántas de las predicciones positivas son realmente correctas. Esta métrica es especialmente importante cuando el coste de los falsos positivos es elevado. Se define como:

$$\text{Precision} = \frac{TP}{TP + FP}$$

En clasificación multiclase, que es el caso que nos concierna, este valor, por clase, es igual a la accuracy.

6.1.3. Recall

El *Recall*, también llamado *Sensitivity*, mide la proporción de verdaderos positivos detectados entre todas las instancias reales positivas:

$$\text{Recall} = \frac{TP}{TP + FN}$$

Esta métrica resulta fundamental cuando el objetivo es minimizar los falsos negativos. En nuestro caso, si el modelo no identifica correctamente mamografías con densidad altamente elevada (BI-RADS 4), puede ser problemático desde el punto de vista clínico, como se mencionó en la sección 3.3. De forma que es una métrica que tenemos que considerar importante.

6.1.4. F1 Score

La puntuación F1 combina dos métricas anteriores, *Precision* y *Recall* en una sola métrica mediante su media armónica. Es especialmente útil

cuando existe un desequilibrio entre clases y es necesario encontrar un equilibrio entre la *Precision* y *recall*:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

Este valor permite evaluar el rendimiento del modelo de forma más robusta que usando únicamente la *Precision* o el *Recall*.

6.1.5. ROC AUC y Curva ROC

La curva ROC (*Receiver Operating Characteristic*) es una representación gráfica que muestra el rendimiento de un modelo de clasificación binaria al variar su umbral de decisión. En el eje X se representa la tasa de falsos positivos (FPR), y en el eje Y la tasa de verdaderos positivos (TPR), también conocida como *Recall*. La curva ROC permite visualizar el compromiso entre la sensibilidad y la especificidad del modelo.

El área bajo esta curva (AUC, por sus siglas en inglés: *Area Under the Curve*) resume en un único valor la capacidad del modelo para distinguir entre clases. Un valor de AUC igual a 1 indica un modelo perfecto, mientras que un valor de 0.5 corresponde a un modelo que predice al azar:

$$\text{AUC} = \int_0^1 TPR(FPR)\, dFPR$$

En el contexto de esta tesis, aunque la clasificación de la densidad mamaria es un problema multiclase (cuatro clases BI-RADS), es posible calcular una curva ROC por cada clase utilizando el enfoque *one-vs-rest* (*uno contra el resto*). Esto permite generar curvas ROC individuales para cada clase de densidad mamaria, proporcionando una evaluación más detallada del comportamiento del modelo para cada categoría.

Cabe destacar que debido a que se trata de un problema de clasificación multiclase, métricas como la *Precision*, el *Recall*, o el *F1 Score* deben ser calculadas para cada clase individualmente, y posteriormente agregadas utilizando estrategias como el promedio macro (media no ponderada), micro (ponderado por el número de muestras) o ponderado (ponderado por

soporte de clase). En los resultados de esta tesis se ha optado por utilizar el promedio macro, ya que permite evaluar de forma más justa el rendimiento sobre clases desbalanceadas. Además, podemos no hacer el promedio y tener esta métrica por cada clase, que también es de mucha utilidad para entender su comportamiento y mejorarlo.

6.2. Modelos entrenados con CBIS-DDSM

A continuación se expondrán tres modelos entrenados con el conjunto CBIS-DDSM. Por cada prueba corresponde una explicación de la misma, las correspondientes métricas de entrenamiento y de test y, finalmente, unas conclusiones finales.

Los hiperparámetros usados en cada prueba se conservan en la siguiente, a no ser que se indique lo contrario.

6.2.1. Baseline

Este modelo corresponde a una primera prueba sin técnicas adicionales, para ver a grandes rasgos cómo se comporta el modelo. Con el objetivo de obtener una línea base de referencia, los modelos posteriores tratarán de implementar mejoras. Las especificaciones del entrenamiento son las siguientes:

- Preprocesado: Cropping, Delabel y orientación (véase Sección 5.3).
- Optimizador: Adam, Learning rate: 10^{-4}.
- Batch size: 30.
- Image size: 800×350.
- Epochs: 60.

En la Figura 6.1 observamos que el entrenamiento sigue un curso normal. Aunque la accuracy en validación se estabiliza demasiado rápido, sobre la epoch 20 la accuracy parece estabilizarse, obteniendo un pico de

0.75 poco después. Además, aunque veamos una pérdida que aún se puede estabilizar, el modelo llega a un 0.95 de accuracy en el entrenamiento. Esto es un claro símbolo de sobreajuste, que además se respalda con la prueba de inferencia.

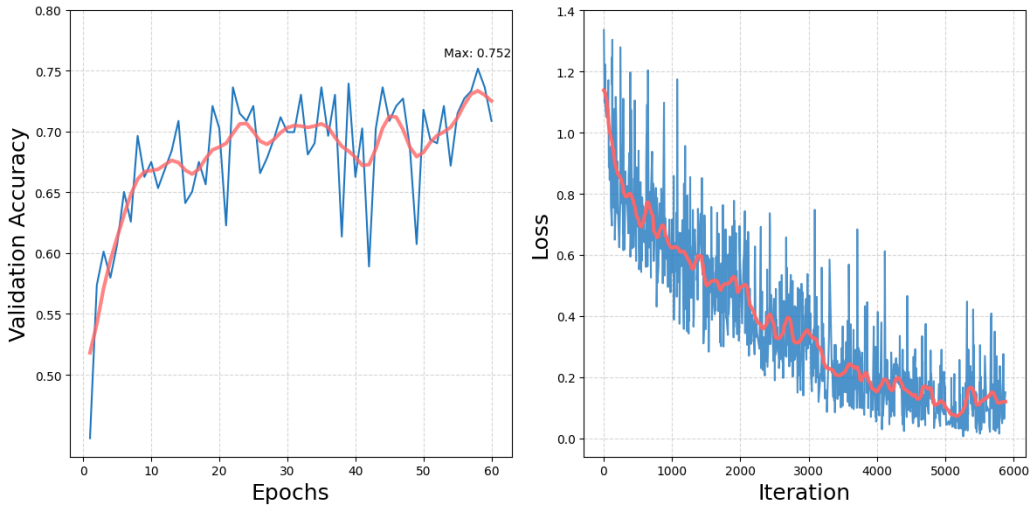


Figura 6.1: Accuracy en validación (izquierda) y pérdida (derecha) durante el entrenamiento básico en el conjunto CBIS-DDSM.

Como se observa en la Figura 6.2 si bien es cierto que se alcanza casi un 80 % de accuracy y precisión en la categoría BI-RADS C, las categorías con una menor presencia de datos en el conjunto (categorías A y D) presentan una accuracy de menos del 50 %. Este modelo no cumpliría el objetivo, ya que, como se mencionó en la sección 3.3, los casos de densidad mamaria extremadamente densos son prioritarios para su clasificación. De todas formas, hay una gran diferencia entre la clase D y la clase C en cuanto al recall. Siendo prácticamente opuestos entre ellos, esto señala que una predicción de clase D es menos fiable que una predicción de clase C; sin embargo, el modelo ha acertado en más proporción las predicciones de la clase extremadamente densa.

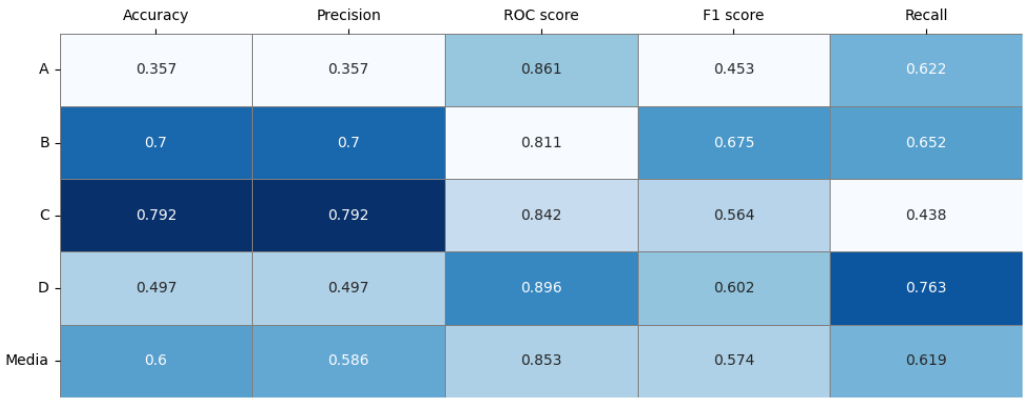


Figura 6.2: Tabla de métricas por clase y media macro del modelo baseline con el conjunto CBIS-DDSM.

En la matriz de confusión mostrada en la Figura 6.3 se aprecia la tendencia del modelo a clasificar las muestras como categorías B y C. Gracias al mapa de calor, se aprecia mayor densidad del color azul oscuro en las filas correspondientes a estas clases. Este comportamiento se puede explicar debido al desbalance del conjunto de datos.

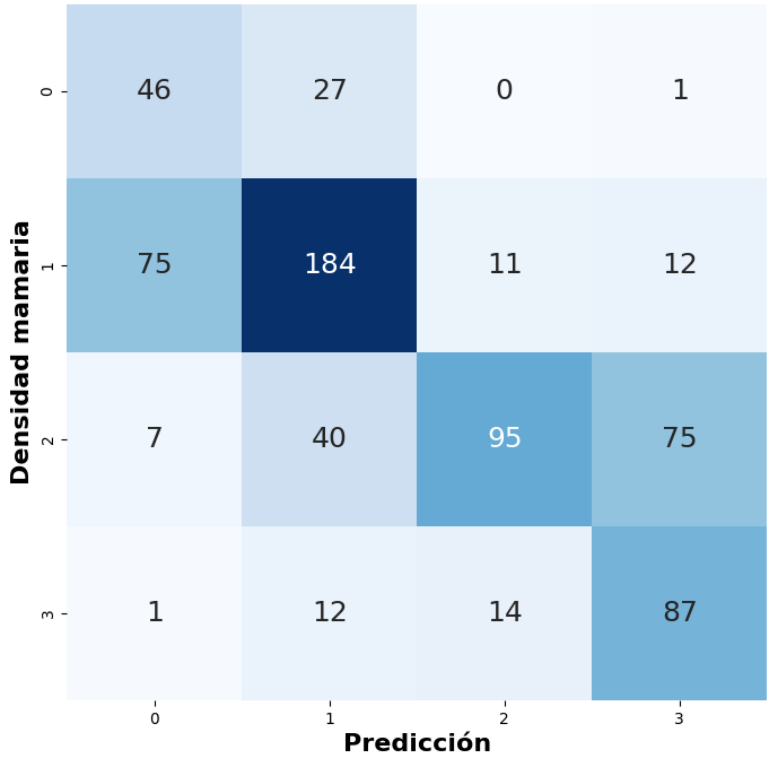


Figura 6.3: Matriz de confusión de la inferencia con el modelo base en el conjunto CBIS-DDSM.

6.2.2. Data augmentation

En esta segunda prueba se ha entrenado el modelo usando data augmentation (véase la subsección 5.4.1). El propósito es ralentizar el overfitting a la vez que conseguir una mejor abstracción de las características, de forma que la inferencia tenga un mejor rendimiento.

Las métricas presentadas en la Figura 6.4 muestran un comportamiento similar al anterior, aunque más prolongado. La pérdida en el entrenamiento converge de forma ligeramente más lenta, pero la accuracy en validación sigue encontrando la estabilidad muy temprano, alcanzando un valor superior (0.81) que en la anterior prueba, lo cual puede indicar una mejor generalización de los datos.

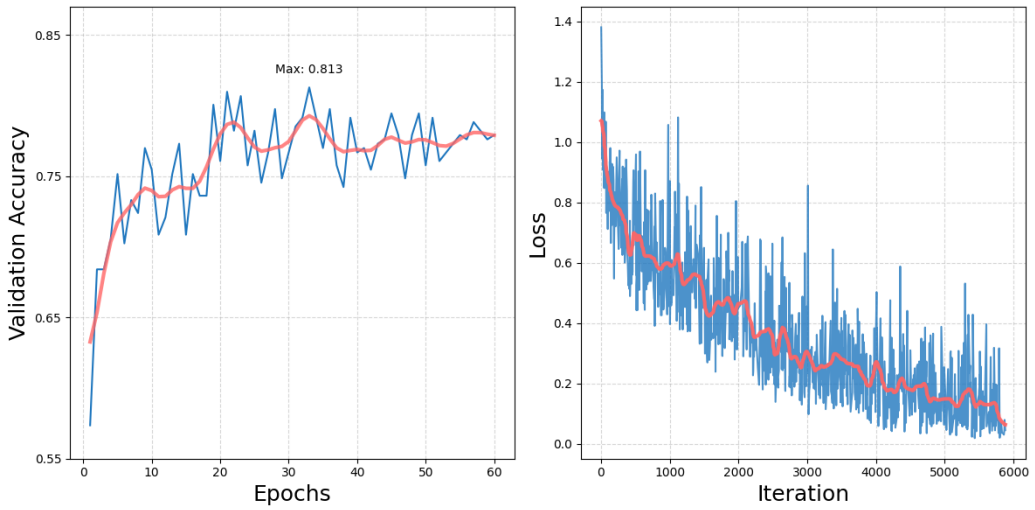


Figura 6.4: Accuracy en validación (izquierda) y pérdida (derecha) durante el entrenamiento con data augmentation en el conjunto CBIS-DDSM.

En las métricas obtenidas de la inferencia de esta prueba mostradas en la Figura 6.5 observamos una distribución más homogénea de la accuracy, aunque la clase menos densa sigue estando por debajo del 50 % y la disminución considerable del valor de la categoría C, el modelo ya no está tan sesgado por el desbalance de clases. También podemos ver un indicativo de estabilidad en la homogeneidad que presenta la métrica F1 score. Este comportamiento también es explicable gracias al data augmentation. Precisamente, esta técnica se utiliza para simular más imágenes de las que realmente hay; de esta forma, el modelo puede generalizar mejor las características y, por lo tanto, adaptarse mejor a otro conjunto de test.

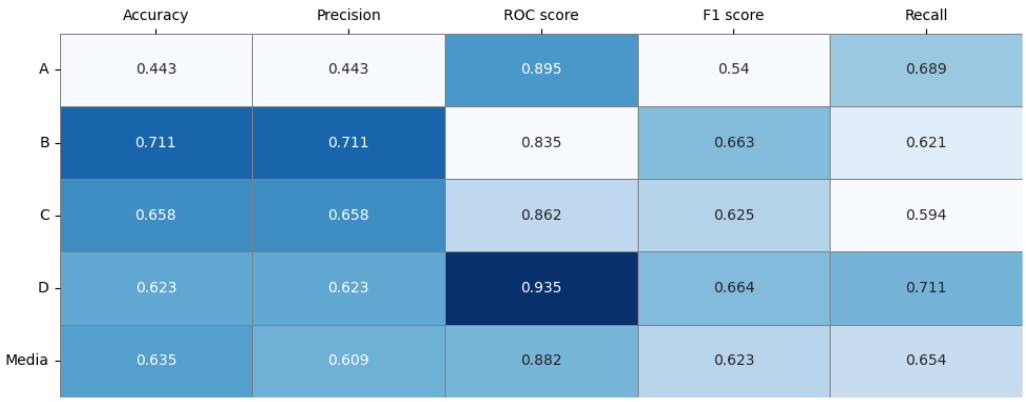


Figura 6.5: Tabla de métricas por clase y media macro del modelo utilizando data augmentation con el conjunto CBIS-DDSM.

Con la matriz de confusión mostrada en la Figura 6.6 observamos menor desviación de las predicciones respecto al valor correcto. Es decir, es menos probable que clasifique una mamografía como una densidad dos o tres categorías menor o mayor a la real.

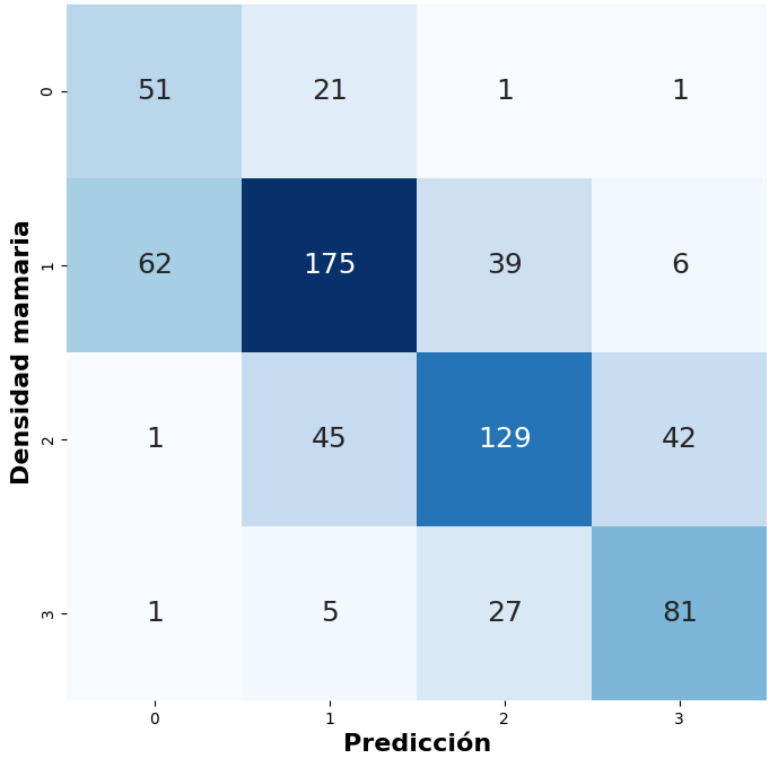


Figura 6.6: Matriz de confusión de la inferencia con el modelo utilizando data augmentation en el conjunto CBIS-DDSM.

6.2.3. Drop out

En este modelo se le añade a la CNN en el proceso de entrenamiento una probabilidad del 20 % de desactivar una fracción de las neuronas en las capas posteriores a los Dense Blocks. Esto ayuda a prevenir o ralentizar el overfitting ya que obliga al modelo a aprender otras características sin tener que depender de neuronas concretas. Adicionalmente, se ha modificado el learning rate a 10^{-5} y se ha añadido un weight decay con valor 10^{-4}. Ambas modificaciones pretenden ralentizar la convergencia del modelo y permitir mejor abstracción de las características aprendidas.

En las métricas de entrenamiento mostradas en la Figura 6.7 se aprecia menor variación en la accuracy de validación durante todo el proceso, el aprendizaje es más robusto y converge más tarde. En cuanto a la pérdida del entrenamiento, observamos cómo la convergencia se prolonga mucho.

Llegando a no converger en las epochs especificadas. Todas estas observaciones son coherentes con los cambios realizados en este modelo.

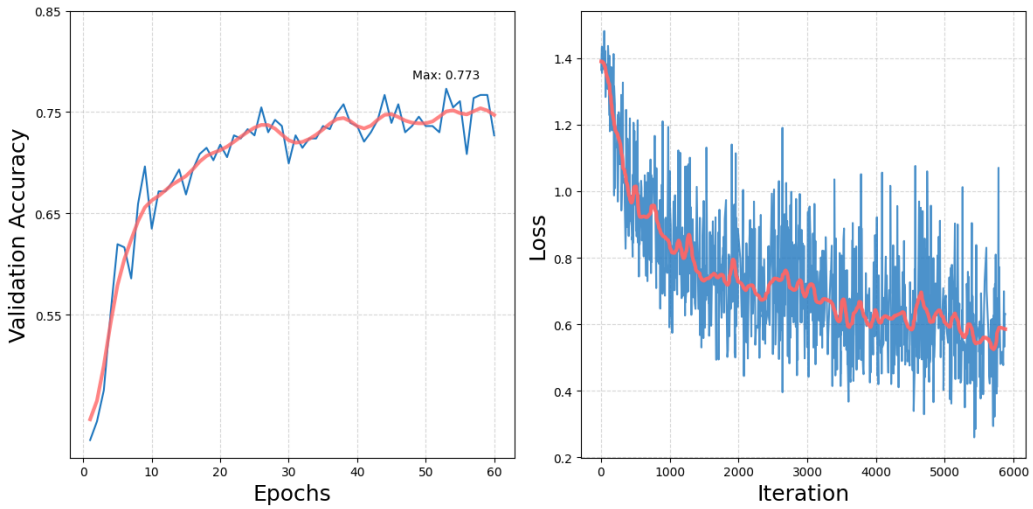


Figura 6.7: Accuracy en validación (izquierda) y pérdida (derecha) durante el entrenamiento con drop out en el conjunto CBIS-DDSM.

Las métricas obtenidas en el conjunto test presentan un valor de accuracy del 0.78 en la clasificación de mamografías con densidad D en la escala BI-RADS. Es un aumento de hasta el 28 % de accuracy. Sin embargo, se observa una notable caída en el recall de 0.15 puntos. Como se explica en la sección 6.1 el recall es más importante que la accuracy en el caso de clasificar mamografías con densidad mamaria extremadamente densa (véase sección 3.3). Aunque menos importante para nuestro caso, es importante notar que la accuracy de las predicciones en la clase A también ha disminuido en comparación con el modelo anterior, pero el recall ha aumentado en 0.135 puntos.

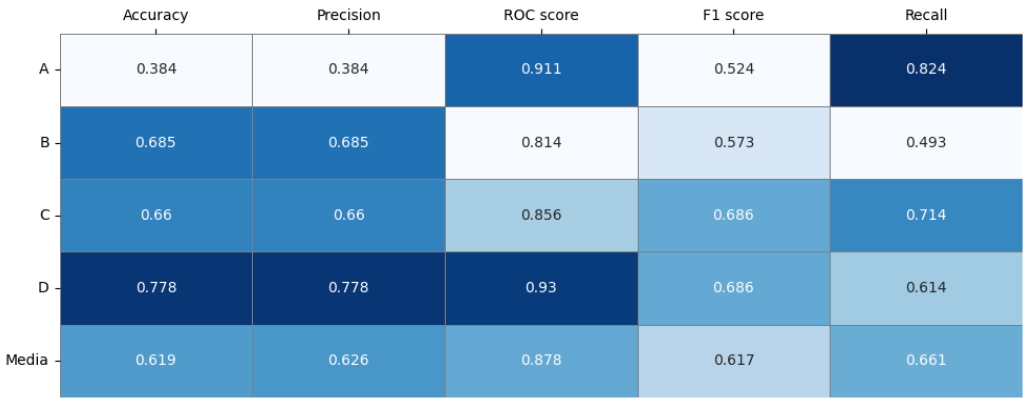


Figura 6.8: Tabla de métricas por clase y media macro del modelo utilizando drop out con el conjunto CBIS-DDSM.

Por último, en la matriz de confusión vemos que el modelo tiende cada vez más a clasificar las mamografías como una clase más alta que la real.

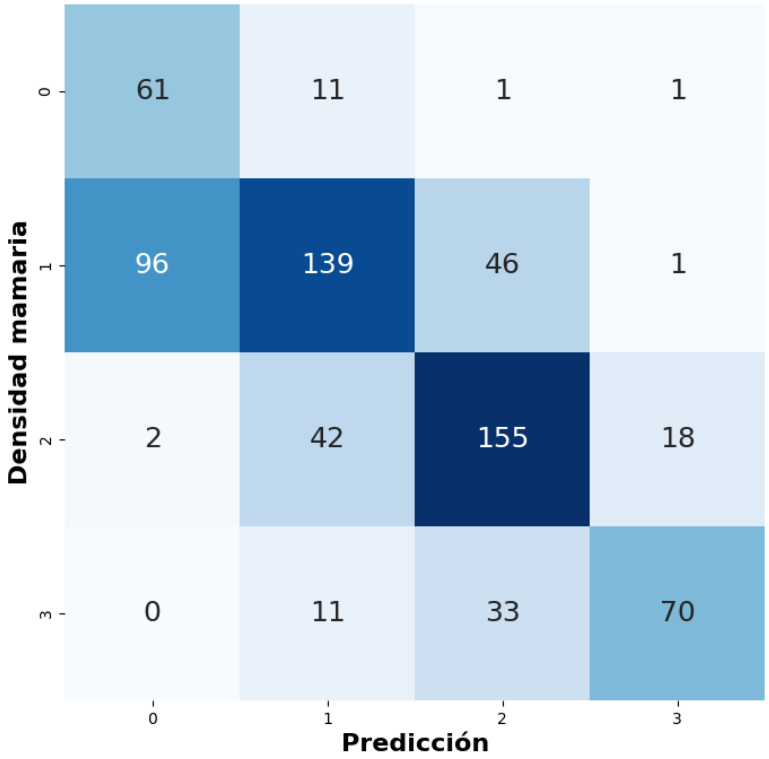


Figura 6.9: Matriz de confusión de la inferencia con el modelo utilizando drop out en el conjunto CBIS-DDSM.

6.2.4. Conclusiones

El rendimiento de los modelos ha sido limitado. No se ha superado el 0.65 de accuracy media en ningún modelo, aunque se han obtenido valores cercanos al 0.8 tanto en accuracy como en recall, pero no en ambos. En la Figura 6.10 se observa cómo el cambio más importante es dado por la técnica de data augmentation, lo que corrobora la importancia de una gran cantidad y variedad de datos para crear un modelo de aprendizaje automático. Además, con las matrices de confusión y las métricas de la inferencia se observa el impacto del desbalance de los datos de entrenamiento en el rendimiento de las redes neuronales.

Por último, es importante recalcar que los valores de la métrica ROC score son cercanos al 0.9. Esto quiere decir que el modelo distingue bien los casos a clasificar. Además, es coherente con el mapa de calor que observamos en la matriz de confusión, donde nos indica que el modelo tiende a equivocarse con la clase adyacente. Por consiguiente, aparece la hipótesis de que la clasificación BI-RADS, al tener un componente subjetivo, es posible que los casos que se encuentren en la línea entre una clasificación y otra sean los más frecuentemente fallidos por la red, de la misma forma que puede ocurrir entre dos doctores.

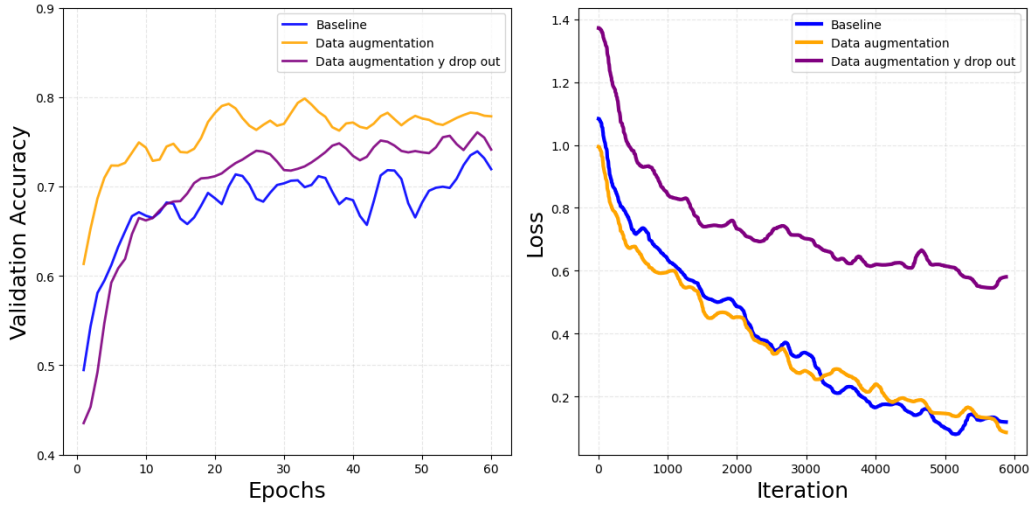


Figura 6.10: Comparación de métricas de entrenamiento de todos los modelos con el conjunto CBIS-DDSM.

6.3. Modelos entrenados con RSNA.

En esta sección se presentan los modelos entrenados con el conjunto de datos RSNA. Los modelos presentarán la misma estructura que en la sección anterior.

Los hiperparámetros usados en cada prueba se conservan en la siguiente, a no ser que se indique lo contrario.

6.3.1. Baseline

De la misma forma que en el baseline del conjunto CBIS-DDSM, este modelo no presenta técnicas adicionales, únicamente la configuración de hiperparámetros que se ha considerado óptima después de varias pruebas. La configuración es la siguiente:

- Preprocesado: Cropping, Delabel y orientación (véase Sección 5.3).
- Optimizador: Adam, Learning rate: 10^{-4}.
- Batch size: 64.
- Image size: 224×224.
- Epochs: 50.

En las gráficas de la Figura 6.11 observamos cómo el modelo hace overfitting desde el principio, el valor máximo de la accuracy en validación se obtiene en la primera epoch, alcanzando el 0.77, lo cual es un valor relativamente alto para una primera epoch.

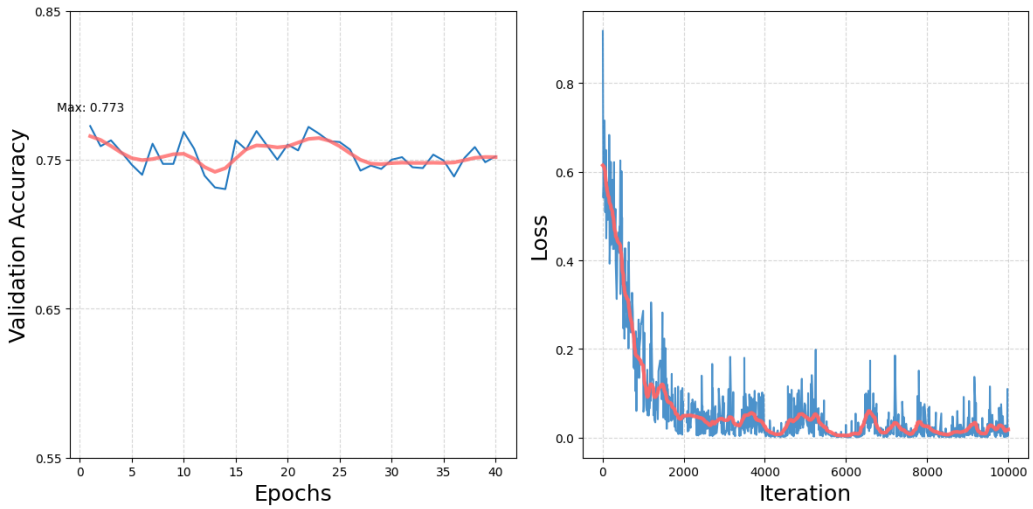


Figura 6.11: Accuracy en validación (izquierda) y pérdida (derecha) durante el entrenamiento base en el conjunto RSNA.

En las métricas de inferencia mostradas en la Figura 6.12 observamos una media macro de la accuracy superior al 0.7. Además, el recall parece ser más homogéneo, descontando la categoría D. Como era previsible, las clases minoritarias en el conjunto de datos presentan peores métricas, adicionalmente, la falta de una curva de aprendizaje y un overfitting precoz fomentan la falta de abstracción para las clases minoritarias. De todas formas, se obtienen mejores métricas con el baseline del modelo RSNA que con el mejor modelo del conjunto CBIS-DDSM.

	Accuracy	Precision	ROC score	F1 score	Recall
A	0.578	0.578	0.925	0.64	0.717
B	0.719	0.719	0.852	0.734	0.75
C	0.787	0.787	0.891	0.765	0.745
D	0.76	0.76	0.897	0.588	0.479
Media	0.725	0.711	0.891	0.682	0.673

Figura 6.12: Tabla de métricas por clase y media macro del modelo base con el conjunto RSNA.

En la matriz de confusión mostrada en la Figura 6.13 se aprecia un comportamiento similar al mencionado con el modelo que utiliza data augmentation de CBIS-DDSM, en el que las predicciones erróneas tienden a ser adyacentes al valor real. De nuevo, observando el ROC score aproximado de 0.9, vuelve a reforzar la hipótesis de la subjetividad de los doctores.

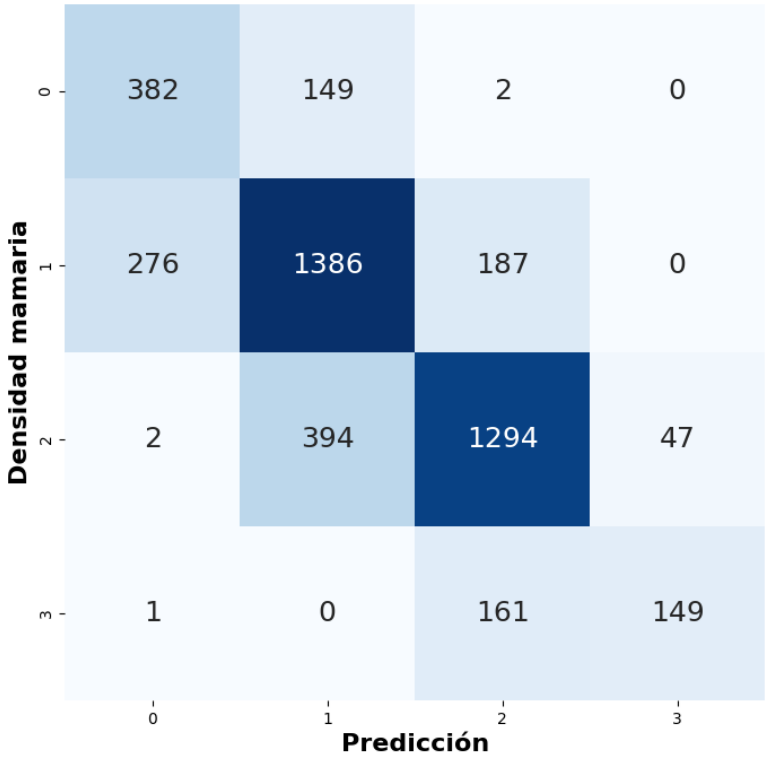


Figura 6.13: Matriz de confusión de la inferencia con el modelo base en el conjunto RSNA.

6.3.2. Data augmentation

Fijando los hiperpárametros anteriores, se añade la técnica de data augmentation propuesta en la subsección 5.4.1 con el propósito de prolongar la convergencia del entrenamiento y, a su vez, permitirle al modelo una mejor generalización para mejorar la fase de inferencia.

Como se observa en la Figura 6.14 la accuracy en validación parece no variar, lo que indica que no esta generalizando a medida que transcurre el

entrenamiento. En cambio, se aprecia una convergencia lenta por parte de la pérdida, lo que ralentiza el overfitting, correspondiendo a la aplicación de data augmentation.

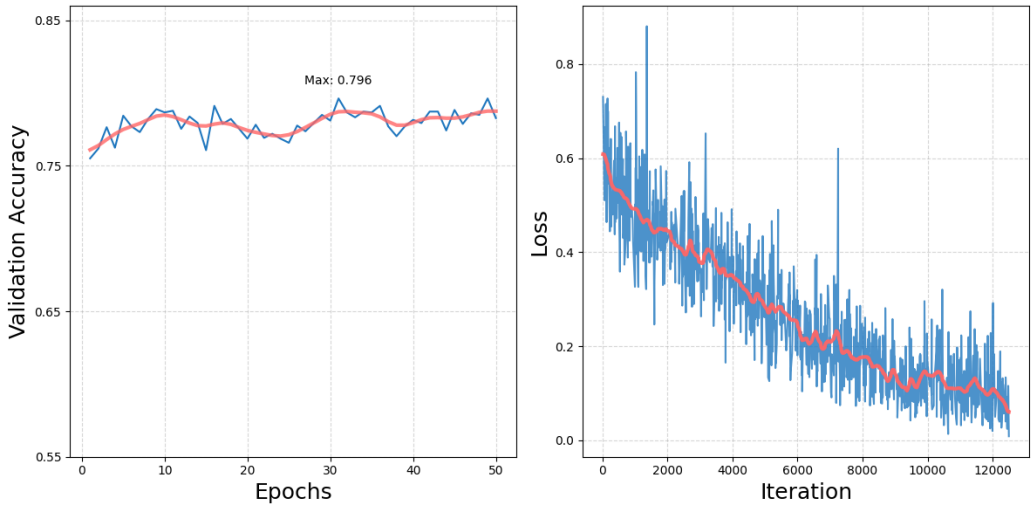


Figura 6.14: Accuracy en validación (izquierda) y pérdida (derecha) durante el entrenamiento utilizando data augmentation en el conjunto RSNA.

En las métricas de la fase de inferencia (Figura 6.15) observamos una mejor accuracy en las clases menos representadas, aumentando ligeramente la accuracy media. De todas formas, el recall en la clase D presenta un valor por debajo del 0.4, aunque una mejor accuracy y el mejor ROC score. La matriz de confusión mostrada en la Figura 6.16 respalda estos datos, mostrando una tendencia del modelo a predecir las clases B y C.

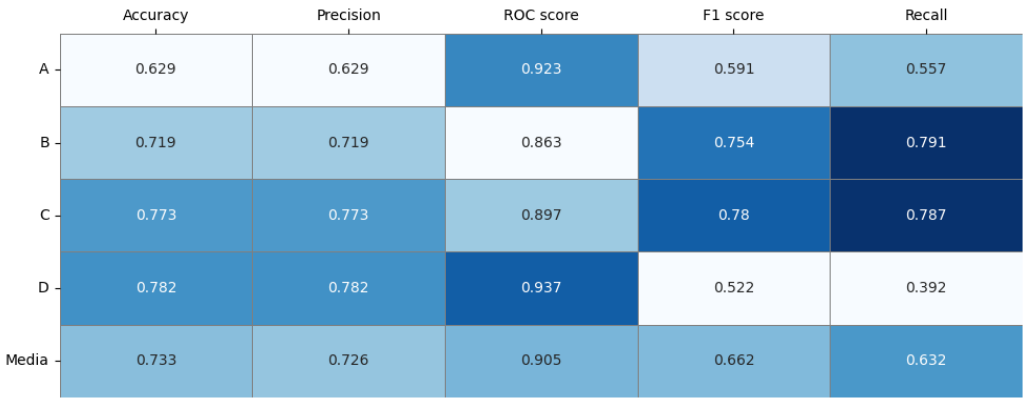


Figura 6.15: Tabla de métricas por clase y media macro del modelo utilizando data augmentation con el conjunto RSNA.

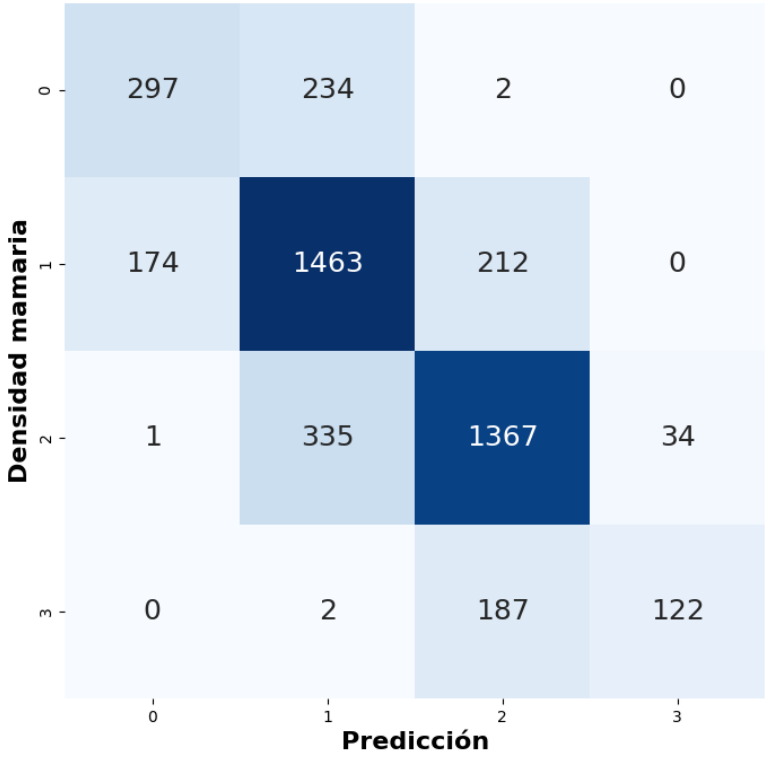


Figura 6.16: Matriz de confusión de la inferencia con el modelo utilizando data augmentation en el conjunto RSNA.

6.3.3. Drop out

En este modelo se implementó drop out con una probabilidad del 20 %, se ha reducido el learning rate a 10^{-5} y se ha añadido un weight decay con valor 10^{-4} con el propósito de ralentizar el overfitting y reforzar el aprendizaje de otras características sin depender de neuronas específicas.

En las métricas de entrenamiento (Figura 6.17) se observa una convergencia tanto de la accuracy de validación como de la pérdida del entrenamiento notablemente más lenta. Aunque alcance un punto máximo igual o menor que el baseline, es probable que el modelo generalice mejor las características en este caso, obteniendo mejor rendimiento en la inferencia.

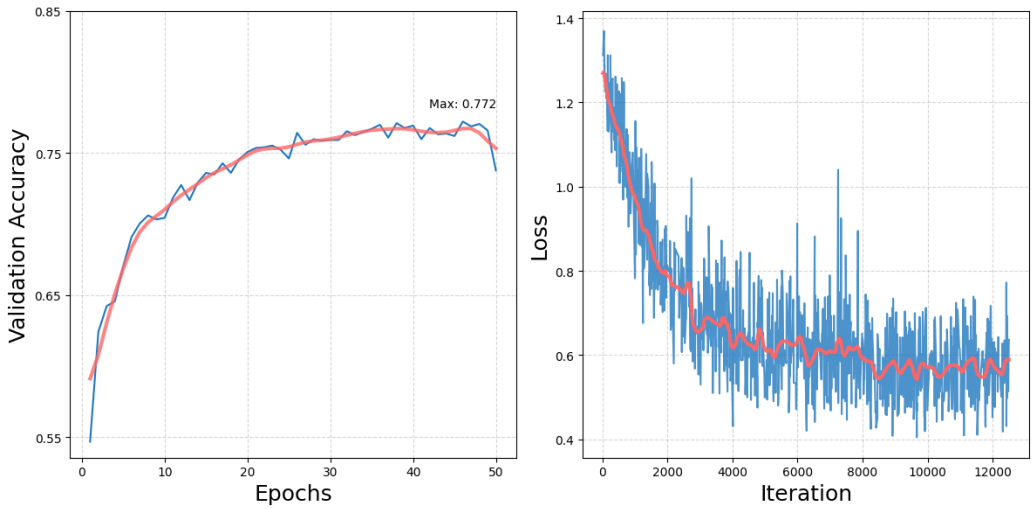


Figura 6.17: Accuracy en validación (izquierda) y pérdida (derecha) durante el entrenamiento con 20 % de probabilidad de drop out en el conjunto RSNA.

Efectivamente, como observamos en las métricas de la inferencia mostradas en la Figura 6.18 la accuracy macro media obtiene mejores valores que en el modelo baseline aunque obtuvieron valores similares en la validación. Presenciamos también un aumento en los valores de todas las métricas. Es importante mencionar que el recall en la categoría de densidad extremadamente densa (D), aunque ha incrementado, es inferior a los modelos entrenados con el conjunto CBIS-DDSM. No obstante, se obtiene un ROC score en esta clase del 0.97, lo que es casi perfecto, es decir,

existen un límite en el que el modelo predice estas clases muy bien, además, refuerza la hipótesis de que las predicciones se ven sesgadas por el desbalance de datos..

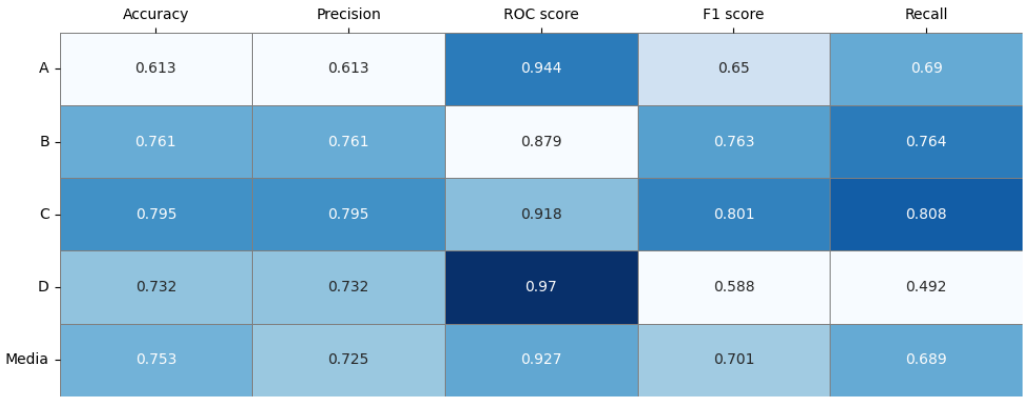


Figura 6.18: Tabla de métricas por clase y media macro del modelo utilizando drop out con el conjunto RSNA.

La matriz de confusión (Figura 6.19) corresponde a las métricas mostradas anteriormente. Si bien es cierto que hay muchos casos de densidad BI-RADS D que no predice correctamente, los predice en gran medida como su clase anterior (BI-RADS C), la métrica ROC score de 0.897 indica que la CNN está ordenando bien los casos, pero falla en la frontera entre las dos clases densas adyacentes. El gran desbalance de datos que presenta el dataset justifica que la red tienda a evaluar en más ocasiones las densidades intermedias (más presencia) que las extremas (menos presencia).

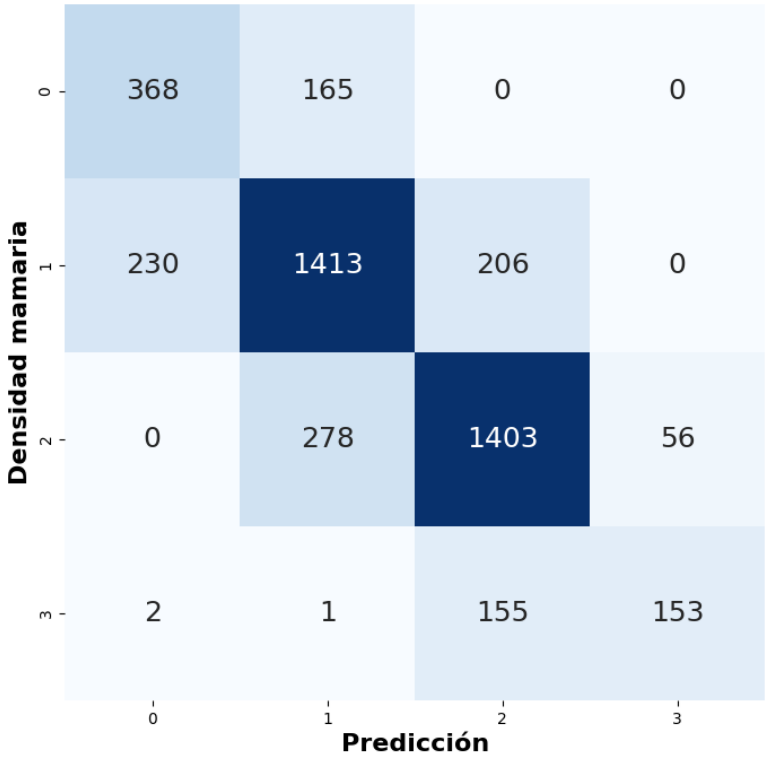


Figura 6.19: Matriz de confusión de la inferencia con el modelo utilizando drop out en el conjunto RSNA.

6.3.4. Conclusiones

Los modelos entrenados con el conjunto RSNA han obtenido valores de accuracy sobre el 70 %. Aunque en las clases con menor presencia de datos encontramos un recall bajo, encontramos también un valor de ROC score más alto que el promedio, indicándonos que las predicciones se ven de alguna forma arrastradas a clases intermedias, correspondiendo al desbalance de datos propio de este caso. Además, el valor cercano a 1.0 obtenido por el ROC score corrobora tanto la hipótesis del desbalance de datos como de la subjetividad de los radiologos.

La introducción del drop out en la fase de entrenamiento prolonga la convergencia tanto de la accuracy en validación como del valor de la función de pérdida durante todo el proceso. Permitiendo al modelo generalizar mejor las características y obteniendo un mejor rendimiento.

De la comparación de todas las métricas de entrenamiento mostradas en la Figura 6.20 determinamos que la inclusión de data augmentation proporciona mejoras notables tanto en la estabilidad como en el rendimiento del modelo. La accuracy de validación aumenta hasta valores cercanos a 0.79, lo que evidencia una mejor capacidad de generalización. El descenso progresivo y sostenido de la función de pérdida indica un aprendizaje más equilibrado.

En cambio, la configuración que combina data augmentation y dropout muestra un comportamiento más limitado. Aunque comienza con una accuracy de validación considerablemente inferior (alrededor de 0.58), logra mejorar gradualmente, alcanzando un valor cercano a 0.76. Sin embargo, su función de pérdida permanece elevada y no decrece de forma pronunciada, lo que sugiere que el modelo está teniendo dificultades para aprender de forma eficiente, posiblemente debido a una regularización excesiva. De todas formas, es el modelo que mejores resultados generales tiene, por lo que, de los tres modelos presentados, es el que mejor capacidad de generalización tiene.

La comparación de las métricas de entrenamiento se muestran en la Figura 6.20:

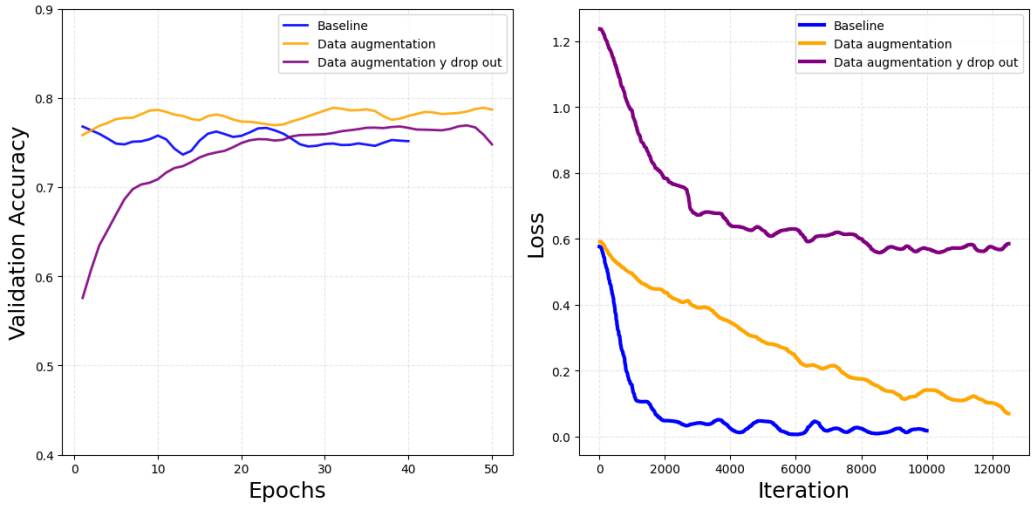


Figura 6.20: Comparación de métricas de entrenamiento de todos los modelos con el conjunto RSNA.

Conclusiones y trabajo futuro

Los modelos entrenados con el conjunto RSNA han mostrado un rendimiento superior respecto a los entrenados con CBIS-DDSM, incluso en configuraciones base. Desde la primera epoch del modelo baseline se alcanza una accuracy del 77 % en validación, y en la inferencia se consigue una accuracy media por clase superior al 70 %. Esta mejora significativa puede explicarse por la mayor cantidad de datos disponibles, aunque incrementa el desbalance de datos. En la Figura 7.1 se observa gracias al mapa de calor como los modelos mejoran en las medias macro de las métricas de la fase de inferencia con los modelos propuestos.

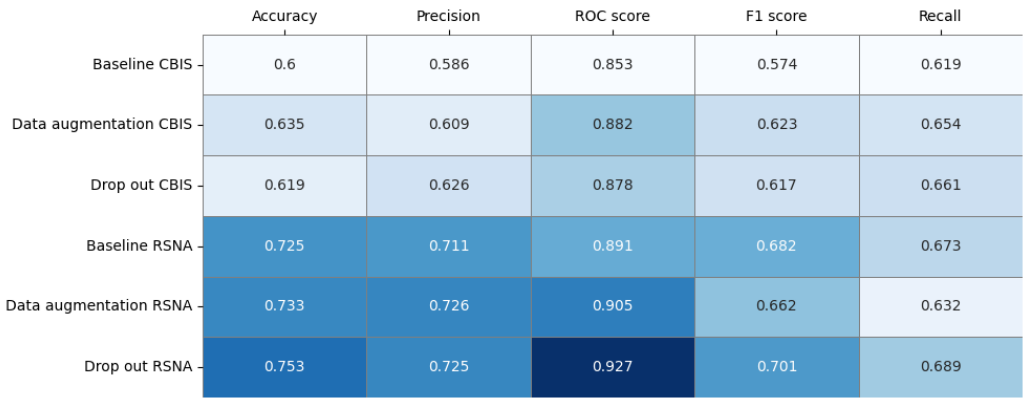


Figura 7.1: Comparación de las medias macro de las métricas de la fase de inferencia de todos los modelos explicados.

Aunque se observa un fuerte sobreajuste desde el inicio del entrenamiento en el modelo baseline, los valores de ROC AUC cercanos a 0.9 y la disposición de la matriz de confusión sugieren que la red es capaz de aprender patrones relevantes, aunque con fallos frecuentes en los casos limítadores entre clases adyacentes. Esta tendencia coincide con la hipótesis planteada anteriormente sobre la naturaleza subjetiva de la clasificación BI-RADS,

en la cual los desacuerdos suelen producirse entre clases contiguas.

Con la incorporación de técnicas como drop out, el aprendizaje se vuelve más robusto y lento, reduciendo el riesgo de sobreajuste y mejorando la generalización del modelo. Este enfoque ha permitido obtener mejores métricas en el conjunto de test respecto al modelo baseline, aunque la mejora no es tan acentuada como en el caso del conjunto CBIS-DDSM al aplicar data augmentation. Aun así, la clase con menor presencia (categoría D) continúa siendo la más difícil de clasificar correctamente, lo que reafirma la necesidad de aplicar técnicas más específicas para combatir el desbalance de clases, como el oversampling, la ponderación de clases o el uso de pérdidas focalizadas.

Posibles ampliaciones

Existen más opciones que se pueden explorar con el propósito de mejorar el rendimiento de los modelos. A continucación se detallan algunas de ellas como posible trabajo futuro para la mejora de estos modelos.

Eliminación del músculo pectoral

En algunas mamografías se puede presenciar una zona de alta intensidad, generalmente con forma rectangular. Esta zona representa el músculo pectoral. Debido a la gran variabilidad en las dimensiones, forma y presencia de este objeto en las mamografías puede afectar negativamente al aprendizaje de la CNN. La solución sería estandarizar los datos de forma que se elimine todo el tejido muscular pectoral de las mamografías. Esta es una tarea complicada debido al ángulo y dimensiones que puede obtener la imágen resultante, pero tiende a mejorar el rendimiento de la CNN.

Estandarización de intensidad

Las imágenes pueden tener histogramas distintos que confunden al modelo y son un factor de descarte o modificación en ocasiones (véase sección 5.2). En este proyecto se han realizado pruebas aplicando la normalización de Nyul[34] en el conjunto de datos. Pero los resultados no fueron los esperados, por este motivo, se decidió prescindir de ellos como propuesta de modelos implementados.

Es posible que la normalización no diera los resultados esperados debido a que la intensidad juega el papel principal en la detección de tejido fibroglandular, al normalizar los histogramas de las imágenes, se puede estar perdiendo información importante para la clasificación. Además, existen otras hipótesis que comprenden la generación de artefactos, o variaciones estadísticas sutíles entre los conjuntos de entrenamiento y tests que se ven amplificadas.

Generación de imágenes sintéticas

Para compensar el desbalance de los datos es posible emplear el uso de inteligencias artificiales o algoritmos de aprendizaje automático para generar imágenes sintéticas de las clases menos representadas en el dataset. Debido a la importancia de la detección de la clase extremadamente densa, se empleó el uso de un modelo *CycleGAN*[1] para generar imágenes sintéticas a partir de imágenes con densidad muy baja. El modelo se puede encontrar en el repositorio *mendigan*[2].

Esta técnica tampoco tuvo resultados positvos. Esta inteligencia artificial fue entrenada con una base de datos distinta, por lo que las imágenes generadas no fueron de la calidad suficiente, de forma que introdujo mayor error en el modelo.

Calibrador

Debido al alto valor de ROC AUC score que presentan los modelos es lícito plantear entrenar un calibrador de forma que aprenda cuál es el mejor límite para cada clase para detectarla. Esto no se implementó por falta de tiempo pero es posible que esta implementación mejore las métricas considerablemente.

[1]Garrucho, L., & Osuala, R. (2022). *CYCLEGAN Model for Mammogram Low-to-High Breast Density Translation ONLY MLO (Trained on OPTIMAM).* Zenodo. `https://doi.org/10.5281/zenodo.7093556`

[2]Osuala, R., Skorupko, G., Lazrak, N., Garrucho, L., García, E., Joshi, S., ... & Lekadir, K. (2023).*medigan: a Python library of pretrained generative models for medical image synthesis.* Journal of Medical Imaging, 10(6), 061403. Repositorio de github: `https://github.com/RichardObi/medigan`

Bibliografía

[1] Instituto Nacional del Cáncer (NIH). *Mamografías*. Disponible en: https://www.cancer.gov/espanol/tipos/seno/hoja-informativa-mamografias. Accedido el 21 de mayo de 2025.

[2] Google Developers. *Accuracy, Precision and Recall*. Machine Learning Crash Course, 2024. Disponible en: https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall. Accedido el 21 de mayo de 2025.

[3] BreastScreen SA. *Breast Density and Cancer Risk*. Disponible en: https://www.breastscreen.sa.gov.au/health-professionals/information-for-gps/breast-density. Accedido el 21 de mayo de 2025.

[4] PyTorch. *PyTorch Official Site*. Disponible en: https://pytorch.org. Accedido el 3 de junio de 2025.

[5] Torch. *Torch Documentation*. Disponible en: https://pytorch.org/docs/stable/torch.html. Accedido el 3 de junio de 2025.

[6] MONAI. *Medical Open Network for AI*. Disponible en: https://monai.io. Accedido el 3 de junio de 2025.

[7] Silva, G. y Silva, P. *Las calcificaciones mamarias como hallazgo radiológico: revisión de la literatura*. Revista chilena de radiología, vol. 22, n.º 2, 2016, pp. 73–79. Disponible en: https://www.scielo.cl/scielo.php?script=sci_arttext&pid=S0717-93082016000200009. Accedido el 21 de mayo de 2025.

[8] Mayo Clinic. *Dense breast tissue: What it means to have dense breasts*. Disponible en: https://www.mayoclinic.org/tests-procedures/mammogram/in-depth/dense-breast-tissue/art-20123968. Accedido el 21 de mayo de 2025.

[9] Cleveland Clinic. *Fibroglandular Density*. Disponible en: https://my.clevelandclinic.org/health/articles/22874-fibroglandular-density. Accedido el 21 de mayo de 2025.

[10] DataCamp. *Introduction to Activation Functions in Neural Networks*. Disponible en: https://www.datacamp.com/tutorial/introduction-to-activation-functions-in-neural-networks. Accedido el 21 de mayo de 2025.

[11] Pyimagesearch. *Convolutional Neural Networks (CNNs) and Layer Types* Disponible en: https://pyimagesearch.com/2021/05/14/convolutional-neural-networks-cnns-and-layer-types/

[12] NVIDIA, *Linear/Fully-Connected Layers User's Guide*, 2023. Disponible en: https://docs.nvidia.com/deeplearning/performance/dl-performance-fully-connected/index.html

[13] Analytics Vidhya. *A Comprehensive Guide on Deep Learning Optimizers*. 2021. Disponible en: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-deep-learning-optimizers/. Accedido el 21 de mayo de 2025.

[14] Raschka Sebasitan. *What are gradient descent and stochastic gradient descent?* https://sebastianraschka.com/faq/docs/gradient-optimization.html

[15] The Cancer Imaging Archive. *Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM)*. Disponible en: https://www.cancerimagingarchive.net/collection/cbis-ddsm/. Accedido el 21 de mayo de 2025.

[16] Zhu, L. *Análisis de imágenes mamográficas usando redes neuronales profundas*. Trabajo de fin de grado, Universitat de Barcelona. Disponible en: https://diposit.ub.edu/dspace/bitstream/2445/186091/3/tfg_zhu_ling.pdf. Accedido el 21 de mayo de 2025.

[17] Mohamed, A. A., Luo, Y., Peng, H., Jankowitz, R. C. y Wu, S. (2018). *Understanding Clinical Mammographic Breast Density Assessment: a Deep Learning Perspective*. Journal of Digital Imaging, 31(4), 387–392. doi: 10.1007/s10278-017-0022-2.

[18] Huang, G., Liu, Z., Van Der Maaten, L., y Weinberger, K. Q. (2016). *Densely Connected Convolutional Networks*. arXiv preprint arXiv:1608.06993. Disponible en: https://arxiv.org/abs/1608.06993.

[19] Gupta, V., Demirer, M., Maxwell, R. W., White, R. D., & Erdal, B. S. (2022). *A multi-reconstruction study of breast density estimation using Deep Learning*. Disponible en: https://arxiv.org/abs/2202.08238

[20] Kingma, D. P., y Ba, J. (2014). *Adam: A Method for Stochastic Optimization*. arXiv preprint arXiv:1412.6980. Disponible en: https://arxiv.org/abs/1412.6980.

[21] Arora, A. (2020). *DenseNet Architecture Explained with PyTorch Implementation from TorchVision*. Disponible en: https://amaarora.github.io/posts/2020-08-02-densenets.html.

[22] SabrePC. *Epochs vs Batch Size vs Iterations: Differences in Deep Learning*. SabrePC Blog, 2021. Disponible en: https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations

[23] Masters, D., & Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. *Pattern Recognition Letters*, 131, 244–250. Disponible en: https://www.sciencedirect.com/science/article/pii/S2405959519303455

[24] SabrePC. (2021). *Epochs vs Batch Size vs Iterations: Differences in Deep Learning*. Recuperado de https://www.sabrepc.com/blog/Deep-Learning-and-AI/Epochs-Batch-Size-Iterations

[25] Masters, D., & Luschi, C. (2018). Revisiting Small Batch Training for Deep Neural Networks. *Pattern Recognition Letters*, 131, 244–250. Recuperado de https://www.sciencedirect.com/science/article/pii/S2405959519303455

[26] Mishra, M. (2023). The Learning Rate: A Hyperparameter That Matters. *Medium*. Recuperado de https://mohitmishra786687.medium.com/the-learning-rate-a-hyperparameter-that-matters-b2f3b68324ab

[27] Goksel, R. (2021). *Breast-Tissue-Cropper-Tools*. Recuperado de https://github.com/RsGoksel/Breast-Tissue-Cropper-Tools

[28] Sociedad Radiológica de Norteamérica. Disponible en: https://www.rsna.org/

[29] Chakravarty, B., et al. (2023). "The Impact of Scanner Domain Shift on Deep Learning Performance in Medical Imaging". *arXiv preprint*. Disponible en: https://arxiv.org/abs/2409.04368.

[30] Rocky Mountain Cancer Centers. *Cancer and Breast Implants: Is There a Connection?*. Disponible en: https://es.rockymountaincancercenters.com/blog/cancer-and-breast-implants-is-there-a-connection. Consultado el 5 de junio de 2025.

[31] B. Triwijoyo, B. I. Nugroho, A. D. Wirastuti, *Analysis of Medical Image Resizing Using Different Interpolation Techniques*, Jurnal RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 6, no. 3, pp. 500–506, 2022. Disponible en: https://d1wqtxts1xzle7.cloudfront.net/86823836/39370-libre.pdf

[32] S. Ahmed, M. S. Islam, F. M. Anwar et al., *Multi-modal fusion in medical imaging: application to neurological disorders*, Frontiers in Neuroscience, vol. 17, 2023. Disponible en: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC10469557/

[33] M. Wulansari, A. Setiawan, A. Ariyanto, *Hybrid CNN and attention mechanism for lung disease detection from chest X-rays*, PLOS ONE, vol. 19, no. 1, 2024. Disponible en: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC11127450/

[34] Kim, Hojin Yoo, Sang Kim, Jin Kim, Yong Lee (2024). *Clinical feasibility of deep learning-based synthetic CT images from T2-weighted MR images for cervical cancer patients compared to MRCAT*. Scientific Reports. 14. 8504. 10.1038/s41598-024-59014-6. Disponible en: https://www.researchgate.net/publication/379777705_Clinical_feasibility_of_deep_learning-based_synthetic_CT_images_from_T2-weighted_MR_images_for_cervical_cancer_patients_compared_to_MRCAT

[35] Garrucho, L., Osuala, R. (2022). *CYCLEGAN Model for Mammogram Low-to-High Breast Density Translation ONLY MLO (Trained on OPTIMAM).* Zenodo. https://doi.org/10.5281/zenodo.7093556

Osuala, R., Skorupko, G., Lazrak, N., Garrucho, L., García, E., Joshi, S., ... Lekadir, K. (2023).*medigan: a Python library of pretrained generative models for medical image synthesis.* Journal of Medical Imaging, 10(6), 061403. Repositorio de github: https://github.com/RichardObi/medigan