Benemérita Universidad Autónoma de Puebla

Facultad de Ciencias Físico Matemáticas

# EXTRACCIÓN DE CARACTERÍSTICAS EN LA CLASIFICACIÓN DE IMÁGENES MÉDICAS MEDIANTE FILTRO BASADO EN REDES NEURONALES ARTIFICIALES

Tesis presentada al

**Posgrado en Física Aplicada**

como requisito parcial para la obtención del grado de

**Maestro en Ciencias**

por

Josué Rodríguez Hernández

Asesorado por

Dr. Jorge Velázquez Castro
Dr. Benito de Celis Alonso

Puebla Pue.
Noviembre 2025

**Título:** EXTRACCIÓN DE CARACTERÍSTICAS EN LA CLASIFICACIÓN DE IMÁGENES MÉDICAS MEDIANTE FILTRO BASADO EN REDES NEURONALES ARTIFICIALES
**Estudiante:** Josué Rodríguez Hernández

COMITÉ

Dr. Javier Miguel Hernández López
Presidente

Dr. Iván Fuentecilla Cárcamo
Secretario

Dra. Beatriz Bonilla Capilla
Vocal

Dr. Jorge Velázquez Castro
Asesor

Dr. Benito de Celis Alonso
Asesor

# Agradecimientos

Me gustaría comenzar agradeciendo a la Secretaría de Ciencias, Humanidades, Tecnología e Innovación (SECIHTI) por el apoyo económico brindado durante esta etapa. Este respaldo fue fundamental para poder realizar mis estudios de maestría, por lo que les estaré eternamente agradecido. Espero que, así como a mí, este apoyo continúe otorgándose a más jóvenes que desean seguir creciendo profesionalmente.

Deseo expresar un agradecimiento muy especial a mi asesor principal, el Dr. Jorge Velázquez Castro, quien fue mi guía a lo largo de todo este camino. En verdad, muchas gracias por el tiempo invertido, por la paciencia que me tuvo y por todos los conocimientos que me compartió. No tengo palabras para expresar el apoyo que recibí de su parte durante esta etapa. Sus observaciones, consejos y, sobre todo, sus palabras de aliento hicieron posible gran parte de este trabajo. Su constante disposición me ayudó a crecer tanto académicamente como en lo personal. De nuevo, muchas gracias; espero poder seguir trabajando con usted en el futuro y, sobre todo, continuar conviviendo y aprendiendo de usted.

Agradezco también a mi coasesor, el Dr. Benito de Celis Alonso, y a mis sinodales, el Dr. Javier Miguel Hernández López, el Dr. Iván Fuentecilla Cárcamo y la Dra. Beatriz Bonilla Capilla, por su disposición para evaluar este trabajo. Sus comentarios y observaciones fueron de gran ayuda.

Quiero hacer un agradecimiento especial a mi amigo y colega, el M.C. Mario Armando Talamantes Johnson, por todo el apoyo brindado durante esta etapa. Aprecio profundamente tus consejos, tus regaños, las risas y, sobre todo, el tiempo compartido. Tu amistad fue clave para que pudiera concluir esta etapa, te convertiste en un hermano mayor para mí, brindándome apoyo moral en aquellos momentos en los que estuve a punto de rendirme. De verdad espero que podamos seguir trabajando y creciendo juntos por muchos años más.

Agradezco profundamente a mi madre, Indira Tania Hernández Sandoval, cuyo amor y apoyo incondicional siempre han estado presentes. Tus enseñanzas y valores me han llevado hasta donde hoy estoy. No existen palabras para expresar el gran trabajo que has hecho criándonos a mi hermano y a mí. Sin duda alguna, sin ti este trabajo no habría sido posible. Gracias por siempre creer en mí. Te amo mucho, mamá.

A mi hermano mayor, Arturo Rodríguez Hernández, quien siempre ha estado para mí sin importar qué. Tal vez no lo expreso tan seguido, pero quiero que sepas que has sido una pieza fundamental tanto en mi desarrollo profesional como personal. Por ello, este trabajo también lleva una parte tuya. Gracias por todo, hermano.

Al resto de mi familia, por su constante apoyo y cariño, y por enseñarme el valor del trabajo, el esfuerzo y la perseverancia. Gracias por estar siempre, en las buenas y en las malas.

A mi mejor amigo, Leonardo Fernández Méndez, por su amistad sincera y por estar siempre que necesito ayuda, un consejo o simplemente alguien con quien distraerme. Gracias por tu apoyo tanto personal como académico.

Sé que aún me faltan muchas personas por mencionar, pero finalmente agradezco de corazón a todas aquellas personas que, de forma directa o indirecta, han contribuido tanto a mi desarrollo profesional como a la realización de este trabajo. Su apoyo y dedicación han sido fundamentales, y les estaré eternamente agradecido por su invaluable colaboración.

# Índice general

# Resumen

El diagnóstico asistido por computadora basado en imágenes médicas ha transformado la detección temprana y precisa de diversas enfermedades, proporcionando una valiosa herramienta de apoyo para la práctica clínica. Sin embargo, a pesar del notable desempeño alcanzado por los modelos de aprendizaje profundo, como las Redes Neuronales Convolucionales (RNC), su falta de interpretabilidad continúa representando una de las principales limitaciones para su adopción en entornos médicos, donde comprender las razones detrás de una predicción resulta esencial.

En este trabajo se propone un filtro de extracción de características basado en el principio de sensibilidad de oclusión, diseñado para visualizar y comprender los procesos de decisión de las RNC. A través de una implementación propia, el método permite generar mapas de activación que resaltan las regiones más relevantes en las imágenes utilizadas por el modelo durante la clasificación. Además, se introducen mejoras orientadas a optimizar la discriminación de las regiones de interés, la calidad visual de los mapas de calor y la eficiencia computacional del proceso.

El filtro propuesto fue evaluado bajo distintos escenarios experimentales, incluyendo su aplicación sobre la base de datos *ChestX-ray14*, donde se comparó su desempeño con el método Grad-CAM. Los resultados obtenidos muestran que el filtro logra generar mapas de activación con un nivel de coherencia espacial y calidad comparable a Grad-CAM, destacando su potencial como herramienta de interpretabilidad y como método de identificación semisupervisada de regiones relevantes.

En conjunto, este trabajo contribuye al desarrollo de modelos de aprendizaje profundo más transparentes y confiables, demostrando que la interpretabilidad puede abordarse desde un enfoque complementario a la optimización del rendimiento del modelo, promoviendo así el uso responsable y explicable de la inteligencia artificial en aplicaciones médicas.

# Capítulo 1

# Introducción

## 1.1. Motivación

Desde el descubrimiento de los rayos X por Wilhelm Conrad Röntgen en 1895, la radiografía se consolidó como una de las herramientas más importantes en el diagnóstico médico, marcando el inicio del diagnóstico por imágenes. Con el paso de los años, se han desarrollado diversas modalidades de imagen médica, como la tomografía computarizada (CT), la resonancia magnética (MRI), la ecografía (US) y la tomografía por emisión de positrones (PET). Todas ellas desempeñan un papel esencial en las distintas etapas del cuidado del paciente: desde la detección y caracterización de enfermedades hasta la planificación, evaluación y seguimiento de tratamientos clínicos y quirúrgicos [1].

La gran cantidad de información contenida en las imágenes médicas y el volumen creciente de estudios generados diariamente han impulsado la búsqueda de herramientas automáticas que asistan al especialista en la interpretación y diagnóstico de dichas imágenes. En este contexto, los avances en *aprendizaje automático* y, particularmente, en *aprendizaje profundo* (*deep learning*), han permitido desarrollar sistemas capaces de analizar de forma automática grandes volúmenes de datos médicos con niveles de precisión comparables e incluso superiores a los del ojo humano.

Entre estos avances, las *Redes Neuronales Convolucionales* (RNC) han demostrado un desempeño sobresaliente en tareas de clasificación y segmentación de imágenes, constituyéndose como el pilar de numerosos sistemas de diagnóstico asistido por computadora (CAD). Sin embargo, la adopción de estas técnicas en el ámbito médico requiere un nivel de transparencia e interpretabilidad que permita comprender cómo y por qué el modelo toma una decisión, lo cual sigue representando uno de los principales desafíos de las RNC modernas.

## 1.2. Problemática general

A pesar de su capacidad para alcanzar altos niveles de precisión, las RNC se caracterizan por su naturaleza de çaja negra", ya que la representación interna de las características aprendidas durante el entrenamiento no es directamente interpretable para el ser humano. Esto dificulta la validación de las decisiones del modelo, particularmente en contextos clínicos donde la confianza, la

trazabilidad y la explicación de las predicciones son tan importantes como la exactitud misma.

En aplicaciones médicas, una predicción incorrecta o injustificada puede tener consecuencias críticas. Por ello, resulta indispensable que las decisiones generadas por una red neuronal sean *explicables* y puedan relacionarse con regiones anatómicas relevantes dentro de la imagen. Si bien se han desarrollado diversas técnicas para mejorar la interpretabilidad de las RNC como *Grad-CAM*, *Guided Backpropagation*, *Layer-wise Relevance Propagation* y *Occlusion Sensitivity*, muchas de estas presentan limitaciones relacionadas con la resolución espacial, la dependencia del gradiente o la sensibilidad al ruido, lo que en ocasiones impide obtener representaciones visuales coherentes con los hallazgos clínicos observados.

En consecuencia, existe la necesidad de desarrollar métodos alternativos o complementarios que permitan **extraer de manera más robusta y estable la información relevante utilizada por las RNC durante la clasificación**, y que al mismo tiempo ofrezcan una representación visual interpretable y clínicamente significativa.

## 1.3. Investigaciones previas

La interpretabilidad en redes neuronales profundas ha sido objeto de un creciente interés en los últimos años, especialmente en el ámbito de las imágenes médicas. Investigaciones previas han explorado múltiples enfoques para identificar las regiones más relevantes en la toma de decisiones de una RNC. Entre los métodos más reconocidos se encuentran *Grad-CAM* [5], que utiliza los gradientes de las activaciones para generar mapas de calor que destacan las áreas que más contribuyen a la clasificación, y *Occlusion Sensitivity* [6], que evalúa la variación en la predicción al ocultar sistemáticamente diferentes regiones de la imagen.

Si bien estas técnicas han demostrado ser útiles, aún presentan desafíos importantes: *Grad-CAM* tiende a generar mapas amplios y difusos que no siempre delimitan con precisión la región anatómica de interés, mientras que la sensibilidad de oclusión puede producir mapas ruidosos y de baja resolución si los parámetros del parche y del desplazamiento no se seleccionan adecuadamente. Además, la mayoría de estos métodos carece de mecanismos de filtrado o refinamiento que permitan eliminar información redundante o no relevante.

Estos aspectos motivan la búsqueda de una **versión modificada o mejorada de los métodos existentes**, capaz de producir mapas de activación más estables, precisos y coherentes con las regiones patológicas reales anotadas por expertos.

## 1.4. Problema específico

Considerando las limitaciones de los métodos actuales de interpretabilidad, este trabajo plantea la necesidad de **desarrollar un nuevo enfoque de filtrado de imágenes** que permita extraer de manera más efectiva las características relevantes aprendidas por las RNC durante la clasificación de imágenes médicas.

El problema se centra, por tanto, en **determinar un mecanismo de filtrado basado en oclusión** que preserve la información esencial para la decisión del modelo, minimizando simultáneamente la influencia de las regiones irrelevantes. Este enfoque debe generar mapas de

calor que sean interpretables, visualmente consistentes y que reflejen de manera más fiel las regiones anatómicas asociadas a la patología detectada.

Asimismo, es necesario **evaluar el desempeño del filtro propuesto** mediante su aplicación en un conjunto de datos clínico, como *ChestX-ray14*, comparando los resultados obtenidos con los de técnicas establecidas como *Grad-CAM* y la versión original de *Occlusion Sensitivity*.

## 1.5. Hipótesis

Si se desarrolla un filtro de imágenes basado en un proceso de oclusión controlado y en el análisis diferencial de las predicciones de una red neuronal convolucional, entonces será posible **resaltar de forma más precisa y estable las regiones anatómicas relevantes** para la clasificación de una patología en una imagen médica.

Este filtro permitirá mejorar la interpretabilidad de las redes neuronales convolucionales y ofrecer una herramienta complementaria que ayude a los especialistas a **comprender y validar las decisiones del modelo** en el contexto del diagnóstico por imágenes.

## 1.6. Objetivos

### 1.6.1. Objetivo general

Desarrollar e implementar una nueva técnica para la visualización y comprensión de los procesos de aprendizaje de las Redes Neuronales Convolucionales (RNC) aplicadas a la clasificación de imágenes médicas.

### 1.6.2. Objetivos específicos

- Desarrollar un filtro de imágenes que, basándose en las clasificaciones realizadas por una RNC previamente entrenada, permita visualizar e interpretar las características relevantes aprendidas por las RNC.
- Implementar el filtro propuesto en problemas de clasificación básicos utilizando conjuntos de datos de imágenes comunes, como MNIST, ImageNet, perros vs gatos, entre otros.
- Evaluar la eficiencia, interpretabilidad y utilidad del filtro desarrollado mediante la comparación de su desempeño con la técnica existente de mapas de calor de activación de clase para la extracción de características aprendidas por las RNC, identificando sus fortalezas y posibles áreas de mejora.
- Replicar alguna de las RNC reportadas en la literatura que realicen tareas de clasificación de imágenes médicas, como se observa en [3, 7].
- Implementar el filtro propuesto en la tarea de clasificación médica abordada.

- Evaluar la eficiencia, interpretabilidad y utilidad clínica de la metodología propuesta, comparando su desempeño con técnicas convencionales y determinando su potencial como herramienta complementaria en el diagnóstico médico.
- Analizar y documentar exhaustivamente los resultados obtenidos, destacando los avances logrados, las limitaciones identificadas y las posibles direcciones futuras para la mejora y expansión de la técnica desarrollada.

# Capítulo 2

# Marco Teórico

## 2.1. Redes Neuronales Artificiales

El desarrollo de herramientas inspiradas en la naturaleza ha sido una constante en la historia de la ciencia, y el campo del aprendizaje automático no es la excepción. Las *redes neuronales artificiales* (RNA) surgieron originalmente como un intento por imitar ciertos principios del sistema nervioso biológico, pero con el tiempo han evolucionado hasta convertirse en modelos matemáticos altamente optimizados para resolver problemas complejos en los que las técnicas tradicionales resultan insuficientes. Hoy en día constituyen la base del *Aprendizaje Profundo*, impulsando aplicaciones que van desde la clasificación masiva de imágenes hasta el análisis médico computarizado, pasando por sistemas de recomendación, reconocimiento de voz y generación de contenido.

Aunque el vínculo conceptual con la biología sigue presente, las redes actuales difieren notablemente de sus inspiraciones neuronales originales. Su relevancia contemporánea se debe a la disponibilidad de grandes volúmenes de datos, a la capacidad de cómputo cada vez mayor y a algoritmos de optimización más eficientes, lo que las ha llevado a experimentar un progreso acelerado durante las últimas dos décadas.

### 2.1.1. De la Neurona Biológica al Modelo Computacional

Los primeros intentos formales por modelar matemáticamente el comportamiento de una neurona datan de 1943, cuando McCulloch y Pitts propusieron una unidad computacional simple capaz de realizar operaciones lógicas a partir de entradas binarias. Aquel modelo rudimentario colocó los cimientos del campo y generó un entusiasmo considerable durante los años cincuenta y sesenta, periodo en el que se creía que el sueño de construir máquinas inteligentes estaba a la vuelta de la esquina.

Sin embargo, las limitaciones teóricas y computacionales de la época desaceleraron el progreso, dando paso a lo que se conoce como una .$^{et}$apa oscura"para las RNA. No fue sino hasta la década de 1980, con el desarrollo del algoritmo de retropropagación y la aparición de nuevas arquitecturas, cuando el interés resurgió. Aun así, durante los años noventa, métodos como las máquinas de soporte vectorial destacaron debido a su solidez teórica y buen rendimiento en problemas de clasificación.

La situación cambió radicalmente con la llegada del siglo XXI. La combinación de tres factores: *datos masivos*, aceleración por *hardware* (particularmente mediante GPU) y mejoras en optimización y regularización, reactivó de forma explosiva a las RNA, permitiendo el entrenamiento de redes profundas con millones de parámetros. Desde entonces, el campo ha entrado en un ciclo de innovación constante que ha impulsado avances en visión computacional, procesamiento de lenguaje natural y, de forma relevante para esta tesis, en el análisis automatizado de imágenes médicas.

### 2.1.2. Neuronas Biológicas

Aunque las RNA modernas no pretenden replicar el funcionamiento biológico con precisión, resulta útil revisar brevemente la estructura básica de una neurona para entender la metáfora conceptual que dio origen a estos modelos.

Una neurona típica está formada por:

- **Un soma o cuerpo celular**, que actúa como centro de integración.
- **Dendritas**, encargadas de recibir señales provenientes de otras neuronas.
- **Un axón**, una prolongación especializada que propaga el impulso eléctrico generado por la célula.
- **Terminales sinápticas**, responsables de transmitir señales hacia otras neuronas.

Cuando una neurona recibe un conjunto de estímulos suficiente, genera un potencial de acción que viaja a lo largo del axón hasta sus sinapsis, influyendo en la actividad de las neuronas conectadas. Este proceso ocurre dentro de una red sumamente densa: una sola neurona puede establecer miles de conexiones, formando estructuras organizadas que suelen agruparse en capas funcionales.

Estas observaciones inspiraron la construcción de modelos artificiales compuestos por unidades simples interconectadas, también organizadas en capas, donde cada unidad realiza una operación matemática elemental y transmite su resultado a las siguientes. A pesar de que las diferencias entre ambos sistemas son significativas (las redes artificiales son modelos estadísticos, mientras que las biológicas constituyen sistemas electroquímicos altamente complejos), la analogía estructural sigue siendo útil para introducir las bases conceptuales de las RNA modernas.

### 2.1.3. El Perceptrón

El modelo simple de neuronas biológicas propuesto por McCulloch y Pitts (neurona artificial) consitía de una o más entradas binarias (on/off) y una salida binaria. Una neurona artificial activa su salida, cuando un cierto número de sus entradas están activas. De esta manera, es posible calcular cualquier proposición lógica a partir de una red de neuronas artificiales.

Por otro lado, en 1957 Frank Rosenblatt inventó una de las arquitecturas más simples de RNA llamado perceptrón. En un perceptrón, las entradas y salidas son números (en lugar de valores binarios) y cada conexión de entrada está asociada con un peso basado en una cierta unidad de umbral lineal (LTU, ver figura 2.3). La LTU calcula una suma ponderada de sus entradas ($z = w_1x_1 + w_2x_2 + \cdots + w_nX_n = W^t \cdot X$), luego aplica una función de paso a esa suma y genera la salida: $h_w(x) = step(z) = step(W^t \cdot X)$.

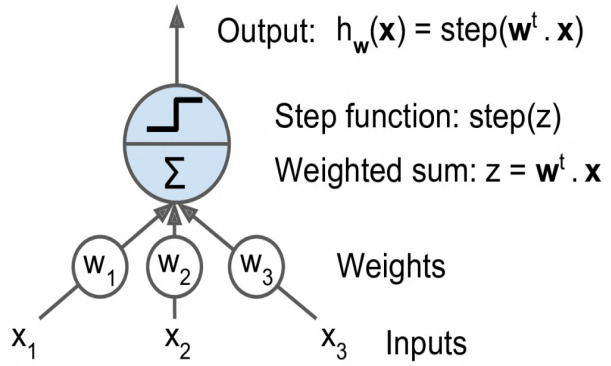


Figura 2.1: Unidad de umbral lineal. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

A continuación, se muestran las dos más comunes funciones de paso utilizadas.

$$\text{heaviside}(z) = \begin{cases} 0 & \text{si } z < 0 \\ 1 & \text{si } z \geq 0 \end{cases}, \quad \text{sgn}(z) = \begin{cases} -1 & \text{si } z < 0 \\ 0 & \text{si } z = 0 \\ +1 & \text{si } z > 0 \end{cases}.$$

Un perceptrón consta de una sola capa de LTU, donde cada neurona está conectada a todas las entradas. Estas conexiones a menudo se representan mediante neuronas de entrada, que simplemente pasan cualquier entrada que reciben. En la figura 2.4 se muestra un perceptrón de dos entradas y tres salidas, el cual es capaz de clasificar instancias simultáneamente en tres clases binarias diferentes (clasificador de salida múltiple).

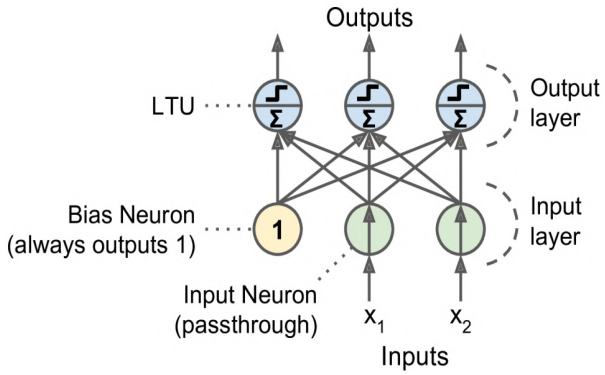


Figura 2.2: Diagrama del perceptrón. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

El entrenamiento de un perceptrón se realiza mediante una variante del aprendizaje hebbiano, inspirado en la idea de que las células que disparan juntas se conectan entre sí, la cual simplemente sigue la regla de que el peso de la conexión entre dos neuronas aumenta siempre que tengan la misma salida. El perceptrón recibe una instancia de entrenamiento a la vez y sus predicciones se

comparan con los resultados esperados. Por cada neurona de salida que produce una predicción incorrecta, se refuerzan los pesos de conexión de las entradas que contribuyen a la predicción correcta. En la ecuación 2.1 se muestra matemáticamente esta regla.

$$w_{i,j} = w_{i,j} + \eta(\hat{y_j} - y_j)x_i, \tag{2.1}$$

donde

- $w_{i,j}$ es el peso de conexión que hay entre la i-ésima neurona de entrada y la j-ésima neurona de salida.
- $x_i$ es el i-ésimo valor de entrada de la instancia de entrenamiento actual.
- $\hat{y_j}$ es la salida en la j-ésima neurona de salida para la instancia de entrenamiento actual.
- $y_j$ es la salida objetivo de la j-ésima neurona de salida para la instancia de entrenamiento actual.
- $\eta$ es la tasa de aprendizaje.

El límite de decisión de cada neurona de salida en un perceptrón es lineal, lo que limita su capacidad para aprender patrones complejos. Esta limitación la comparte con otros modelos de clasificación lineal como la regresión logística. Sin embargo, si las instancias de entrenamiento son linealmente separables, se demostró a través del teorema de convergencia de perceptrón que el algoritmo encontrará una solución.

Las debilidades de los perceptrones se destacaron en la monografía titulada "Perceptrones"de Marvin Minsky y Seymour Papert en 1969. Mostraron que los perceptrones no podían resolver ciertos problemas triviales, como el problema de clasificación XOR (ver figura 2.5, lado izquierdo). Esto llevó a una disminución en el interés por las redes neuronales en ese momento. Sin embargo, más tarde se descubrió que al apilar varios perceptrones, un Perceptrón Multicapa (MLP, figura 2.5, lado derecho) podría superar estas limitaciones. Los MLP son capaces de resolver problemas complejos, incluido el problema XOR, mediante la introducción de capas ocultas entre las capas de entrada y salida.

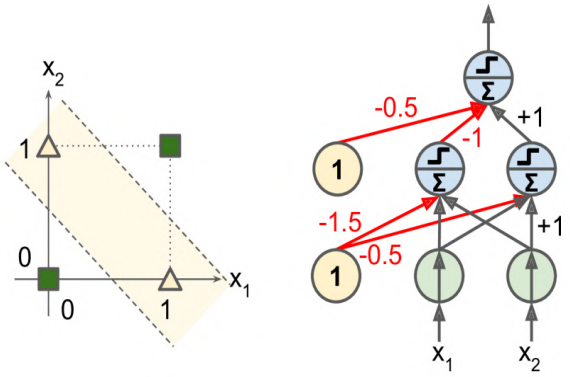


Figura 2.3: Problema de clasificación XOR y un MLP que lo resuelve. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

### 2.1.4. Perceptrón multicapa y el algoritmo Backpropagation

Un MLP consta de diferentes capas que trabajan en conjunto. Comienza con una capa de entrada (capa de paso), seguida de una o más capas ocultas compuestas por LTU y termina con una capa de salida que también esta compuesta por LTU (ver figura 2.6). Cada capa, excepto la capa de salida, incluye una neurona de sesgo (bias neuron) y está conectada en su totalidad a la siguiente capa. Cuando una RNA tiene dos o más capas ocultas, se denomina red neuronal profunda (DNN, por sus siglas en inglés).

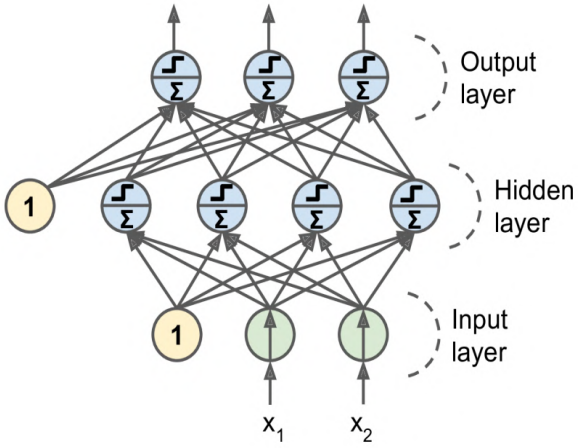


Figura 2.4: Diagrama de un perceptrón multicapa. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

Los investigadores lucharon por encontrar una forma de entrenar a los MLP hasta 1986, año en que se introdujo el algoritmo de entrenamiento de Backpropagation por D. E. Rumelhart et al. Este algoritmo, similar a Gradient Descent, implica un paso hacia adelante para calcular las salidas de las neuronas, medir el error de salida y luego propagar el error hacia atrás a través de las capas para calcular las contribuciones de error de cada neurona.

El algoritmo Backpropagation realiza un paso inverso para medir el gradiente de error en todos los pesos de conexión de la red. De esta manera, propaga eficientemente el gradiente de error hacia atrás, de ahí el nombre del algoritmo. El paso final implica el uso de Gradient Descent para ajustar los pesos de conexión en función de los gradientes de error. Por lo que, en resumen, el algoritmo de Backpropagation hace predicciones para cada instancia de entrenamiento, mide el error, calcula las contribuciones de error en un paso inverso a través de las capas y luego ajusta los pesos de conexión para minimizar el error usando Gradient Descent.

Por otro lado, para hacer que todo esto funcione, los autores reemplazaron la función de paso en la arquitectura del MLP con la función logística (función sigmoide), $\sigma(z) = 1/(1 + exp(-z))$ para permitir que Gradient Descent pueda avanzar en cada paso y no tenga problemas con los segmentos planos de la función de paso, en donde no hay gradiente. Esto se debe a que la función sigmoide tiene una derivada distinta de cero en todas partes, a diferencia de la función escalonada.

El algoritmo de Backpropagation también se puede utilizar con otras funciones de activación. Dos populares son la función tangente hiperbólica, $tanh(z) = 2\sigma(2z) - 1$ y la función ReLU (unidad lineal rectificada), $ReLU(z) = max(0, z)$. La función de tangente hiperbólica produce valores de salida que van de -1 a 1 y puede ayudar a acelerar la convergencia. La función ReLU es rápida de

calcular y funciona bien en la práctica, aunque no es diferenciable en z = 0. En la figura 2.7, se representan estas dos de funciones de activación junto con sus derivadas.

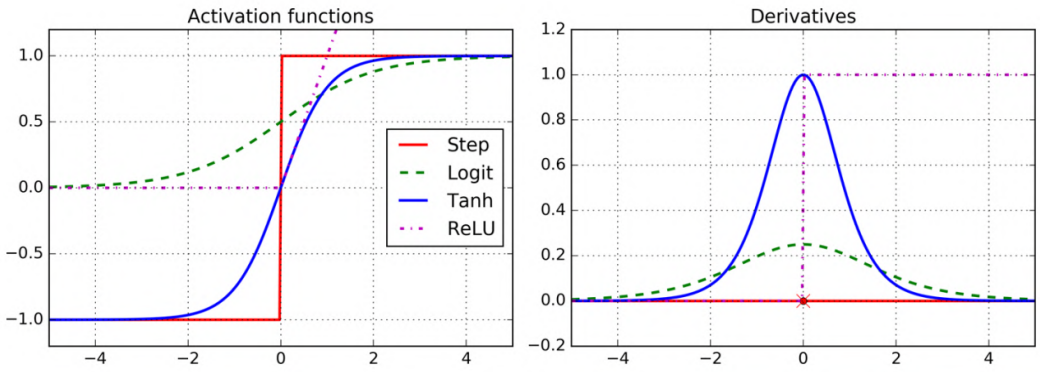


Figura 2.5: Funciones de activación y sus derivadas. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

La arquitectura MLP se usa a menudo para la clasificación, con cada neurona de salida correspondiente a una clase binaria diferente (por ejemplo, perros/gatos, correos deseados/no deseados, etc). En clases exclusivas (por ejemplo, clasificación de imágenes de dígitos), la capa de salida se modifica utilizando ahora la función softmax (ver figura 2.8), donde la salida de cada neurona se interpreta como la probabilidad estimada para cada clase. Es importante destacar que la señal se mueve en una sola dirección, desde las entradas hacia las salidas. Este tipo de arquitectura es un ejemplo de una red neuronal secuencial (FN).

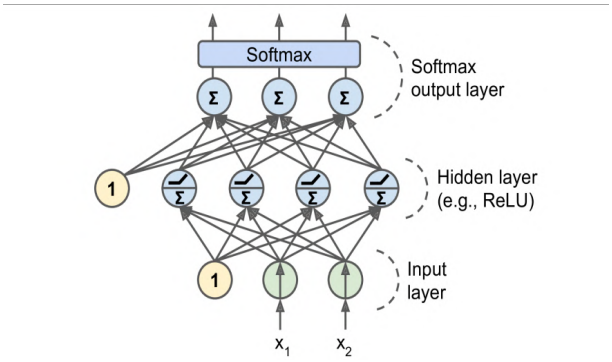


Figura 2.6: Un MLP moderno para la clasificación que incluya ReLU y softmax. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

Es importante tener mencionar que la función de activación sigmoide se usaba comúnmente en RNA debido a su parecido con las neuronas biológicas, pero la función de activación ReLU generalmente funciona mejor en la práctica.

### 2.1.5. Hiperparámetros de una red neuronal

La flexibilidad de las redes neuronales, las lleva a la gran desventaja de tener muchos hiperparámetros para ajustar. Algunos de estos pueden ser la topología de la red (cómo se conectan las neuronas),la cantidad de capas ocultas, la cantidad de neuronas por capa, el tipo de función de activación, el método de inicialización de pesos entre otros varios. Esto hace que encontrar la mejor combinación de hiperparámetros sea todo un desafío.

Para darle una solución a esto la búsqueda en cuadrícula con validación cruzada es un enfoque común para encontrar los hiperparámetros adecuados, pero requiere mucho tiempo. Por otro lado, una mejor opción puede ser la búsqueda aleatoria o emplear herramientas especializadas como Oscar, que se pueden utilizar para una optimización de hiperparámetros más rápida. Sin embargo, también es útil tener una comprensión de los valores adecuados para cada hiperparámetro, lo cual permite limitar el rango de búsqueda. Empecemos analizando la cantidad de capas ocultas.

#### Número de capas ocultas

El número de capas ocultas es un hiperparámetro importante. Mientras que una sola capa oculta puede modelar funciones complejas con suficientes neuronas, las redes profundas con múltiples capas ocultas tienen una mayor eficiencia de parámetros y pueden modelar funciones complejas con menos neuronas que las redes superficiales. Esto a su vez, hace que sea posible entrenarlas en muy poco tiempo.

Las redes neuronales profundas aprovechan las estructuras jerárquicas de los datos. Las capas inferiores modelan estructuras de bajo nivel, las capas intermedias las combinan y las capas superiores modelan estructuras de alto nivel. Esta arquitectura jerárquica ayuda con la convergencia y la generalización a nuevos conjuntos de datos. Una herramienta muy poderosa para ayudar en la generalización de la red, es la reutilización de capas inferiores de una red preentrenada, ya que esto puede acelerar el entrenamiento de nuevas redes para tareas similares. Esto permite que la red se concentre en aprender estructuras de nivel superior en lugar de comenzar desde cero.

Por lo general, comenzar con una o dos capas ocultas puede funcionar bien para muchos problemas. A medida que aumenta la complejidad de la tarea, el número de capas ocultas se puede aumentar gradualmente hasta que se produzca un sobreajuste en los datos de entrenamiento. Tareas complejas como la clasificación de imágenes o el reconocimiento de voz pueden requerir redes con docenas de capas y una cantidad significativa de datos de entrenamiento.

#### Número de neuronas por capa oculta

Evidentemente, el número de neuronas en las capas de entrada y salida siempre será un valor fijo, puesto que depende únicamente del tipo de entrada y salida que requiera la tarea o problema en el que se éste trabajando. Por otro lado, el número de neuronas por capa oculta puede seguir una estructura similar a la de un embudo, disminuyendo en número de una capa a la siguiente. Esto se debe a que múltipes características de menor nivel pueden combinarse en un número reducido de características de nivel superior. Sin embargo, hoy en día es preferible optar por usar el mismo tamaño para todas las capas ocultas, y así tener un solo hiperparámetro que ajustar, en lugar de uno por cada capa. No obstante, aumentar el número de capas tiende a proporcionar más beneficios que aumentar el número de neuronas por capa.

**Funciones de activación**

La función de activación de ReLU, se usa comúnmente en las capas ocultas, debido a su eficiencia computacional y resistencia a quedarse atascada en mesetas (regiones planas), ya que no se satura con valores de entrada grandes, a diferencia de las funciones sigmoide o tangente hiperbólica que se saturan en 1. Para la capa de salida, La función softmax es adecuada cuando se está trabajando con tareas de clasificación, mientras que si se trata de un problema de regresión, no es necesario poner una función de activación.

Estos son los tres hiperpárametros más simples que se pueden ajustar al momento de entrenar una red neuronal. Sin embargo, existen muchos otros más hiperparámetros o técnicas que se pueden emplear para facilitar el entrenamiento de una RNA, principalmente cuando se trata de redes profundas. A continuación, se describen dos técnicas importantes.

### 2.1.6. Programación de la tasa de aprendizaje

Establecer una tasa de aprendizaje óptima para la tarea con la que se éste trabajando, puede ser todo un desafío. Si se toma demasiado alta puede hacer que el entrenamiento diverja, mientras que establecerla demasiado baja conduce a una convergencia lenta. Una tasa de aprendizaje ligeramente alta puede hacer un progreso rápido inicialmente, pero no lograr establecerse en torno al óptimo. Sin embargo, generalmente se opta por encontrar soluciones subóptimas para la tasa de aprendizaje.

Encontrar una buena tasa de aprendizaje puede implicar entrenar la red varias veces con diferentes tasas de aprendizaje y comparar las curvas de aprendizaje como se muestra en la figura 2.9. La tasa de aprendizaje ideal logra un aprendizaje rápido y converge a una buena solución.

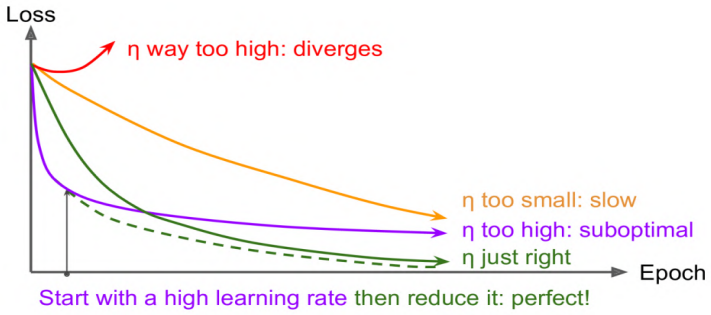


Figura 2.7: Curvas de aprendizaje para varias tasas de aprendizaje. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

Por otro lado, usar una tasa de aprendizaje constante no siempre es óptimo. Es posible llegar a una buena solución más rápido comenzando con una tasa de aprendizaje alta y reduciéndola cuando el progreso se ralentiza. Las diferentes estrategias para reducir la tasa de aprendizaje durante el entrenamiento se conocen como programas de aprendizaje. Las más comunes son:

- **Tasa de aprendizaje constante por partes predeterminada**. Esta implica establecer diferentes tasas de aprendizaje en puntos específicos del entrenamiento. Requiere un ajuste manual para encontrar las tasas de aprendizaje correctas y cuándo cambiarlas.

- **Programación del rendimiento.** Aquí, se mide el error de validación periódicamente y se reduce la tasa de aprendizaje en un factor $\lambda$ cuando el error deja de mejorar, de forma similar a la detención temprana.

- **Programación exponencial.** Esta establece la tasa de aprendizaje en función del número de iteraciones. La tasa de aprendizaje disminuye exponencialmente, normalmente por un factor de 10 cada determinados pasos. Requiere ajustar la tasa de aprendizaje inicial y la tasa de caída exponencial.

- **Programación de energía.** Esta establece la tasa de aprendizaje mediante una función de energía. Esto hace que la tasa de aprendizaje disminuya más lentamente en comparación con la programación exponencial.

Un estudio que comparó la eficacia de algunos de los programas de aprendizaje más utilizados al entrenar redes neuronales profundas en reconocimiento de voz encontró que tanto la programación de desempeño como la programación exponencial funcionaron bien. Sin embargo, se favoreció la programación exponencial debido a su simplicidad, facilidad de ajuste y convergencia ligeramente más rápida a la solución óptima.

Sin embargo, los algoritmos de optimización como AdaGrad, RMSProp y Adam ajustan automáticamente la tasa de aprendizaje durante el entrenamiento, lo que hace innecesaria la programación de tasas de aprendizaje adicionales. Para otros algoritmos de optimización, el uso de la programación exponencial o la programación del rendimiento puede acelerar significativamente la convergencia.

### 2.1.7. Técnicas de Regularización

Las redes neuronales profundas generalmente suelen tener millones de parámetros, lo cual hace que tengan una gran flexibilidad para adaptarse a distintos conjuntos de datos complejos. No obstante, tienden a sobreajustarse al conjunto de entrenamiento con gran facilidad. Para solucionar este problema, se emplean distintas técnicas de regularización, a continuación se presentan algunas de ellas.

#### Detención anticipada

La detención anticipada es una técnica de regularización común en la que el entrenamiento se interrumpe cuando el rendimiento del modelo en el conjunto de validación comienza a disminuir. Esta técnica es muy fácil de implementar al entrenar una RNA y además, se ha observado que proporciona mejores resultados cuando se combina con otras técnicas de regularización.

#### Regularizaciones $l_1$ y $l_2$

La regularización $l_1$, agrega un término a la función de costo del modelo que es proporcional a la suma de los valores absolutos de los coeficientes del modelo. Esto tiene el efecto de penalizar los coeficientes más pequeños y, en algunos casos, establecer algunos de ellos en cero. En consecuencia, $l_1$ tiende a producir modelos más dispersos y puede ayudar en la selección automática de características.

La regularización $l_2$, agrega un término a la función de costo del modelo que es proporcional a la suma de los cuadrados de los coeficientes del modelo. Al igual que $l_1$, $l_2$ penaliza los coeficientes más grandes, pero a diferencia de $l_1$, no establece los coeficientes en cero de forma automática. En su lugar, $l_2$ tiende a empujar los coeficientes hacia valores más pequeños y distribuirlos de manera más uniforme, reduciendo la complejidad del modelo y evitando el sobreajuste.

En resumen, la regularización $l_1$ y $l_2$ son similares a los modelos lineales y se utilizan para restringir los pesos de conexión de las redes neuronales.

#### Dropout

Dropout, es probablemente la técnica de regularización más popular al entrenar RNAs. Su funcionamiento es bastante simple, pues consiste solo en descarta neuronas aleatoriamente (excluyendo las neuronas de salida) durante cada paso de entrenamiento. Esto obliga a la red a ser más robusta y menos dependiente de neuronas específicas, lo que lleva a una mejor generalización. Se ha demostrado que el uso de esta técnica mejora ligeramente la precisión de las redes neuronales.

En otras palabras, Dropout genera una red neuronal única en cada paso de entrenamiento, creando un conjunto de diversos modelos. Este conjunto contribuye a mejorar el rendimiento y la capacidad de generalización de la red. Por otro lado, durante el entrenamiento, para compensar la deserción de neuronas, los pesos de conexión de entrada deben multiplicarse por la probabilidad de mantenimiento (1 - tasa de deserción) o dividir la salida de la neurona por la probabilidad de mantenimiento.

Existe una variante de Dropout, llamada Dropconnect, donde las conexiones individuales se eliminan aleatoriamente en lugar de neuronas completas. Sin embargo, Dropout en general ha mostrado dar mejores resultados que Dropconnect.

### 2.1.8. Redes Neuronales Convolucionales

Es momento de hablar de una de las arquitecturas de RNA más impresionantes y poderosas en la actualidad. Para ello, primero hablemos de un dato histórico interesante. Mientras que, nosotros los humanos podemos realizar tareas como detectar un cachorro en una imagen o reconocer tareas habladas sin esfuerzo alguno, las computadoras lucharon con tales tareas hasta hace poco. Esto se debe a que la percepción ocurre en gran medida fuera de nuestra conciencia, dentro de módulos sensoriales especializados en nuestros cerebros. Comprender estos módulos sensoriales es clave para comprender la percepción.

Las redes neuronales convolucionales (CNN), fueron inspiradas en la corteza visual del cerebro y se han utilizado para el reconocimiento de imágenes desde la década de 1980. Con los avances en el poder computacional, los datos de entrenamiento disponibles y las técnicas de entrenamiento de redes neuronales profundas, las CNN han logrado un rendimiento sobrehumano en tareas visuales complejas. Impulsan aplicaciones como la búsqueda de imágenes, los vehículos autónomos y la clasificación de videos. Además, no solo son buenas en tareas visuales, también son buenas en el reconocimiento de voz y en el análisis de series de tiempo.

Una razón importante por la cual optar por una CNN, en lugar de una red neuronal profunda regular con capas completamente conectadas para el reconocimiento de imágenes, es debido a

la gran cantidad de parámetros que se requieren para imágenes grandes. Las CNN abordan este problema mediante el uso de capas parcialmente conectadas, lo que reduce la cantidad de conexiones y mejora la eficiencia del modelo.

### Capa Convolucional

La pieza clave de una CNN, radica en la capa convolucional, la cual consta de neuronas que no están conectadas a cada píxel de la imagen de entrada (como lo estarían con capas planas), sino solo a los píxeles dentro de sus campos receptivos como se muestra en la figura 2.10. Esta estructura jerárquica permite que la red se centre en características de bajo nivel en las primeras capas y las reúna en características de nivel superior en capas posteriores, lo que hace que las CNN sean efectivas para el reconocimiento de imágenes.

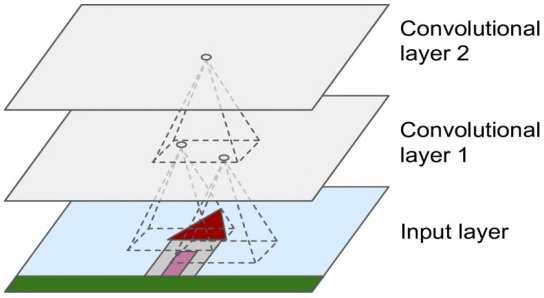


Figura 2.8: Capas CNN con campos receptivos locales rectangulares. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

Cada neurona de una capa está conectada a una región específica de la capa anterior, determinada por la altura $f_h$ y el ancho $f_w$ del campo receptivo (ver figura 2.11). El relleno con ceros (zero padding) se usa comúnmente para mantener la misma altura y ancho entre las capas, y como su nombre lo dice, implica agregar ceros alrededor de las entradas. Otro parámetro importante en una capa convolucional es el paso (stride), que es la distancia entre dos campos receptivos consecutivos. Esta distancia, se puede ajustar para conectar una capa de entrada más grande con una capa más pequeña. Las neuronas de la capa superior están conectadas a regiones específicas de la capa anterior en función de los valores del paso.

### Filtros

Los pesos de las neuronas en una CNN se pueden representar como filtros o núcleos de convolución. Los filtros son pequeñas imágenes que coinciden con el tamaño del campo receptivo, los cuales resaltan patrones específicos en la imagen de entrada. Por ejemplo, un filtro de línea vertical enfatiza las líneas verticales, mientras que un filtro de línea horizontal enfatiza las líneas horizontales (ver figura 2.12). Las neuronas que usan estos filtros ignoran otras partes de su campo receptivo.

Cuando todas las neuronas de una capa usan el mismo filtro, la capa genera un mapa de características que resalta las áreas en la imagen similares al filtro, como se muestra en la figura 2.12. La CNN aprende a combinar estos filtros en patrones más complejos durante el entrenamiento.

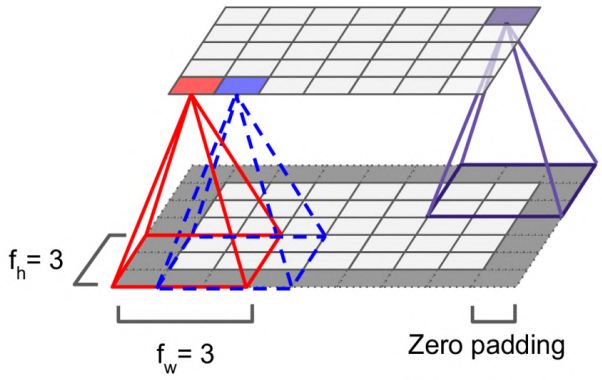


Figura 2.9: Conexiones entre capas y padding cero. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

Por ejemplo, un patrón cruzado puede activar neuronas que responden a filtros tanto verticales como horizontales, lo que indica la presencia de un patrón en forma de cruz en una imagen.

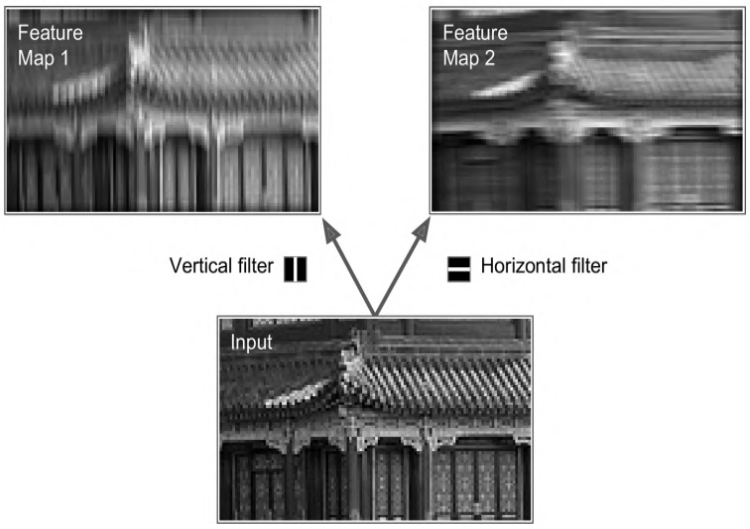


Figura 2.10: Aplicando dos filtros diferentes para obtener dos mapas de características. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

### Múltiples mapas de características

En realidad, una capa convolucional se compone de varios mapas de características del mismo tamaño representados en 3D, en lugar de una fina capa 2D (ver figura 2.13). Cada mapa de características consta de neuronas que comparten los mismos parámetros (pesos y términos de sesgo), mientras que diferentes mapas de características pueden tener parámetros diferentes. El campo receptivo de una neurona se extiende a través de todos los mapas de características de las capas anteriores, lo que permite que la capa convolucional aplique simultáneamente múltiples filtros

para detectar varias características en las entradas.

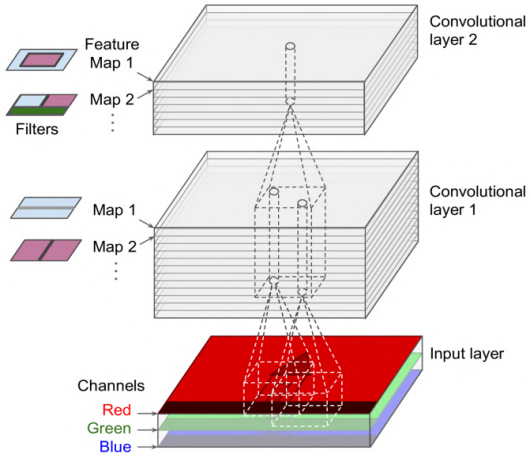


Figura 2.11: Capa de convolución con múltiples mapas de característica e imágenes con tres canales. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

Compartir parámetros entre neuronas en un mapa de características reduce significativamente la cantidad de parámetros en el modelo. Además, una vez que la CNN aprende a reconocer un patrón en una región, puede reconocerlo en cualquier otra región. A diferencia de una red neuronal profunda regular, que solo puede reconocer un patrón en la región específica que lo aprendió.

Por otro lado, las imágenes de entrada, también se suelen componer de varias subcapas, donde cada una de ellas representa un canal de color. Generalmente, las imágenes están en formato RGB, por lo que tienen 3 canales (rojo, verde y azul). Sin embargo, las imágenes en escala de grises tienen un solo canal, mientras que otras imágenes, como las imágenes satelitales que capturan diferentes frecuencias de luz, pueden tener una mayor cantidad de canales.

Finalmente, la conexión entre una neurona ubicada en el mapa de características $k$ de la capa convolucional $l$ y las salidas de las neuronas en la capa anterior $l-1$, está dada por la ecuación 2.2, la cual es un poco confusa por todos los índices que tiene, pero lo único que hace es calcular la suma ponderada de todas las entradas, más el término de sesgo.

$$z_{i,j,k} = b_k + \sum_{u=1}^{f_h} \sum_{v=1}^{f_w} \sum_{k'=1}^{f'_n} (x_{i',j',k'} \cdot w_{u,v,k',k}) \quad \text{con} \begin{cases} i' = u \cdot s_h + f_h - 1 \\ j' = v \cdot s_w + f_w - 1 \end{cases} \tag{2.2}$$

- $z_{i,j,k}$ es la salida de la neurona localizada en la fila $i$, columna $j$ en el mapa de características $k$ de la capa convolucional actual (capa $l$).
- $s_h$ y $s_w$ son los pasos verticales y horizontales (tamaño del stride), $f_h$ y $f_w$ son la altura y el ancho del campo receptivo, y $f_{n'}$ es el número total de mapas de características de la capa anterior (capa $l-1$).
- $x_{i',j',k'}$ es la salida de la neurona localizada en la capa $l-1$, fila $i'$, columna $j'$, mapa de características $k'$ (o canal $k'$ si la capa anterior es la capa de entrada).

- $b_k$ es el término de sesgo para el mapa de características $k$ (en la capa $l$).

- $w_{u,v,k',k}$ es el peso de conexión entre cualquier neurona en el mapa de características $k$ de la capa $l$ y su entrada ubicada en la fila $u$, columna $v$ (relativo al campo receptivo de la neurona) y el mapa de características $k'$

### Capa de Agrupación (Pooling Layer)

Las capas de agrupación tienen como objetivo reducir los mapas de características o los canales de la imagen (para el caso de la capa de entrada), lo que reduce la carga computacional, el uso de memoria y la cantidad de parámetros. Esto ayuda a evitar el sobreajuste y permite que la red tolere ligeros cambios en la imagen (invariancia de ubicación).

La conexión entre una neurona en una capa de agrupación y las neuronas de la capa anterior, es muy similar a las descritas para capas convolucionales, es decir, se conecta a un número limitado de neuronas en la capa anterior, dentro de un pequeño campo receptivo rectangular. Es necesario definir el tamaño del paso y el tipo de relleno de la capa de agrupación. No obstante, las neuronas agrupadas no tienen pesos; agregan las entradas usando una función de agregación como el máximo o la media.

En la figura 2.14, se ilustra una capa de agrupación máxima (maxpooling), que comúnmente es el tipo más usado de estas capas. En este ejemplo, se utiliza un núcleo de agrupación de 2x2, un paso de 2 y ningún tipo de relleno. Solo el valor de entrada máximo dentro de cada kernel se pasa a la siguiente capa, mientras que las otras entradas se descartan. Esto da como resultado una reducción de la muestra de la salida por un factor de 2 en ambas dimensiones, reduciendo efectivamente el área de la imagen cuatro veces y por lo tanto eliminando el 75 % de los valores de entrada.

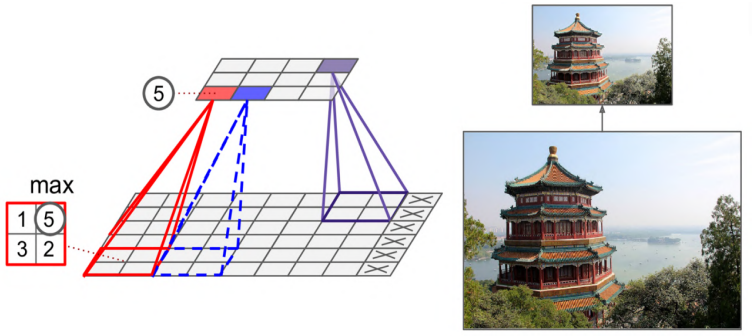


Figura 2.12: capa maxpooling (núcleo de agrupación 2x2, paso 2, sin relleno). Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

Las capas de agrupación normalmente funcionan de forma independiente en cada canal de entrada, manteniendo la misma profundidad en la salida que en la entrada. Alternativamente, la agrupación se puede realizar sobre la dimensión de profundidad, reduciendo el número de canales mientras se mantienen las dimensiones espaciales (alto y ancho) sin cambios.

### 2.1.9. Redes Neuronales Residuales (ResNet)

Finalmente, pero no menos importante, es turno de hablar de las redes neuronales residuales (ResNet). Presentadas por primera vez por Kaimig He y su equipo en 2015, las ResNet son una de las arquitecturas de RNA más populares en el campo del aprendizaje profundo. De hecho, un modelo realizado con esta nueva arquitectura de red neuronal fue capaz de ganar el primer lugar en el desafío ILSVRC ImageNet de ese mismo año.

La innovación clave de las ResNet es el uso de conexiones de salto, también conocidas como conexiones de acceso directo o simplemente conexiones residuales. En estas conexiones residuales a la entrada que alimenta a una capa también se le agrega (es decir, suma) la salida de una capa ubicada un poco antes de ella. Lo cual hace, que en lugar de esperar que cada capa aprenda directamente la representación deseada, las capas aprendan las diferencias entre la representación deseada y la representación actual.

Cuando se agrega una conexión de residual, la red modela la función f(x) = h(x) - x, en lugar de solo la función h(x), donde h(x) es la función objetivo que se quiere modelar. Esto recibe el nombre de aprendizaje residual (ver figura 2.15). Este enfoque acelera el entrenamiento, especialmente cuando la función objetivo es similar a la función de identidad.

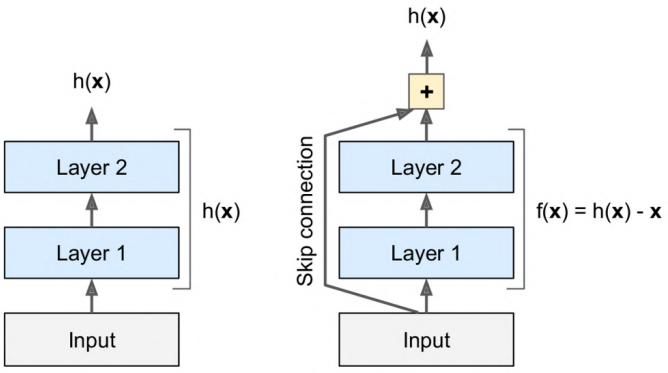


Figura 2.13: Aprendizaje residual. Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

Emplear muchas conexiones residuales permite que la señal se propague a través de la red, incluso si algunas capas aún no han comenzado a aprender. Esto permite que la red progrese al transferir fácilmente la señal a través de ella sin llegar a ser atenuada. Las ResNet profundas, están construidas como una pila de unidades residuales, donde cada unidad es una pequeña red neuronal con una conexión de salto. En la figura 2.16, se muestra una comparación entre una red neuronal secuencial y una red residual (ambas redes son profundas)

La arquitectura de ResNet es simple y se le puede implementar tanto a una red densa (Perceptrón multicapa), como a una red convolucional. Actualmente, las ResNet se considera la arquitectura más potente y sencilla. Sin embargo, también vale la pena explorar otras arquitecturas como VGGNet e Inception-v4, con un rendimiento competitivo en el desafío ILSVRC.

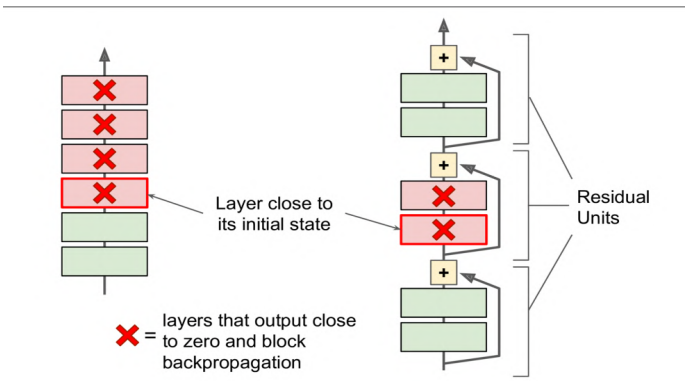


Figura 2.14: Red neuronal secuencial profunda (lado izquierdo) comparada con una red residual (lado derecho). Figura tomada de: "Hands On Machine Learning with Scikit-Learn and TensorFlow", Géron, A. (2017).

## 2.2. Métodos de Interpretabilidad

En los modelos de aprendizaje profundo, especialmente en aquellos basados en Redes Neuronales Convolucionales (RNC), la capacidad de interpretar las decisiones del modelo constituye un aspecto fundamental para su aplicación en contextos sensibles como el diagnóstico médico. A pesar de su elevado rendimiento en tareas de clasificación y detección de objetos, las RNC suelen considerarse "cajas negras" debido a la dificultad para comprender qué características o regiones de una imagen influyen en sus predicciones. Esta limitación ha motivado el desarrollo de diversos métodos de interpretabilidad, los cuales buscan proporcionar una representación visual o conceptual del proceso interno de decisión del modelo.

De manera general, los métodos de interpretabilidad pueden agruparse en dos grandes categorías: *métodos basados en activaciones* y *métodos basados en perturbaciones*. Los primeros analizan directamente las respuestas de las capas convolucionales internas del modelo para identificar qué regiones de la imagen provocan una activación más intensa. Dentro de esta categoría se encuentran técnicas como *Grad-CAM*, *Guided Backpropagation* y *Layer-wise Relevance Propagation* (LRP). Por otro lado, los métodos basados en perturbaciones evalúan los cambios en la salida del modelo cuando se modifica o elimina información de la entrada, como ocurre con los métodos de *Occlusion Sensitivity* o mapas de sensibilidad.

### 2.2.1. Grad-CAM (Gradient-weighted Class Activation Mapping)

El método Grad-CAM, propuesto por Selvaraju et al. (2017), es uno de los enfoques más utilizados para la interpretabilidad de redes convolucionales. Su principio consiste en utilizar los gradientes de la clase de interés con respecto a las activaciones de una capa convolucional específica, generalmente la última capa antes de la clasificación, para determinar qué regiones de la imagen contribuyeron más a la predicción.

Formalmente, Grad-CAM calcula los gradientes de la puntuación de una clase $c$ con respecto a los mapas de activación $A^k$ de una capa convolucional, obteniendo un peso $\alpha_k^c$ para cada mapa según la media espacial de dichos gradientes. Estos pesos representan la importancia de cada filtro para la predicción de la clase. Posteriormente, se genera un mapa de calor $L^c_{\text{Grad-CAM}}$ como una

combinación lineal ponderada de las activaciones, seguido de una función ReLU para mantener únicamente las contribuciones positivas:

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha^c_k A^k\right)$$

El resultado es un mapa de activación que puede superponerse a la imagen original, mostrando las regiones visuales más relevantes para la decisión del modelo. Una de sus principales ventajas es que puede aplicarse a una amplia variedad de arquitecturas sin necesidad de modificar el modelo ni acceder a los pesos de sus capas internas. Sin embargo, su resolución espacial depende directamente del tamaño del mapa de características de la capa seleccionada, lo que puede limitar la precisión de las regiones destacadas.

### 2.2.2. Sensibilidad de Oclusión (Occlusion Sensitivity)

El método de Sensibilidad de Oclusión, introducido por Zeiler y Fergus (2014), pertenece a los enfoques de tipo perturbativo. Su idea central consiste en medir cómo varía la salida del modelo cuando se ocultan sistemáticamente pequeñas regiones de la imagen de entrada. Si la ocultación de una región provoca una disminución significativa en la probabilidad de la clase objetivo, se infiere que dicha región contiene información relevante para la predicción.

Para implementar este método, se define una cuadrícula de tamaño fijo sobre la imagen y se sustituyen progresivamente las regiones cubiertas por un valor constante (por ejemplo, cero o el valor medio del conjunto de datos). Para cada posición de oclusión, se calcula la diferencia entre la probabilidad original y la probabilidad obtenida con la región ocluida. Estas diferencias se almacenan en un mapa que refleja la importancia de cada zona de la imagen, permitiendo identificar las áreas de mayor influencia en la clasificación final.

A diferencia de Grad-CAM, la Sensibilidad de Oclusión no depende de los gradientes ni de las activaciones internas del modelo, lo que la convierte en una técnica más general y directamente interpretable. Sin embargo, su principal desventaja es el alto costo computacional, ya que requiere evaluar el modelo un gran número de veces, proporcional al número de regiones en las que se divide la imagen.

# Capítulo 3

# Desarrollo Experimental del Filtro de Extracción de Características

Este capítulo presenta el diseño, implementación y evolución del filtro propuesto para la visualización y compresión de los procesos de aprendizaje de las redes neuronales convolucionales (RNC). A lo largo del desarrollo del filtro, se llevaron a cabo distintas pruebas experimentales con versiones sucesivas del método, cada una diseñada para abordar limitaciones identificadas en etapas previas. Las pruebas se realizaron en dos bases de datos con distintos niveles de complejidad (MNIST e ImageNet), lo que permitió analizar el comportamiento del filtro en escenarios controlados y en contextos más realistas. Los resultados obtenidos, así como los ajustes metodológicos implementados en cada fase, se presentan en orden cronológico, resaltando cómo las dificultades encontradas contribuyeron al diseño de la versión final del filtro propuesto en este trabajo. Asimismo, se incluye un análisis comparativo frente a dos métodos ampliamente utilizados en la literatura, sensibilidad de oclusión y Grad-CAM, con el fin de evaluar su desempeño relativo en términos de interpretabilidad y eficiencia. Con ello, no solo se delimitaron las fortalezas y limitaciones del filtro propuesto, sino que también se establecieron las bases para su implementación en un escenario de mayor relevancia y complejidad, la clasificación de imágenes médicas. Este desafío será abordado en el próximo capítulo.

## 3.1. Formulación inicial del filtro

La idea detrás del filtro surge al reflexionar sobre cómo los modelos basados en RNC logran clasificar una imagen identificando una clase específica presente en ella. Por ejemplo, un modelo entrenado para detectar perros en imágenes, en un problema de clasificación binaria donde 1 indica la presencia de un perro y 0 su ausencia, con una arquitectura de red adecuada y suficientes instancias de entrenamiento, puede identificar un perro en una imagen incluso cuando también hay personas u otros objetos presentes. En bases de datos variadas, donde algunas imágenes solo muestran una parte del perro, el modelo aún puede clasificar correctamente. Esto nos llevó a preguntarnos: ¿cuál es la menor cantidad de información necesaria en la imagen asociada a la clase objetivo para que el modelo sea capaz de clasificarla correctamente?

Con esta idea en mente, decidimos desarrollar un filtro de imágenes que, basándose en las

clasificaciones realizadas por uno de estos modelos, minimizara la información presente en la imagen, conservando la clasificación de la red. Este enfoque, aunque simple, permitiría identificar indirectamente las regiones más relevantes de la imagen en las que el modelo se basa para realizar la clasificación.

La metodología seguida para implementar esta idea consistió en formular un problema de optimización con restricciones, cuyo objetivo es minimizar una función de costo definida por dos términos (ver Ecuación 3.1). El primer término se enfoca en reducir la información contenida en la imagen, calculando el error cuadrático medio (ECM) entre las intensidades de los píxeles de la imagen $I$ y una imagen del mismo tamaño completamente negra (es decir, compuesta únicamente de ceros). El segundo término busca mantener la predicción realizada por el modelo basado en RNC, evaluando el ECM entre el valor real de la imagen original, $y$, y la salida de la red, $\hat{y}$. Ambos valores están definidos en términos de la probabilidad de pertenencia a la clase objetivo, por lo que se cumple que $y = 1$, dado que este es el valor verdadero correspondiente a la clase en cuestión.

$$J(I) = \frac{1}{N}\sum_{i=1}^{N}(I_i - 0)^2 + (y - \hat{y}(I))^2, \tag{3.1}$$

Donde, $N$ es el número total de píxeles en la imagen y $I_i$ es la intensidad del i-ésimo píxel en la imagen. Para resolver este problema de optimización, se utilizó el método de descenso de gradiente, calculando el gradiente de la función de costo $J(I)$ con respecto a las intensidades de los píxeles de la imagen $I$. El funcionamiento de este filtro se puede resumir en los siguientes pasos:

1. Seleccionar una imagen de prueba.
2. Realizar la predicción de la imagen de prueba mediante el modelo de RNC preentrenado.
3. Calcular el valor de la función de costo $J(I)$, considerando dos restricciones: mantener la predicción de la red y minimizar la cantidad de información presente en la imagen.
4. Calcular el gradiente de $J(I)$ con respecto a $I$.
5. Actualizar las intensidades de los píxeles de la imagen utilizando el método de descenso de gradiente.
6. Repetir los pasos 3, 4 y 5 durante un número predeterminado de iteraciones, hasta que el valor de la función de costo converja a un valor constante.
7. Mostrar en pantalla la comparación entre la imagen original y la imagen filtrada.

## 3.2. MNIST

Para evaluar el funcionamiento del filtro propuesto, primero se utilizó la base de datos **MNIST**. MNIST es un conjunto de datos recopilado por el Instituto Nacional de Estándares y Tecnología (NIST, por sus siglas en inglés) en la década de 1980. Esta base de datos es ampliamente reconocida por la comunidad de aprendizaje automático (ML, por sus siglas en inglés) y es considerada el *"Hola Mundo"* del ML debido a su simplicidad y popularidad. El problema detrás de MNIST consiste en clasificar imágenes en escala de grises de dígitos escritos a mano (28×28 píxeles) en 10 categorías distintas, representando los números del 0 al 9. Por esta razón, cuando se propone un nuevo modelo

de clasificación o un método para interpretar representaciones aprendidas, como es el caso de este trabajo, una buena práctica es evaluarlo primero con MNIST.

Para este propósito, se entrenó una red neuronal convolucional (RNC) capaz de resolver MNIST. La arquitectura del modelo empleado consistió en una sola capa convolucional con 5 filtros y un kernel de 2×2, seguida de una capa de *maxpooling* de 2×2 y una red densa de 100 unidades. Durante el entrenamiento, se utilizó la función de costo *categorical crossentropy*, el optimizador *RMSprop* y la métrica *accuracy*, todos ellos hiperparámetros comúnmente aplicados en problemas de clasificación multiclase. Tras 50 épocas de entrenamiento, el modelo logró un porcentaje de aciertos del 98 % en el conjunto de prueba de MNIST, un resultado satisfactorio para este conjunto de datos.

Una vez obtenido el modelo, se aplicó el filtro propuesto a 10 imágenes del conjunto de prueba de MNIST. En la Figura 3.1 se muestran los resultados obtenidos.

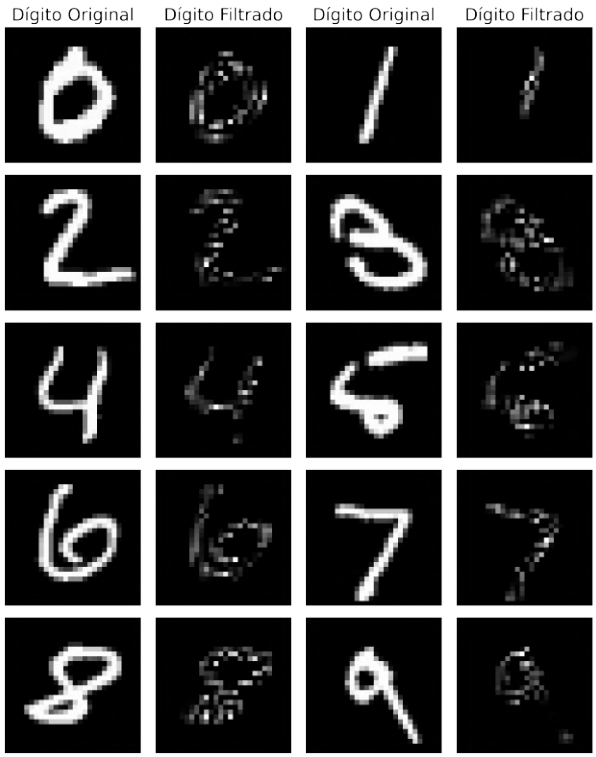


Figura 3.1: Resultados del filtro propuesto aplicados a imágenes de MNIST utilizando una RNC con una sola capa convolucional de 5 filtros de 2×2, seguida de un maxpooling de 2×2 y una red densa de 100 unidades.

A partir de estos resultados obtenidos surge la interrogante de si el filtro está funcionando correctamente. Responder a esta pregunta no es trivial, ya que la idea del filtro es reducir la cantidad de información en la imagen mientras se conserva la clasificación del modelo. En el caso de MNIST, esto equivale a determinar cuántos píxeles pueden eliminarse de una imagen sin que el dígito representado pierda su identidad. En la Figura 3.1, se observa que, tras el filtrado, el modelo sigue clasificando correctamente los dígitos y estos aún son reconocibles visualmente, lo que sugiere que el filtro está funcionando adecuadamente. Sin embargo, para evaluar su comportamiento en un escenario más desafiante, se generaron imágenes con pequeñas manchas artificiales. Estas manchas no deberían afectar la clasificación, ya que, por ejemplo, un 7 sigue siendo un 7 independientemente

de la presencia de ruido visual en la imagen. En consecuencia, si el filtro opera correctamente, debería eliminar las manchas, ya que no aportan información relevante para la clasificación.

Para evaluar esta hipótesis, primero se procesaron estas imágenes con manchas generadas artificialmente asegurando que el modelo de RNC fuese capaz de clasificarlas correctamente a pesar del ruido. Posteriormente, se aplicó el filtrado y se obtuvieron los resultados mostrados en la Figura 3.2. En general, el filtro logró eliminar las manchas en la mayoría de los casos, lo que confirma que tiende a suprimir información irrelevante. No obstante, para los dígitos *2* y *7* las manchas únicamente fueron atenuadas, sin llegar a desaparecer por completo. Por otro lado, en el caso del *1* ocurrió un comportamiento inesperado, ya que la mancha se intensificó mientras que el trazo del dígito se redujo notablemente. En lugar de replantear el diseño del filtro, consideramos que estas limitaciones podrían deberse a las representaciones aprendidas por el modelo durante su entrenamiento.

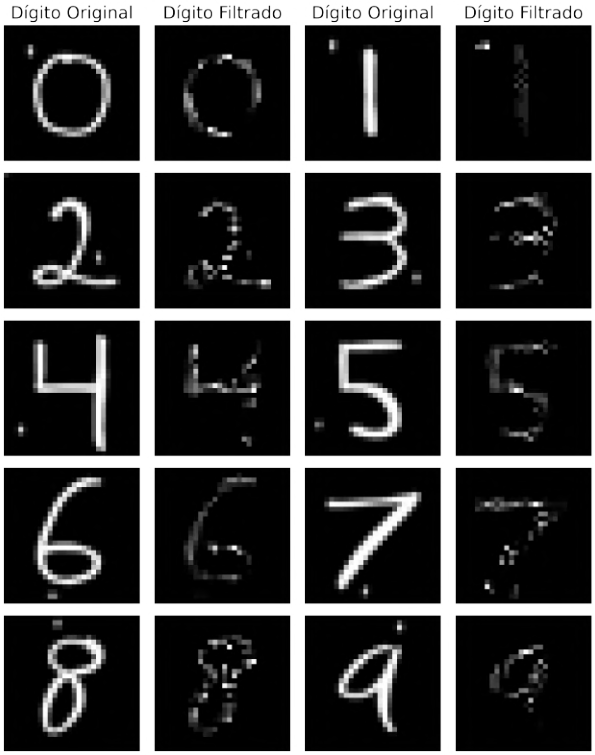


Figura 3.2: Resultados del filtro propuesto aplicado a dígitos de MNIST con manchas artificiales.

El proceso de extracción de características en una RNC está determinado por las capas convolucionales y los filtros que las conforman. La cantidad de filtros afecta la diversidad de las características extraídas, mientras que el tamaño de los filtros influye en el nivel de detalle de dichas características. En este caso, la incapacidad del filtro para eliminar completamente las manchas podría deberse a que el modelo aprendió representaciones demasiado locales y no es capaz de diferenciar entre una mancha y una parte del dígito.

Para poner a prueba esta hipótesis, se entrenó una nueva RNC desde cero con la misma arquitectura base, pero con una modificación clave: en lugar de emplear 5 filtros de tamaño $2 \times 2$, se utilizaron 10 filtros de $10 \times 10$ en la capa convolucional. El modelo fue entrenado con los mismos hiperparámetros previamente descritos y alcanzó un rendimiento similar sobre el conjunto de prueba

de MNIST. Una vez verificado que este nuevo modelo también era capaz de clasificar correctamente los dígitos con manchas generadas artificialmente, se aplicó el filtro propuesto utilizando sus predicciones como referencia. Los resultados obtenidos se presentan en la Figura 3.3.

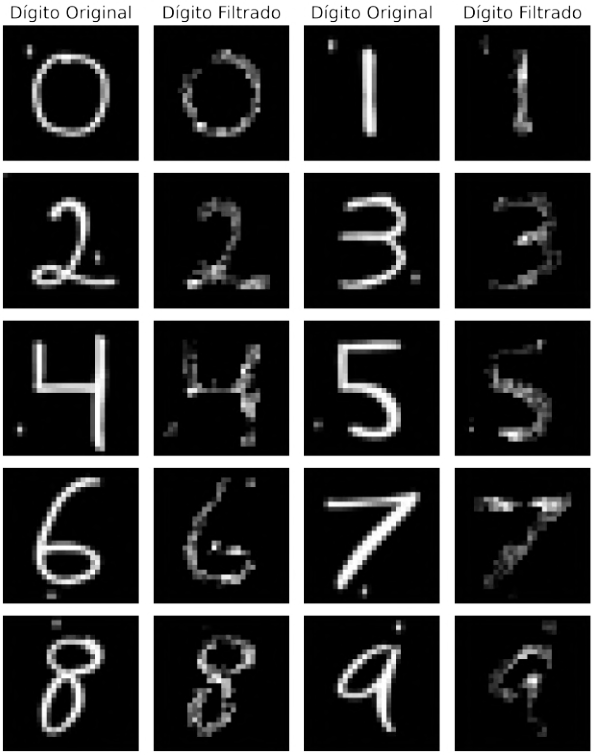


Figura 3.3: Resultados del filtro propuesto en dígitos de MNIST con manchas artificiales, utilizando una RNC con una sola capa convolucional de 10 filtros de $10 \times 10$, seguida de una operación de max pooling de $2 \times 2$ y una red densa de 100 unidades.

Se observa una mejora notable en la atenuación de las manchas con respecto al modelo anterior. Además, la atenuación de los píxeles que conforman los dígitos es considerablemente menor, es decir, el filtro logra eliminar buena parte del ruido preservando al mismo tiempo la mayor parte de la estructura original del dígito. Este comportamiento sugiere que las representaciones aprendidas por el segundo modelo permiten distinguir con mayor precisión entre información relevante (la forma del dígito) e irrelevante (las manchas), lo cual se refleja en el resultado del filtrado. Aunque no se consigue una eliminación completa del ruido en todos los casos, los contornos de los dígitos se conservan de forma más nítida. Esto contrasta con el primer modelo, donde si bien las manchas eran parcialmente eliminadas, también se veían afectadas zonas importantes del dígito, lo que podría comprometer la interpretabilidad del modelo.

Estos hallazgos refuerzan la idea de que el diseño de la arquitectura de la RNC influye directamente en la efectividad del filtro propuesto. En consecuencia, la calidad del filtrado no depende exclusivamente del método utilizado, sino también de la manera en que el modelo ha aprendido a representar la información durante su entrenamiento. Esto está en línea con el objetivo principal de este trabajo: desarrollar e implementar una técnica novedosa para la visualización y comprensión de los procesos de aprendizaje en redes neuronales convolucionales.

Con el objetivo de eliminar de forma más efectiva el ruido restante en las imágenes filtradas (es

decir, las manchas), se implementó una máscara que conserva únicamente aquellos píxeles cuya intensidad, en la imagen filtrada, fuera mayor a un umbral $\lambda = 0.05$. Esta estrategia se basa en el comportamiento observado durante el proceso iterativo del filtrado: en cada iteración, los píxeles son atenuados en función del gradiente de la salida del modelo con respecto a su intensidad. Conforme avanzan las iteraciones, este gradiente disminuye, lo que produce una estabilización en la intensidad de ciertos píxeles. Se asumió entonces que, al alcanzar este punto de estabilidad, los píxeles cuya intensidad permanece por debajo del umbral $\lambda$ ya no aportan información significativa para la clasificación y, por lo tanto, pueden ser descartados sin afectar la interpretación del modelo. En la Figura 3.4 se muestran los resultados obtenidos tras aplicar esta máscara.

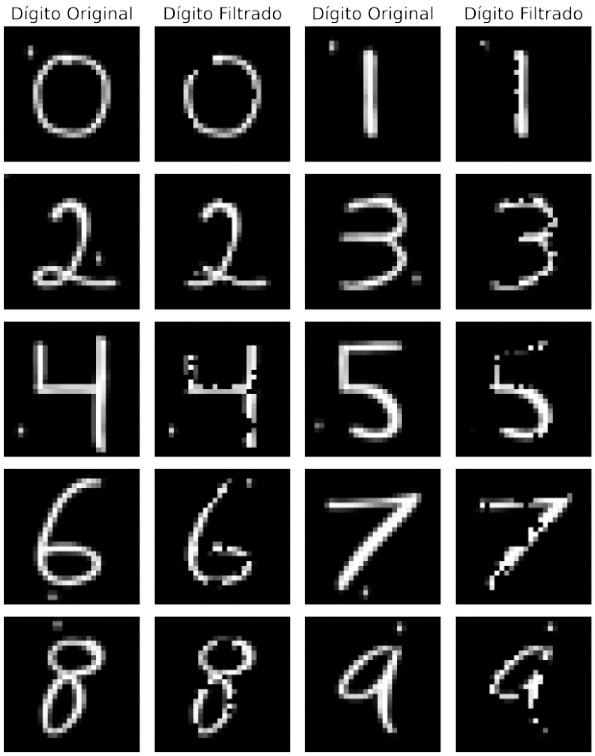


Figura 3.4: Resultados obtenidos al aplicar una máscara que conserva únicamente los píxeles de las imágenes filtradas (mostradas en la Figura 3.3) cuya intensidad es mayor al umbral $\lambda = 0.05$.

Como prueba final dentro de este conjunto de experimentos, se evaluó el desempeño del filtro propuesto ante un escenario con mayor nivel de ruido: imágenes de dígitos de MNIST con manchas artificiales más grandes que las utilizadas previamente. El procedimiento seguido fue el mismo que en los casos anteriores. Primero se verificó que el modelo fuera capaz de clasificar correctamente estas nuevas imágenes; posteriormente, se aplicó el filtro y, finalmente, se utilizó la máscara con el umbral $\lambda = 0.05$ para eliminar los píxeles residuales. Los resultados obtenidos fueron aún más satisfactorios: el filtro logró atenuar o eliminar completamente las manchas en la mayoría de los casos, al tiempo que preservó los contornos y la forma general de los dígitos originales (ver Figura 3.5).

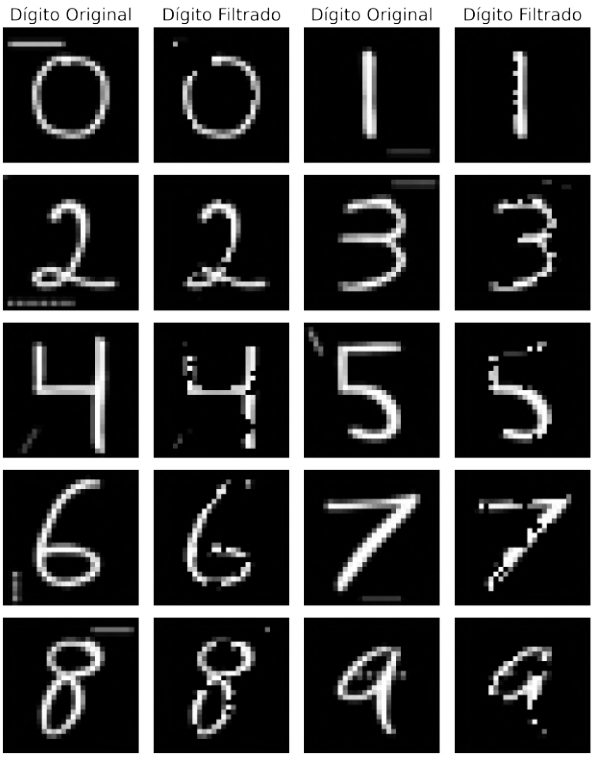


Figura 3.5: Resultados del filtro propuesto aplicado a dígitos de MNIST con manchas artificiales más grandes.

Con esta última prueba se concluyó la fase experimental con la base de datos MNIST. A pesar de su simplicidad, esta base permitió identificar y analizar con claridad el comportamiento del filtro en distintos escenarios, incluyendo casos donde las representaciones aprendidas por la red dificultaban la separación entre información relevante y ruido. Los resultados obtenidos mostraron el potencial del método propuesto para aislar regiones significativas de una imagen sin comprometer la salida del modelo. Esto motivó su aplicación en un entorno más desafiante, por lo que en la siguiente sección se explora el desempeño del filtro en una base de datos más compleja y representativa del problema real de clasificación de imágenes médicas.

## 3.3. ImageNet

Una vez evaluado el comportamiento del filtro propuesto en un entorno controlado y relativamente simple como MNIST, el siguiente paso consistió en probar su desempeño en un escenario más desafiante y realista. Para ello, se seleccionó la base de datos **ImageNet** [11], ampliamente utilizada en el campo del aprendizaje profundo debido a su diversidad y complejidad. ImageNet está compuesta por aproximadamente 1.4 millones de imágenes a color de alta resolución, distribuidas en mil categorías distintas. Estas imágenes presentan una gran variabilidad en términos de escala, orientación, iluminación y contenido, lo que convierte a ImageNet en un entorno ideal para evaluar la capacidad del filtro propuesto para identificar las regiones más relevantes en imágenes naturales. Algunas imágenes pertenecientes a ImageNet, son mostradas en la Figura 3.6.

Por otro lado, en lugar de entrenar un modelo desde cero con esta base de datos, se optó por utilizar un modelo preentrenado en ImageNet. Esta decisión se basa en la disponibilidad de múltiples modelos ya entrenados sobre este conjunto de datos, los cuales han demostrado un alto desempeño en tareas de clasificación de imágenes. El modelo seleccionado fue **Xception** [12], una red neuronal convolucional profunda propuesta por François Chollet en 2017. Xception (abreviatura de *Extreme Inception*) se fundamenta en la hipótesis de que una representación más eficiente del aprendizaje puede lograrse mediante una arquitectura que sustituya las convoluciones estándar por *depthwise separable convolutions*, las cuales consisten en una operación de convolución por canal (*depthwise*) seguida de una combinación lineal de los canales resultantes (*pointwise*). Esta arquitectura permite una mayor eficiencia computacional y, al mismo tiempo, un mejor rendimiento en tareas de clasificación de imágenes. El modelo Xception preentrenado fue cargado con los pesos aprendidos sobre ImageNet, con los cuales se reportó un porcentaje de aciertos del 94.5 % en la métrica *top-5 accuracy*.

Figura 3.6: Ejemplos de imágenes del conjunto de datos ImageNet. Imagen tomada de *Papers with Code* [13].

### 3.3.1. Experimento I

Para la primera prueba del filtro propuesto con el conjunto de datos ImageNet, se seleccionaron cuatro imágenes pertenecientes a cuatro clases distintas: *African elephant*, *Hamster*, *Mexican hairless* y *Zebra*. Todas estas clases corresponden a animales, ya que se considera más relevante, desde el punto de vista del problema médico abordado en este trabajo, analizar el comportamiento del filtro en imágenes que representen seres vivos, en lugar de categorías como vehículos o herramientas, cuya identificación tiene menor relación con la detección de patrones clínicos. Al igual que en el caso de MNIST, el objetivo fue evaluar la capacidad del filtro para identificar las regiones más relevantes que el modelo Xception considera para realizar su predicción.

En esta etapa inicial, se decidió no aplicar la máscara que conserva únicamente los píxeles con una intensidad mayor a un umbral $\lambda = 0.05$. Esto se debe a que, al tratarse de imágenes mucho más complejas y de mayor resolución, como lo son imágenes de animales, los resultados obtenidos tras el filtrado fueron considerablemente más variados entre clases en comparación con los dígitos de MNIST. Aplicar una máscara basada en un umbral fijo podía eliminar regiones relevantes o producir interpretaciones menos claras. Por esta razón, se optó por mostrar directamente los resultados

del filtro sin umbralización, con el fin de analizar de forma más libre las regiones atenuadas o preservadas por el filtro.

Un aspecto que se observó durante esta primera prueba fue que el comportamiento del filtro resultó ser más sensible a los parámetros utilizados para la actualización de los píxeles, en particular, el número de iteraciones y la tasa de aprendizaje (*learning rate*) en comparación con lo observado en MNIST. Mientras que en el caso de los dígitos fue posible utilizar un mismo conjunto de valores para todas las imágenes sin afectar significativamente el resultado, en ImageNet esta configuración tenía un impacto notable en la calidad y estructura de las imágenes filtradas. Por esta razón, se optó por realizar una exploración sistemática de estos parámetros. Para cada una de las cuatro imágenes seleccionadas, se generaron 12 variantes aplicando el filtro propuesto con todas las combinaciones posibles entre cuatro tasas de aprendizaje ($\text{lr} = \{2.5, 5.0, 7.5, 10.0\}$) y tres valores del número de iteraciones ($n = \{100, 150, 200\}$), lo que dio como resultado un total de 48 imágenes filtradas.

Debido a la cantidad de imágenes generadas, en la Figura 3.7 se presentan únicamente algunos ejemplos representativos que ilustran las principales diferencias observadas entre distintas configuraciones de parámetros.

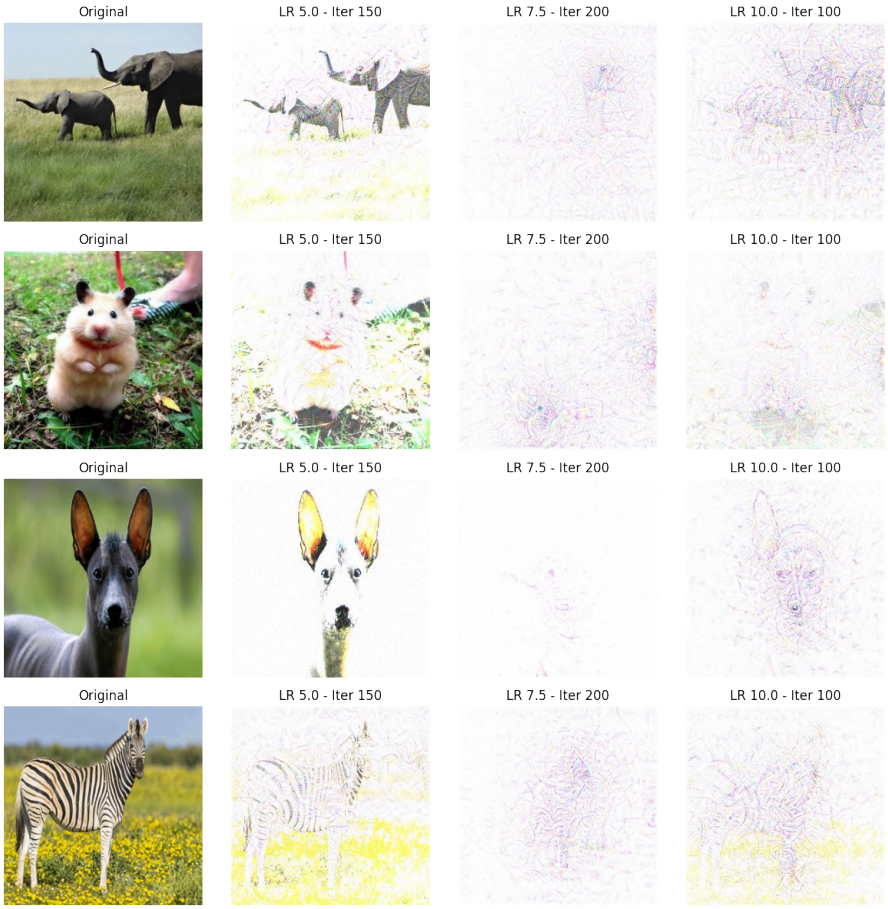


Figura 3.7: Para cada clase se presentan tres imágenes filtradas usando combinaciones distintas de la tasa de aprendizaje y número de iteraciones, seleccionadas por su valor ilustrativo. Estos valores se muestran como títulos sobre cada subimagen.

Los resultados de esta primera prueba muestran una clara dependencia del comportamiento del filtro respecto a la tasa de aprendizaje y al número de iteraciones empleadas. Para valores bajos de la tasa de aprendizaje (por ejemplo, lr = 2.5), independientemente del número de iteraciones, las imágenes filtradas presentan apenas diferencias con respecto a las originales. El filtro apenas atenúa el fondo y mantiene todos los detalles del animal prácticamente intactos. Esto indica que con una tasa de aprendizaje demasiado baja el gradiente aplicado a los píxeles no es suficiente para producir cambios perceptibles, al menos después de 200 iteraciones.

A medida que se incrementa la tasa de aprendizaje hasta valores medios (por ejemplo, lr = 5.0 con 150 iteraciones), comienza a observarse una atenuación notable del fondo, conservándose la silueta del animal de forma reconocible. Sin embargo, esta atenuación no responde a un criterio claramente interpretable, no queda evidente si el filtro prioriza la cabeza, el contorno corporal o detalles específicos para la clasificación, lo que plantea la pregunta de qué regiones considera realmente relevantes el modelo.

Cuando se emplean tasas de aprendizaje más altas (lr > 5.0), el filtro tiende a eliminar gran parte de la imagen de forma abrupta, borrando casi por completo el fondo y dejando apenas algunos trazos de colores (principalmente azules y morados), correspondientes a regiones que el modelo aún considera relevantes para mantener la predicción correcta. En algunos casos, como con la clase *mexican_ hairless* bajo una configuración de lr = 10.0 y 100 iteraciones, aún es posible distinguir contornos sutiles de orejas, ojos u hocico entre los trazos de colores (ver Figura 3.7, fila 3, columna 4), lo que sugiere que el filtro todavía conserva patrones que el modelo asocia con la clase objetivo. No obstante, la excesiva pérdida de información hace que dichos patrones resulten difíciles de interpretar o incluso desaparezcan por completo en algunas imágenes.

## Regularización por intensidad promedio

Los resultados obtenidos en la primera prueba con imágenes de ImageNet evidenciaron una limitación importante del filtro en su versión original: al emplear tasas de aprendizaje elevadas, el proceso de optimización tiende a eliminar gran parte de la información visualmente útil para la clasificación. Si bien el filtro logra preservar algunas regiones relevantes, la pérdida generalizada de contenido dificulta su interpretación visual.

Con el objetivo de mitigar esta pérdida excesiva de información, se propuso una versión mejorada del filtro que incorpora un término adicional de regularización en la función de costo descrita en la Ecuación 3.1. Este nuevo término busca mantener un equilibrio entre la eliminación de ruido y la conservación de detalles visuales que permitan una mejor comprensión del funcionamiento del modelo, penalizando los cambios excesivos en la intensidad promedio de los píxeles entre la imagen original y la imagen filtrada.

Sea $I$ la imagen original y $J$ la imagen filtrada, ambas de tamaño $n \times m$ píxeles y con tres canales de color (RGB). La intensidad promedio de cada canal se puede definir como:

$$\overline{I_R} = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} R_{ij}, \quad \overline{I_G} = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} G_{ij}, \quad \overline{I_B} = \frac{1}{nm}\sum_{i=1}^{n}\sum_{j=1}^{m} B_{ij}$$

donde $\overline{I_R}$, $\overline{I_G}$ y $\overline{I_B}$ corresponden a las intensidades promedio de los píxeles en cada canal RGB respectivamente. A partir de estos valores, se evaluaron dos estrategias para cuantificar la diferencia

de intensidades promedio entre $I$ y $J$:

- **Canal por canal:** se calcula la suma de las diferencias absolutas entre las intensidades promedio de cada canal:

$$J_{\text{pix}}(I,J) = \left|\overline{I_R} - \overline{J_R}\right| + \left|\overline{I_G} - \overline{J_G}\right| + \left|\overline{I_B} - \overline{J_B}\right| \tag{3.2}$$

- **Global:** se calcula primero la intensidad promedio global de cada imagen como el promedio de los tres canales, y luego se obtiene la diferencia absoluta entre estos promedios:

$$\overline{I} = \frac{1}{3}\left(\overline{I_R} + \overline{I_G} + \overline{I_B}\right), \quad \overline{J} = \frac{1}{3}\left(\overline{J_R} + \overline{J_G} + \overline{J_B}\right)$$

  entonces,

$$J_{\text{pix}}(I,J) = \left|\overline{I} - \overline{J}\right| \tag{3.3}$$

Con este nuevo término de regularización, la función de costo original (Ecuación 3.1) se puede reescribir como:

$$J_{\text{total}}(I,J) = \underbrace{\frac{1}{N}\sum_{i=1}^{N}(I_i - 0)^2}_{\text{Pérdida de la imagen}} + \underbrace{(y - \hat{y}(I))^2}_{\text{Pérdida de la predicción}} + \underbrace{J_{\text{pix}}(I,J)}_{\text{Regularización por intensidad}} \tag{3.4}$$

donde:

- $N$ es el número total de píxeles en la imagen original,
- $I_i$ es la intensidad del i-ésimo píxel en la imagen original,
- $y$ es el valor real para la imagen original,
- $\hat{y}(I)$ es la salida del modelo para la imagen filtrada,
- $J_{\text{pix}}(I,J)$ cuantifica el cambio en la intensidad promedio entre $I$ y $J$.

### 3.3.2. Experimento II

Una vez incorporado el nuevo término de regularización en la función de costo (ver Ecuación 3.4), se llevaron a cabo dos experimentos independientes, denominados **Experimento II.a** y **Experimento II.b**, con el objetivo de evaluar si dicho término contribuye a preservar mejor la información relevante de la imagen original sin comprometer la interpretabilidad del filtrado. En el Experimento II.a, el término de regularización se calculó utilizando la formulación global de la intensidad promedio (Ecuación 3.3), mientras que en el Experimento II.b se empleó la formulación canal por canal (Ecuación 3.2).

En ambos casos se utilizaron las mismas cuatro imágenes de prueba seleccionadas en el **Experimento I**. Se exploraron cuatro valores de la tasa de aprendizaje ($\text{lr} = \{5.0, 10.0, 20.0, 50.0\}$)

y cuatro valores del número de iteraciones ($n = \{20, 50, 100, 200\}$). Para cada combinación de parámetros se generó una imagen filtrada, lo que resultó en un total de 16 imágenes filtradas por cada imagen de prueba y por cada experimento.

Dado que los resultados obtenidos con ambas formulaciones fueron prácticamente indistinguibles, en esta sección solo se muestran los resultados del caso global (Experimento II.a), mientras que la variante por canal (Experimento II.b) se omite por motivos de brevedad. La Figura 3.8 presenta algunos ejemplos representativos de este experimento.

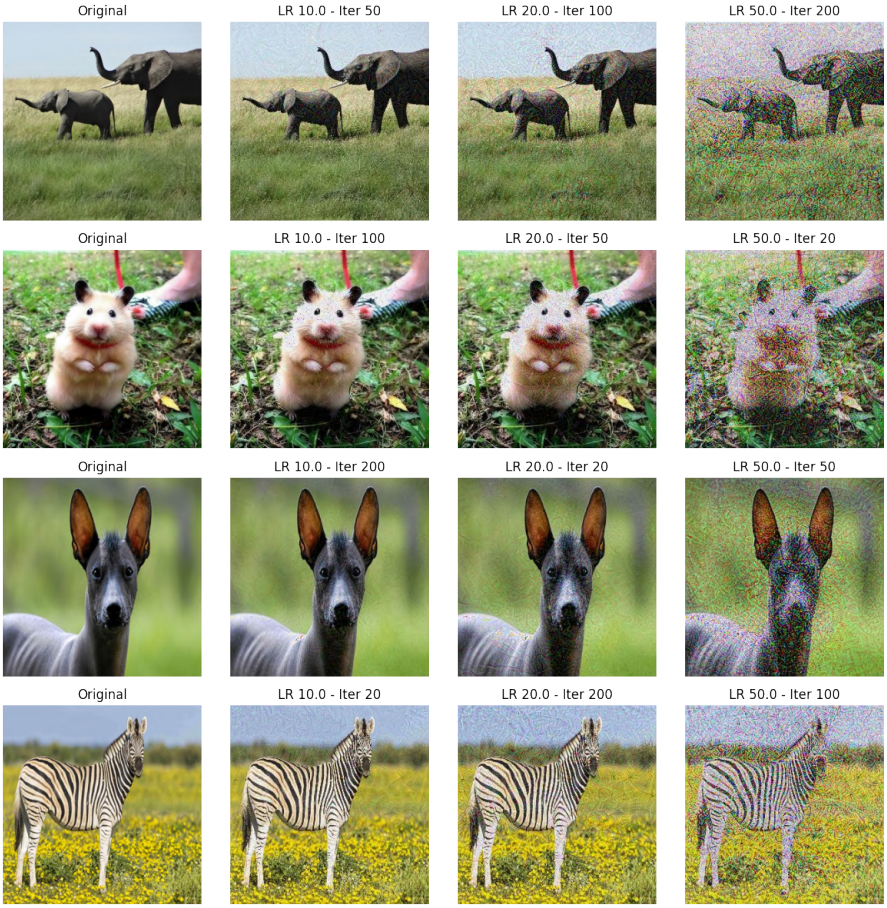


Figura 3.8: Ejemplos representativos del Experimento II.a. El término de regularización por intensidad se calcula según la Ecuación 3.3. Las combinaciones de parámetros utilizadas se indican sobre cada subimagen.

El análisis cualitativo de los resultados muestra que ambas estrategias de regularización producen comportamientos muy similares en el filtro. En ambos casos, para tasas de aprendizaje bajas (por ejemplo, $\text{lr} < 20$), las imágenes filtradas presentan diferencias mínimas respecto a las originales, independientemente del número de iteraciones. Al aumentar la tasa de aprendizaje a valores intermedios ($\text{lr} = 20$), comienzan a aparecer distorsiones visuales, como manchas de color que deforman y/o pixelan parcialmente la figura, aunque el fondo se mantiene prácticamente intacto.

Con tasas de aprendizaje altas ($\text{lr} = 50$), las distorsiones se intensifican y tienden a cubrir toda la imagen; sin embargo, tanto el animal como el fondo siguen siendo visualmente distinguibles. Este comportamiento sugiere que el nuevo término de regularización podría estar ejerciendo un

peso dominante sobre el término que busca minimizar la información presente en la imagen. Como consecuencia, el filtro no logra eliminar por completo las regiones irrelevantes para la clasificación y, en su lugar, genera artefactos visuales, tales como patrones de color y ruido estructurado.

A partir de estas observaciones, se concluye que la forma de calcular la intensidad promedio (global o por canal) tiene un impacto marginal en el resultado final del filtrado. Por esta razón, y debido a su menor complejidad computacional, para los experimentos posteriores se adoptó la versión descrita en la Ecuación 3.3.

## Coeficiente de ponderación

Los resultados obtenidos en el segundo experimento con imágenes de ImageNet sugieren que, si bien el nuevo término de regularización por intensidad promedio permite conservar mayor información en la imagen filtrada, también puede limitar la capacidad del filtro para eliminar de forma efectiva las regiones no relevantes para la clasificación. Ante este nuevo desafío, se introduce una nueva modificación al filtro mediante la incorporación de un coeficiente de ponderación que permita controlar la influencia relativa de cada término de la función de costo.

Recordando la formulación presentada en la Ecuación 3.4, la función de costo puede expresarse como la suma de tres componentes:

$$\text{loss total} = \text{loss img} + \text{loss pred} + \text{loss pix}$$

donde:

- **loss img**: término que penaliza la magnitud de los valores de los píxeles, favoreciendo imágenes cercanas a cero.
- **loss pred**: término que asegura que la predicción del modelo para la imagen filtrada se mantenga consistente con la imagen original.
- **loss pix**: término de regularización que penaliza grandes variaciones en la intensidad promedio de los píxeles.

Si definimos a $\gamma$ como un coeficiente de ponderación ($0 \leq \gamma \leq 1$), es posible reescribir la pérdida total de la siguiente manera:

$$\text{loss total} = (1-\gamma)\,\text{loss img} + \text{loss pred} + \gamma\,\text{loss pix} \tag{3.5}$$

Esta formulación permite ajustar de forma controlada el peso relativo de los términos **loss img** y **loss pix**, los cuales influyen directamente en la cantidad y el tipo de información que se conserva en la imagen filtrada. El término **loss pred** permanece sin ponderación, ya que su función es mantener la coherencia entre la predicción del modelo y la imagen filtrada, y no se desea reducir su importancia relativa.

### 3.3.3. Experimento III

Tras la incorporación del coeficiente de ponderación $\gamma$ en la función de costo (ver Ecuación 3.5), se realizó un tercer experimento con el objetivo de analizar cómo la interacción entre los tres parámetros del filtro (tasa de aprendizaje, número de iteraciones y el coeficiente de ponderación) afecta el resultado del proceso de filtrado.

A partir de los hallazgos obtenidos en los Experimentos I y II, se seleccionaron los siguientes rangos de valores para cada uno de los parámetros:

- **Tasa de aprendizaje ($lr$):** {10.0, 15.0, 20.0, 25.0, 30.0}
- **Número de iteraciones ($n$):** {100, 125, 150, 175, 200}
- **Coeficiente de ponderación ($\gamma$):** {0.1, 0.2, 0.3, 0.4, 0.5}

Por un lado, se evitaron tasas de aprendizaje demasiado bajas, que ya habían mostrado tener un efecto casi neutro en las imágenes filtradas, así como valores demasiado altos que producían comportamientos erráticos. Por otro lado, se exploró una escala amplia de valores para $\gamma$, con el fin de observar de manera detallada su impacto en la pérdida total.

Dado que el número total de combinaciones posibles asciende a 125 (5 tasas de aprendizaje $\times$ 5 números de iteraciones $\times$ 5 valores de $\gamma$), se optó por seleccionar aleatoriamente 50 configuraciones distintas utilizando la librería `random` de Python. Cada una de estas combinaciones fue aplicada a las cuatro imágenes de prueba, generando un total de 200 imágenes filtradas. Es importante mencionar que se utilizaron las mismas 50 combinaciones para cada una de las imágenes de prueba. En la Figura 3.9 se muestran algunos ejemplos representativos de los resultados obtenidos.

Los resultados muestran que, para $\gamma = 0.5$, el comportamiento del filtro es análogo al observado en el Experimento II: las imágenes filtradas conservan gran parte de la información original y únicamente presentan ligeras distorsiones visuales (como manchas de color y pixeleo parcial), independientemente del valor de la tasa de aprendizaje y el número de iteraciones. Esto confirma que valores altos de $\gamma$ refuerzan el término de regularización por intensidad promedio, limitando el efecto de los otros términos de la función de costo.

A medida que se reduce el valor de $\gamma$, la influencia relativa de los tres parámetros se vuelve más notoria. En estos casos, el resultado del filtrado depende fuertemente de la combinación específica de $lr$, $n$ y $\gamma$. A continuación, se describen las observaciones más relevantes organizadas por grupos de tasas de aprendizaje:

- **$lr = 10.0$.**

  Para $n = 100$, las imágenes filtradas resultantes conservan la silueta del animal con claridad, aunque el fondo presenta distintos niveles de atenuación según el valor de $\gamma$. A medida que aumenta el número de iteraciones (125, 150, 175), el fondo tiende a desaparecer casi por completo, y la figura del animal comienza a formarse por trazos de colores parcialmente estilizados (principalmente tonos azulados y morados) que, permiten en muchos casos identificar claramente al objeto de interés. Por ejemplo, con la configuración $lr = 10.0$, $n = 175$, $\gamma = 0.4$, la silueta de las clases *elephant* y *mexican hairless* se conserva de forma clara, e incluso se destacan rasgos distintivos como ojos, orejas, hocico o trompa (ver Figura 3.9, columna 3, filas 1 y 3). Para $n = 200$, se observa nuevamente el comportamiento errático descrito en el

Experimento I: las imágenes se reducen a trazos dispersos, con mínima información visual interpretable.

- ***lr* = 15.0.**

  Para $n = 100$ y 125, el fondo se atenúa de forma considerable en la mayoría de los casos; sin embargo, la silueta del animal tiende a perder definición debido a la aparición de distorsiones visuales más agresivas, como manchas de color y pixeleo. Aun así, bajo ciertas combinaciones específicas de parámetros, el filtro logra conservar estructuras relevantes. Por ejemplo, con $lr = 15.0$, $n = 125$ y $\gamma = 0.4$, es posible distinguir visualmente la silueta del animal en las cuatro imágenes de prueba (ver Figura 3.9, columna 4). Además, en las imágenes correspondientes a las clases *elephant* y *mexican hairless*, se destacan regiones faciales como los ojos, la trompa, el hocico o las orejas, lo que sugiere que el filtro logra identificar áreas que el modelo podría considerar relevantes para la clasificación. En contraste, para las clases *zebra* y *hamster*, algunas regiones del fondo permanecen visibles, lo cual indica que, en ciertos casos, el filtro no consigue eliminar completamente la información no relevante. A partir de $n = 150$, la figura se compone de trazos más abstractos que, aunque resaltan ciertas regiones, dificultan una interpretación visual precisa.

- ***lr* >15.0.**

  Independientemente del valor de $n$ o $\gamma$, la mayoría de los resultados presentan el mismo comportamiento errático documentado previamente: las imágenes se reducen a trazos de colores dispersos que, en general, no permiten reconocer fácilmente elementos de la clase objetivo. Una excepción notable se da en las imágenes de la clase *mexican hairless*, donde el rostro del animal es parcialmente reconocible en varios casos, aunque la claridad de sus rasgos depende fuertemente de la combinación de parámetros utilizada (ver Figura 3.9, columna 5, fila 3). En contraste, para el resto de las clases, el resultado suele estar dominado por artefactos visuales sin relación aparente con el objeto de interés.

Estos resultados muestran que, si bien la inclusión del coeficiente de ponderación $\gamma$ ofrece un mayor control sobre el comportamiento del filtro, en la práctica no se logró un funcionamiento consistente para todas las clases analizadas. Aunque para las clases *elephant* y *mexican hairless* se observaron en algunos casos indicios de detección o resaltado de patrones relevantes, para las demás clases el filtro tendió a no eliminar completamente el fondo o a mostrar el comportamiento errático descrito previamente, en el que las imágenes se reducen a trazos de colores dispersos que ocupan gran parte del área y dificultan la identificación de elementos de la clase objetivo. Este resultado contradice la expectativa inicial de que la imagen filtrada conservara únicamente las regiones correspondientes a la clase objetivo, lo que sugiere que el fondo podría estar contribuyendo parcialmente a la clasificación o que el proceso de filtrado introduce un nivel de ruido significativo.

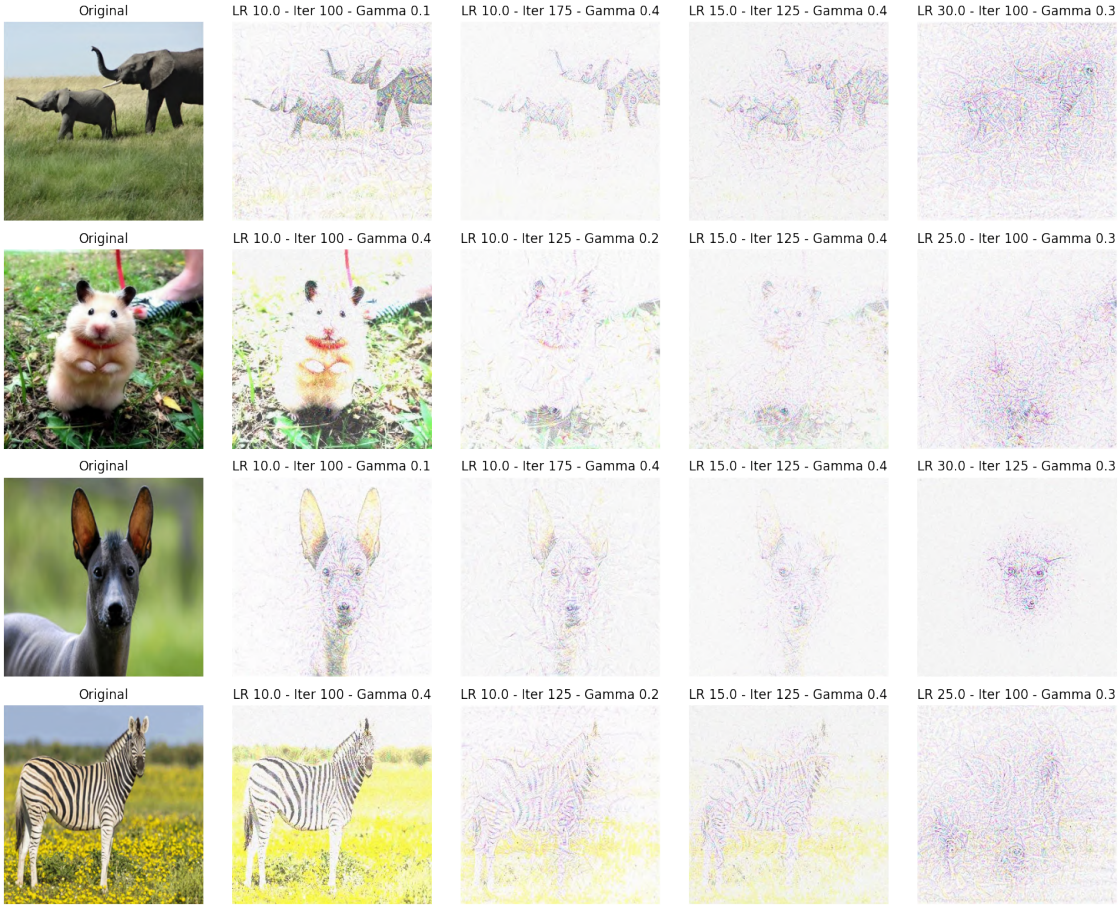


Figura 3.9: Ejemplos representativos del Experimento III, en el que se incorpora el coeficiente de ponderación $\gamma$ a la función de costo. Las combinaciones de parámetros utilizadas en cada caso se indican sobre las subimágenes correspondientes.

## Visualización de gradientes

Con base en los resultados obtenidos en el Experimento III, se decidió visualizar los gradientes de las imágenes de prueba como una alternativa para comprender mejor la relación entre la entrada y la salida del modelo, y así identificar posibles causas de los comportamientos observados. Para cada imagen se calculó el gradiente de la predicción de la clase objetivo con respecto a los píxeles de la imagen original, empleando la API `GradientTape` de `TensorFlow`, y se generó un mapa de calor que resalta las regiones de la imagen con mayor influencia en la salida del modelo (ver Figura 3.10).

La visualización de gradientes reveló una relación directa entre las regiones con gradientes de alta magnitud y la respuesta del modelo, lo que sugiere que dichas zonas contienen características relevantes para la clasificación. Sin embargo, también se observan múltiples zonas dispersas con valores altos o intermedios fuera del contorno del objeto de interés, lo que indica un posible efecto de *ruido* en los gradientes. Este fenómeno podría estar interfiriendo con el funcionamiento del filtro, ya que el procedimiento de optimización modifica los píxeles proporcionalmente a la magnitud del gradiente. Así, la existencia de gradientes elevados en áreas irrelevantes provoca que el filtro altere el fondo o introduzca trazos de color dispersos, en lugar de concentrar las modificaciones en las

regiones más representativas del objeto.

En términos técnicos, esto refleja una *baja relación señal/ruido* en los mapas de calor: la señal (gradientes útiles en el objeto) se encuentra mezclada con un nivel significativo de gradientes no informativos en el resto de la imagen. Este comportamiento no solo dificulta el aislamiento de las zonas realmente discriminativas, sino que también podría explicar las inconsistencias y el comportamiento errático descritos en el Experimento III.

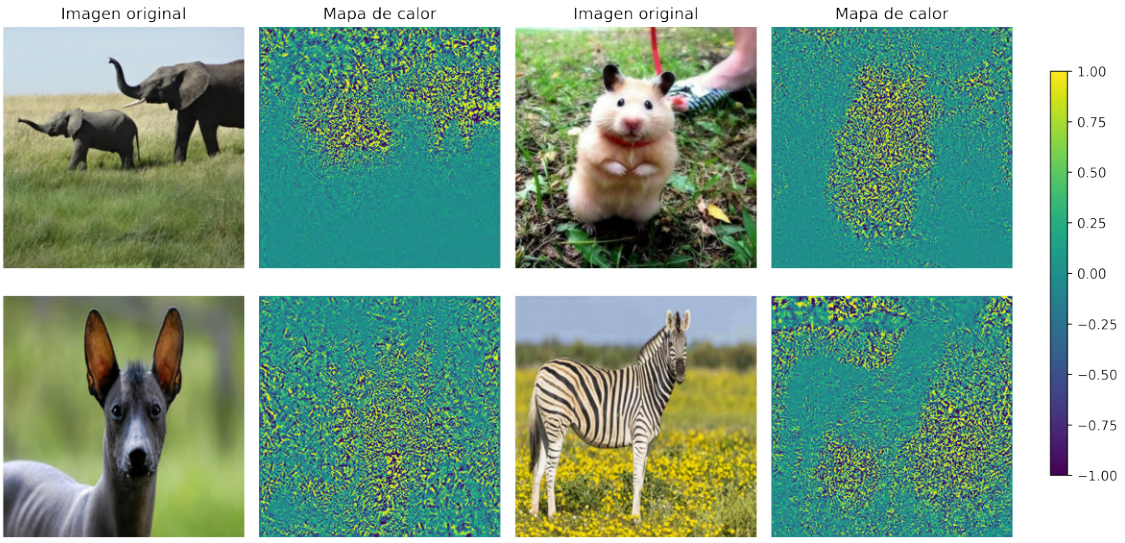


Figura 3.10: Visualización de gradientes para las imágenes de prueba. En la primera y tercera columna se muestran las imágenes originales, mientras que en la segunda y cuarta columna se presentan sus correspondientes mapas de calor, calculados a partir de los gradientes de la salida del modelo con respecto a la imagen de entrada. Los mapas fueron normalizados en el rango de -1 a 1 y se empleó el cmap *viridis* para su visualización. La barra de color indica la magnitud relativa del gradiente, donde los valores más altos (tonos amarillos) representan una mayor influencia sobre la salida del modelo, y los valores más bajos (tonos oscuros) corresponden a regiones con menor contribución.

Estas observaciones motivaron la exploración de un enfoque alternativo: medir el cambio en la predicción del modelo al ocultar regiones específicas de la imagen, con el objetivo de cuantificar su contribución a la clasificación final. Esta idea, que posteriormente se identificó como el método de *sensibilidad de oclusión*, se presenta en la siguiente sección, donde además se propone una versión modificada de dicho método.

## 3.4. Sensibilidad de Oclusión

Como se describió en la Sección 2.2.2, la sensibilidad de oclusión (o *occlusion maps*) es una técnica de interpretabilidad que analiza cómo varía la respuesta de un modelo de visión por computadora cuando se ocultan pequeñas regiones de la imagen de entrada. El principio fundamental consiste en modificar de forma controlada la entrada, bloqueando o reemplazando secciones específicas de píxeles, y observar el cambio en la probabilidad asignada a la clase objetivo. Si la eliminación de cierta región provoca una disminución significativa en dicha probabilidad, se infiere que esa zona

contiene información relevante para la decisión del modelo. El resultado de este procedimiento es un *mapa de oclusión*, que indica qué partes de la imagen tienen un mayor impacto en la clasificación.

Este método fue propuesto originalmente por Zeiler y Fergus [14] como una forma intuitiva y directa de visualizar las características discriminativas aprendidas por redes neuronales convolucionales. A diferencia de enfoques basados en gradientes, como Grad-CAM [10], la sensibilidad de oclusión no requiere calcular derivadas de la salida respecto a los píxeles de entrada, lo que la hace menos susceptible a problemas de ruido en los gradientes, como los observados en la sección anterior, donde los mapas de calor presentaron un nivel de ruido que dificultaba la identificación fiable de las zonas discriminativas. Estas limitaciones motivaron la implementación de este método en el presente trabajo.

Si bien existen implementaciones de sensibilidad de oclusión en librerías como el submódulo `tf-explain` de TensorFlow, en nuestras pruebas esta opción no ofreció resultados satisfactorios para las imágenes y modelos utilizados, ya que los mapas generados presentaban una baja discriminación espacial y un contraste insuficiente entre regiones relevantes e irrelevantes, dificultando su interpretación. Por ello, se optó por desarrollar una implementación propia en Python, basada en el principio fundamental del método (ocluir regiones y medir la variación en la probabilidad de la clase objetivo), pero con ajustes específicos de diseño. En particular, nuestra implementación asigna valores de importancia al mapa de oclusión considerando únicamente las diferencias positivas entre la predicción de la imagen original y la imagen ocluida, garantizando así que se contabilicen solo las regiones que realmente aportan a la clasificación.

Adicionalmente, tras generar el mapa de oclusión, se calcula el primer cuartil de su distribución de intensidades y se conservan únicamente aquellos valores que se encuentran por encima de este umbral, de forma que se refuercen las contribuciones más relevantes para la predicción. Finalmente, el mapa de oclusión se normaliza en el rango $[0, 1]$ para facilitar su interpretación visual.

### 3.4.1. Experimento I

Con el objetivo de evaluar el desempeño de la implementación propuesta del método de sensibilidad de oclusión, se llevó a cabo un primer experimento empleando el mismo conjunto de datos y modelo utilizados en los experimentos previos descritos en la Sección 3.3, es decir, la base de datos ImageNet y el modelo Xception. Para este análisis, el conjunto de imágenes de prueba se amplió con dos nuevas muestras correspondientes a las clases *mantis* y *three-toed sloth*, sumándose a las cuatro empleadas anteriormente (*elephant*, *hamster*, *mexican hairless* y *zebra*), con el fin de contar con una mayor diversidad visual y de patrones a identificar.

En este experimento, la única variable controlada fue el parámetro *grid size*, que en nuestra implementación determina el tamaño del parche cuadrado utilizado para ocluir distintas regiones de la imagen. Se consideraron cuatro configuraciones de este parámetro: 5, 10, 15 y 20, donde cada valor corresponde a un parche de tamaño $n \times n$ píxeles. Para cada imagen de prueba se generaron, por lo tanto, cuatro mapas de oclusión, lo que permitió examinar cómo el tamaño del parche influye en la localización y definición de las regiones relevantes para la clasificación. En la Figura 3.11 se muestran los mapas de oclusión generados para cada una de las imágenes de prueba.

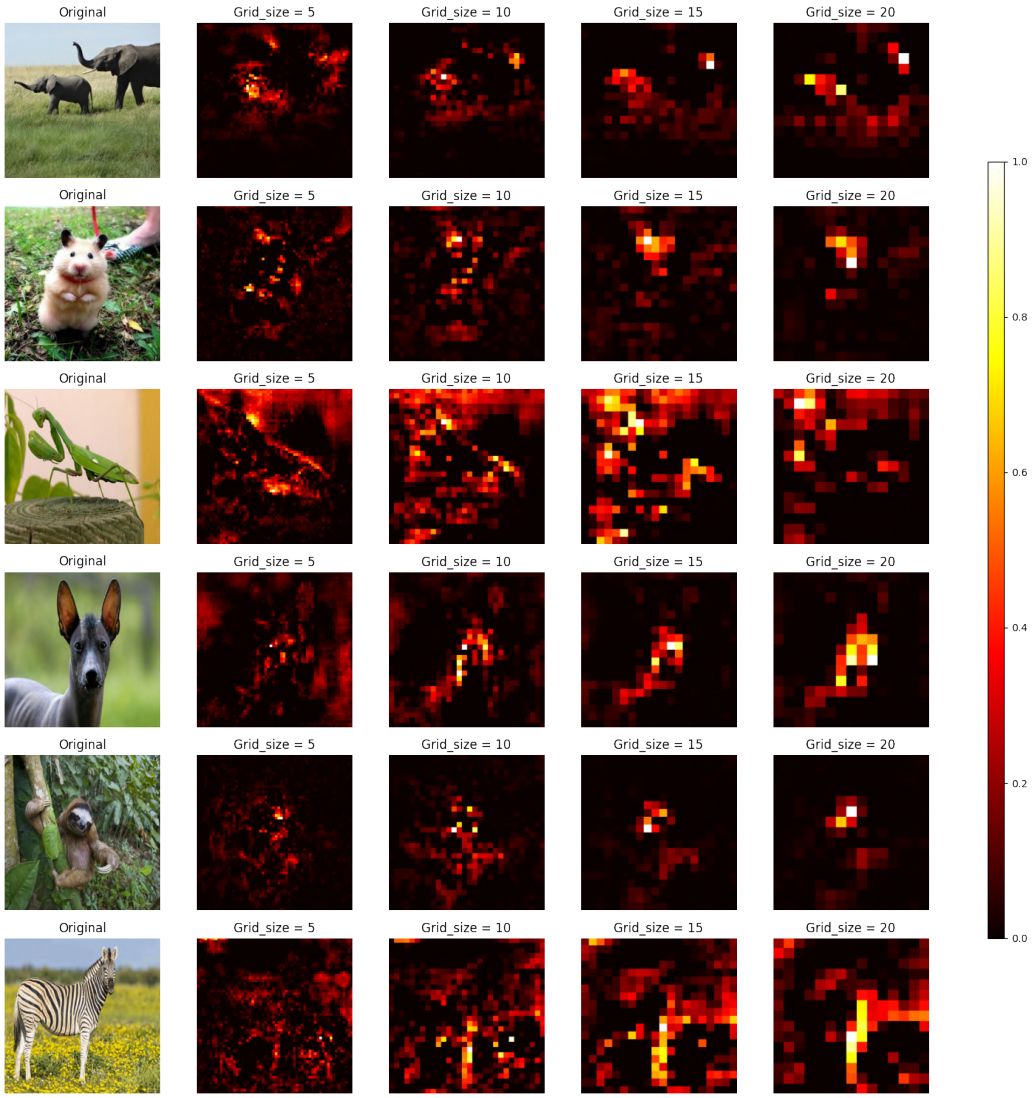


Figura 3.11: Mapas de oclusión generados en el Experimento I para las seis imágenes de prueba (*elephant*, *hamster*, *mexican hairless*, *zebra*, *mantis* y *three-toed sloth*). Para cada imagen (primera columna) se muestran los resultados obtenidos con cuatro valores del parámetro *grid size* (5, 10, 15 y 20 píxeles), donde cada valor corresponde a un parche cuadrado de $n \times n$ píxeles utilizado para ocluir distintas regiones de la imagen.

Los resultados obtenidos en este primer experimento muestran una clara dependencia del mapa de oclusión respecto al tamaño del parche definido por el parámetro *grid size*. Para valores pequeños de este parámetro (*grid size* = 5 y 10), los mapas de calor presentan un mayor nivel de detalle y una distribución de intensidades más uniforme, lo que facilita la identificación de patrones finos asociados a la clase objetivo. Sin embargo, en estas configuraciones es más notoria la presencia de ruido, evidenciada por la aparición de regiones con intensidades intermedias o altas en zonas que no corresponden al objeto de interés. Este fenómeno sugiere que el modelo podría estar utilizando información del fondo de la imagen para la clasificación. Un ejemplo representativo se observa en las clases *elephant* y *mexican hairless* con *grid size* = 10, donde el mapa resalta patrones anatómicos

específicos como la cabeza, trompa u orejas, lo que incrementa la interpretabilidad del resultado (ver Figura 3.11, columna 3, filas 1 y 4).

En contraste, para valores grandes del parámetro (*grid size* = 15 y 20), los mapas de oclusión tienden a ser más discriminativos respecto a la clase objetivo, ya que las zonas de mayor activación se concentran con mayor claridad en las regiones correspondientes al objeto de interés. No obstante, esta ganancia en discriminación se logra a costa de una pérdida significativa de detalle y suavidad, observándose saltos bruscos de intensidad entre regiones adyacentes y una menor capacidad para localizar áreas más específicas dentro de la imagen.

## Superposición de Oclusión

A partir de las observaciones realizadas en el Experimento I, se identificaron dos problemas principales en los mapas de oclusión generados: (1) la presencia de ruido en regiones ajenas al objeto de interés, lo que sugiere que el modelo podría estar utilizando información del fondo para apoyar su decisión, y (2) la ausencia de suavidad manifestada en transiciones abruptas de intensidad entre regiones adyacentes, que dificulta una delimitación precisa y continua de las regiones relevantes.

Para abordar estas limitaciones, se propuso una modificación en la implementación del método, consistente en utilizar un paso de oclusión (*stride*) menor que el tamaño del parche (*grid size*). De este modo, se produce un barrido más denso y uniforme sobre la imagen, generando un mayor número de regiones evaluadas y, por ende, un mapa de calor potencialmente más suave y detallado.

La reducción del *stride* provoca la superposición de múltiples parches sobre una misma región de la imagen. Para asignar el valor de importancia asignado a cada píxel, se optó por calcular el promedio de todos los valores que recibe a partir de las distintas oclusiones que lo cubren, que como se había descrito anteriormente cada uno de esos valores se calcula considerando únicamente las diferencias positivas entre la predicción para la imagen original y la imagen ocluida. Este criterio fue elegido por ser una aproximación intuitiva que pondera de manera equilibrada el efecto acumulado de las distintas superposiciones. No obstante, cabe señalar que podrían explorarse métricas alternativas para la asignación de estos valores, tales como el máximo o la mediana, dependiendo del grado de suavidad o discriminación que se desee enfatizar.

Este proceso se ilustra en la Figura 3.12, donde se muestra gráficamente cómo un mismo píxel puede ser cubierto por múltiples parches de oclusión, recibiendo un valor de importancia distinto de cada uno de ellos y cómo se calcula su valor final mediante el promedio de estos.

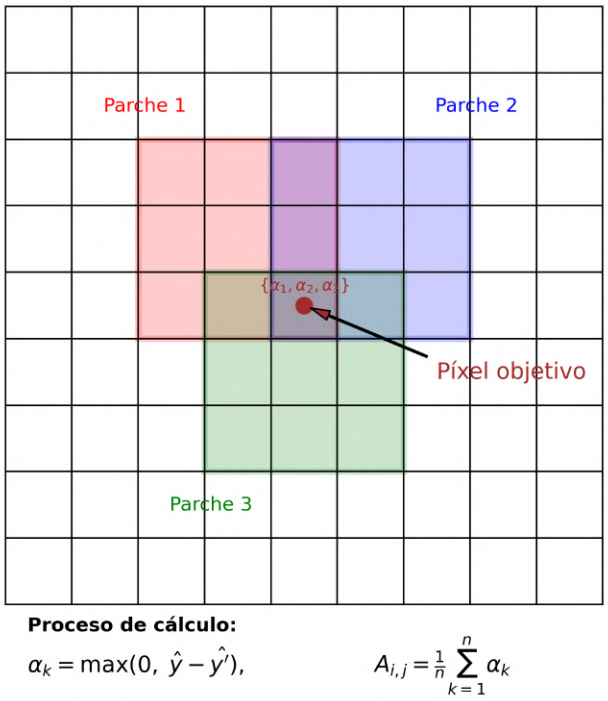


Figura 3.12: Diagrama ilustrativo del proceso de superposición de oclusión. En este ejemplo, tres parches diferentes cubren un mismo píxel, asignándole los valores de importancia $\alpha_1$, $\alpha_2$ y $\alpha_3$, calculados según $\alpha_k = \text{máx}(0, \hat{y} - \hat{y}')$, donde $\hat{y}$ y $\hat{y}'$ corresponden a las predicciones del modelo para la imagen original y la imagen ocluida, respectivamente. El valor de importancia final asignado a ese píxel se obtiene como el promedio de estos valores, de acuerdo con la expresión $A_{i,j} = \frac{1}{n} \sum_{k=1}^{n} \alpha_k$.

### 3.4.2. Experimento II

Con el propósito de evaluar el impacto de la estrategia de superposición de oclusiones en la calidad y la interpretabilidad de los mapas de calor generados por nuestra implementación de sensibilidad de oclusión, se diseñó un segundo experimento en el que se incorporó el uso de un *stride* menor que el tamaño del parche (*grid size*). Para cada una de las seis imágenes de prueba, se generaron un total de seis mapas de oclusión, combinando dos tamaños de parche (10 y 15 píxeles) con tres valores de *stride* (5, 2 y 1 píxel).

Esta configuración experimental permite analizar cómo la densidad del barrido (determinada por el *stride*) y el tamaño del parche interactúan para modificar la suavidad, el nivel de detalle y la discriminación de las regiones relevantes en la clasificación. En la Figura 3.13 se muestran ejemplos representativos de los mapas de oclusión obtenidos para cada imagen de prueba, seleccionados de entre las seis configuraciones generadas en este experimento. Cabe destacar que, a partir de los resultados del Experimento I, se decidió trabajar únicamente con valores de *grid size* de 10 y 15 píxeles, ya que mostraron un mejor equilibrio entre detalle y suavidad.

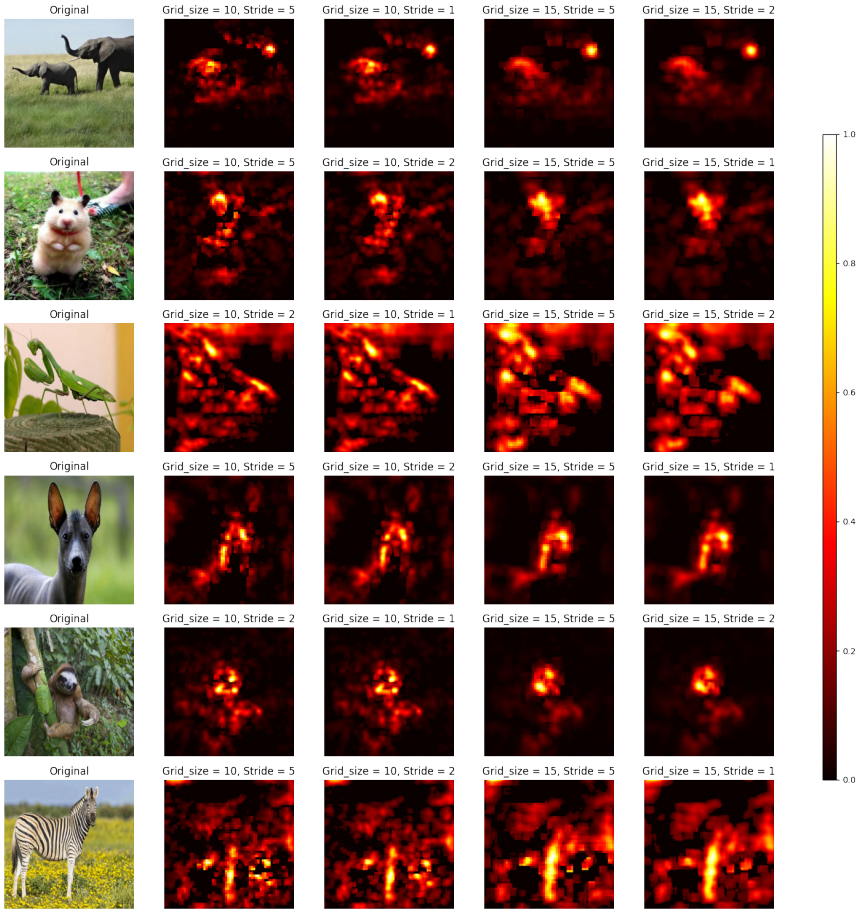


Figura 3.13: Mapas de oclusión generados en el Experimento II para las seis imágenes de prueba (*elephant*, *hamster*, *mantis*, *mexican hairless*, *three-toed sloth* y *zebra*). Para cada imagen (primera columna) se muestran ejemplos representativos obtenidos con las combinaciones de *grid size* = {10, 15} píxeles y *stride* = {5, 2, 1} píxeles. La primera columna presenta la imagen original, mientras que las siguientes muestran los mapas de oclusión normalizados, en los que las regiones más relevantes para la predicción del modelo se indican en colores cálidos.

En comparación con los resultados obtenidos en el Experimento I, los mapas generados en este segundo experimento presentan una notable mejoría en la definición de detalles y en la suavidad de las transiciones de intensidad. La estrategia de superposición de oclusiones produce mapas con una distribución de intensidades más uniforme, lo que facilita la identificación de patrones finos asociados a la clase objetivo. En muchos casos, elementos anatómicos de los animales tales como contornos de cabeza, cuerpo, patas, orejas o trompa se distinguen con mayor claridad y aparecen representados con valores de mayor intensidad en la escala del mapa de calor.

No obstante, la presencia de ruido sigue siendo un problema, principalmente en clases como *mantis* y *zebra*, donde amplias zonas del fondo aparecen resaltadas, dificultando la discriminación de la clase objetivo (ver Figura 3.13, filas 3 y 6). Por el contrario, en clases como *elephant* y *mexican hairless*, los mapas presentan un mayor grado de detalle y discriminación, llegando incluso a delinear de forma clara la silueta del animal (ver Figura 3.13, filas 1 y 4).

En cuanto a los parámetros evaluados, se observó que reducir el *stride* incrementa significativamente el costo computacional, debido al incremento en el número de iteraciones requeridas por el algoritmo para barrer toda la imagen, especialmente en el caso extremo de *stride* = 1. Sin embargo, no se observaron mejoras sustanciales en la calidad de los mapas de calor entre los tres valores de *stride* evaluados en este experimento, lo que hace menos conveniente el uso de valores muy bajos debido al aumento en el tiempo de procesamiento que generan.

Por otro lado, el tamaño del parche (*grid size*) continúa siendo un factor relevante. Al igual que en el experimento anterior, un *grid size* igual a 10 ofrece una mayor definición de detalles pero introduce más ruido, mientras que un *grid size* igual a 15 reduce el ruido a costa de una ligera pérdida de detalle. Sin embargo, con la evidencia disponible no es posible establecer de forma concluyente cuál de estos valores produce resultados más óptimos de manera general.

Como parte del análisis complementario del Experimento II, se evaluó una métrica alternativa para el cálculo del valor de importancia en las superposiciones de oclusión. En lugar de utilizar el promedio de las contribuciones recibidas por cada píxel, se aplicó una operación de máximo (*max pooling*), de modo que únicamente se considerara la mayor magnitud entre las superposiciones que lo cubren. El objetivo de esta modificación era reducir el ruido y resaltar con mayor claridad las regiones relevantes, ya que esta métrica enfatiza las áreas donde el impacto de la oclusión es más pronunciado, ignorando contribuciones menores o menos significativas.

Sin embargo, los resultados obtenidos fueron muy similares a los del enfoque original empleando el promedio. La cantidad de ruido presente en los mapas de calor se mantuvo prácticamente inalterada, y la única diferencia perceptible fue un incremento en la intensidad de las regiones resaltadas incluyendo tanto aquellas asociadas a la clase objetivo como las ajenas a ella. Esto sugiere que, si bien el *max pooling* puede intensificar la visibilidad de ciertas estructuras anatómicas relevantes, también amplifica de igual forma las áreas no relacionadas con la clase objetivo, lo que en conjunto produce mapas de calor visualmente muy similares a los obtenidos empleando el promedio.

### 3.4.3. Experimento III

A pesar de las mejoras obtenidas en el Experimento II, la presencia de ruido en los mapas de calor sigue siendo un problema recurrente, dificultando en algunos casos la correcta identificación de las regiones relevantes asociadas a la clase objetivo. Para atacar este problema, se procedió a analizar la distribución de intensidades de los mapas de oclusión generados previamente. Este análisis reveló que la gran mayoría de los valores situados por debajo del percentil 95 correspondían a zonas del fondo de la imagen, es decir, a regiones que no aportaban información relevante para la predicción del modelo.

Con base en esta observación, se diseñó un último experimento que incorpora dos modificaciones principales: (1) la filtración de intensidades conservando únicamente los valores por encima del percentil 95, y (2) la aplicación de un filtro gaussiano como técnica de suavizado. Esta última operación contribuye a reducir variaciones abruptas y ruido local, preservando las estructuras y bordes más relevantes del mapa de calor.

Para este experimento, se eligieron los valores de *stride* = 5 y *grid size* = $\{10, 15\}$ píxeles, basados en los resultados del Experimento II, donde se observó que valores más bajos de *stride* no aportaban mejoras significativas en la calidad de los mapas pero, sí incrementaban considerablemente el costo computacional. Además, dado que el filtrado por percentil reduce considerablemente la cobertura

espacial del mapa, dificultando su interpretación cuando se presenta de forma aislada, se optó por visualizar los resultados como *superposiciones* sobre la imagen original.

En la Figura 3.14 se muestran los mapas de oclusión superpuestos generados para cada imagen de prueba después de aplicar este nuevo procedimiento. Los resultados permiten comparar visualmente el impacto de las técnicas de filtrado y suavizado en la reducción de ruido y en la claridad de las regiones relevantes.

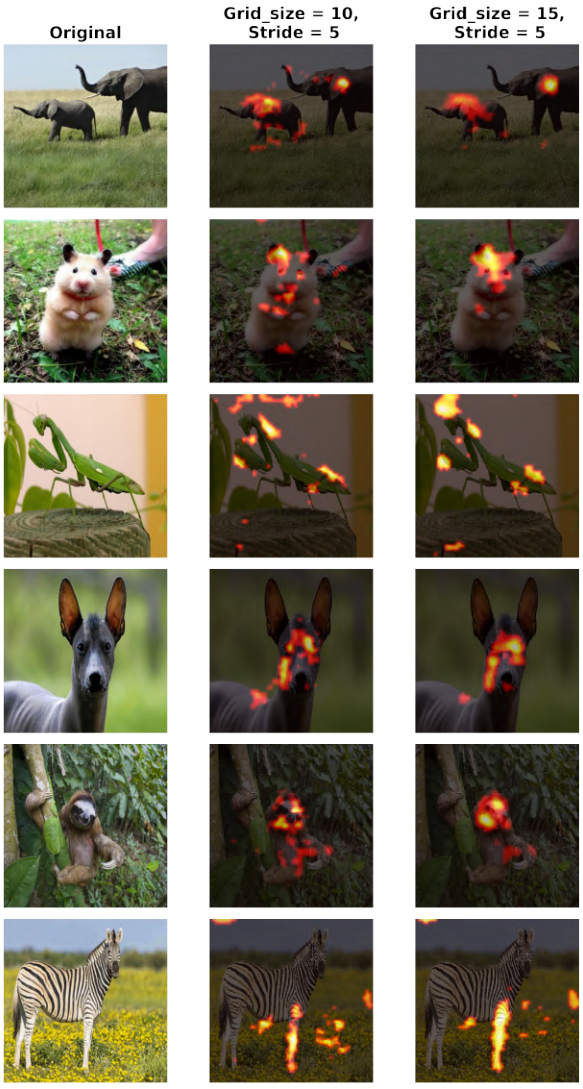


Figura 3.14: Mapas de oclusión generados en el Experimento III aplicando filtrado por percentil 95 y suavizado gaussiano. Se emplearon los parámetros de *stride* = 5 y *grid size* = {10, 15} píxeles. Los mapas se muestran superpuestos a la imagen original.

En comparación con los resultados obtenidos en el Experimento II, la reducción de ruido lograda en este tercer experimento es sustancial, llegando en muchos casos a prácticamente erradicarlo. El filtrado por percentil 95 provoca que solo zonas muy puntuales de la imagen sean resaltadas en el mapa de oclusión y, al superponer estos mapas con las imágenes originales (Figura 3.14), dichas zonas caen en su mayoría sobre regiones pertenecientes a la clase objetivo. Un ejemplo claro se

observa en las clases *mantis* y *zebra* (filas 3 y 6 de la figura), donde en el Experimento II el ruido cubría amplias zonas del fondo, mientras que en este experimento se reduce casi por completo, permaneciendo únicamente algunas manchas dispersas en el fondo, pero resaltando de manera clara regiones de interés de la clase objetivo.

Otro aspecto importante es la diferencia observada entre los mapas generados con *grid size* = 10 y aquellos obtenidos con *grid size* = 15. En general, en los mapas con *grid size* = 10, las regiones resaltadas tienden a estar más dispersas alrededor de las áreas de interés, cubriendo distintas partes del objeto con pequeñas manchas de calor. En contraste, los mapas con *grid size* = 15 se concentran en zonas específicas de la clase objetivo, resaltando con mayor intensidad únicamente aquellas regiones que parecen ser más determinantes para la clasificación. Por ejemplo, en el caso de la clase *three-toed sloth* (fila 5), el *grid size* = 10 produce múltiples manchas distribuidas en casi todo el cuerpo del animal, mientras que con *grid size* = 15 el mapa se concentra principalmente en la cara, con apenas algunas marcas adicionales en las garras. Una tendencia similar puede observarse en la clase *hamster* (fila 2), donde el *grid size* = 10 resalta de manera dispersa cabeza y extremidades, mientras que el *grid size* = 15 concentra la activación casi exclusivamente en la cabeza. Esto sugiere que valores mayores de *grid size* promueven una discriminación más localizada de las regiones relevantes.

Finalmente, aunque los mapas de calor generados en este experimento muestran menos detalles finos y contornos sutiles en comparación con los obtenidos en experimentos anteriores, presentan un mayor grado de discriminación de las regiones relevantes. Gracias a la superposición con la imagen original, es posible visualizar de manera más clara estructuras anatómicas asociadas a la clase objetivo. De esta manera, el método no solo mejora la interpretabilidad del modelo, sino que también abre la posibilidad de emplearse como una aproximación preliminar a la localización de objetos en un marco semi-supervisado.

## 3.5. Evaluación comparativa con técnicas de interpretabilidad

Una vez completado el desarrollo experimental del filtro basado en sensibilidad de oclusión, y tras observar que en el Experimento III se alcanzó una notable reducción de ruido y una mayor discriminación de las regiones relevantes, el siguiente paso consiste en evaluar su desempeño en relación con técnicas de interpretabilidad ya establecidas. En particular, se plantea comparar la versión modificada de sensibilidad de oclusión propuesta en esta tesis con la versión original de sensibilidad de oclusión y los mapas de activación ponderada por gradiente (Grad-CAM). El objetivo de esta comparación es identificar las fortalezas y limitaciones del método propuesto, así como explorar sus posibles áreas de mejora en el marco de la interpretabilidad de modelos de aprendizaje profundo.

### 3.5.1. Versión modificada de sensibilidad de oclusión

La versión implementada en esta tesis se basa en el método original de sensibilidad de oclusión, pero incorpora una serie de mejoras y adaptaciones diseñadas para aumentar la capacidad de discriminación de las regiones relevantes y reducir el ruido presente en los mapas generados. En particular, se introdujeron las siguientes modificaciones:

1. **Asignación de importancia:** los valores de importancia se calculan considerando únicamente

las diferencias positivas entre la predicción de la imagen original y la imagen ocluida, evitando que las diferencias negativas introduzcan ruido en el mapa.

2. **Superposición de oclusión:** se emplea un paso de oclusión (*stride*) menor al tamaño del parche de oclusión (*grid size*), con el fin de hacer un barrido más denso y uniforme sobre la imagen, generando un mayor número de regiones evaluadas.

3. **Promediado de las superposiciones de oclusión:** la reducción del *stride* genera múltiples parches superpuestos sobre una misma región de la imagen. El valor de importancia de cada píxel se define como el promedio de todas las contribuciones recibidas, lo que atenúa la variabilidad local y resalta de forma más consistente las zonas relevantes.

4. **Filtrado por percentil 95:** se descartan los valores de intensidad por debajo del percentil 95, preservando únicamente las regiones más relevantes según la distribución de intensidades.

5. **Suavizado gaussiano:** se aplica un filtro gaussiano para reducir variaciones abruptas y suavizar el mapa, manteniendo las estructuras más importantes.

Estas modificaciones permiten obtener mapas más claros, con una reducción significativa del ruido y una mejor correspondencia con las regiones anatómicas relevantes para la clasificación.

### 3.5.2. Análisis comparativo

Para evaluar la eficiencia, interpretabilidad y utilidad del filtro propuesto en comparación con los dos métodos mencionados anteriormente, se empleó la base de datos *ImageNet* y el modelo *Xception*, asegurando consistencia con los experimentos previos. El procedimiento consistió en aplicar los tres métodos de interpretabilidad sobre las mismas seis imágenes de prueba utilizadas anteriormente.

De esta manera, para cada imagen se generó un mapa de calor correspondiente a cada método, lo que permite una comparación visual y cualitativa entre ellos. Esta estrategia facilita identificar las diferencias en la definición de detalles, el nivel de ruido presente y la capacidad de cada enfoque para resaltar regiones relevantes asociadas a la clase objetivo.

Cabe señalar que, mientras en los experimentos anteriores se utilizó la paleta de colores *hot*, para la comparación final se empleó la paleta *jet*, con el fin de garantizar homogeneidad con Grad-CAM y mantener consistencia con la literatura, ya que este esquema de color es el más utilizado para representar los resultados de dicho método. Asimismo, es importante destacar que en la implementación propuesta de sensibilidad de oclusión se introdujo el procesamiento por lotes de imágenes ocluidas, a diferencia de la versión original en la que cada imagen se procesaba de manera individual. Esta mejora redujo significativamente el tiempo de cómputo, pasando de un promedio de 50 segundos por mapa de calor a solo 4.5 segundos al emplear un *grid size* de 15 píxeles. Esta reducción en el tiempo de ejecución fue fundamental para hacer factible el empleo de superposiciones de oclusión, cuyo costo computacional hubiese resultado prohibitivo de no aplicar esta optimización.

En cuanto a los parámetros empleados, tanto la versión original como la propuesta de sensibilidad de oclusión utilizaron un *grid size* igual a 15 píxeles, diferenciándose únicamente en el *stride* y en las modificaciones introducidas en este trabajo, descritas previamente en la Sección 3.5.1. Finalmente, los mapas de calor de Grad-CAM se generaron siguiendo la implementación propuesta en el

trabajo original titulado *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization* [10].

En la Figura 3.15 se muestran los mapas de calor generados por la versión original de sensibilidad de oclusión, Grad-CAM y los obtenidos con la versión modificada propuesta en esta tesis.

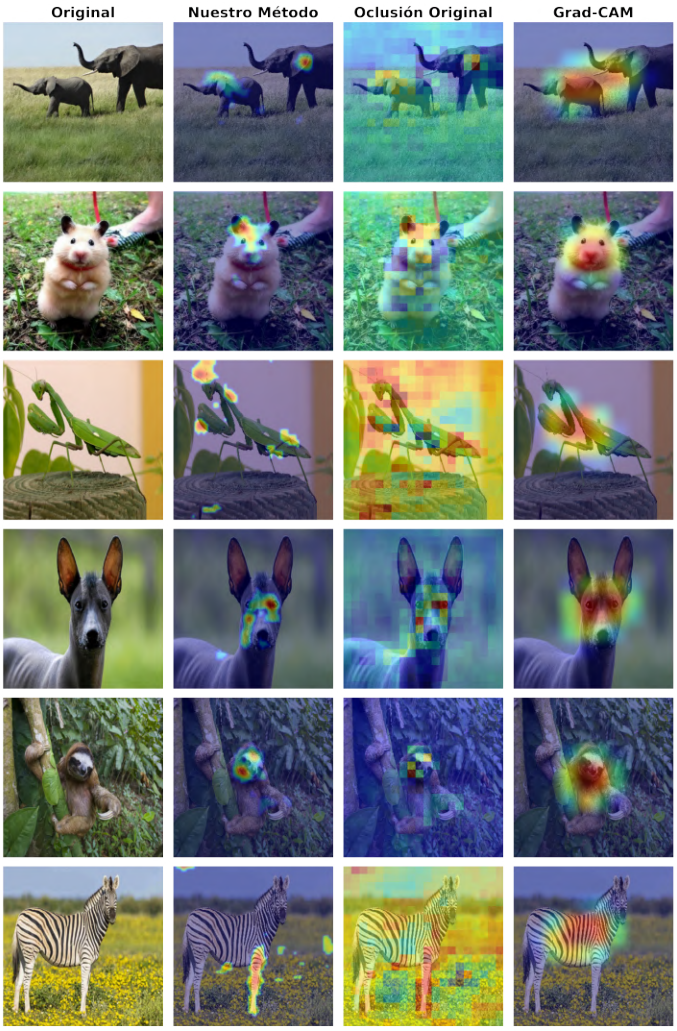


Figura 3.15: Comparación visual de los mapas de calor obtenidos con los tres métodos de interpretabilidad. La primera columna corresponde a la imagen original, la segunda muestra los resultados de la versión modificada de sensibilidad de oclusión propuesta en esta tesis, la tercera presenta los mapas generados con la versión original de sensibilidad de oclusión y la cuarta los mapas obtenidos mediante Grad-CAM. Se emplearon seis imágenes de prueba seleccionadas de la base de datos *ImageNet*, con el fin de evaluar la capacidad de cada método para resaltar de manera consistente las regiones relevantes asociadas a la clase objetivo.

A partir de la comparación visual mostrada en la Figura 3.15, es posible analizar los resultados de los tres métodos de acuerdo con dos criterios fundamentales señalados en la literatura para evaluar explicaciones visuales de modelos de visión por computadora: (a) ser *class-discriminative*, es decir, localizar adecuadamente la categoría de interés dentro de la imagen, y (b) mantener una resolución suficientemente alta para capturar detalles finos relevantes. En este sentido, la versión modificada

de sensibilidad de oclusión propuesta en este trabajo (segunda columna) produce activaciones que cumplen en gran medida con el primer criterio, ya que resaltan regiones anatómicas claramente asociadas con la clase objetivo (por ejemplo, la cabeza y orejas en *elephant* o *mexican hairless*). Además, gracias a la estrategia de superposición y filtrado, las activaciones son más puntuales y con menor ruido de fondo en comparación con la versión original, lo que facilita la interpretación de regiones específicas discriminativas para la clasificación.

Por otro lado, la versión original de sensibilidad de oclusión (tercera columna) también es *class-discriminative*, ya que mide directamente el impacto de ocultar regiones en la predicción del modelo, pero su resolución espacial es claramente inferior en comparación con la propuesta presentada en este trabajo. El uso de parches sin superposición genera mapas con bordes abruptos y activaciones dispersas, lo que introduce ruido y reduce la claridad interpretativa. En casos como *zebra* o *mantis*, este efecto se traduce en la presencia de múltiples regiones de fondo activadas que no aportan evidencia real para la clasificación. Grad-CAM (cuarta columna), en cambio, genera mapas de activación más suaves y de mayor soporte espacial que suelen identificar el contexto y la zona general del objeto, cumpliendo bien con el primer criterio (por ejemplo, al resaltar de forma consistente el torso de la *zebra* o el cuerpo de la *mantis*). Sin embargo, sus mapas tienden a ser de menor resolución, con una menor capacidad para delimitar con precisión pequeños componentes discriminativos. En este sentido, los resultados de la versión propuesta y de Grad-CAM pueden considerarse complementarios: mientras la primera aporta explicaciones más detalladas y puntuales, la segunda ofrece una visión más amplia y robusta del contexto discriminativo del modelo.

Es importante destacar que, en la mayoría de los casos, los mapas de calor obtenidos con el método propuesto y con Grad-CAM resaltan regiones muy similares, aunque el primero tiende a hacerlo con mayor nivel de detalle y sobre áreas más puntuales. Esta diferencia se aprecia con mayor claridad en las clases *mantis* y *zebra*, donde la versión original de sensibilidad de oclusión presenta un nivel elevado de ruido de fondo que, aunque se reduce en la versión propuesta, sigue afectando la calidad de los mapas obtenidos. Grad-CAM, por su parte, muestra una mayor resistencia a dicho ruido intrínseco de la imagen, generando mapas más amplios y consistentes, aunque menos precisos en la delimitación de zonas pequeñas y discriminativas.

Finalmente, al considerar el costo computacional, si bien Grad-CAM sigue siendo el método más eficiente al requerir únicamente una pasada por el modelo, la implementación propuesta en esta tesis reduce de manera significativa la desventaja computacional de la sensibilidad de oclusión tradicional. Gracias al procesamiento por lotes de regiones ocluidas, el tiempo promedio de generación por mapa de calor es ahora de 4.5 segundos para la versión original y 45.5 segundos para la versión propuesta en este trabajo, siendo comparables con los 1.5 segundos que tarda Grad-CAM bajo el mismo entorno de cómputo. De esta forma, el método propuesto no solo mejora la calidad visual y la resolución de los mapas, sino que también se convierte en una alternativa computacionalmente viable para tareas de interpretabilidad.

En síntesis, este capítulo permitió consolidar el desarrollo del filtro de extracción de características, mostrando su evolución, validación experimental y ventajas frente a métodos de referencia como Grad-CAM y la sensibilidad de oclusión. Con el filtro ya definido y optimizado, el siguiente capítulo se centra en su aplicación práctica en el ámbito médico, específicamente sobre la base de datos *ChestX-ray14*, con el fin de evaluar su utilidad en el análisis e interpretación de imágenes radiográficas.

# Capítulo 4

# Aplicación del Filtro de Extracción de Características en Imágenes Médicas

Este capítulo presenta la aplicación del filtro de extracción de características, basado en el método de sensibilidad de oclusión y desarrollado en el capítulo anterior, ahora orientado a la clasificación de imágenes médicas. Con este propósito, se describe de manera detallada el proceso seguido para su implementación en este nuevo contexto. En primer lugar, se llevó a cabo una revisión de las bases de datos públicas disponibles y se seleccionó aquella que mejor se ajustara a los objetivos planteados en este trabajo. Posteriormente, se realizó una revisión del estado del arte, identificando trabajos previos que abordaron con éxito la tarea de clasificación definida en la base de datos elegida. A partir de esta revisión, se buscó replicar alguna de las arquitecturas de redes neuronales convolucionales (RNC) reportadas en la literatura, con el objetivo de alcanzar un rendimiento comparable al de dichos estudios.

Durante este proceso surgieron diversos retos relacionados con el preprocesamiento de los datos, el entrenamiento y la evaluación de los modelos, lo cual motivó la exploración de múltiples enfoques hasta obtener un desempeño aceptable en la base de datos seleccionada. Finalmente, una vez consolidado un modelo de clasificación adecuado, se procedió a implementar el filtro propuesto, siguiendo una metodología similar a la aplicada en el caso de *ImageNet*, y se realizó una comparación directa con los mapas de activación generados mediante Grad-CAM.

## 4.1. ChestX-ray14

Tras una búsqueda exhaustiva de bases de datos de imágenes médicas de acceso público, se seleccionó **ChestX-ray14**, un conjunto de radiografías de tórax desarrollado por el Instituto Nacional de la Salud (NIH) [15]. Esta base de datos contiene un total de 112,120 radiografías frontales de tórax correspondientes a 30,805 pacientes únicos. Cada imagen está etiquetada con una o más de catorce patologías torácicas comunes, entre las que se incluyen atelectasia, consolidación, infiltración, neumotórax, edema, enfisema, fibrosis, derrame pleural, neumonía, engrosamiento pleural, cardiomegalia, nódulo, masa y hernia. Las etiquetas fueron obtenidas a partir de los informes radiológicos asociados a las imágenes mediante técnicas de minería de texto y procesamiento de lenguaje natural, por lo que una misma radiografía puede presentar múltiples enfermedades. Algunas

imágenes pertenecientes a ChesX-ray14 son mostradas en la Figura 4.1

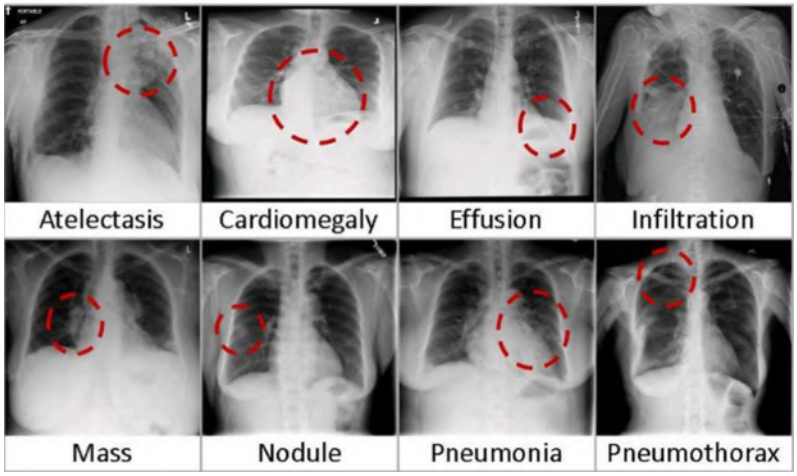


Figura 4.1: Ejemplos de radiografías de tórax con ocho patologías torácicas comunes, tomadas de la base de datos ChestX-ray14 (reproducido de [15]).

El problema de clasificación planteado por esta base de datos consiste en detectar y clasificar estas catorce patologías, lo que se traduce en una tarea de *clasificación multiclase y multietiqueta*. Este enfoque refleja de manera más realista la práctica clínica, donde un mismo paciente puede presentar varias condiciones de forma simultánea.

La elección de esta base de datos respondió a dos criterios principales. En primer lugar, su gran tamaño, a diferencia de la mayoría de los conjuntos de datos de imágenes médicas, **ChestX-ray14** dispone de más de cien mil radiografías, lo que la convierte en un recurso particularmente valioso para entrenar redes neuronales profundas con un menor riesgo de sobreajuste. En segundo lugar, y más relevante para los objetivos de esta tesis, aproximadamente mil imágenes de este conjunto cuentan con cuadros delimitadores (*bounding boxes*) anotados manualmente por expertos. Estas anotaciones permiten contar con una referencia objetiva para evaluar la capacidad del filtro propuesto en la localización de regiones anatómicas y en la discriminación de enfermedades. Sin estas referencias, sería necesario recurrir a la validación por parte de radiólogos, lo cual resultaría inviable dentro del alcance de este trabajo.

## 4.2. Antecedentes en la clasificación de ChestX-ray14

Antes de entrenar un modelo de red neuronal convolucional con un rendimiento competitivo en la base de datos *ChestX-ray14*, se llevó a cabo una revisión bibliográfica exhaustiva de trabajos previos que abordaron este mismo problema. Este paso resulta fundamental, ya que permite conocer los enfoques más exitosos, identificar las métricas de referencia reportadas en la literatura, y establecer una línea base contra la cual evaluar los modelos desarrollados en esta tesis.

Como resultado de esta revisión, se identificó el artículo titulado *Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists* [16]. El objetivo principal de este trabajo fue comparar el rendimiento de un algoritmo de aprendizaje profundo con el de radiólogos en ejercicio para la detección de patologías en radiografías de tórax. Para ello, los autores desarrollaron **CheXNeXt**, un ensamble de 10 redes neuronales, cada una

basada en la arquitectura **DenseNet-121**, compuesta por 121 capas distribuidas en 4 bloques densos. Como métrica principal de evaluación se utilizó el área bajo la curva ROC (AUROC), que mide la capacidad del modelo para discriminar entre clases. Los resultados mostraron que CheXNeXt alcanzó un rendimiento comparable al de radiólogos en 10 de las 14 patologías, superándolos en el caso de atelectasia, aunque fue superado por ellos en cardiomegalia, enfisema y hernia hiatal. Los resultados detallados se presentan en la Tabla 4.1. No obstante, la replicación de CheXNeXt no fue factible en este trabajo debido a las limitaciones computacionales, ya que se trata de un ensamble de redes profundas cuyo entrenamiento resulta altamente costoso en tiempo y recursos.

A partir de esta referencia, se identificó un trabajo previo de los mismos autores titulado *CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning* [17]. En este trabajo, el objetivo inicial fue desarrollar un algoritmo capaz de detectar neumonía a partir de radiografías de tórax frontal y comparar su rendimiento con el de radiólogos en ejercicio. Para ello, entrenaron una sola red **DenseNet-121** preentrenada en ImageNet, la cual posteriormente se extendió para abarcar la clasificación de las 14 patologías pulmonares presentes en la base de datos. Entre los resultados más relevantes, **CheXNet** superó el desempeño promedio de cuatro radiólogos en la detección de neumonía (F1 Score 0.435 vs 0.387), además de obtener valores de AUC superiores a los mejores resultados publicados hasta ese momento para cada una de las 14 patologías. Los valores completos pueden observarse en la Tabla 4.2. Este trabajo aportó una línea base ampliamente utilizada en la literatura, al proponer un modelo de referencia tanto en términos de desempeño como de metodología de entrenamiento para la clasificación de imágenes médicas a gran escala.

En conclusión, aunque CheXNeXt representa un estado del arte más avanzado, CheXNet constituyó la opción más adecuada para este trabajo debido a su menor complejidad computacional y a su probada eficacia en la clasificación de *ChestX-ray14*. Por esta razón, se tomó como base de referencia para los experimentos de clasificación desarrollados en esta tesis. En la siguiente sección se describe el proceso seguido para replicar este modelo, así como las dificultades encontradas y los resultados obtenidos.

| **Patología** | **Radiólogos (95 % CI)** | **Algoritmo (95 % CI)** | **Ventaja** |
|---|---|---|---|
| Atelectasia | 0.808 (0.777–0.838) | 0.862 (0.825–0.895) | Algoritmo |
| Cardiomegalia | 0.888 (0.863–0.910) | 0.831 (0.790–0.870) | Radiólogos |
| Consolidación | 0.841 (0.815–0.870) | 0.893 (0.859–0.924) | No diferencia |
| Edema | 0.910 (0.886–0.930) | 0.924 (0.886–0.955) | No diferencia |
| Derrame | 0.900 (0.876–0.921) | 0.901 (0.868–0.930) | No diferencia |
| Enfisema | 0.911 (0.866–0.947) | 0.704 (0.567–0.833) | Radiólogos |
| Fibrosis | 0.897 (0.840–0.936) | 0.806 (0.719–0.884) | No diferencia |
| Hernia | 0.985 (0.974–0.991) | 0.851 (0.785–0.909) | Radiólogos |
| Infiltración | 0.734 (0.688–0.779) | 0.721 (0.651–0.786) | No diferencia |
| Masa | 0.886 (0.856–0.913) | 0.909 (0.864–0.948) | No diferencia |
| Nódulo | 0.899 (0.869–0.924) | 0.894 (0.853–0.930) | No diferencia |
| Engrosamiento pleural | 0.779 (0.740–0.809) | 0.798 (0.744–0.849) | No diferencia |
| Neumonía | 0.823 (0.779–0.856) | 0.851 (0.781–0.911) | No diferencia |
| Neumotórax | 0.940 (0.912–0.962) | 0.944 (0.915–0.969) | No diferencia |

Tabla 4.1: Resultados de CheXNeXt comparados con radiólogos en ejercicio, medidos en términos de AUC para cada una de las 14 patologías presentes en la base de datos *ChestX-ray14* [16].

| Patología | Wang et al. (2017) | Yao et al. (2017) | CheXNet |
|---|---|---|---|
| Atelectasia | 0.716 | 0.772 | **0.8094** |
| Cardiomegalia | 0.807 | 0.904 | **0.9248** |
| Derrame | 0.784 | 0.859 | **0.8638** |
| Infiltración | 0.609 | 0.695 | **0.7345** |
| Masa | 0.706 | 0.792 | **0.8676** |
| Nódulo | 0.671 | 0.717 | **0.7802** |
| Neumonía | 0.633 | 0.713 | **0.7680** |
| Neumotórax | 0.806 | 0.841 | **0.8887** |
| Consolidación | 0.708 | 0.788 | **0.7901** |
| Edema | 0.835 | 0.882 | **0.8878** |
| Enfisema | 0.815 | 0.829 | **0.9371** |
| Fibrosis | 0.769 | 0.767 | **0.8047** |
| Engrosamiento pleural | 0.708 | 0.765 | **0.8062** |
| Hernia | 0.767 | 0.914 | **0.9164** |

Tabla 4.2: Resultados de CheXNet en términos de AUC para las 14 patologías de la base de datos *ChestX-ray14*, comparados con los valores reportados previamente por Wang et al. y Yao et al. En todas las patologías, CheXNet supera los valores previos del estado del arte [17].

## 4.3. CheXNet

Como se mencionó en la sección anterior, el modelo **CheXNet** fue propuesto originalmente por Rajpurkar et al. [17] con el objetivo de desarrollar un sistema automático para la detección de neumonía en radiografías de tórax frontal, utilizando la base de datos *ChestX-ray14*. Posteriormente, el modelo fue extendido para clasificar de manera simultánea las 14 patologías presentes en la base de datos.

### 4.3.1. Entrenamiento original de CheXNet

En el trabajo original, los autores emplearon una red neuronal convolucional densa (*DenseNet*) de 121 capas, sobre la cual reemplazaron la última capa completamente conectada por otra que produjera una salida de 14 dimensiones con activación sigmoide. De esta manera, CheXNet genera un vector de probabilidades asociadas a la presencia de cada una de las siguientes patologías: atelectasia, cardiomegalia, consolidación, edema, derrame, enfisema, fibrosis, hernia, infiltración, masa, nódulo, engrosamiento pleural, neumonía y neumotórax. El valor de cada componente del vector refleja la probabilidad predicha de la presencia de la patología correspondiente.

Para entrenar el modelo, se utilizó la función de costo *Binary Cross-Entropy*, adecuada para problemas multietiqueta. Los pesos iniciales de la red se tomaron de un modelo preentrenado en ImageNet, y fue entrenada de extremo a extremo empleando el optimizador Adam. El entrenamiento se realizó con un *batch size* de 16 imágenes y una tasa de aprendizaje inicial de 0.001, que se reducía en un factor de 10 cada vez que la pérdida de validación se estabilizaba tras una época. El modelo final correspondió a aquel con la menor pérdida de validación registrada.

El conjunto de datos se dividió aleatoriamente en entrenamiento (70 %), validación (10 %) y prueba (20 %), siguiendo trabajos previos sobre *ChestX-ray14* (Wang et al., 2017; Yao et al., 2017), asegurándose de que no hubiera solapamiento de pacientes entre subconjuntos. Antes del entrenamiento las imágenes fueron redimensionadas a $224 \times 224$ píxeles y normalizadas utilizando

la media y desviación estándar de ImageNet. Además, se aplicó una estrategia de aumento de datos en el conjunto de entrenamiento consistente en un giro horizontal aleatorio, con el fin de mejorar la capacidad de generalización del modelo. La métrica principal de evaluación fue el área bajo la curva ROC (AUROC), calculada de forma independiente para cada patología. Con esta configuración, CheXNet logró superar el estado del arte previo en las 14 enfermedades pulmonares, como se muestra en la Tabla 4.2.

### 4.3.2. Implementación en CheXNet-Keras

Para poder replicar el algoritmo **CheXNet**, se empleó como punto de partida el repositorio **CheXNet-Keras** [18], el cual sus autores describen como una herramienta para construir modelos similares a CheXNet, escritos en Keras. En dicho repositorio, se puede entrenar un modelo **DenseNet-121** con salida multietiqueta sobre la base de datos *ChestX-ray14* siguiendo solo una corta lista de instrucciones relativamente simples. Además, se incluye la posibilidad de seleccionar entre otras arquitecturas de redes como VGG16, ResNet50 o InceptionV3. Los autores mencionan que, siguiendo su guía de inicio rápido, se puede obtener un modelo con un rendimiento similar al de CheXNet. Sin embargo, al haber sido publicado hace más de siete años, presentaba múltiples incompatibilidades con versiones actuales de TensorFlow, Keras y otras dependencias. Para poder utilizarlo, fue necesario actualizar el repositorio completo a versiones recientes, corrigiendo llamadas obsoletas, mejorando el flujo de carga de los datos y adaptando la manera en que se construía y entrenaba el modelo.

Tras resolver estas incompatibilidades, se procedió a entrenar un modelo tratando de ser lo más fieles posible al proceso de entrenamiento descrito en el trabajo original de CheXNet (ver sección 4.3.1). En esta implementación se conservaron varios aspectos fundamentales del modelo original, mientras que otros fueron adaptados o modificados para asegurar la compatibilidad y estabilidad del flujo de trabajo. A continuación, se describen los principales puntos de esta configuración:

- Se conservó la misma arquitectura base (**DenseNet-121** preentrenada en ImageNet, con una salida sigmoide de 14 neuronas).
- Las imágenes fueron redimensionadas a $512 \times 512$ píxeles en lugar de $224 \times 224$, siguiendo el criterio utilizado posteriormente en CheXNeXt. Posteriormente, fueron normalizadas utilizando la media y desviación estándar de ImageNet.
- Para el aumento de datos, además del giro horizontal aleatorio, se aplicó un giro con un valor aleatorio en el rango de $[-10\,\%, +10\,\%]$ de un círculo completo (equivalente a $[-36°, +36°]$). Estas operaciones se implementaron mediante las funciones *RandomFlip* y *RandomRotation* del submódulo `keras.layers`.
- Se empleó la misma función de costo (*Binary Cross-Entropy*) y el mismo optimizador (Adam) que en CheXNet, manteniendo también la tasa de aprendizaje inicial (0.001) y el *batch size* (16). El ajuste dinámico de la tasa de aprendizaje se implementó mediante el callback `ReduceLROnPlateau`. Adicionalmente, para prevenir el sobreajuste, se incluyeron los callbacks `EarlyStopping` (deteniendo el entrenamiento si la pérdida de validación no disminuía tras tres épocas) y `ModelCheckpoint` (guardando el modelo con la menor pérdida de validación registrada).
- Para la división del conjunto de datos se utilizaron directamente las listas proporcionadas por los autores de *ChestX-ray14* (`train_val_list.txt` y `test_list.txt`). Con ello, se

garantizó que todos los estudios de un mismo paciente aparecieran únicamente en el conjunto de entrenamiento, validación o en el de prueba, siguiendo una proporción de 70 % para entrenamiento, 10 % para validación y 20 % para prueba.

- Al igual que en CheXNet, la métrica principal de evaluación fue el área bajo la curva ROC (AUROC), calculada de forma independiente para cada patología.

### 4.3.3. Resultados del modelo CheXNet-Keras

Con el objetivo de evaluar el rendimiento del modelo entrenado con la implementación descrita anteriormente, se diseñó un protocolo de evaluación basado en métricas estándar para problemas de clasificación. En esta sección se presentan los resultados obtenidos, comparándolos cuando es posible con los reportados en el trabajo original de CheXNet.

#### Curvas de entrenamiento

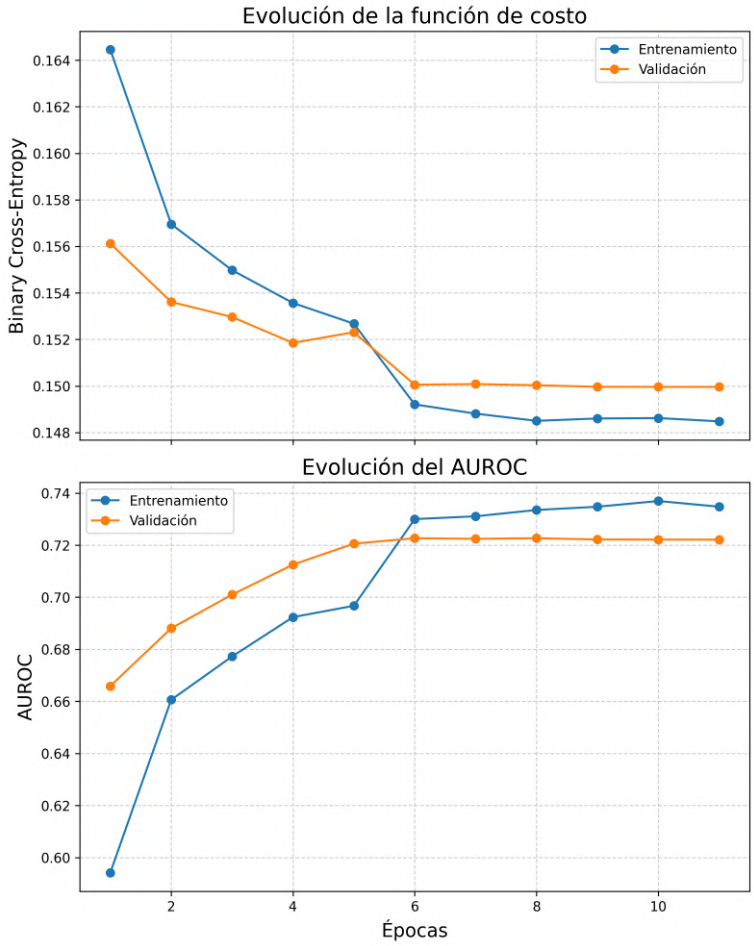


Figura 4.2: Evolución de la función de costo y del AUROC durante el entrenamiento y validación del modelo CheXNet-Keras.

En la Figura 4.2 se muestran las gráficas de la función de costo (*Binary Cross-Entropy*) y del AUROC obtenidas durante las 12 épocas de entrenamiento y validación. Se observa que tanto la pérdida de entrenamiento como la de validación disminuyeron de forma progresiva hasta estabilizarse alrededor de la sexta época. En paralelo, el AUROC mostró un incremento sostenido en ambas particiones, alcanzando valores cercanos a 0.72 en el conjunto de validación. Estos resultados reflejan un aprendizaje estable y consistente, sin indicios marcados de sobreajuste en las primeras etapas del entrenamiento.

### Comparación de AUROC por clase con CheXNet

Dado que el AUROC fue la métrica principal empleada por los autores de **CheXNet** [17] para evaluar su modelo, en la Tabla 4.3 se muestran los valores obtenidos por nuestra implementación en el conjunto de prueba y se comparan directamente con los reportados en el trabajo original para cada una de las 14 patologías presentes en ChestX-ray14.

| **Patología** | **CheXNet** | **Nuestro modelo** |
|---|---|---|
| Atelectasia | 0.8094 | 0.6615 |
| Cardiomegalia | 0.9248 | 0.6609 |
| Derrame | 0.8638 | 0.7054 |
| Infiltración | 0.7345 | 0.6576 |
| Masa | 0.8676 | 0.6025 |
| Nódulo | 0.7802 | 0.6288 |
| Neumonía | 0.7680 | 0.6067 |
| Neumotórax | 0.8887 | 0.7678 |
| Consolidación | 0.7901 | 0.6712 |
| Edema | 0.8878 | 0.7502 |
| Enfisema | 0.9371 | 0.7377 |
| Fibrosis | 0.8047 | 0.7618 |
| Engrosamiento pleural | 0.8062 | 0.6738 |
| Hernia | 0.9164 | 0.7934 |

Tabla 4.3: Comparación de valores de AUROC obtenidos por clase entre **CheXNet** y el modelo entrenado en este trabajo sobre la base de datos *ChestX-ray14*.

Los resultados muestran que, aunque nuestro modelo alcanzó valores de AUROC superiores a 0.5 en todas las patologías (indicando que fue capaz de aprender patrones discriminativos básicos), el desempeño fue consistentemente inferior al de **CheXNet**. Si bien se alcanzaron valores razonablemente altos en algunas clases, como *Hernia* (0.7934 frente a 0.9164) y *Fibrosis* (0.7618 frente a 0.8047), en la mayoría de los casos el rendimiento resultó considerablemente menor, particularmente en patologías como *Cardiomegalia* (0.6609 frente a 0.9248) y *Masa* (0.6025 frente a 0.8676). Esto sugiere que el modelo entrenado logró identificar ciertos patrones relevantes en los datos, pero sin alcanzar la capacidad discriminativa del modelo original.

Por otro lado, dado que el objetivo principal de esta investigación no es mejorar el algoritmo de clasificación, sino aplicar y evaluar el filtro de imágenes propuesto, resulta suficiente contar con un modelo que proporcione predicciones razonablemente confiables, ya que estas constituyen la base sobre la cual opera el filtro. Sin embargo, aunque los valores de AUROC obtenidos pueden considerarse aceptables en términos generales, esta métrica por sí sola no garantiza la calidad de las inferencias del modelo. En consecuencia, se recurrió a métricas adicionales con el fin de evaluar de manera más directa la capacidad del modelo de realizar inferencias útiles para el propósito de este trabajo.

Finalmente, es importante señalar que, antes de adoptar este modelo, se llevaron a cabo múltiples experimentos adicionales, variando hiperparámetros como la tasa de aprendizaje, el tamaño del *batch* e incluso añadiendo capas densas adicionales antes de la salida sigmoide. Ninguno de estos ajustes produjo mejoras significativas respecto a los valores obtenidos, por lo que los resultados presentados en la Tabla 4.3 corresponden al mejor desempeño alcanzado en este trabajo al intentar replicar CheXNet.

### Evaluación mediante métricas adicionales

Como se mencionó anteriormente, aunque el AUROC permite evaluar el desempeño global del modelo de manera independiente del umbral de decisión, puede resultar potencialmente engañoso, especialmente cuando se trabaja con conjuntos de datos desbalanceados (es decir, cuando algunas clases son mucho más frecuentes que otras), como es el caso de *ChestX-ray14*. Por este motivo, se incorporaron métricas adicionales que permiten un análisis más directo de la capacidad del modelo para clasificar correctamente las imágenes. En particular, se calcularon la precisión (*precision*), la sensibilidad (*recall*) y el puntaje $F_1$ (*F1-score*) para cada una de las 14 clases presentes, además de emplear matrices de confusión para algunas patologías representativas. Estas métricas ofrecen una perspectiva complementaria, ya que evidencian con mayor claridad el número de aciertos y errores cometidos en el proceso de clasificación.

Cabe destacar que todas estas métricas dependen del umbral de decisión utilizado para transformar las probabilidades predichas en etiquetas binarias. En este trabajo se adoptó un umbral fijo de 0.5 para todas las patologías, siguiendo la práctica común en problemas de clasificación multietiqueta. Aunque hubiera sido posible calcular umbrales específicos por clase, por ejemplo mediante el análisis de curvas *precision-recall* en función del umbral, esta opción se descartó para mantener el foco del trabajo. El objetivo no es optimizar el clasificador, sino evaluar si el modelo entrenado ofrece inferencias lo suficientemente consistentes como para permitir la aplicación del filtro de imágenes propuesto.

Los resultados obtenidos para precisión, sensibilidad y puntaje $F_1$ muestran que, en la mayoría de las patologías, estas métricas resultaron nulas (0.0), lo que indica que el modelo no fue capaz de generar predicciones positivas correctas en dichas clases al emplear un umbral fijo de 0.5. Este comportamiento contrasta con los valores de AUROC reportados previamente, evidenciando que, aunque el modelo parecía captar ciertas señales en los datos, no logró traducirlas en predicciones válidas.

Las únicas excepciones se presentaron en las clases *Derrame* e *Infiltración*, donde se obtuvieron valores bajos pero distintos de cero, como se resume en la Tabla 4.4. Estos resultados refuerzan la conclusión de que, pese a mostrar AUROC aceptables en varias patologías, el modelo carece de la capacidad necesaria para realizar clasificaciones confiables en el conjunto de prueba.

| **Patología** | **Precisión** | **Sensibilidad** | **$F_1$** |
|---|---|---|---|
| Derrame | 0.4217 | 0.0618 | 0.1078 |
| Infiltración | 0.4572 | 0.0402 | 0.0740 |

Tabla 4.4: Resultados de precisión, sensibilidad y puntaje $F_1$ para las clases con valores distintos de cero utilizando un umbral fijo de 0.5 en el conjunto de prueba de *ChestX-ray14*.

Para visualizar con mayor detalle el tipo de errores cometidos por el modelo, en la Figura 4.3 se presentan las matrices de confusión correspondientes a las patologías *Derrame* e *Infiltración*. En ambos casos la gran mayoría de las muestras negativas fueron correctamente clasificadas como tales, con porcentajes de acierto superiores al 98 %. Sin embargo, el número de falsos negativos es considerablemente alto, el modelo clasificó erróneamente más del 90 % de las imágenes positivas como negativas en ambas clases, lo que confirma que el modelo tiende a predecir la ausencia de enfermedad. Este comportamiento explica los bajos valores de sensibilidad y $F_1$ reportados en la Tabla 4.4, y pone de manifiesto la dificultad del modelo para distinguir correctamente las clases positivas en un entorno de datos altamente desbalanceado.

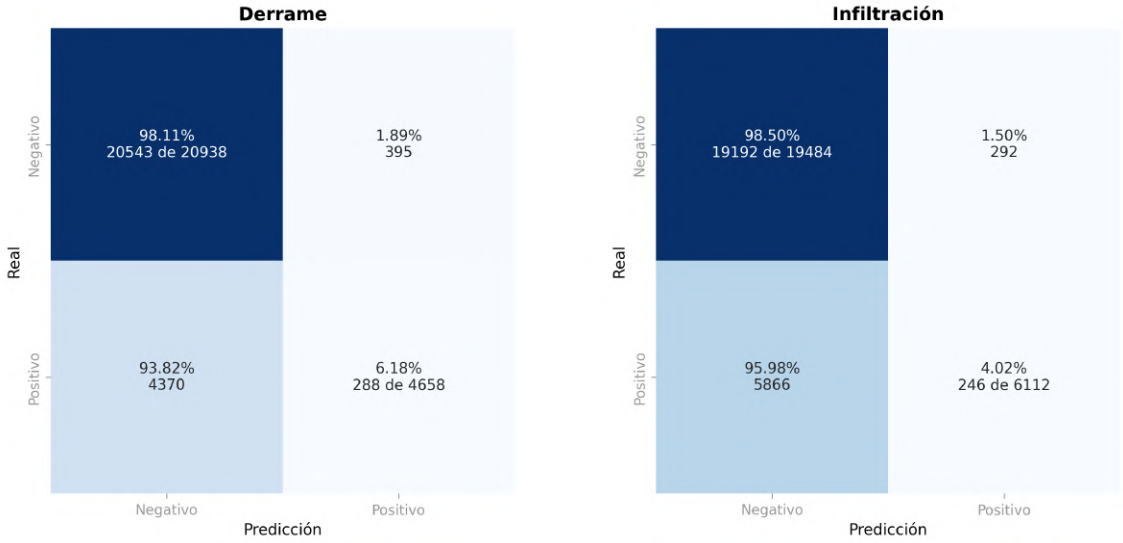


Figura 4.3: Matrices de confusión obtenidas para las patologías *Derrame* (izquierda) e *Infiltración* (derecha) en el conjunto de prueba, utilizando un umbral fijo de 0.5. Los valores dentro de cada celda indican el porcentaje de ejemplos respecto al total de su clase real, seguido del número absoluto de muestras.

En conclusión, los resultados obtenidos al intentar replicar **CheXNet** mostraron que, si bien fue posible entrenar una red con un comportamiento estable y valores de AUROC aceptables en el conjunto de prueba, el modelo no logró generalizar adecuadamente hacia las clases positivas. Las métricas de precisión, sensibilidad y puntaje $F_1$ evidenciaron que el modelo tiende a predecir sistemáticamente la ausencia de enfermedad, siendo incapaz de identificar con fiabilidad las imágenes que presentan alguna patología. Este comportamiento se reflejó también en las matrices de confusión, donde más del 90 % de las muestras positivas fueron clasificadas erróneamente como negativas.

Una de las posibles causas de este comportamiento es el severo desbalance de clases presente en la base de datos *ChestX-ray14*. De las más de 100,000 imágenes disponibles, solo alrededor del 20 % presentan al menos una patología, y entre estas, existe una gran disparidad en el número de ejemplos por clase. Por ejemplo, mientras la clase más representada, *Infiltración*, cuenta con cerca de 20,000 ejemplos positivos, la menos frecuente, *Hernia*, apenas alcanza poco más de 200 (Figura 4.4). Esta distribución desigual puede inducir a que el modelo aprenda un sesgo hacia la clase negativa (*sin hallazgos*), obteniendo métricas aparentemente competitivas como el AUROC, sin que ello implique una verdadera capacidad de detección de patologías.

Estos resultados evidencian que, bajo el enfoque de clasificación multietiqueta original, el modelo tiende a comportarse como un clasificador trivial que predice mayormente la ausencia de enfermedad.

Para mitigar este problema, se optó por reformular el problema de clasificación hacia un esquema binario, donde cada modelo se entrene para distinguir la presencia o ausencia de una patología específica. Este cambio de enfoque, descrito en la siguiente sección, busca mitigar los efectos del desbalance de clases y facilitar la obtención de inferencias más coherentes y útiles para la aplicación del filtro de imágenes.

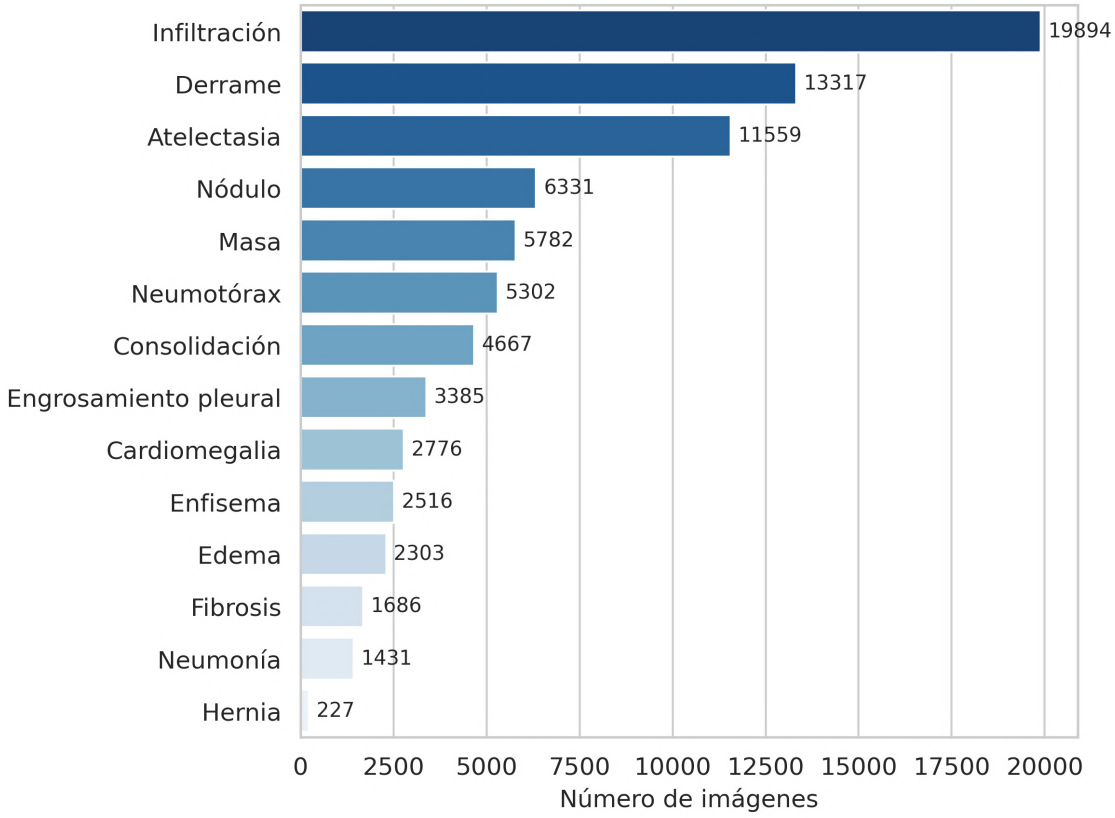


Figura 4.4: Distribución de las 14 patologías en la base de datos *ChestX-ray14*. Se observa un marcado desbalance, con una alta proporción de imágenes sin hallazgos y una gran variación en la frecuencia de las distintas enfermedades.

## 4.4. CheXNet Binario

Como se mencionó en la sección 4.2, el modelo **CheXNet** fue concebido originalmente como un clasificador binario para la detección de neumonía a partir de radiografías de tórax, y posteriormente extendido para la clasificación simultánea de 14 patologías en la base de datos *ChestX-ray14*. Esta ampliación introdujo un grado adicional de complejidad al problema, asociado tanto al severo desbalance de clases como a la coexistencia de múltiples enfermedades en una misma imagen. Ambos factores dificultan que el modelo aprenda representaciones discriminativas efectivas y tienden a favorecer predicciones negativas, como se evidenció en la sección anterior.

Inspirados en la formulación original de CheXNet para la detección de neumonía, se propuso retomar un enfoque similar, reformulando el problema multietiqueta como una serie de problemas de clasificación binaria independientes. En este nuevo planteamiento, cada modelo se entrena para

distinguir entre la presencia o ausencia de una patología específica. Para ello, se seleccionaron dos enfermedades con una alta disponibilidad de ejemplos positivos en el conjunto de datos: *Derrame pleural* (*Effusion*) y *Atelectasia*.

Aunque *Infiltración* es la clase con mayor número de ejemplos positivos, la elección de *Derrame* y *Atelectasia* se justifica por la disponibilidad de anotaciones con cuadros delimitadores (*bounding boxes*) que indican la localización aproximada de la enfermedad en las radiografías. Esta información resulta esencial para las etapas posteriores de evaluación del filtro de imágenes propuesto, ya que permite validar visualmente las regiones de interés identificadas por el modelo.

En cuanto a la definición de las etiquetas, se consideraron dos enfoques complementarios. En el primero, denominado **enfoque inclusivo**, se etiquetaron como positivas todas las imágenes que contenían la enfermedad objetivo, incluso si coexistían con otras patologías, y como negativas todas las demás imágenes. Cabe destacar que este fue el enfoque empleado en el trabajo original de CheXNet para la detección de neumonía. En el segundo, denominado **enfoque exclusivo**, las imágenes positivas fueron aquellas que contenían únicamente la enfermedad objetivo, mientras que las negativas correspondieron a radiografías sin hallazgos patológicos. Este segundo enfoque busca reducir el ruido en el aprendizaje asociado con la presencia de otras enfermedades y favorecer la identificación de patrones visuales más específicos.

Para ambos enfoques, se construyeron conjuntos de datos balanceados con igual número de ejemplos positivos y negativos. Esto permitió utilizar la métrica *exactitud* (*accuracy*) como una medida válida del rendimiento del modelo, ya que en un escenario balanceado un valor de 0.5 equivale al desempeño de un clasificador aleatorio. De esta manera, se evitó la dependencia de métricas potencialmente engañosas, como el AUROC, que pueden resultar poco representativas en presencia de un desbalance severo.

En el trabajo original de *CheXNet*, el desbalance entre las clases positiva y negativa se abordó mediante el uso de una función de costo ponderada denominada *Weighted Binary Cross Entropy*, una versión modificada de la entropía cruzada binaria que ajusta la contribución de cada clase en función de su frecuencia. En este trabajo, se optó por prescindir de dicho ajuste, empleando en su lugar conjuntos balanceados para simplificar la implementación y mantener un control más claro sobre el impacto de las modificaciones introducidas al modelo.

### 4.4.1. Entrenamiento

El proceso de entrenamiento seguido para los modelos binarios propuestos fue, en gran medida, similar al descrito previamente para el modelo **CheXNet-Keras** (secciones 4.3.1 y 4.3.2). No obstante, se introdujeron algunas modificaciones importantes para adaptar el modelo al nuevo enfoque de clasificación binaria.

En ambos casos, se empleó como base la arquitectura **DenseNet-121** preentrenada en *ImageNet*, manteniendo la misma configuración general del modelo anterior: la función de costo utilizada fue *Binary Cross-Entropy*, el optimizador *Adam*, un tamaño de lote (*batch size*) de 16, y una tasa de aprendizaje inicial de $10^{-3}$, la cual se redujo automáticamente mediante el callback *ReduceLROnPlateau* cuando la pérdida de validación se estabilizaba. Además, se continuó utilizando el callback *ModelCheckpoint* para conservar el modelo con la menor pérdida de validación registrada, así como *EarlyStopping* para detener el entrenamiento cuando no se observaban mejoras significativas después de tres épocas consecutivas.

De igual forma, se aplicaron las mismas técnicas de aumento de datos descritas previamente: giros horizontales aleatorios y rotaciones dentro del rango de $[-10\,\%, +10\,\%]$, implementadas mediante las capas *RandomFlip* y *RandomRotation* de Keras. Las imágenes fueron redimensionadas a $512 \times 512$ píxeles y normalizadas empleando la media y desviación estándar de *ImageNet*.

Sin embargo, a pesar de conservar esta configuración base, se introdujeron tres modificaciones relevantes respecto al modelo original:

1. **Capa de salida:** dado que el problema fue reformulado como una tarea de clasificación binaria, la última capa completamente conectada se sustituyó por una única neurona con activación sigmoide, en lugar de las 14 utilizadas en la versión multietiqueta.
2. **Métrica de evaluación:** aunque el AUROC continuó empleándose como métrica de validación, se incorporó la métrica *exactitud* como medida principal del desempeño del modelo, ya que al tratarse de conjuntos de datos balanceados esta métrica resulta más representativa.
3. **Ajuste fino (*Fine-tuning*):** tras completar el entrenamiento inicial, se realizó una segunda etapa de optimización en la que se descongelaron los dos últimos bloques de la arquitectura *DenseNet-121*, permitiendo así refinar los pesos de las capas convolucionales superiores con una tasa de aprendizaje reducida ($10^{-5}$). Este proceso tuvo como objetivo mejorar la capacidad del modelo para capturar patrones visuales más específicos asociados a cada patología.

En cuanto a la construcción de los conjuntos de datos balanceados, dependiendo del enfoque empleado (inclusivo o exclusivo) y de la enfermedad seleccionada, se obtuvieron tamaños de conjuntos distintos. Del total de ejemplos considerados en cada caso, el 90 % se destinó al entrenamiento y el 10 % restante a la validación, asegurando así una adecuada representación de ambas clases en cada partición. En la Tabla 4.5 se resumen las dimensiones finales de los conjuntos de entrenamiento, validación y prueba empleados para cada caso.

| Enfermedad | Enfoque | Conjunto | Total | Positivas | Negativas |
|---|---|---|---|---|---|
| Atelectasia | 1 | Entrenamiento/Validación | 16,560 | 8,280 | 8,280 |
| Atelectasia | 1 | Prueba | 6,558 | 3,279 | 3,279 |
| Atelectasia | 2 | Entrenamiento/Validación | 6,828 | 3,414 | 3,414 |
| Atelectasia | 2 | Prueba | 1,602 | 801 | 801 |
| Derrame | 1 | Entrenamiento/Validación | 17,318 | 8,659 | 8,659 |
| Derrame | 1 | Prueba | 9,316 | 4,658 | 4,658 |
| Derrame | 2 | Entrenamiento/Validación | 5,576 | 2,788 | 2,788 |
| Derrame | 2 | Prueba | 2,334 | 1,167 | 1,167 |

Tabla 4.5: Tamaños de los conjuntos de entrenamiento, validación y prueba empleados para cada enfoque de clasificación binaria. El **Enfoque 1** corresponde a la definición inclusiva de clases, mientras que el **Enfoque 2** corresponde a la definición exclusiva.

### 4.4.2. Resultados

En esta sección se presentan los resultados obtenidos tras el entrenamiento de los modelos binarios correspondientes a las patologías *Atelectasia* y *Derrame*, bajo los dos enfoques de etiquetado propuestos en la sección anterior. Se analizan tanto las curvas de entrenamiento y validación como las métricas obtenidas en el conjunto de prueba, con el objetivo de evaluar la estabilidad del entrenamiento y la capacidad del modelo para realizar inferencias confiables.

El proceso de entrenamiento de cada modelo se llevó a cabo en dos fases consecutivas. En la primera fase, se entrenaron las capas finales de la red manteniendo congelados los pesos del modelo base *DenseNet121*, mientras que en la segunda fase se realizó un ajuste fino (*fine-tuning*) descongelando los dos últimos bloques convolucionales. En ambos casos se utilizó la estrategia de *Early Stopping* para detener automáticamente el entrenamiento cuando la pérdida de validación dejara de mejorar, por lo que el número total de épocas varió entre modelos. Por ejemplo, para la patología *Atelectasia*, en el primer enfoque, la primera fase de entrenamiento duró 8 épocas y la etapa de ajuste fino se extendió por 11 épocas adicionales.

Durante el entrenamiento se registraron tres métricas principales: la función de costo Binary Cross-Entropy, el área bajo la curva ROC (AUROC) y la exactitud (accuracy), con el fin de evaluar simultáneamente la convergencia, la capacidad discriminativa y la proporción de predicciones correctas. Para efectos de presentación, únicamente se muestran las curvas correspondientes a la segunda fase de entrenamiento, ya que reflejan de manera más representativa el comportamiento final del modelo tras el ajuste fino.

### Resultados para Atelectasia

En la Figura 4.5 se muestran las curvas de entrenamiento correspondientes a los dos enfoques evaluados para la patología *Atelectasia*. En ambos casos se observa un comportamiento estable y una convergencia progresiva de la función de costo, acompañada por incrementos sostenidos en las métricas de exactitud y AUROC a lo largo de las épocas. Esto sugiere que los modelos fueron capaces de aprender representaciones discriminativas sin evidenciar un sobreajuste significativo.

En el **Enfoque 1**, donde las imágenes positivas incluyen aquellas con *Atelectasia* coexistiendo con otras patologías, se observa una reducción continua de la función de costo hasta estabilizarse alrededor de un valor de 0.54, mientras que la exactitud de validación alcanza un valor cercano a 0.71 y el AUROC se aproxima a 0.78. Si bien las curvas de entrenamiento superan levemente a las de validación, la diferencia es moderada, lo que indica un ajuste razonable del modelo.

Por otro lado, el **Enfoque 2**, que restringe las imágenes positivas a aquellas que contienen exclusivamente *Atelectasia* y las negativas a radiografías sin hallazgos patológicos, muestra un rendimiento ligeramente superior. La función de costo disminuye de manera más pronunciada durante las primeras épocas y alcanza valores finales cercanos a 0.52, mientras que la exactitud y el AUROC de validación se estabilizan alrededor de 0.70 y 0.78, respectivamente. Este comportamiento sugiere que el modelo logra una mejor separación entre las clases al reducir el ruido asociado a la coexistencia de múltiples enfermedades en las imágenes.

Una vez finalizado el entrenamiento, cada modelo fue evaluado en su conjunto de prueba correspondiente. La Figura 4.6 muestra las matrices de confusión obtenidas para los dos enfoques propuestos en la clasificación binaria de *Atelectasia*, mientras que en la Tabla 4.6 se resumen las métricas cuantitativas correspondientes. En ambos casos, las métricas fueron calculadas utilizando un umbral de decisión estándar de 0.5, siguiendo la misma metodología aplicada en las secciones anteriores.

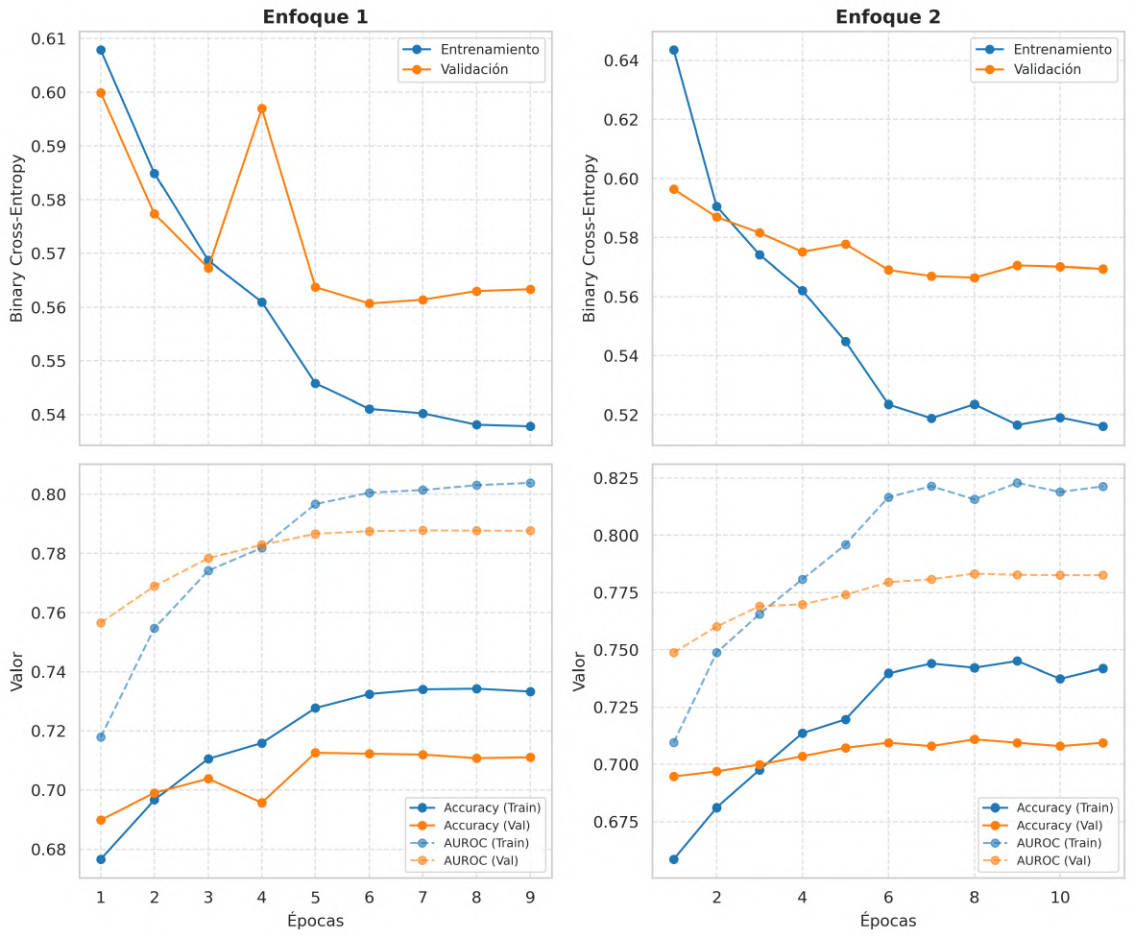


Figura 4.5: Evolución de la función de costo, la exactitud (*accuracy*) y el AUROC durante el entrenamiento de los modelos para la detección de *Atelectasia*, bajo los enfoques 1 y 2. Las líneas continuas representan la exactitud (*accuracy*) y las líneas punteadas el AUROC, tanto para los conjuntos de entrenamiento como de validación.

A partir de las matrices de confusión puede observarse que los dos modelos presentan un patrón de comportamiento muy similar. En general, alrededor del 73 % de las imágenes positivas fueron correctamente clasificadas, mientras que menos del 60 % de las imágenes negativas fueron identificadas correctamente. Esto indica que los modelos tienden ligeramente a predecir la presencia de enfermedad, generando una proporción moderada de falsos positivos.

En términos cuantitativos, los valores de precisión, sensibilidad, puntaje $F_1$, exactitud y AUROC son comparables entre ambos enfoques, con diferencias menores al 1 % en la mayoría de los casos. Aunque el segundo enfoque muestra valores marginalmente superiores en precisión, exactitud y AUROC, estas diferencias no son lo suficientemente significativas como para afirmar una mejora sustancial en el desempeño general. En conjunto, ambos enfoques alcanzan un rendimiento equivalente para la detección de *Atelectasia*, lo que sugiere que la definición más estricta de las etiquetas positivas en el Enfoque 2 no altera de manera notable la capacidad de clasificación del modelo.

| Enfoque | Precisión | Sensibilidad | $F_1$ | Exactitud | AUROC |
|---|---|---|---|---|---|
| Enfoque 1 | 0.6333 | 0.7304 | 0.6784 | 0.6537 | 0.7142 |
| Enfoque 2 | 0.6393 | 0.7303 | 0.6818 | 0.6592 | 0.7275 |

Tabla 4.6: Resultados de precisión, sensibilidad, puntaje $F_1$, exactitud y AUROC para los modelos binarios entrenados en la detección de *Atelectasia* bajo ambos enfoques.

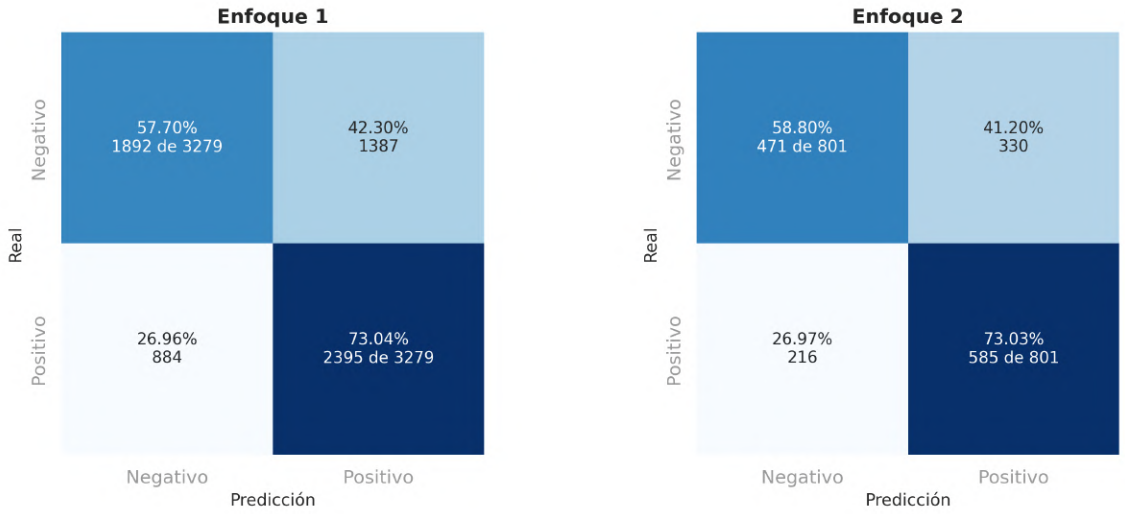


Figura 4.6: Matrices de confusión obtenidas en el conjunto de prueba para los modelos binarios entrenados en la detección de *Atelectasia* bajo los dos enfoques propuestos. Los valores dentro de cada celda indican el porcentaje de ejemplos respecto al total de su clase real, seguido del número absoluto de muestras.

### Resultados para Derrame

De manera análoga al caso de *Atelectasia*, en la Figura 4.7 se presentan las curvas de entrenamiento obtenidas para los dos enfoques evaluados en la detección de *Derrame*. En ambos modelos se observa una disminución sostenida de la función de costo acompañada por incrementos consistentes en las métricas de exactitud y AUROC, reflejando un aprendizaje estable y una adecuada capacidad de generalización. En comparación con los resultados obtenidos para *Atelectasia*, las curvas de entrenamiento y validación muestran una convergencia más rápida y valores finales superiores, especialmente en el segundo enfoque, donde la función de costo desciende hasta aproximadamente 0.41 y las métricas de validación alcanzan valores cercanos a 0.80 en exactitud y 0.87 en AUROC. Estos resultados indican que el modelo logra una separación más clara entre clases y una representación más robusta de los patrones asociados al *Derrame*, posiblemente debido a la menor variabilidad visual de esta patología en comparación con la *Atelectasia*.

No obstante, como en el caso anterior, será necesario analizar el desempeño sobre los conjuntos de prueba correspondientes a cada modelo para determinar si estas diferencias durante el entrenamiento se traducen en mejoras efectivas en la capacidad de generalización.

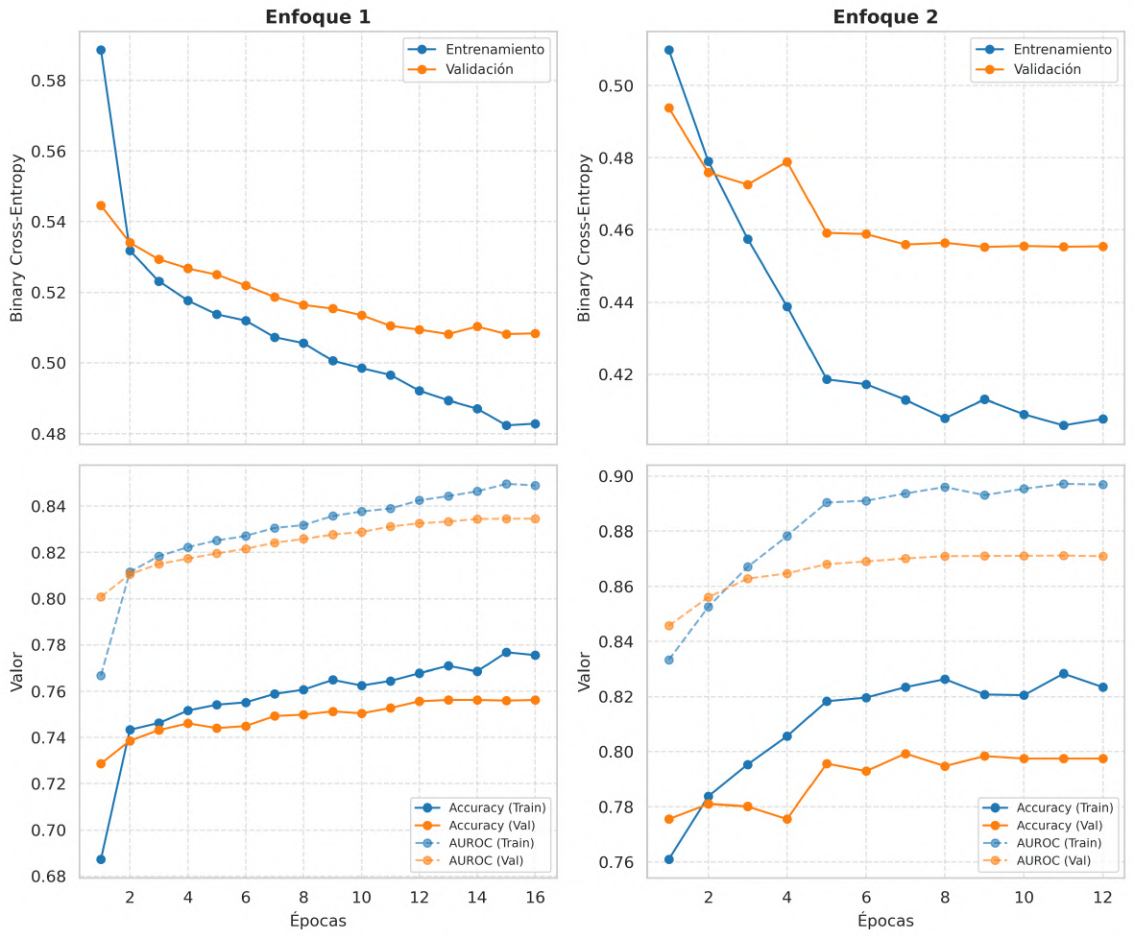


Figura 4.7: Evolución de la función de costo, la exactitud (*accuracy*) y el AUROC durante el entrenamiento de los modelos para la detección de *Derrame*, bajo los enfoques 1 y 2. Las líneas continuas representan la exactitud (*accuracy*) y las líneas punteadas el AUROC, tanto para los conjuntos de entrenamiento como de validación.

La Figura 4.8 muestra las matrices de confusión obtenidas para los dos enfoques propuestos en la clasificación binaria de *Derrame*, mientras que la Tabla 4.7 resume las métricas cuantitativas calculadas en el conjunto de prueba. Al igual que en el caso anterior, todas las métricas fueron estimadas utilizando un umbral de decisión estándar de 0.5.

En ambos enfoques puede observarse un patrón común: los modelos presentan una elevada sensibilidad, con aproximadamente el 91 % de las imágenes positivas correctamente clasificadas, pero una menor capacidad para identificar los casos negativos. En el Enfoque 1, únicamente el 42.79 % de las imágenes sin derrame fueron correctamente clasificadas, mientras que el Enfoque 2 mejora este valor hasta un 50.73 %. Este comportamiento sugiere una ligera reducción de falsos positivos al adoptar una definición más estricta de las etiquetas, aunque ambos modelos mantienen una tendencia a sobrestimar la presencia de la patología.

En términos cuantitativos, los resultados del Enfoque 2 superan consistentemente a los del Enfoque 1 en todas las métricas, con incrementos particularmente notables en precisión (de 0.6140 a 0.6475), puntaje $F_1$ (de 0.7333 a 0.7548), exactitud (de 0.6690 a 0.7061) y AUROC (de 0.7624 a

0.7967). Si bien las diferencias no son drásticas, reflejan una mejora sistemática en la capacidad del modelo para discriminar entre imágenes positivas y negativas. Comparado con los resultados obtenidos para *Atelectasia*, ambos modelos muestran un desempeño superior, lo que refuerza la idea de que el *Derrame* es una patología visualmente más consistente y, por tanto, más fácilmente identificable por el modelo.

| Enfoque | Precisión | Sensibilidad | $F_1$ | Accuracy | AUROC |
|---|---|---|---|---|---|
| Enfoque 1 | 0.6140 | 0.9100 | 0.7333 | 0.6690 | 0.7624 |
| Enfoque 2 | 0.6475 | 0.9049 | 0.7548 | 0.7061 | 0.7967 |

Tabla 4.7: Resultados de precisión, sensibilidad, puntaje $F_1$, exactitud y AUROC para los modelos binarios entrenados para la detección de *Derrame* bajo ambos enfoques.

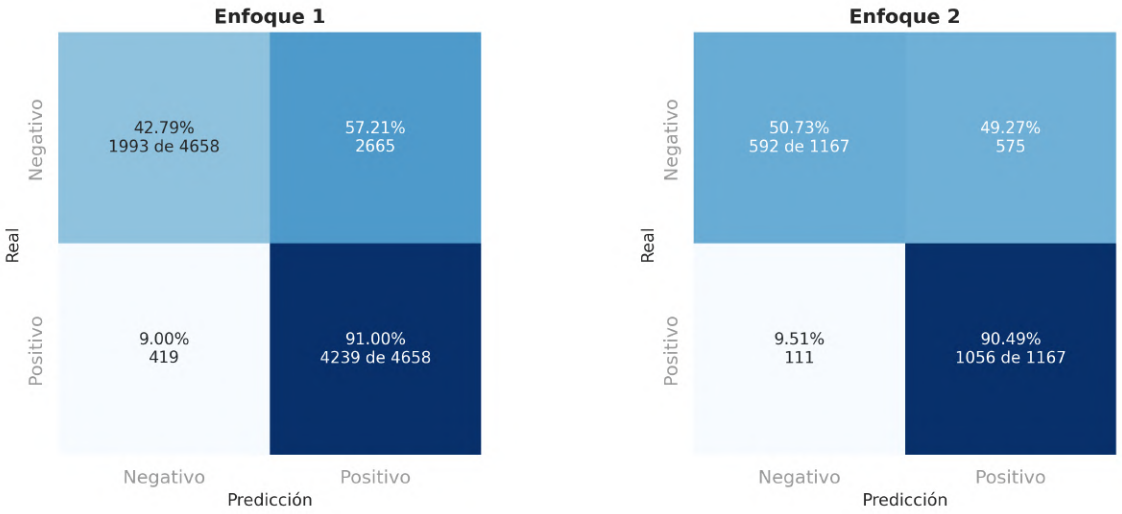


Figura 4.8: Matrices de confusión obtenidas en el conjunto de prueba para los modelos binarios entrenados para la detección de *Derrame* bajo los dos enfoques propuestos. Los valores dentro de cada celda indican el porcentaje de ejemplos respecto al total de su clase real, seguido del número absoluto de muestras.

### Análisis general

En conjunto, los resultados obtenidos para *Atelectasia* y *Derrame* muestran un comportamiento coherente entre ambos enfoques. La reformulación del problema de clasificación multietiqueta a binario permitió obtener un entrenamiento más estable y métricas de desempeño más consistentes, reflejando una mejor capacidad del modelo para distinguir entre imágenes con y sin la patología objetivo. Sin embargo, los valores de exactitud y AUROC se mantienen en un rango moderado (0.65–0.80), lo cual indica que aún existe un grado de confusión entre las clases, posiblemente derivado de la similitud visual entre la patología objetivo (*Atelectasia* o *Derrame*) y otras patologías pulmonares, así como de la variabilidad inherente a las radiografías del conjunto *ChestX-ray14*. Esto evidencia que el problema sigue siendo complejo incluso bajo esta simplificación.

En términos comparativos, el **Enfoque 2** tiende a ofrecer resultados ligeramente superiores, especialmente en la detección de *Derrame*, donde la definición más estricta de las etiquetas positivas reduce el ruido en el aprendizaje y mejora la separación entre clases. En contraste, para *Atelectasia*

las diferencias son mínimas, lo que sugiere que la efectividad de cada enfoque depende de la variabilidad visual y la frecuencia de la enfermedad en el conjunto de datos.

Finalmente, aunque los modelos binarios entrenados en esta sección no alcanzan métricas de rendimiento sobresalientes, es decir, no logran clasificar correctamente el 100 % de los casos, los resultados obtenidos demuestran que son capaces de distinguir entre las patologías pulmonares más representativas del conjunto de datos con un grado de acierto razonable (entre el 65 % y el 80 %). Esto contrasta con el comportamiento del modelo multietiqueta, que mostraba una marcada tendencia a predecir únicamente la ausencia de enfermedad. Dado que el objetivo central de este trabajo no radica en optimizar el desempeño del clasificador, sino en aplicar y evaluar el filtro de imágenes propuesto, las predicciones generadas por estos modelos se consideran lo suficientemente confiables para servir como base en la siguiente etapa. En la próxima sección se presenta la aplicación del filtro y se analiza la capacidad de los mapas de calor generados para resaltar las regiones anatómicas asociadas a la patología de interés.

## 4.5. Aplicación del filtro propuesto con el modelo binario

A diferencia de las imágenes naturales utilizadas en el capítulo anterior, donde era relativamente sencillo determinar si el filtro de extracción de características lograba identificar correctamente la clase objetivo, en el caso de las imágenes médicas el problema se vuelve considerablemente más complejo. La interpretación de una radiografía de tórax requiere no solo de conocimiento especializado, sino también del contexto clínico del paciente, lo que hace que la evaluación visual directa de los resultados de un modelo de interpretación sea una tarea poco trivial incluso para expertos.

Con el fin de disponer de un punto de referencia objetivo para la validación del filtro propuesto, se seleccionó el conjunto de datos *ChestX-ray14*, el cual, como se describió en la sección 4.1, incluye alrededor de mil imágenes con anotaciones manuales en forma de cuadros delimitadores (*bounding boxes*). Estas anotaciones fueron realizadas por radiólogos certificados y señalan la región anatómica donde se manifiesta la patología correspondiente, permitiendo así evaluar la capacidad del filtro de identificar correctamente las zonas de interés asociadas a cada enfermedad.

Cada una de estas anotaciones se encuentra registrada en el archivo `BBox_List_2017.csv`, que contiene el índice de la imagen, la etiqueta de la patología y las coordenadas del cuadro delimitador en formato `Bbox[x, y, w, h]`, donde `[x, y]` representa la esquina superior izquierda del cuadro y `[w, h]` su ancho y altura, respectivamente.

La Figura 4.9 muestra un conjunto de ejemplos de estas anotaciones para las patologías *Atelectasia* y *Derrame pleural*. En ella se puede observar la gran variabilidad presente en las regiones afectadas, tanto en ubicación como en extensión, lo que refleja la complejidad inherente al problema de detección y localización de estas enfermedades a partir de imágenes radiográficas.

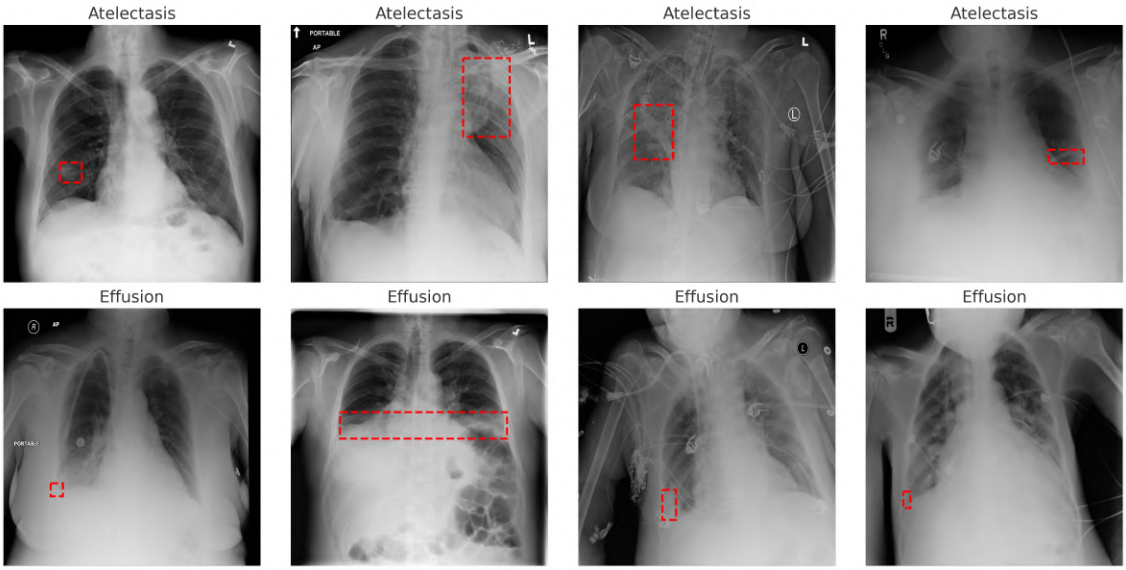


Figura 4.9: Ejemplos de imágenes con anotaciones manuales (*bounding boxes*) pertenecientes al conjunto de datos *ChestX-ray14*. Se muestran cuatro ejemplos de la patología *Atelectasia* (arriba) y cuatro de *Derrame pleural* (abajo). La variabilidad en el tamaño, posición y forma de las regiones afectadas evidencia la dificultad del problema de localización en radiografías de tórax.

### 4.5.1. Procedimiento de aplicación del filtro

Con el fin de evaluar la capacidad del filtro propuesto para identificar las regiones anatómicas asociadas a las patologías de interés, se procedió primero a identificar el número total de imágenes con cuadros delimitadores disponibles por enfermedad (*Atelectasia* y *Derrame pleural*). Posteriormente, se evaluó cada uno de los modelos binarios en este subconjunto de imágenes con anotaciones, considerando únicamente aquellas correspondientes a la enfermedad que cada modelo es capaz de clasificar. A continuación, se seleccionaron las imágenes que fueron clasificadas correctamente por ambos modelos binarios entrenados para una misma patología. Este criterio garantiza que el análisis del filtro se base en predicciones válidas, evitando que errores de clasificación afecten la interpretación de los mapas de calor generados. En la Tabla 4.8 se resume el número total de imágenes con cuadros delimitadores disponibles por enfermedad, la sensibilidad (tasa de imágenes clasificadas correctamente) obtenida por cada modelo al evaluarse sobre dicho subconjunto y el número total de imágenes que ambos modelos clasificaron correctamente para una misma patología.

| **Parámetro** | **Atelectasia** | **Derrame pleural** |
|---|---|---|
| N. Imágenes con BBox | 180 | 153 |
| Sensibilidad (E1) | 0.7389 | 0.9412 |
| Sensibilidad (E2) | 0.7556 | 0.9216 |
| N. Correctas por ambos modelos | 121 | 140 |

Tabla 4.8: Resumen de las imágenes con cuadros delimitadores disponibles por enfermedad, las sensibilidades obtenidas por los modelos entrenados bajo los enfoques 1 y 2, y el número total de imágenes clasificadas correctamente por ambos modelos para cada patología.

Por otro lado, debido al alto costo computacional del filtro propuesto, se seleccionaron únicamente seis imágenes por patología, tomadas aleatoriamente del subconjunto de radiografías clasificadas correctamente por ambos modelos y con anotaciones manuales disponibles. Este número permitió mantener una diversidad anatómica suficiente para el análisis visual de resultados, reduciendo al mismo tiempo el tiempo total de procesamiento a un nivel manejable (aproximadamente cuatro horas para el conjunto total de imágenes analizadas).

Una vez definidos los dos conjuntos de imágenes sobre los que el filtro propuesto sería aplicado (uno por enfermedad), se procedió a su implementación empleando las predicciones de cada uno de los modelos binarios correspondientes. En otras palabras, primero se aplicó el filtro utilizando las predicciones del modelo binario entrenado con el primer enfoque de etiquetado y, posteriormente, se aplicó el mismo filtro sobre esas mismas imágenes, pero empleando las predicciones del modelo binario correspondiente al segundo enfoque (esto se realizó para ambas enfermedades).

La aplicación del filtro se llevó a cabo siguiendo los mismos principios y etapas descritos en la Sección 3.5.1, conservando características como la superposición de oclusión, el promediado de superposiciones, el filtrado por percentil 95 y el suavizado gaussiano. Sin embargo, considerando que las radiografías de tórax utilizadas en este estudio fueron redimensionadas a una resolución significativamente mayor ($512 \times 512$ píxeles) que las imágenes empleadas en los experimentos previos con *ImageNet* ($299 \times 299$ píxeles), se ajustaron proporcionalmente los parámetros principales del filtro: el tamaño del parche de oclusión (*grid size*) y el tamaño del paso (*stride*). De esta manera, se establecieron los valores $\mathbf{grid\_size} = \mathbf{30}$ y $\mathbf{stride} = \mathbf{10}$, los cuales mantienen una densidad de muestreo equivalente a la usada en el conjunto *ImageNet* y reducen de forma considerable el tiempo total de procesamiento sin afectar notablemente la calidad de los mapas generados.

Finalmente, para cada imagen seleccionada se generaron los mapas de oclusión correspondientes utilizando el filtro propuesto. Posteriormente, los mapas fueron suavizados y normalizados antes de su superposición sobre las radiografías originales, empleando el mapa de colores *jet* para representar la escala de activación. Esta representación, coherente con la metodología empleada en el capítulo anterior, facilita la comparación visual entre las regiones destacadas por el filtro y las áreas delimitadas en las anotaciones originales.

### 4.5.2. Análisis de resultados

Las Figuras 4.10 y 4.11 muestran los resultados obtenidos tras aplicar el filtro propuesto sobre las imágenes seleccionadas de ambas patologías. Cada figura muestra las seis radiografías utilizadas para la evaluación del filtro, organizadas en una cuadrícula de tres filas por cuatro columnas. En cada par de columnas se observa la misma radiografía procesada mediante los dos modelos binarios entrenados: el modelo del *Enfoque 1* (columnas impares) y el modelo del *Enfoque 2* (columnas pares).

Esta disposición facilita la comparación directa entre los mapas de oclusión generados por ambos modelos, permitiendo identificar similitudes y diferencias en las regiones anatómicas resaltadas como relevantes para la clasificación. Además, los cuadros delimitadores en color resaltan las áreas de referencia proporcionadas por las anotaciones manuales, lo que permite evaluar visualmente el grado de correspondencia entre las zonas de activación detectadas por el filtro y las regiones de interés señaladas por los especialistas.

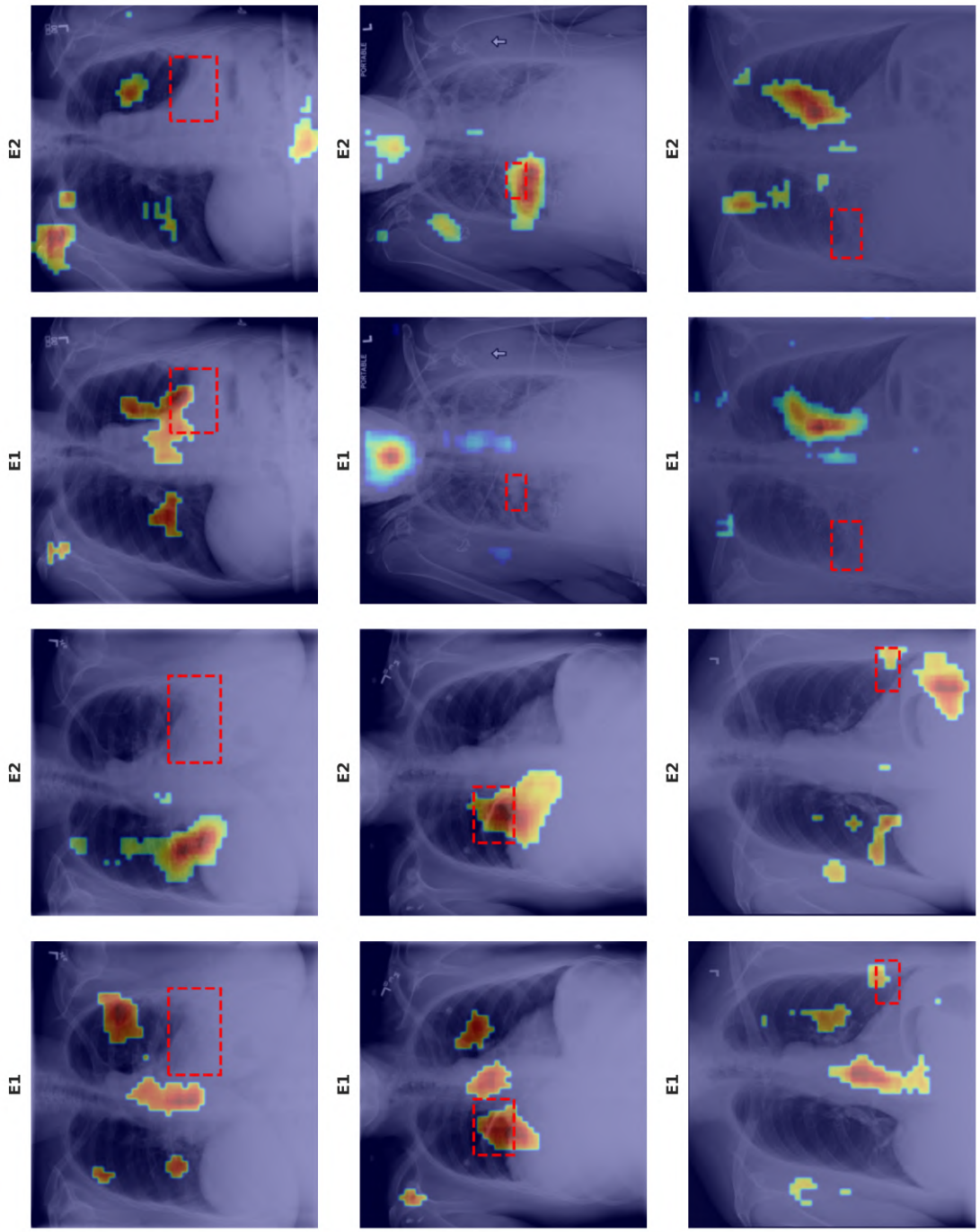


Figura 4.10: Resultados del filtro propuesto aplicados sobre las radiografías con diagnóstico de *Atelectasia*. Las columnas impares muestran las imágenes filtradas obtenidas mediante el modelo binario del *Enfoque 1*, mientras que las columnas pares presentan los resultados del modelo correspondiente al *Enfoque 2*. Cada par de columnas corresponde a una misma radiografía.

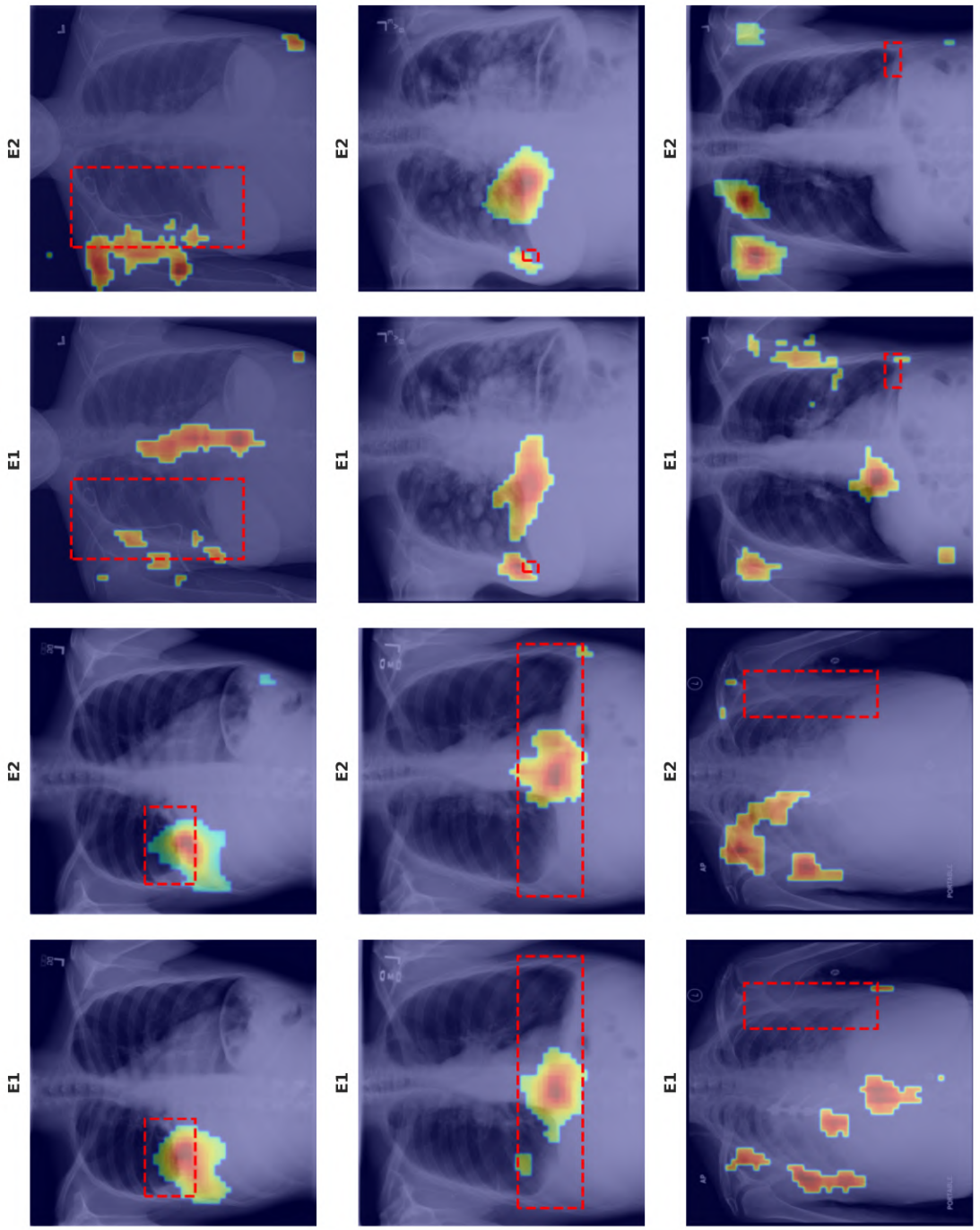


Figura 4.11: Resultados del filtro propuesto aplicados sobre las radiografías con diagnóstico de *Derrame pleural*. Al igual que en la Figura 4.10, las columnas impares corresponden al *Enfoque 1* y las pares al *Enfoque 2*.

En términos generales, los resultados revelan que ninguno de los modelos binarios fue capaz de identificar de manera precisa la ubicación de la patología en la mayoría de los casos. Las

activaciones obtenidas tienden a distribuirse de forma dispersa en distintas zonas de la radiografía, sin concentrarse exclusivamente dentro de los cuadros delimitadores. Este comportamiento sugiere que los modelos aún presentan un nivel considerable de ruido en sus salidas, posiblemente debido a la coexistencia de múltiples patologías en una misma imagen y a la superposición de patrones visuales entre clases.

Aun así, pueden observarse algunos casos en los que el filtro logra resaltar adecuadamente las regiones anatómicas relevantes. Por ejemplo, en la Figura 4.10 (fila 2, columnas 1 y 2), las activaciones se alinean de manera razonablemente precisa con las áreas afectadas en ambas configuraciones (E1 y E2), aunque persisten pequeñas zonas activas fuera del cuadro delimitador. Un comportamiento similar se aprecia en la Figura 4.11 (fila 1, columnas 1 y 2), donde ambos modelos logran concentrar la activación en las regiones correctas, reduciendo en buena medida la dispersión observada en otros ejemplos. Destaca también el caso mostrado en la Figura 4.11 (fila 2, columnas 1 y 2), en el cual las activaciones se encuentran casi completamente contenidas dentro del cuadro delimitador, mostrando un desempeño sobresaliente en comparación con los demás ejemplos.

Por otro lado, existen ejemplos en los que solo uno de los modelos binarios (E1 o E2) detecta correctamente la ubicación de la patología, como se observa en la Figura 4.10 (fila 2, columnas 3 y 4). Esto indica que las diferencias entre los enfoques de etiquetado pueden influir en la forma en que el modelo aprende a identificar las regiones relevantes. No obstante, no se observan discrepancias significativas en la calidad global de los mapas entre ambos enfoques.

Finalmente, al comparar los resultados entre ambas patologías, se aprecia una ligera tendencia a que los mapas de Atelectasia presenten una mayor dispersión y ruido en comparación con los de Derrame pleural. Aun así, esta diferencia no es muy marcada, ya que ambos modelos muestran patrones de activación parcialmente inconsistentes, lo cual evidencia las limitaciones de los modelos actuales para localizar con precisión las regiones patológicas en radiografías de tórax, y justifica la necesidad de seguir perfeccionando tanto el filtro propuesto como las estrategias de entrenamiento utilizadas.

## 4.6. Modelo Multiclase

Tras el análisis de los resultados obtenidos con los modelos binarios, se identificó que, si bien estos eran capaces de distinguir entre la presencia y ausencia de una patología específica con un rendimiento razonable, las representaciones aprendidas no parecían ser lo suficientemente discriminativas como para localizar con precisión las regiones anatómicas donde se encontraba la enfermedad. Este comportamiento se atribuyó, en parte, a que los modelos podrían estar basando sus decisiones en patrones o características no específicas de la patología objetivo, posiblemente influenciados por la presencia de otras enfermedades en las radiografías, lo cual generaba ruido en las salidas y dificultaba la interpretación de los mapas de calor.

Ante esta limitación, se exploraron diversas estrategias para mejorar el rendimiento de los modelos binarios, incluyendo la modificación de hiperparámetros (como la tasa de aprendizaje, el tamaño del lote y la profundidad de las capas densas) y el ajuste de la arquitectura de la red. Sin embargo, a pesar de múltiples experimentos, los resultados no mostraron mejoras significativas en la precisión ni en la coherencia espacial de las activaciones obtenidas mediante el filtro propuesto. Esta situación llevó a considerar la necesidad de un cambio más profundo en el enfoque de clasificación.

Durante la búsqueda de alternativas, se identificó el proyecto *Xray Xplorer* [19], un desarrollo

orientado a la detección de COVID-19 a partir de radiografías de tórax mediante redes neuronales convolucionales (RNC), técnicas de amplificación por gradiente (XGBoost) y el método de interpretabilidad Grad-CAM. Dicho proyecto entrena un modelo multiclase para distinguir entre tres categorías: casos positivos de COVID-19, casos normales y casos con otras infecciones pulmonares, alcanzando un rendimiento sobresaliente (precisión del 94.05 % y puntuación F1 del 94.08 %). Además, los mapas de calor obtenidos mediante Grad-CAM mostraron activaciones concentradas y visualmente coherentes, lo que evidencia una mejora notable en la interpretabilidad del modelo.

Este planteamiento guarda una estrecha relación con los objetivos del presente trabajo, ya que ambos problemas comparten la misma naturaleza: clasificar una patología específica a partir de radiografías de tórax y generar explicaciones visuales sobre las decisiones del modelo. A diferencia del caso de COVID-19, la base de datos *ChestX-ray14* empleada en este trabajo, como se ha mencionado anteriormente, cuenta con cuadros delimitadores (*Bounding Boxes*) que indican la ubicación exacta de la enfermedad, lo cual permite evaluar de manera cuantitativa la capacidad del modelo y del filtro propuesto para identificar las regiones anatómicas relevantes.

Inspirado en la metodología de *Xray Xplorer*, se propuso reformular el problema de clasificación hacia un esquema **multiclase**, en el cual el modelo aprende a distinguir entre tres categorías distintas: *Sin hallazgos*, *Patología objetivo* y *Otra enfermedad*. La primera categoría corresponde a radiografías sin indicios de anormalidades; la segunda, a aquellas que presentan exclusivamente la patología de interés; y la tercera, a imágenes que contienen únicamente alguna de las otras trece enfermedades presentes en la base de datos. Este cambio de enfoque busca proporcionar al modelo un contexto clínico más rico y balanceado, promoviendo la adquisición de representaciones más específicas y, en consecuencia, una localización más precisa de las regiones afectadas. A diferencia de los enfoques multietiqueta y binario utilizados previamente, esta reformulación permite que el modelo aprenda a distinguir explícitamente entre la ausencia de enfermedad, la presencia de la patología objetivo y la manifestación de otras afecciones pulmonares, reduciendo así la confusión entre clases.

### 4.6.1. Entrenamiento del modelo multiclase

En el proyecto original *Xray Xplorer* [19], los autores emplearon tres arquitecturas convolucionales preentrenadas en ImageNet: *VGG16*, *ResNet50* e *Inception V3*, adaptadas a un problema de clasificación multiclase con tres categorías: *Normal*, *COVID-19* e *Infección sin COVID*. Para ello, cada red base fue extendida con una capa de agrupación de promedios globales (*Global Average Pooling*), seguida de una capa densa de 1024 unidades con activación *ReLU*, y una capa de salida con tres unidades y activación *softmax*. De esta forma, la salida del modelo corresponde a un vector de tres probabilidades normalizadas, donde la componente con mayor valor determina la clase predicha para la imagen de entrada.

El modelo fue entrenado utilizando el conjunto de datos **COVID-QU-Ex** (Conjunto de datos ampliado de la Universidad de Qatar sobre la enfermedad del coronavirus), disponible en Kaggle, el cual incluye 33,900 radiografías de tórax clasificadas en las tres categorías antes mencionadas. Dichas imágenes fueron divididas en subconjuntos de entrenamiento (21,705), validación (5,410) y prueba (6,785), y redimensionadas a $256 \times 256$ píxeles. Previo al entrenamiento, las imágenes se normalizaron al rango [0, 1] y se aplicó aumentación de datos mediante rotaciones aleatorias, traslaciones, volteo horizontal, ajustes de brillo y contraste, y transformaciones de zoom, con el fin de incrementar la robustez del modelo frente a las variaciones que pueden presentarse en la adquisición de las radiografías.

El entrenamiento se llevó a cabo en dos etapas: una inicial con los pesos del modelo base congelados, y una segunda fase de ajuste fino en la que se descongelaron las últimas cuatro capas convolucionales para adaptar las representaciones aprendidas a las características específicas de las radiografías de tórax (un proceso similar al que se hizo en la Sección 4.4). En ambas fases, se utilizó el optimizador *Adam* con tasas de aprendizaje de $10^{-3}$ y $10^{-4}$, respectivamente, la función de pérdida *sparse categorical cross-entropy*, adecuada para clasificación multiclase con etiquetas enteras, y un tamaño de lote (*batch size*) de 32 imágenes.

El diseño y entrenamiento del modelo multiclase desarrollado en este trabajo siguió, en gran medida, los principios y mecanismos descritos en el proyecto *Xray Xplorer*, con algunas diferencias adaptadas a los objetivos particulares de esta investigación. En primer lugar, se empleó la base de datos **ChestX-ray14**, la cual como se ha mencionado anteriormente, ofrece anotaciones exhaustivas sobre distintas patologías pulmonares, incluyendo cuadros delimitadores (BBoxes) que permiten evaluar la localización de las regiones anatómicas asociadas a cada enfermedad. En este contexto, el problema de clasificación se reformuló para distinguir entre tres categorías clínicas: *Sin hallazgos*, *Patología objetivo* y *Otra enfermedad*.

En segundo lugar, se decidió trabajar exclusivamente con la arquitectura *VGG16*, debido a que esta fue la que mostró el mejor rendimiento en el proyecto original. En consecuencia, se empleó la función de preprocesamiento `applications.vgg16.preprocess_input`, la cual convierte las imágenes del espacio de color RGB a BGR y centra cada canal en torno a cero, conforme a las estadísticas del conjunto ImageNet, en lugar de normalizarlas al rango [0, 1]. Este paso es fundamental para mantener la compatibilidad con los pesos preentrenados del modelo VGG16.

El entrenamiento del modelo se llevó a cabo utilizando el mismo esquema de dos etapas descrito anteriormente: una fase inicial con las capas convolucionales congeladas y una segunda de ajuste fino, descongelando las últimas 4 capas del modelo base. En ambas etapas se aplicaron mecanismos de regularización automática para prevenir el sobreajuste, incluyendo la reducción automática de la tasa de aprendizaje (*ReduceLROnPlateau*), la detención temprana (*EarlyStopping*) y el guardado del mejor modelo en función del desempeño en el conjunto de validación (*ModelCheckpoint*). Mecanismos que ya fueron utilizados y descritos anteriormente (ver Secciones 4.3.2 y 4.4).

Finalmente, se seleccionaron las patologías *Atelectasia* y *Derrame pleural* para mantener coherencia con los experimentos realizados con los modelos binarios. Para cada una de ellas se construyó un conjunto de datos multiclase balanceado, seleccionando un número igual de muestras por categoría con el fin de reducir el sesgo hacia las clases mayoritarias y promover un entrenamiento más estable. En la Tabla 4.6.1 se resumen las dimensiones finales de los conjuntos de entrenamiento, validación y prueba empleados para cada caso.

| **Clase** | **Índice** | **Entrenamiento** | **Validación** | **Prueba** |
|---|---|---|---|---|
| | | *Atelectasia* | | |
| Sin hallazgos | 0 | 2757 | 657 | 801 |
| Atelectasia | 1 | 2743 | 671 | 801 |
| Otra enfermedad | 2 | 2724 | 690 | 801 |
| | | *Derrame pleural* | | |
| Sin hallazgos | 0 | 2265 | 523 | 1167 |
| Derrame | 1 | 2212 | 576 | 1167 |
| Otra enfermedad | 2 | 2243 | 545 | 1167 |

Tabla 4.9: Distribución balanceada de imágenes por clase y conjunto para los modelos multiclase.

### 4.6.2. Resultados del modelo multiclase

En esta sección se presentan los resultados obtenidos tras el entrenamiento de los modelos multiclase correspondientes a las patologías *Atelectasia* y *Derrame pleural*. Se analizan las curvas de entrenamiento y validación, las matrices de confusión y las métricas cuantitativas obtenidas en el conjunto de prueba, con el objetivo de evaluar la estabilidad del entrenamiento, la capacidad del modelo para discriminar entre las tres clases y la consistencia de su comportamiento frente a diferentes patologías.

#### Análisis de las curvas de entrenamiento

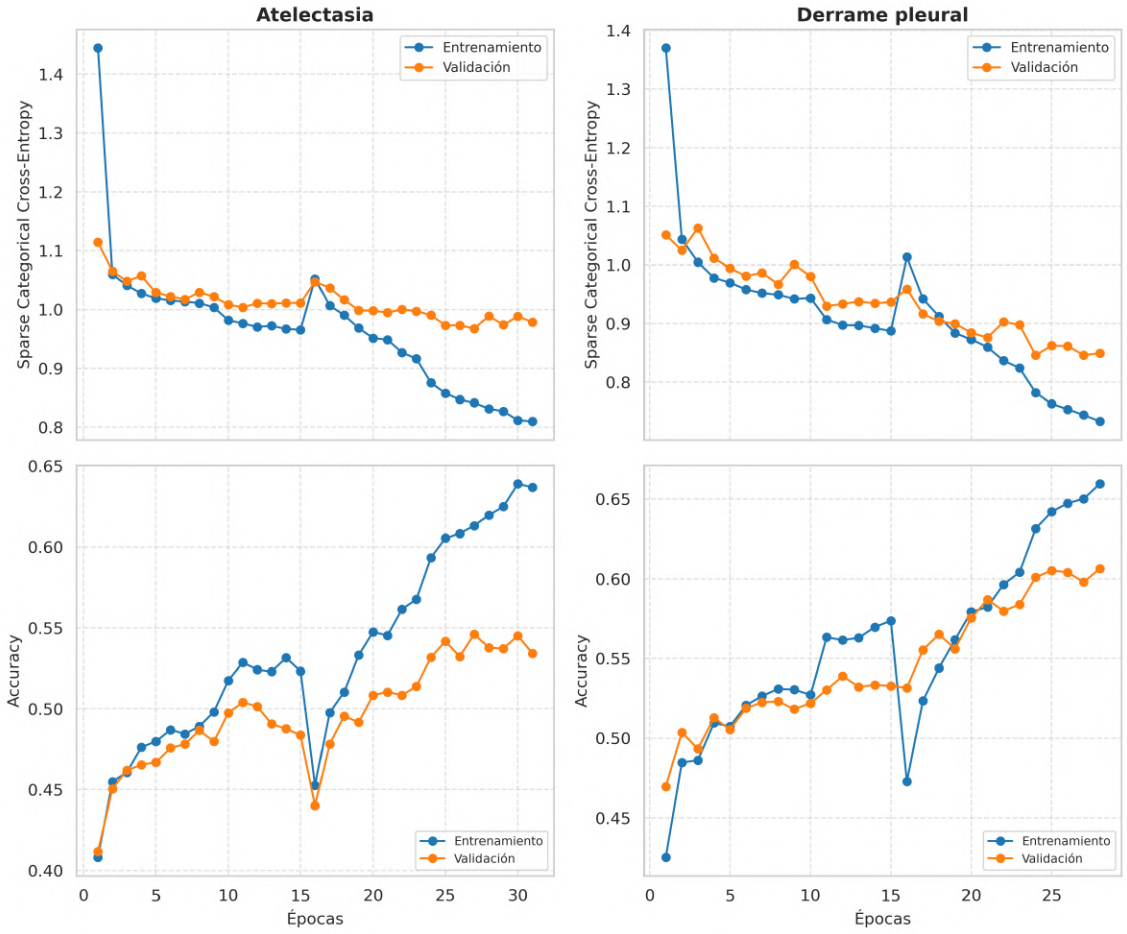


Figura 4.12: Evolución de la función de pérdida (*sparse categorical cross-entropy*) y la exactitud (*accuracy*) durante el entrenamiento de los modelos multiclase para *Atelectasia* y *Derrame pleural*. Las curvas representan el entrenamiento completo (ambas fases), razón por la cual se observan picos alrededor de las épocas 15–16.

En la Figura 4.12 se muestran las curvas de entrenamiento y validación correspondientes a los modelos multiclase desarrollados para las patologías *Atelectasia* y *Derrame pleural*. En esta ocasión, se empleó únicamente la métrica *accuracy* como medida del rendimiento del modelo durante

el entrenamiento, junto con la función de pérdida *sparse categorical cross-entropy*. Las gráficas representan el proceso completo de entrenamiento, es decir, incluyen tanto la primera etapa con las capas convolucionales congeladas como la segunda etapa de ajuste fino (*fine-tuning*), motivo por el cual se observan ligeros picos alrededor de las épocas 15–16, correspondientes a la transición entre ambas fases.

En ambos modelos se aprecia una disminución progresiva y consistente de la función de pérdida a lo largo de las épocas, acompañada por incrementos sostenidos en la exactitud tanto en el conjunto de entrenamiento como en el de validación. Este comportamiento indica que los modelos fueron capaces de aprender representaciones discriminativas entre las tres clases sin evidenciar un sobreajuste significativo.

Comparativamente, el modelo entrenado para la detección de *Derrame pleural* muestra una convergencia más rápida y una pérdida final ligeramente inferior en relación con el modelo de *Atelectasia*. Asimismo, su curva de validación exhibe una evolución más estable y una menor brecha entre las métricas de entrenamiento y validación. Este comportamiento sugiere que la clasificación de *Derrame pleural* resulta más sencilla para la red, posiblemente debido a la naturaleza más homogénea y visualmente distintiva de esta patología en comparación con la *Atelectasia*, que suele presentar patrones radiográficos más difusos y variables.

En general, ambos modelos mantienen una tendencia de mejora constante hasta las últimas épocas, sin indicios de divergencia entre las métricas de entrenamiento y validación, lo cual refleja una adecuada capacidad de generalización y estabilidad en el proceso de aprendizaje.

### Análisis de las matrices de confusión

La Figura 4.13 muestra las matrices de confusión obtenidas en el conjunto de prueba para los modelos multiclase entrenados en la detección de *Atelectasia* y *Derrame pleural*. Cada matriz refleja la distribución de predicciones correctas e incorrectas entre las tres clases definidas, expresadas en porcentaje y número absoluto de muestras.

En el caso de **Atelectasia**, el modelo presenta una distribución de predicciones dominada por la clase *Otra enfermedad*, la cual concentra una gran proporción de ejemplos mal clasificados provenientes de las otras dos clases. Aproximadamente el 39 % de las imágenes realmente etiquetadas como "Sin hallazgos" fueron correctamente identificadas, mientras que un 49 % fueron clasificadas erróneamente como "Otra enfermedad". De manera similar, solo el 39 % de las imágenes con *Atelectasia* fueron correctamente detectadas, mientras que el 45 % se confundieron con otras patologías. Este comportamiento sugiere que el modelo logra detectar la presencia de anormalidades radiológicas, pero tiene dificultades para diferenciar los patrones específicos de la *Atelectasia* frente a los de otras afecciones pulmonares.

Por otro lado, el modelo de **Derrame pleural** muestra un patrón de clasificación más concentrado y una mayor coherencia entre clases, aunque con una distribución de predicciones dominada por la clase *Patología objetivo*. En este caso, el 90.4 % de las imágenes con derrame fueron correctamente clasificadas, mientras que las clases "Sin hallazgos" y "Otra enfermedad" alcanzaron precisiones del 54.8 % y 7.5 %, respectivamente. Aun cuando persiste cierta confusión entre las tres categorías, el modelo demuestra una clara capacidad para reconocer los patrones visuales característicos del *Derrame pleural*.

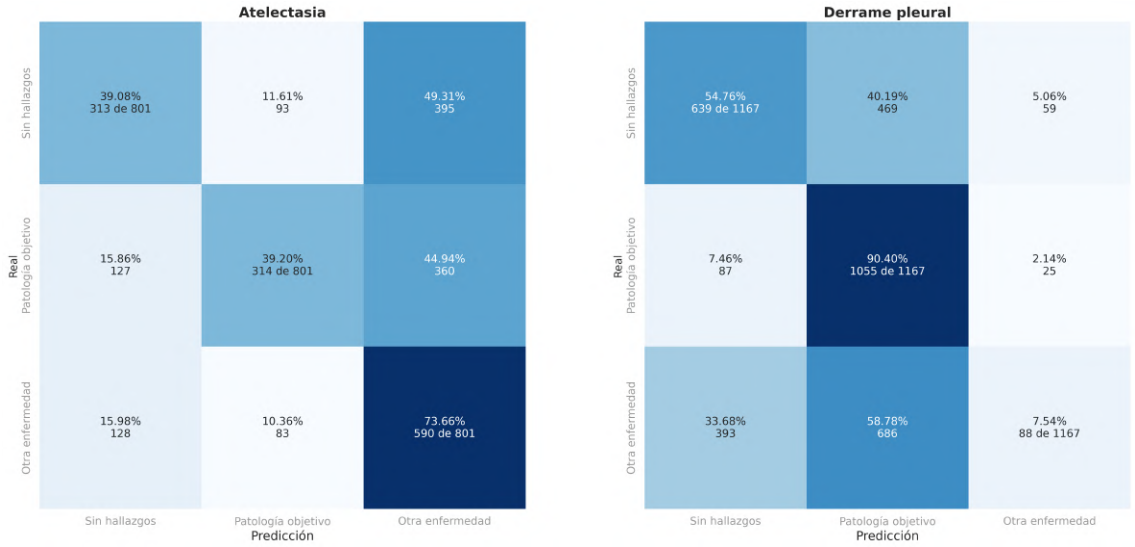


Figura 4.13: Matrices de confusión obtenidas en el conjunto de prueba para los modelos multiclase correspondientes a *Atelectasia* y *Derrame pleural*. Los valores indican el porcentaje y número de ejemplos respecto al total de su clase real.

### Métricas cuantitativas

La Tabla 4.10 resume los valores de precisión, sensibilidad, puntaje $F_1$, exactitud y AUROC obtenidos para ambos modelos multiclase. Todas las métricas se calcularon de manera global (*macro average*), considerando el equilibrio entre las tres clases definidas.

| **Patología** | **Precisión** | **Sensibilidad** | **$F_1$** | **Exactitud** | **AUROC** |
|---|---|---|---|---|---|
| Atelectasia | 0.5435 | 0.5065 | 0.4979 | 0.5065 | 0.6886 |
| Derrame pleural | 0.5200 | 0.5090 | 0.4384 | 0.5090 | 0.7166 |

Tabla 4.10: Resultados globales de precisión, sensibilidad, puntaje $F_1$, exactitud y AUROC para los modelos multiclase correspondientes a *Atelectasia* y *Derrame pleural*.

En términos cuantitativos, ambos modelos presentan un rendimiento moderado, con valores de exactitud y $F_1$ cercanos al 0.50. No obstante, los valores de AUROC superiores a 0.68 en ambos casos indican una capacidad discriminativa no trivial. El modelo de *Derrame pleural* alcanza los mejores resultados globales (AUROC = 0.7166), lo que refleja una mejor separación entre las tres categorías clínicas en comparación con el modelo de *Atelectasia*.

### Discusión de los resultados

Aunque los valores absolutos de las métricas son moderados, los resultados obtenidos son adecuados para los fines de este trabajo. Ambos modelos evidencian un aprendizaje efectivo de patrones radiológicos y una capacidad parcial para distinguir entre casos normales, casos con la patología objetivo y casos con otras enfermedades. Este nivel de rendimiento es coherente con la complejidad inherente del problema de clasificación multiclase en radiografías de tórax, donde múltiples patologías pueden presentar características visuales similares o solapadas.

Si bien los modelos binarios desarrollados previamente alcanzaron valores de precisión, sensibilidad y AUROC más altos (por ejemplo, el modelo binario de *Derrame pleural* obtuvo un AUROC de 0.79 frente al 0.72 del modelo multiclase), esta diferencia es esperada debido a la naturaleza más sencilla del problema binario. En el caso multiclase, el modelo no solo debe identificar la presencia o ausencia de una enfermedad, sino también distinguirla de otras patologías con manifestaciones radiográficas parcialmente coincidentes. Por lo tanto, aunque el rendimiento cuantitativo es menor, las representaciones aprendidas tienden a ser más discriminativas entre clases.

Cabe resaltar que el propósito de esta etapa no es alcanzar un rendimiento clínicamente competitivo, sino disponer de modelos con representaciones internas consistentes y diferenciadas que permitan analizar las regiones de activación mediante el filtro propuesto. Los valores de AUROC superiores a 0.68 confirman que las redes han aprendido correlaciones válidas entre patrones visuales y las tres categorías definidas, proporcionando así una base sólida para aplicar el filtro de interpretabilidad y evaluar si las activaciones se concentran en las zonas anatómicas correspondientes a las patologías, de acuerdo con los cuadros delimitadores de la base de datos.

## 4.7. Aplicación del filtro propuesto con el modelo multiclase

Siguiendo la misma metodología descrita en la Sección 4.5, en esta etapa se evalúa nuevamente la capacidad del filtro propuesto para identificar las regiones anatómicas relevantes en las radiografías de tórax, pero empleando ahora los modelos multiclase desarrollados en la Sección 4.6.

El objetivo principal es analizar si el entrenamiento conjunto en tres categorías clínicas (*Sin hallazgos*, *Patología objetivo* y *Otra enfermedad*) permite generar mapas de activación más focalizados y coherentes con las regiones donde se manifiestan las patologías de interés, en comparación con los resultados obtenidos por los modelos binarios.

### 4.7.1. Procedimiento de aplicación del filtro

De manera análoga al procedimiento seguido con los modelos binarios, la aplicación del filtro propuesto se realizó empleando las mismas seis radiografías seleccionadas previamente para las patologías *Atelectasia* y *Derrame pleural*. Este criterio permite establecer una comparación visual directa entre ambos enfoques, evaluando si el modelo multiclase, al ser entrenado para discriminar entre tres categorías clínicas, logra una mayor precisión en la localización de las zonas de interés.

Antes de aplicar el filtro, se verificó que las radiografías seleccionadas fueran correctamente clasificadas por los modelos multiclase correspondientes, con el fin de garantizar que el análisis se base exclusivamente en predicciones válidas. En la Tabla 4.11 se presentan los resultados de esta verificación, donde se indica el identificador único de cada imagen, la clase predicha por el modelo multiclase y si la predicción fue correcta. Como puede observarse, el modelo multiclase para

*Atelectasia* clasificó correctamente cuatro de las seis radiografías (67 %), mientras que el modelo correspondiente a *Derrame pleural* alcanzó una exactitud del 83 %. Por lo tanto, únicamente las imágenes correctamente clasificadas en ambos enfoques fueron consideradas para la aplicación del filtro propuesto.

| **Identificador de imagen** | **Predicción** | **Correcta** | |
|---|---|---|---|
| | *Atelectasia* | | |
| 00007124_008 | Otra enfermedad | No | |
| 00008554_009 | Patología objetivo | Sí | |
| 00017582_003 | Patología objetivo | Sí | |
| 00019124_045 | Otra enfermedad | No | |
| 00028620_000 | Patología objetivo | Sí | |
| 00029631_006 | Patología objetivo | Sí | |
| | | **Exactitud** | **0.67** |
| | *Derrame pleural* | | |
| 00014015_003 | Patología objetivo | Sí | |
| 00014346_010 | Patología objetivo | Sí | |
| 00015078_013 | Patología objetivo | Sí | |
| 00018102_001 | Patología objetivo | Sí | |
| 00021132_000 | Patología objetivo | Sí | |
| 00026810_001 | Otra enfermedad | No | |
| | | **Exactitud** | **0.83** |

Tabla 4.11: Resultados de la clasificación de las imágenes de prueba utilizadas previamente en el caso binario, evaluadas con los modelos multiclase correspondientes a las patologías *Atelectasia* y *Derrame pleural*. Se muestra el identificador único de cada imagen, la clase predicha y si la predicción fue correcta.

Una vez más, la aplicación del filtro se realizó siguiendo exactamente los mismos principios y etapas descritos en la Sección 3.5.1. Sin embargo, debido a que las radiografías utilizadas en los modelos multiclase fueron redimensionadas a una resolución de $256 \times 256$ píxeles, se mantuvieron los valores de los parámetros empleados en los experimentos previos con el conjunto *ImageNet*, es decir, **grid_size = 15** y **stride = 5**. Estos valores proporcionan una resolución espacial adecuada en los mapas de activación y reducen el tiempo de procesamiento sin afectar significativamente la calidad de los resultados obtenidos.

### 4.7.2. Análisis de resultados

Las Figuras 4.14 y 4.15 muestran los resultados de la aplicación del filtro propuesto para las patologías *Atelectasia* y *Derrame pleural*, respectivamente. Cada figura se compone de tres columnas: la primera corresponde a la radiografía original con su cuadro delimitador manual (*bounding box*), la segunda presenta el mapa de activación generado por el modelo binario (segundo enfoque), y la tercera muestra el mapa generado por el modelo multiclase.

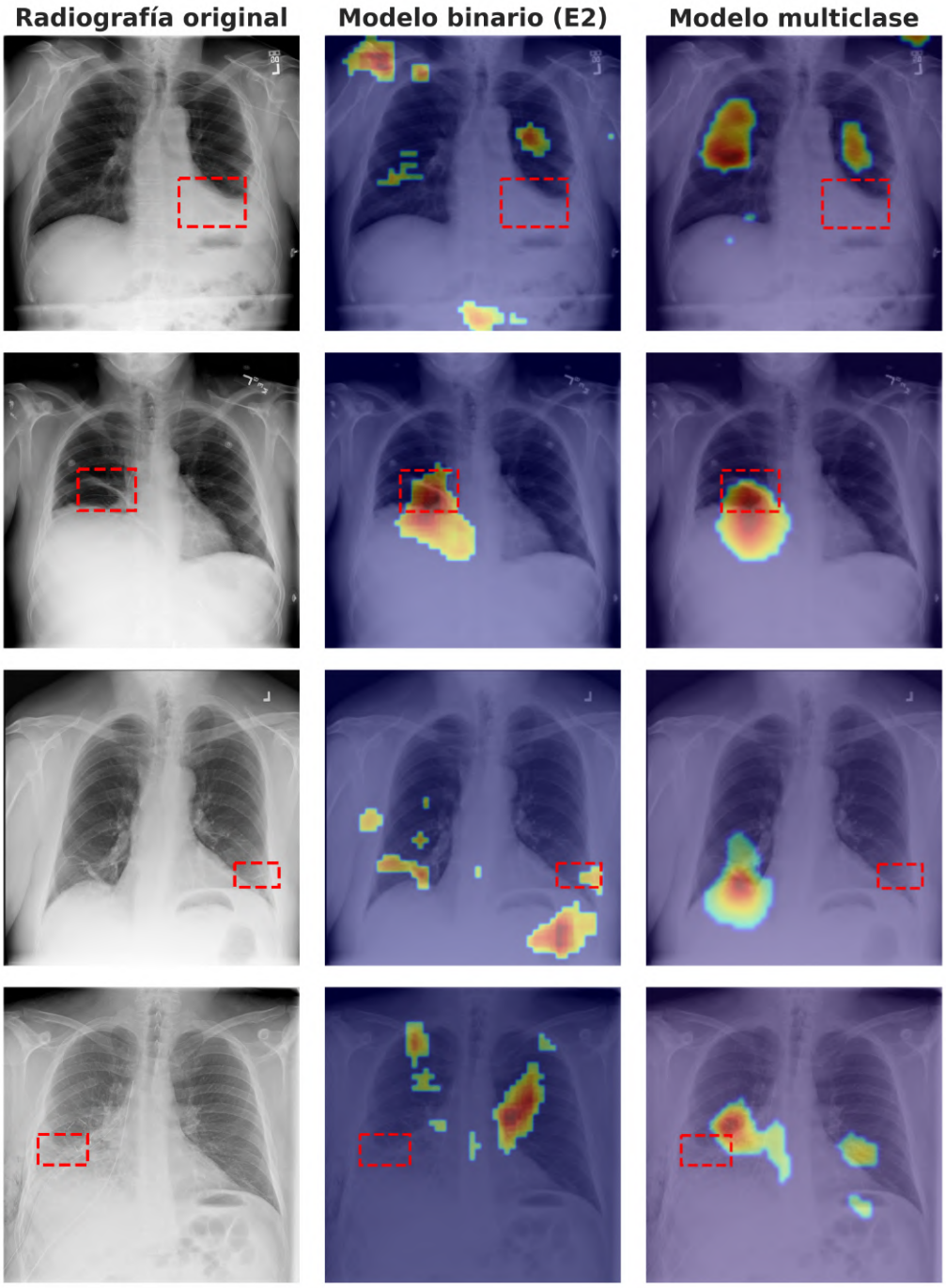


Figura 4.14: Comparación visual de los resultados obtenidos por el filtro propuesto aplicado a las radiografías de prueba correspondientes a la patología *Atelectasia*. Se muestran, de izquierda a derecha: la radiografía original con su cuadro delimitador, el mapa generado por el modelo binario (enfoque 2) y el mapa generado por el modelo multiclase.

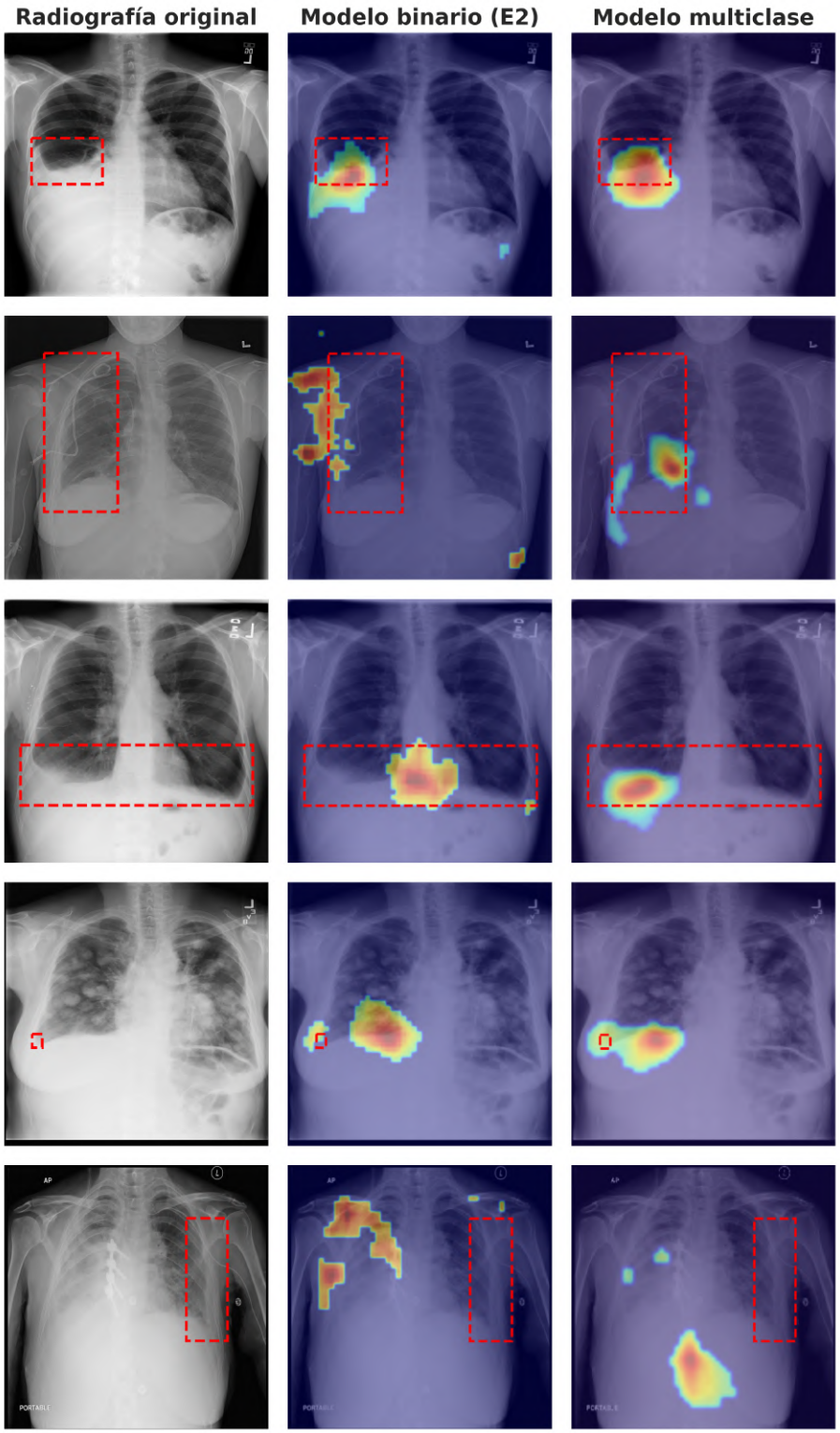


Figura 4.15: Comparación visual de los resultados obtenidos por el filtro propuesto aplicado a las radiografías de prueba correspondientes a la patología *Derrame pleural*. Se muestran, de izquierda a derecha: la radiografía original con su cuadro delimitador, el mapa generado por el modelo binario (enfoque 2) y el mapa generado por el modelo multiclase.

En general, los mapas de activación producidos por los modelos multiclase presentan una apariencia más suave y menos dispersa en comparación con los obtenidos mediante los modelos binarios. Aunque no se observa una mejora sustancial en la localización exacta de las regiones de interés, las activaciones generadas por los modelos multiclase tienden a concentrarse con mayor coherencia en las zonas delimitadas, reduciendo la presencia de activaciones aisladas o ruidosas.

Cuando las activaciones coinciden con los cuadros delimitadores, ambos enfoques en general, resaltan correctamente las regiones relevantes. Sin embargo, los mapas del modelo binario suelen mostrar una respuesta más extendida, mientras que los del modelo multiclase concentran la activación en áreas más definidas y continuas. Esta diferencia sugiere que el entrenamiento multiclase promueve representaciones internas más consistentes, lo que se refleja en una distribución de activaciones más ordenada y focalizada.

En conjunto, los resultados visuales indican que el modelo multiclase contribuye a generar mapas de activación con una mejor calidad visual, caracterizados por una menor dispersión y una mayor correspondencia con las regiones señaladas en las anotaciones manuales. Si bien las mejoras no son drásticas, la mayor coherencia observada en los mapas producidos respalda la utilidad del enfoque multiclase como base para el análisis interpretativo del filtro propuesto.

En síntesis, los resultados obtenidos en esta sección, permiten concluir que la calidad y precisión de los mapas de activación dependen en gran medida del modelo utilizado para generarlos. Las discrepancias observadas entre las regiones resaltadas por el filtro propuesto y las áreas delimitadas manualmente no se deben necesariamente al funcionamiento del filtro, sino al grado de especificidad de las representaciones aprendidas por cada modelo. En efecto, cuando las características internas del clasificador son demasiado generales, los mapas tienden a presentar activaciones dispersas y ruidosas, mientras que representaciones más discriminativas conducen a mapas más coherentes y focalizados sobre las zonas relevantes.

Con el fin de validar esta hipótesis y evaluar la coherencia de los resultados obtenidos, en la siguiente sección se realiza una comparación directa entre los mapas de calor generados por el filtro propuesto y aquellos producidos mediante el método Grad-CAM. Si ambos métodos reflejan un comportamiento similar en cuanto a la localización y extensión de las activaciones, se confirmaría que las diferencias observadas se originan principalmente en las representaciones internas del modelo y no en la metodología de filtrado, respaldando así la correcta funcionalidad del filtro propuesto.

## 4.8. Evaluación comparativa con Grad-CAM

Con el propósito de validar la consistencia de los resultados obtenidos y analizar la calidad de los mapas de activación generados por el filtro propuesto, en esta sección se realiza una comparación directa con los mapas obtenidos mediante el método Grad-CAM. Este análisis busca determinar si las diferencias observadas en las secciones previas se deben al funcionamiento del filtro o, por el contrario, a las representaciones internas aprendidas por los modelos multiclase.

Siguiendo un procedimiento análogo al aplicado con los modelos binarios, se comenzó evaluando los clasificadores multiclase entrenados para *Atelectasia* y *Derrame pleural* sobre los subconjuntos de radiografías con cuadros delimitadores disponibles, con el fin de determinar cuántas de ellas fueron clasificadas correctamente por cada modelo. En la Tabla 4.12 se resumen estos resultados, mostrando el número total de imágenes con anotaciones (*bounding boxes*) disponibles por enfermedad, la sensibilidad alcanzada por los modelos al evaluarse sobre dicho subconjunto y el número de radiografías clasificadas correctamente.

A partir de estos resultados, se seleccionaron aleatoriamente 10 radiografías correctamente clasificadas por cada modelo, y para cada una de ellas se generó un mapa de calor correspondiente a los dos métodos de interpretabilidad considerados. Esto permitió realizar una comparación visual directa entre los mapas generados por el filtro propuesto y aquellos obtenidos mediante Grad-CAM.

Para la aplicación del filtro propuesto, se mantuvieron los parámetros de muestreo definidos en los experimentos realizados sobre *ImageNet* (**grid_size = 15** y **stride = 5**), al igual que en la sección anterior. Por su parte, los mapas de calor de Grad-CAM se generaron siguiendo la implementación descrita en el trabajo original, tal como se realizó al final del capítulo anterior.

| **Parámetro** | **Atelectasia** | **Derrame pleural** |
|---|---|---|
| N. Imágenes con BBox | 180 | 153 |
| Sensibilidad (modelo multiclase) | 0.4722 | 0.8301 |
| N. Imágenes clasificadas correctamente | 85 | 127 |

Tabla 4.12: Resumen de las imágenes con cuadros delimitadores disponibles por enfermedad, la sensibilidad obtenida por los modelos multiclase y el número de imágenes correctamente clasificadas en cada caso.

### 4.8.1. Análisis comparativo

En las Figuras 4.16 y 4.17 se presentan los resultados obtenidos al aplicar el filtro propuesto y el método Grad-CAM sobre los modelos multiclase entrenados para las patologías *Atelectasia* y *Derrame pleural*, respectivamente. En cada figura se muestran, para un conjunto de cinco radiografías seleccionadas aleatoriamente, la imagen original con sus cuadros delimitadores, el mapa de calor generado por el filtro propuesto y el obtenido mediante Grad-CAM.

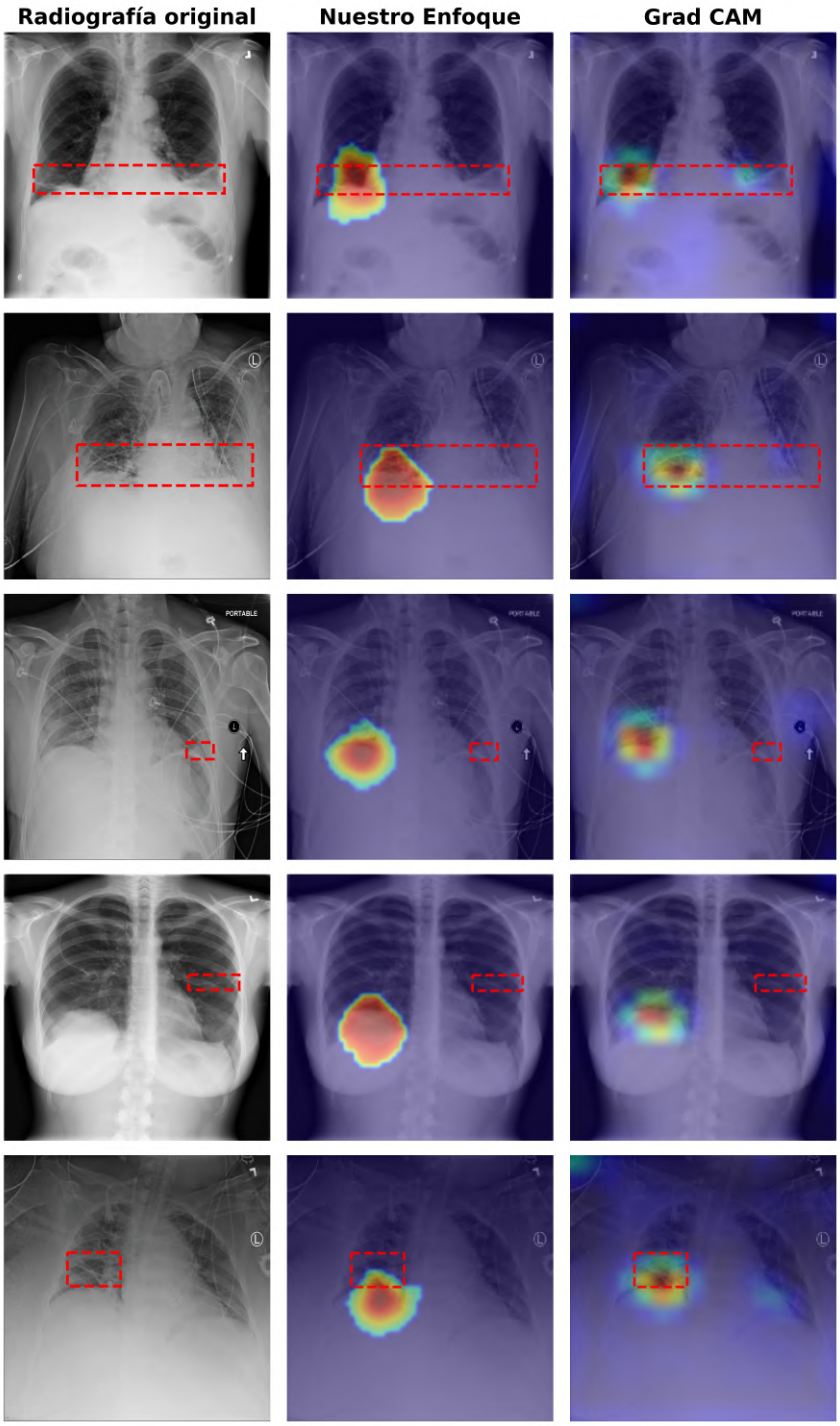


Figura 4.16: Comparación visual entre los mapas de activación generados por el filtro propuesto y por Grad-CAM para el modelo multiclase entrenado en la detección de *Atelectasia*. En cada fila se muestra la radiografía original con su cuadro delimitador, seguida por los mapas obtenidos mediante ambos enfoques. Las imágenes corresponden a cinco casos seleccionados aleatoriamente del conjunto de radiografías correctamente clasificadas.

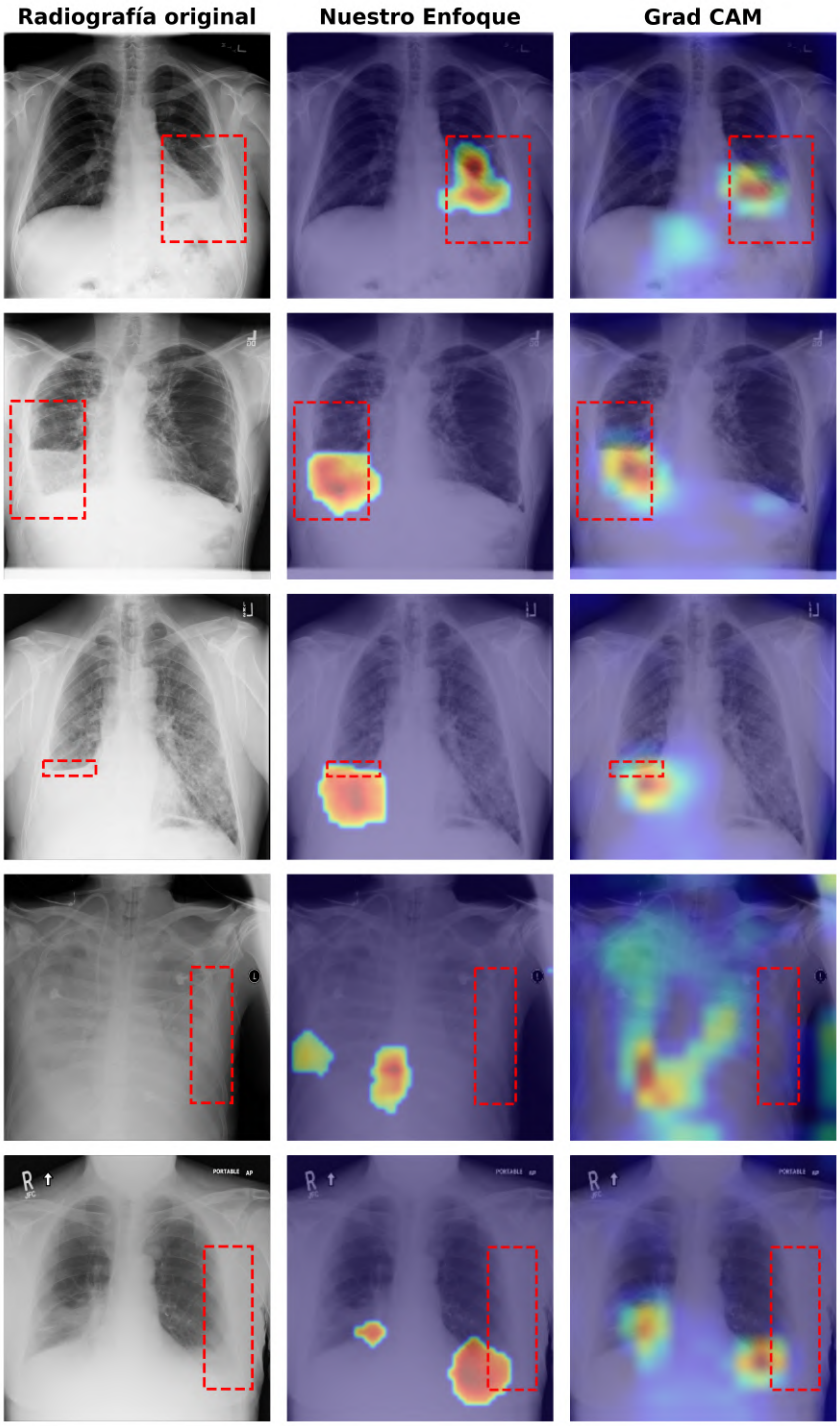


Figura 4.17: Comparación visual entre los mapas de activación generados por el filtro propuesto y por Grad-CAM para el modelo multiclase entrenado en la detección de *Derrame pleural*. En cada fila se muestra la radiografía original con su cuadro delimitador, seguida por los mapas obtenidos mediante ambos enfoques. Las imágenes corresponden a cinco casos seleccionados aleatoriamente del conjunto de radiografías correctamente clasificadas.

De manera general, se observa que ambos enfoques producen mapas de calor de buena calidad, con activaciones suaves y sin presencia significativa de regiones aisladas o ruidosas. En la mayoría de los casos, las activaciones generadas por ambos métodos resaltan zonas anatómicas similares, lo que indica una alta consistencia en la identificación de las regiones más relevantes para la decisión del modelo. Este comportamiento sugiere que tanto el filtro propuesto como Grad-CAM capturan las mismas características discriminativas aprendidas por los modelos multiclase.

Asimismo, se evidencia que cuando las activaciones coinciden con las áreas delimitadas por los cuadros manuales, ambos métodos lo hacen de manera simultánea; mientras que, cuando las activaciones caen fuera de dichas regiones, ambos tienden a presentar la misma desviación. Esto refuerza la idea de que las regiones resaltadas corresponden efectivamente a las representaciones internas utilizadas por el modelo para clasificar cada imagen, aunque no siempre estas regiones coincidan con las zonas anatómicas donde se localiza la patología. En otras palabras, el modelo puede basar su decisión en patrones visuales indirectos, pero relevantes según su propio proceso de aprendizaje.

Por otro lado, si bien los mapas obtenidos mediante Grad-CAM suelen presentar activaciones ligeramente más suaves y concentradas, también muestran en algunos casos comportamientos erráticos, con regiones activadas que se extienden más allá de las zonas de interés o que abarcan grandes áreas de la radiografía. En contraste, las activaciones producidas por el filtro propuesto tienden a mantenerse más estables y menos dispersas, lo que podría atribuirse al muestreo exhaustivo que realiza el filtro sobre toda la imagen y que lo hace menos sensible al ruido local presente en los datos de entrada.

En conjunto, los resultados muestran una correspondencia significativa entre los mapas generados por ambos enfoques, lo que confirma que la calidad de los mapas obtenidos con el filtro propuesto depende principalmente de las representaciones internas aprendidas por el modelo y no del método de visualización en sí. Este hallazgo respalda la validez del filtro propuesto como una alternativa efectiva y coherente con Grad-CAM para la interpretación visual de redes neuronales convolucionales y su potencial uso en el área clínica.

# Capítulo 5

# Discusión y Conclusiones

El objetivo principal de este trabajo fue proponer una nueva técnica para la visualización e interpretación de los procesos de aprendizaje en Redes Neuronales Convolucionales (RNC). A lo largo del desarrollo de la tesis, se diseñó e implementó un método basado en la sensibilidad de oclusión que, tras una serie de mejoras, logró convertirse en una alternativa funcional y eficiente frente a técnicas consolidadas como Grad-CAM. El filtro propuesto demostró ser capaz de generar mapas de activación con una calidad comparable a los obtenidos con Grad-CAM, conservando una interpretación coherente de las regiones relevantes empleadas por la red en el proceso de clasificación.

Durante el proceso de investigación, el filtro fue sometido a diversos escenarios experimentales, desde pruebas controladas en conjuntos de datos genéricos hasta su aplicación final en imágenes médicas de la base de datos *ChestX-ray14*. En este último entorno, el método evidenció su potencial de uso en contextos clínicos, permitiendo resaltar las regiones anatómicas más relevantes asociadas con las predicciones realizadas por los modelos. Estos resultados confirman la versatilidad del filtro, así como su aplicabilidad en dominios donde la interpretabilidad y la transparencia de los modelos son factores críticos.

Un hallazgo importante derivado de este trabajo es que el filtro propuesto no solo puede emplearse como herramienta de visualización e interpretación de redes neuronales, sino también como un método de identificación de objetos de tipo semisupervisado. A partir de las predicciones de una RNC, el filtro es capaz de localizar las regiones correspondientes a la clase objetivo dentro de la imagen, lo que abre la posibilidad de extender su uso a distintas áreas, desde la localización de objetos en visión por computadora hasta la identificación de posibles patrones patológicos en radiografías médicas.

Por otro lado, los resultados obtenidos también ponen en evidencia la relevancia de continuar investigando y mejorando los métodos de interpretabilidad en modelos de aprendizaje profundo. A lo largo de los experimentos se observó que la calidad y la precisión de los mapas de activación dependen en gran medida de las representaciones internas aprendidas por el modelo subyacente. Esto implica que un modelo puede alcanzar métricas cuantitativas satisfactorias, como una alta precisión o sensibilidad, pero basar sus decisiones en características que no corresponden a las regiones relevantes de la clase objetivo. Este fenómeno refuerza la necesidad de evaluar la confianza y validez de los modelos no solo a través de métricas globales de desempeño, sino también mediante un análisis cualitativo de las regiones que sustentan sus predicciones, especialmente en contextos médicos donde la interpretación clínica es esencial.

Entre las principales limitaciones encontradas durante el desarrollo de esta tesis se destacan la disponibilidad y accesibilidad de bases de datos médicas de calidad, el alto costo computacional asociado al procesamiento de imágenes de alta resolución y la falta de modelos preentrenados específicos para el dominio médico. Aunque existen numerosos trabajos en el estado del arte que reportan arquitecturas y resultados sobresalientes, en la mayoría de los casos no se proporciona el código fuente ni los pesos de los modelos, lo que dificulta su replicación y comparación objetiva. Estas limitaciones subrayan la importancia de promover la apertura de datos y modelos en el campo de la inteligencia artificial aplicada a la medicina.

Finalmente, los resultados obtenidos permiten concluir que el filtro propuesto constituye una contribución significativa en el ámbito de la interpretabilidad de redes neuronales, al ofrecer una alternativa práctica, eficiente y de fácil implementación para la generación de mapas de activación. Aunque el método aún puede beneficiarse de mejoras en su capacidad de localización y reducción de ruido, los resultados demuestran su validez y potencial de aplicación en problemas reales. Asimismo, este trabajo abre nuevas líneas de investigación orientadas a optimizar la integración entre modelos clasificadores e interpretabilidad, con el fin de avanzar hacia sistemas de inteligencia artificial más confiables, explicables y éticamente responsables.

# Bibliografía


[1] Chan, H.-P., Samala, R. K., Hadjiiski, L. M., & Zhou, C. (2020). Deep Learning in Medical Image Analysis. En G. Lee & H. Fujita (Eds.), Deep Learning in Medical Image Analysis (Avances en Medicina Experimental y Biología, Vol. 1213). Springer Nature Switzerland AG.

[2] José Gerardo Suárez-García, Javier Miguel Hernández-López, Eduardo Moreno-Barbosa, Benito de Celis-Alonso, .ª simple model for glioma grading based on texture analysis applied to conventional brain MRI", PLoS ONE 15(5): e0228972. https://doi.org/10.1371/journal.pone.0228972

[3] Mehmood, A., Yang, S., Feng, Z., Wang, M., Ahmad, A. S., Khan, R., Maqsood, M., & Yaqub, M. (2021). A Transfer Learning Approach for Early Diagnosis of Alzheimer's Disease on MRI Images. Neuroscience, 460, 43-52. https://doi.org/10.1016/j.neuroscience.2021.01.002.

[4] Chen, X., Wang, X., Zhang, K., Fung, K.-M., Thai, T. C., Moore, K., Mannel, R. S., Liu, H., Zheng, B., & Qiu, Y. (2022). Recent advances and clinical applications of deep learning in medical image analysis. Medical Image Analysis, 79, 102444. https://doi.org/10.1016/j.media.2022.102444.

[5] Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Lecture Notes in Computer Science Computer Vision – European Conference on Computer Vision (ECCV) 2014, vol 8689, pp 818–833.

[6] Yosinski J, Clune J, Nguyen A, Fuchs T, Lipson H (2015) Understanding neural networks through deep visualization. arXiv:1506.06579v1.

[7] Samala RK, Chan H-P, Hadjiiski LM, Helvie MA, Cha KH, Richter CD (2017) Multi-task transfer learning deep convolutional neural network: application to computer-aided diagnosis of breast cancer on mammograms. Phys Med Biol 62:8894–8908.

[8] Luetkens, J.A., Nowak, S., Mesropyan, N. et al. Deep learning supports the differentiation of alcoholic and other-than-alcoholic cirrhosis based on MRI. Sci Rep 12, 8297 (2022). https://doi.org/10.1038/s41598-022-12410-2.

[9] Géron, A. (2017). Hands-On Machine Learning with Scikit-Learn and TensorFlow: Techniques and Tools to Build Learning Machines. O'Reilly Media.

[10] Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 618–626. https://doi.org/10.1109/ICCV.2017.74

[11] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision, 115(3), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[12] Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (pp. 1251–1258). https://doi.org/10.1109/CVPR.2017.195

[13] Papers with Code. (s.f.). *ImageNet Dataset*. Recuperado el 8 de julio de 2025, de https://paperswithcode.com/dataset/imagenet

[14] Zeiler, M. D., & Fergus, R. (2014). Visualizing and Understanding Convolutional Networks. En D. Fleet, T. Pajdla, B. Schiele, & T. Tuytelaars (Eds.), *European Conference on Computer Vision (ECCV), Lecture Notes in Computer Science* (vol. 8689, pp. 818–833). Springer. https://doi.org/10.1007/978-3-319-10590-1_53

[15] Wang, X., Peng, Y., Lu, L., Lu, Z., Bagheri, M., & Summers, R. M. (2017). ChestX-ray8: Hospital-scale Chest X-ray Database and Benchmarks on Weakly-Supervised Classification and Localization of Common Thorax Diseases. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2097–2106. https://doi.org/10.1109/CVPR.2017.369.

[16] Rajpurkar, P., Irvin, J., Ball, R. L., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C. P., Patel, B. N., Yeom, K. W., Shpanskaya, K., Blankenberg, F. G., Seekins, J., Amrhein, T. J., Mong, D. A., Halabi, S. S., Zucker, E. J., Ng, A. Y., & Lungren, M. P. (2018). Deep learning for chest radiograph diagnosis: A retrospective comparison of the CheXNeXt algorithm to practicing radiologists. PLOS Medicine, 15(11), e1002686. https://doi.org/10.1371/journal.pmed.1002686

[17] Rajpurkar, P., Irvin, J., Zhu, K., Yang, B., Mehta, H., Duan, T., Ding, D., Bagul, A., Langlotz, C. P., Shpanskaya, K., Lungren, M. P., & Ng, A. Y. (2017). CheXNet: Radiologist-Level Pneumonia Detection on Chest X-Rays with Deep Learning. arXiv preprint arXiv:1711.05225.

[18] Chou, B. (2017). *CheXNet-Keras*. GitHub repository. Disponible en: `https://github.com/brucechou1983/CheXNet-Keras`.

[19] Pitumbur, A. (2023). *Xray Xplorer: CNN, XGBoost y Grad-CAM para la detección de COVID-19 en radiografías de tórax mediante algoritmos de amplificación explicables*. Repositorio en GitHub. Disponible en: `https://github.com/Abhijeet-Pitumbur/Xray-Xplorer`.