



UNIVERSIDAD DE EXTREMADURA

ESCUELA POLITÉCNICA

INGENIERÍA INFORMÁTICA EN INGENIERÍA DEL SOFTWARE

TRABAJO FIN DE GRADO

**Predicción de tumores en imágenes radiológicas mediante *Deep Learning*  
con redes neuronales *U-net***

Miguel Jara Arroyo

Septiembre, 2024





UNIVERSIDAD DE EXTREMADURA

ESCUELA POLITÉCNICA

INGENIERÍA INFORMÁTICA EN INGENIERÍA DEL SOFTWARE

TRABAJO FIN DE GRADO

**Predicción de tumores en imágenes radiológicas mediante *Deep Learning*  
con redes neuronales *U-net***

Autor: Miguel Jara Arroyo

Tutor: Félix Rodríguez Rodríguez



# Agradecimientos

Quiero expresar mi más sincero agradecimiento a todas las personas e instituciones que han contribuido a la realización de este proyecto. En primer lugar, agradezco a mi tutor, Félix Rodríguez Rodríguez, por su orientación, apoyo y paciencia a lo largo de este proceso. A mis compañeros de clase y amigos, por su constante ánimo y colaboración. A mi familia, por su amor incondicional y su apoyo inquebrantable en cada etapa de mi vida. También quiero agradecer a la Universidad de Extremadura por proporcionar los recursos y el entorno necesarios para llevar a cabo esta investigación. Finalmente, agradezco a la comunidad de *The Cancer Imaging Archive* (TCIA) por el acceso a los datos utilizados en este estudio.



# Resumen

Usando como fuente de datos mamografías digitales y tomisíntesis digitales de mama, este trabajo de fin de grado se adentrará en el mundo del *Deep Learning* y aprendizaje automático, una de las vanguardias más rompedoras de la informática en los últimos años, realizando un análisis del uso de estas tecnologías para dar solución al problema de detección y categorización del cáncer de mama. Esta investigación busca representar cómo las técnicas de *Deep Learning* pueden y deben utilizarse para solucionar los problemas que más afectan a nuestra sociedad, desprendiéndose de la banalidad de las redes sociales.

En el siguiente documento se presentarán cuatro arquitecturas de redes neuronales diferentes, las cuales tendrán la finalidad de clasificar en casos de cáncer, tumores benignos o pacientes sanos un conjunto de datos de mamografías digitales. Se hará uso de técnicas como el *transfer learning*, el aumento de datos, la validación cruzada, la creación de funciones de pérdida personalizadas y *dropout* entre muchas otras para obtener el mejor resultado posible en esta tarea.

Finalmente, veremos las conclusiones de cada modelo, viendo como dos de estas arquitecturas logran superar el rendimiento de todos los antecedentes propuestos, entre los cuales se encuentran algunos de los sistemas CAD utilizados en la detección del cáncer por radiólogos.

*Palabras clave:* detección del cáncer de mama, *Deep Learning*, aprendizaje automático.



# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Contexto . . . . .	1
1.2. Objetivos . . . . .	3
1.3. Planificación . . . . .	3
1.4. Organización de contenidos . . . . .	6
<b>2. Fundamentos médicos y tecnológicos del cáncer de mama</b>	<b>11</b>
2.1. Estado actual del cáncer . . . . .	11
2.2. Estado de la detección del cáncer . . . . .	12
2.2.1. Características del cáncer determinado mediante exámenes de detección . . . . .	12
2.2.2. Concepto de sesgos en los exámenes de detección . . . . .	13
2.2.3. Variables relacionadas con la precisión . . . . .	14
2.3. Mamografía digital . . . . .	16
2.4. Tomosíntesis digital de mama . . . . .	17
2.4.1. Ejemplos de aplicación . . . . .	19
<b>3. Antecedentes</b>	<b>21</b>
3.1. Contextualización . . . . .	21
3.2. Sistemas CAD . . . . .	22
<b>4. Herramientas <i>software</i> y <i>hardware</i></b>	<b>27</b>
4.1. <i>Hardware</i> . . . . .	27

## ÍNDICE GENERAL

---

4.1.1.	Unidad de Procesamiento Gráfico, <i>GPU</i> . . . . .	27
4.1.2.	Almacenamiento . . . . .	28
4.2.	<i>Software</i> . . . . .	29
4.2.1.	<i>Anaconda</i> . . . . .	29
4.2.2.	<i>CUDA</i> . . . . .	30
4.2.3.	<i>cuDNN</i> . . . . .	30
4.2.4.	<i>TensorFlow</i> . . . . .	30
<b>5.</b>	<b>Análisis de los datos</b>	<b>33</b>
5.1.	Fuentes de datos . . . . .	33
5.1.1.	Fuente de datos <i>Breast Cancer Screening (DBT)</i> . . . . .	34
5.1.2.	<i>The Chinese Mammographies Database (CMMMD)</i> . . . . .	40
<b>6.</b>	<b>Procesado de datos</b>	<b>45</b>
6.1.	Detalles sobre la recolección de datos . . . . .	45
6.1.1.	Formato de los datos . . . . .	46
6.2.	Preprocesamiento de datos . . . . .	47
6.2.1.	<i>Breast Cancer Screening DBT</i> . . . . .	47
6.2.2.	El conjunto de datos <i>CMMMD</i> . . . . .	49
6.2.3.	Unión de los dos conjuntos de datos . . . . .	52
6.3.	Conjuntos utilizados en el entrenamiento de modelos . . . . .	54
6.3.1.	Conjunto de datos base . . . . .	55
6.3.2.	Conjunto de datos con aumento de datos . . . . .	58
<b>7.</b>	<b>Desarrollo del proyecto</b>	<b>63</b>
7.1.	Modelos de referencia iniciales . . . . .	63
7.1.1.	Modelo de referencia <i>Breast Cancer Screening (DBT)</i> . . . . .	64
7.1.2.	Modelo de referencia <i>Inception V3</i> inicial . . . . .	65
7.2.	Aspectos comunes de los modelos implementados . . . . .	69
7.2.1.	Función de pérdida ponderada . . . . .	69
7.2.2.	Técnica de validación cruzada . . . . .	70

## ÍNDICE GENERAL

---

7.3.	Red <i>UNet3D-Densa</i> . . . . .	72
7.3.1.	Aspectos teóricos de las redes <i>U-Net</i> . . . . .	73
7.3.2.	Aplicación de <i>U-Net</i> en la clasificación de radiografías de mama	77
7.3.3.	Implementación del modelo <i>UNet3D-Densa</i> . . . . .	78
7.4.	Red <i>UNet-Paralela</i> . . . . .	84
7.5.	Red <i>Inceptionv3-Paralela</i> . . . . .	92
7.5.1.	Origen y evolución de <i>Inception</i> . . . . .	92
7.5.2.	Detalles técnicos de <i>Inception V3</i> . . . . .	93
7.5.3.	Aplicación de <i>Inception V3</i> en la clasificación de radiografías de mama . . . . .	94
7.5.4.	Implementación del modelo <i>Inceptionv3-Paralela</i> . . . . .	95
7.5.5.	Compilación del modelo <i>Inceptionv3-Paralela</i> . . . . .	98
7.6.	Red <i>ResNetv2-Paralela</i> . . . . .	99
7.6.1.	Origen y evolución de <i>ResNet V2</i> . . . . .	99
7.6.2.	Detalles técnicos de <i>Inception-ResNet V2</i> . . . . .	100
7.6.3.	Implementación del modelo <i>ResNetv2-Paralela</i> . . . . .	104
<b>8.</b>	<b>Resultados</b>	<b>107</b>
8.1.	Red UNet3D-Dense . . . . .	107
8.1.1.	Conjunto de datos base . . . . .	108
8.1.2.	Conjunto de datos con aumento de datos . . . . .	115
8.2.	Red UNet-Paralela . . . . .	123
8.2.1.	Conjunto de datos base . . . . .	123
8.2.2.	Conjunto de datos con aumento de datos . . . . .	130
8.3.	Red Inceptionv3-Paralela . . . . .	138
8.3.1.	Conjunto de datos base . . . . .	138
8.3.2.	Conjunto de datos con aumento de datos . . . . .	147
8.4.	Red ResNetv2-Paralela . . . . .	156
8.4.1.	Conjunto de datos base . . . . .	156
8.4.2.	Conjunto de datos con aumento de datos . . . . .	164



## ÍNDICE GENERAL

---

8.5. Resumen de resultados . . . . .	172
<b>9. Conclusiones</b>	<b>175</b>
9.1. Conclusiones del proyecto . . . . .	175
9.1.1. Comparación con el modelo de referencia . . . . .	176
9.1.2. Comparativa con los sistemas CADe . . . . .	177
9.2. Conclusiones Personales . . . . .	178
9.3. Desarrollos futuros . . . . .	180
9.3.1. Aumento de la cantidad de datos . . . . .	180
9.3.2. Posibilidades de optimización futura . . . . .	181
9.3.3. Segmentación de imágenes . . . . .	181
<b>Anexos</b>	<b>183</b>
<b>A. Contenido teórico</b>	<b>185</b>
A.1. Matrices de confusión . . . . .	185
<b>B. Código</b>	<b>189</b>
B.1. Weighted categorical crossentropy . . . . .	189
B.2. Validación cruzada . . . . .	190
<b>Bibliografía y referencias</b>	<b>193</b>

# Índice de tablas

2.1. Comparación de diferentes sistemas de mamografía . . . . .	18
3.1. Estadísticas de rendimiento de sistemas CADe seleccionados para la detección de anomalías . . . . .	24
3.2. Estadísticas de rendimiento de sistemas CADx seleccionados para la clasificación de masas . . . . .	25
5.1. Primeras cuatro filas de <i>BCS-DBT-file-paths-subcolección-v2.csv</i> . . . . .	37
5.2. Primeras cinco filas de <i>BCS-DBT-labels-subcolección-v2.csv</i> . . . . .	38
5.3. Primeras cinco filas del archivo CSV proporcionado - Parte 1. . . . .	39
5.4. Primeras cinco filas del archivo CSV proporcionado - Parte 2. . . . .	39
5.5. Primeras cinco filas del archivo XLSX. . . . .	42
8.1. Resultados de los modelos evaluados en el conjunto de datos de prueba. . . . .	111
8.2. Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase <i>actionable</i> . . . . .	113
8.3. Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	114
8.4. Resultados de los modelos DA evaluados en el conjunto de datos de prueba. . . . .	119
8.5. Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase <i>actionable</i> . . . . .	121

## ÍNDICE DE TABLAS

---

8.6. Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	121
8.7. Resultados de los modelos evaluados en el conjunto de datos de prueba. . . . .	126
8.8. Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase <i>actionable</i> . . . . .	128
8.9. Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	129
8.10. Resultados de los modelos DA evaluados en el conjunto de datos de prueba. . . . .	134
8.11. Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase <i>actionable</i> . . . . .	136
8.12. Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	136
8.13. Resultados de los modelos evaluados en el conjunto de datos de prueba. . . . .	143
8.14. Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase <i>actionable</i> . . . . .	145
8.15. Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	146
8.16. Resultados de los modelos DA evaluados en el conjunto de datos de prueba. . . . .	152
8.17. Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase <i>actionable</i> . . . . .	154
8.18. Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	154
8.19. Resultados de los modelos evaluados en el conjunto de datos de prueba. . . . .	160

## ÍNDICE DE TABLAS

---

8.20. Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase <i>actionable</i> . . . . .	162
8.21. Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	163
8.22. Resultados de los modelos evaluados en el conjunto de datos de prueba. . . . .	168
8.23. Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase <i>actionable</i> . . . . .	170
8.24. Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	171
8.25. Comparación de desempeño de las diferentes redes neuronales con los dos conjuntos de datos, con la precisión de la clase <i>normal</i> y la precisión en las clases <i>benign</i> y <i>cancer</i> . . . . .	173



# Índice de figuras

1.1.	Cronología para llevar a cabo el trabajo desarrollado. . . . .	4
1.2.	Flujo de trabajo con las etapas clave de cada uno de los modelos implementados. . . . .	8
2.1.	Diferentes geometrías de sistemas clínicos de DBT . . . . .	18
2.2.	RCC . . . . .	20
2.3.	RMLO . . . . .	20
2.4.	Falso positivo identificado por DBT . . . . .	20
2.5.	Tumor detectado por DBT. . . . .	20
4.1.	NVIDIA GeForce RTX 3070 Ti . . . . .	28
5.1.	Volumen DBT-P00002 LCC, el cual cuenta con 70 rodajas . . . . .	35
5.2.	Diagrama circular con la distribución de casos para el conjunto <i>Breast Cancer Screening DBT</i> . . . . .	36
5.3.	<i>DBT-P00431 LCC Box</i> . . . . .	37
5.4.	Distribución de casos en el conjunto <i>CMMMD</i> . . . . .	41
5.5.	Caso <i>malignant</i> del conjunto <i>CMMMD</i> . . . . .	42
6.1.	Ejemplo de un paciente con cáncer identificado con DBT-S00029_DBT-P00659. . . . .	49
6.2.	Ejemplo de un paciente con cáncer del conjunto <i>CMMMD</i> . . . . .	51
6.3.	Casos combinados de <i>CBIS-CMMMD</i> y <i>Breast Cancer Screening DBT</i> . . . . .	53
6.4.	Distribución de los datos en el conjunto con validación cruzada. . . . .	56

## ÍNDICE DE FIGURAS

---

6.5. División de los datos entre la fase de entrenamiento y de <i>test</i> . . . . .	57
7.1. Valores de la pérdida ( <i>validation loss</i> ) y <i>accuracy</i> ( <i>validation accuracy</i> ) obtenidos durante el entrenamiento para el modelo <i>Inception V3</i> . . . . .	67
7.2. Matriz de confusión con los datos de <i>test</i> para el modelo <i>Inception V3</i> . . . . .	68
7.3. Validación cruzada en $K = 5$ fases. . . . .	71
7.4. Esquema de una arquitectura <i>U-Net</i> . . . . .	73
7.5. Ejemplo de codificador en una <i>U-Net</i> . . . . .	74
7.6. Ejemplo de <i>bottleneck</i> en una <i>U-Net</i> . . . . .	75
7.7. Ejemplo de decodificador en una <i>U-Net</i> . . . . .	76
7.8. Camino de codificación en el modelo <i>U-Net</i> implementado. . . . .	79
7.9. <i>Bottleneck</i> en el modelo <i>U-Net</i> implementado. . . . .	80
7.10. Camino de expansión en el modelo <i>U-Net</i> implementado. . . . .	81
7.11. Capas para clasificar los resultados obtenidos en la <i>U-Net</i> implementada. . . . .	83
7.12. Capas encargadas de la obtención y división de los diferentes <i>arrays</i> . . . . .	84
7.13. Capas que componen un módulo del camino de codificación de cada una de las cuatro redes <i>U-Net</i> del modelo <i>UNet-Paralela</i> . . . . .	85
7.14. Capas que componen el <i>bottleneck</i> de cada una de las cuatro redes <i>U-Net</i> del modelo <i>UNet-Paralela</i> . . . . .	87
7.15. Capas que componen un módulo del camino de expansión de cada una de las cuatro redes <i>U-Net</i> del modelo <i>UNet-Paralela</i> . . . . .	88
7.16. Capas que se encargan de reducir las dimensiones del mapa de características de las cuatro redes <i>U-Net</i> del modelo <i>UNet-Paralela</i> . . . . .	90
7.17. Capas que se encargan de combinar la salida de las cuatro redes <i>U-Net</i> y realizar la clasificación de los distintos casos del modelo <i>UNet-Paralela</i> . . . . .	91
7.18. Ejemplo de factorización de una convolución de $5 \times 5$ en dos convoluciones de $3 \times 3$ . . . . .	93

## ÍNDICE DE FIGURAS

---

7.19. Estructura de un módulo <i>Inception</i> con diferentes tamaños de filtros operando en paralelo. . . . .	94
7.20. Capas <i>Lambda</i> en el modelo <i>Inceptionv3-Paralela</i> implementado. . . . .	95
7.21. Capas <i>GlobalAveragePooling2D</i> , <i>Concatenate</i> , <i>Reshape</i> y <i>Conv2D</i> en el modelo <i>Inceptionv3-Paralela</i> implementado. . . . .	96
7.22. Últimas capas del modelo <i>Inceptionv3-Paralela</i> implementado. . . . .	97
7.23. Estructura de la arquitectura <i>Inception-ResNet V2</i> . . . . .	101
7.24. Estructura de la capa <i>Inception-ResNet-A</i> . . . . .	103
7.25. Capas modificadas en el modelo <i>ResNetv2-Paralela</i> implementado con respecto al modelo <i>Inception V3</i> . . . . .	105
 8.1. Pérdida durante el entrenamiento. . . . .	108
8.2. <i>Accuracy</i> durante el entrenamiento. . . . .	109
8.3. Pérdida de los diferentes modelos. . . . .	110
8.4. <i>Accuracy</i> de los diferentes modelos. . . . .	111
8.5. Matriz de confusión del modelo “best_model5.keras”. . . . .	112
8.6. Matriz de confusión del modelo “best_model5.keras” sin la categoría <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	115
8.7. Pérdida durante el entrenamiento. . . . .	116
8.8. <i>Accuracy</i> durante el entrenamiento. . . . .	117
8.9. Pérdida de los diferentes modelos. . . . .	118
8.10. <i>Accuracy</i> de los diferentes modelos. . . . .	118
8.11. Matriz de confusión del modelo “best_model1_20.keras”. . . . .	120
8.12. Matriz de confusión del modelo “model_fold_2_20_epochs.keras” sin la categoría <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	122
8.13. Pérdida durante el entrenamiento . . . . .	123
8.14. <i>Accuracy</i> durante el entrenamiento . . . . .	124
8.15. Pérdida de los diferentes modelos. . . . .	125
8.16. <i>Accuracy</i> de los diferentes modelos. . . . .	125
8.17. Matriz de confusión del modelo “model_fold_5_20_epochs.keras”. . . . .	127

## ÍNDICE DE FIGURAS

---

8.18. Matriz de confusión del modelo “model_fold_5_20_epochs.keras” sin la categoría <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	130
8.19. Pérdida durante el entrenamiento. . . . .	131
8.20. <i>Accuracy</i> durante el entrenamiento. . . . .	132
8.21. Pérdida de los diferentes modelos. . . . .	133
8.22. <i>Accuracy</i> de los diferentes modelos. . . . .	134
8.23. Matriz de confusión del modelo “model_fold_2_20_epochs.keras”. . . . .	135
8.24. Matriz de confusión del modelo “model_fold_2_20_epochs.keras” sin la categoría <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	137
8.25. Pérdida durante el entrenamiento. . . . .	139
8.26. <i>Accuracy</i> durante el entrenamiento. . . . .	140
8.27. Precisión durante el entrenamiento (primeros dos <i>folds</i> ). . . . .	141
8.28. Precisión durante el entrenamiento (últimos tres <i>folds</i> ). . . . .	141
8.29. Pérdida de los diferentes modelos. . . . .	142
8.30. <i>Accuracy</i> de los diferentes modelos. . . . .	143
8.31. Matriz de confusión del modelo “best_model3.keras”. . . . .	144
8.32. Matriz de confusión del modelo “best_model3.keras” sin la categoría <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	147
8.33. Pérdida durante el entrenamiento. . . . .	148
8.34. <i>Accuracy</i> durante el entrenamiento. . . . .	149
8.35. Precisión durante el entrenamiento. . . . .	150
8.36. Pérdida de los diferentes modelos. . . . .	151
8.37. <i>Accuracy</i> de los diferentes modelos. . . . .	151
8.38. Matriz de confusión del modelo “best_model3.keras”. . . . .	153
8.39. Matriz de confusión del modelo “best_model3.keras” sin la categoría <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	155
8.40. Pérdida durante el entrenamiento . . . . .	157
8.41. <i>Accuracy</i> durante el entrenamiento . . . . .	158
8.42. Precisión durante el entrenamiento . . . . .	158

## ÍNDICE DE FIGURAS

---

8.43. Pérdida de los diferentes modelos . . . . .	159
8.44. <i>Accuracy</i> de los diferentes modelos . . . . .	160
8.45. Matriz de confusión del modelo “best_model5.keras”. . . . .	161
8.46. Matriz de confusión del modelo “best_model5.keras” sin la categoría <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	164
8.47. Pérdida durante el entrenamiento. . . . .	165
8.48. <i>Accuracy</i> durante el entrenamiento. . . . .	166
8.49. Precisión durante el entrenamiento. . . . .	166
8.50. Pérdida de los diferentes modelos . . . . .	167
8.51. <i>Accuracy</i> de los diferentes modelos . . . . .	168
8.52. Matriz de confusión del modelo “model_fold_5_50_epochs.keras”. . .	169
8.53. Matriz de confusión del modelo “model_fold_5_50_epochs.keras” sin la categoría <i>actionable</i> y con datos ampliados para la clase <i>normal</i> . . . . .	172
A.1. Matriz de confusión de ejemplo. . . . .	186

# Capítulo 1

## Introducción

### 1.1. Contexto

Hace más de 5.000 años se redactó la mención más antigua conocida del cáncer en el antiguo Egipto. Los egipcios ya diferenciaban entre tumores benignos y malignos, cancerosos. Alrededor del 460 a.C., en la antigua Grecia, Hipócrates pensaba que había cuatro fluidos en el cuerpo que influían en la salud: sangre, flema, bilis amarilla y bilis negra. Desde entonces pasaron 1.400 años en los que la humanidad pensaba que el cáncer estaba provocado por la bilis negra; hasta que en 1860, el cirujano alemán Karl Thiersch fue la primera persona en demostrar que el cáncer se propaga a través de células malignas. El cáncer es una de las áreas de estudio más importantes en la investigación médica, haciendo progresos hasta nuestros días.

Paralelo al desarrollo de la medicina, las ciencias concibieron el sector de la informática. Todo empezó cuando el matemático alemán Wilhelm Schickard inventó la primera calculadora en 1623; continuó con Ada Lovelace, convirtiéndose en la primera programadora en 1830. En los años 40, el famoso matemático londinense, conocido como padre de la inteligencia artificial, Alan Mathison Turing, ideó la *Máquina de Turing* y desarrolló la máquina *Bombe* durante la Segunda Guerra Mundial ([Wikipedia - Alan Turing, 2024](#)). En 1957, Frank Rosenblatt inventó el perceptrón, que es el componente básico de una red neuronal; y desde hace 5 años estamos viviendo una

## 1.1. CONTEXTO

---

explosión en este sector, con el lanzamiento y fama mundial de páginas webs que trabajan con modelos de inteligencia artificial, como *ChatGPT*, *Gemini* y *DALL-E*.

El avance en el conocimiento del cáncer viene acompañado por el desarrollo de tecnologías que permiten un diagnóstico y tratamiento más preciso. La integración de la informática y la medicina ha permitido a los científicos y médicos entender mejor la complejidad de la enfermedad, identificando patrones y desarrollando tratamientos personalizados.

El cáncer es uno de los desafíos globales más importantes a los que se enfrenta el mundo de la medicina, el cual afecta a millones de personas en todo el mundo. Aunque haya evidencias de su relación con malos hábitos (el tabaco, el alcohol, la obesidad etc), con ciertos estilos de vida o incluso con la genética de cada individuo. Precisamente, su marcado carácter aleatorio, el saber que puede afectar a cualquier persona y no se pueda hacer nada para evitarlo, es lo que pone en manifiesto uno de los mayores miedos de la humanidad: la incertidumbre, el saber que no hay forma de predecir el fin de la existencia de un individuo.

En este contexto, los sistemas de Detección Asistida por Computador, CADe<sup>1</sup>, han demostrado un rendimiento notable. Ayudan a los radiólogos menos experimentados a igualar, o incluso superar, el rendimiento y la precisión de los más experimentados en la detección del cáncer hoy en día. La investigación médica continúa trabajando intensamente para mejorar la precisión de los sistemas CADe, con el objetivo de detectar la mayor cantidad de casos de cáncer posibles e incluso superar el rendimiento y la precisión de los radiólogos más experimentados.

Los algoritmos de Inteligencia Artificial (IA) están siendo desarrollados para interpretar mamografías y biopsias de mama, con el objetivo de mejorar la precisión y eficiencia de los sistemas CADe. La IA ha demostrado excelentes resultados en otros campos de la visión por ordenador, destacándose en áreas como el reconocimiento facial, reconocimiento de señales de tráfico o la clasificación de objetos en imágenes.

---

<sup>1</sup>CADe: *Computer-Aided Detection* o Detección Asistida por Computador. Sistema *software* para ayudar a buscar áreas de tejido sospechosas en una imagen radiológica.

## 1.2. Objetivos

Durante el desarrollo de este Trabajo Fin de Grado (TFG) se analizará el uso de la creciente tecnología de aprendizaje profundo, *Deep Learning*, para la identificación del cáncer y tumores benignos mediante la mamografía digital y la tomosíntesis digital de mama. Se realizarán diferentes aproximaciones para solucionar esta problemática, desde el desarrollo de dos redes neuronales basadas en la arquitectura *U-net*, como modelos que utilizan la técnica de *Transfer learning*<sup>2</sup>, con los modelos *Inception v3* y *ResNet v2* de *Google*.

Se compararán los modelos implementados con los sistemas utilizados en la actualidad con los sistemas CADe, los cuales ayudan en la detección de masas malignas. Los resultados obtenidos servirán para proporcionar una hoja de ruta para futuras investigaciones y mejoras del trabajo realizado.

El cumplimiento de estos objetivos ha necesitado de un análisis exhaustivo de las diferentes técnicas de aprendizaje automático utilizando *deep learning*, con una formación personal previa en este campo, así como del estudio de antecedentes o trabajos previos de relevancia en la detección del cáncer de mama a través de fuentes confiables.

## 1.3. Planificación

La figura 1.1 muestra la planificación cronológica del proyecto, detallando las actividades mensuales y las horas dedicadas a cada una de ellas. Esta planificación es crucial para asegurar un desarrollo ordenado y eficiente del proyecto, permitiendo la monitorización del progreso y ajustar tiempos según ha sido necesario.

---

<sup>2</sup>*Transfer learning*: técnica de aprendizaje automático en la que un modelo preentrenado en una tarea con abundante información se adapta para mejorar su desempeño en otra tarea relacionada, generalmente con menos datos disponibles, permitiendo así aprovechar el conocimiento adquirido previamente.

### 1.3. PLANIFICACIÓN

Mes	Formación	Conjuntos de datos	Desarrollo Inception v3	Desarrollo ResNet v2	Desarrollo UNet	Memoria
Septiembre	Comienzo del libro Python Deep Learning (Torres, 2020)					
Octubre	Fin del Libro Python Deep Learning (Torres, 2020) Comienzo del libro Deep Learning with Python, 2nd ed. (Chollet, 2021)					
Noviembre	Lectura del libro Deep Learning with Python, 2nd ed. (Chollet, 2021)					
Diciembre	Lectura del libro Deep Learning with Python, 2nd ed. (Chollet, 2021)					
Enero	Fin del libro Deep Learning with Python, 2nd ed. (Chollet, 2021)					
Febrero		Obtención y procesamiento del conjunto de datos DBT				
Marzo		Obtención y procesamiento del conjunto de datos CMMD				
Abril		Procesamiento de los datos	Comienzo del desarrollo			Comienzo documentación de memoria
Mayo	Lectura documentación de Keras sobre las diferentes capas	Procesamiento de los datos	Mejora del modelo	Comienzo del desarrollo		Documentación de memoria
Junio	Lectura de artículos sobre el modelo U-Net Lectura documentación de Keras sobre las diferentes capas		Entrenamiento final	Entrenamiento final	Comienzo del desarrollo de la red UNet-Densa	Documentación de memoria
Julio	Lectura de la documentación de Keras sobre las diferentes capas				Final del desarrollo de UNet-Densa y comienzo de UNet-Paralela	Documentación de memoria
Agosto	Lectura de la documentación de Keras sobre las diferentes capas				Entrenamiento final	Fin documentación de memoria

Figura 1.1: Cronología para llevar a cabo el trabajo desarrollado.

A continuación, se describen las actividades planificadas mes a mes, junto con el número de días y las horas promedio dedicadas diariamente a cada tarea.

#### Septiembre a enero:

- Formación: lectura de los textos "Python Deep Learning" de Torres (2020) y "Deep Learning with Python", 2nd ed. de Chollet, Francois (2021).

*Días: 50; horas/día: 5.*

#### Febrero:

## CAPÍTULO 1. INTRODUCCIÓN

---

- Conjuntos de datos: obtención y procesamiento del conjunto de datos DBT.

*Días: 25; horas/día: 2.*

### **Marzo:**

- Conjuntos de datos: obtención y procesamiento del conjunto de datos CMMD.

*Días: 10; horas/día: 2.*

### **Abril:**

- Conjuntos de datos: procesamiento de los datos.

*Días: 5; horas/día: 3. - Desarrollo Inception v3: comienzo del desarrollo.*

*Días: 15; horas/día: 4. - Memoria: comienzo de la documentación.*

*Días: 3; horas/día: 2.*

### **Mayo:**

- Formación: lectura de la documentación del paquete Keras para las diferentes capas.

*Días: 20; minutos/día: 15.*

- Conjuntos de datos: procesamiento de los datos.

*Días: 3; horas/día: 5.*

- Desarrollo Inception v3: mejora del modelo.

*Días: 15; horas/día: 4.*

- Desarrollo ResNet v2: comienzo del desarrollo.

*Días: 5; horas/día: 4.*

- Memoria: documentación de la memoria.

*Días: 5; horas/día: 2.*

### **Junio:**

- Formación: lectura de artículos sobre el modelo *U-Net* y la documentación de *Keras* sobre las diferentes capas.

*Días: 10; horas/día: 2.*

- Desarrollo Inception v3: entrenamiento final.

*Días: 15; horas/día: 4.*

- Desarrollo ResNet v2: entrenamiento final.

*Días: 15; horas/día: 4.*

#### 1.4. ORGANIZACIÓN DE CONTENIDOS

---

- Desarrollo UNet-Densa: comienzo del desarrollo.

*Días: 25; horas/día: 6.*

- Memoria: documentación de la memoria.

*Días: 20; horas/día: 3.*

##### **Julio:**

- Formación: lectura de la documentación de *Keras* sobre las diferentes capas.

*Días: 1; minutos/día: 30.*

- Desarrollo UNet-Densa: entrenamiento final.

*Días: 11; horas/día: 8.*

- Desarrollo UNet-Paralela: comienzo del desarrollo.

*Días: 5; horas/día: 8.*

- Memoria: documentación de la memoria.

*Días: 12; horas/día: 6.*

##### **Agosto:**

- Formación: lectura de la documentación de *Keras* sobre las diferentes capas.

*Días: 1; minutos/día: 30.*

- Desarrollo UNet-Paralela: comienzo del desarrollo.

*Días: 10; horas/día: 8.*

- Memoria: fin de la documentación de la memoria.

*Días: 12; horas/día: 6.*

## 1.4. Organización de contenidos

La presente memoria está estructurada en nueve capítulos que guían al lector a través de los distintos aspectos del proyecto, desde su base teórica hasta las conclusiones finales. A continuación, se describe brevemente el contenido de cada uno de estos capítulos.

En el [capítulo 1](#), se introduce al lector en el contexto general del proyecto, estableciendo los objetivos, la motivación y el alcance del mismo. Este capítulo proporciona una visión global que permite entender la relevancia del estudio en el

## CAPÍTULO 1. INTRODUCCIÓN

---

ámbito del cáncer de mama, situando el trabajo dentro de un marco más amplio.

El **capítulo 2** está dedicado a los *Fundamentos médicos y tecnológicos del cáncer de mama*. Aquí, se abordan las bases biológicas y clínicas del cáncer de mama, con un enfoque en su diagnóstico y tratamiento, además de una revisión de las tecnologías más relevantes en el campo. Este capítulo es esencial para comprender el contexto médico en el que se enmarca el proyecto.

Posteriormente, en el **capítulo 3**, se presentan los *Antecedentes* que han servido de base para el desarrollo del presente trabajo. En esta sección, se revisan trabajos previos y se destacan aquellos estudios que han influido de manera significativa en la construcción del proyecto actual, proporcionando así una justificación sólida de la metodología adoptada.

El **capítulo 4** describe las *Herramientas software y hardware* utilizadas en el desarrollo del proyecto. Se detallan las plataformas tecnológicas, entornos de desarrollo, y equipos utilizados, estableciendo las bases técnicas necesarias para la implementación del proyecto.

A partir de aquí, la memoria se centra en el trabajo realizado. En el **capítulo 5**, titulado *Análisis de los datos*, se describe en detalle el origen de las fuentes de datos empleadas, así como el proceso de recopilación y selección de las mismas. Esta sección es crucial para entender la calidad y relevancia de los datos que se han utilizado para entrenar y validar los modelos.

El siguiente es el **capítulo 6**, donde se explica el *Procesado de datos*. En este capítulo, se aborda el tratamiento y transformación de los datos, describiendo las técnicas y metodologías aplicadas para preparar los datos de cara a su uso en los modelos implementados. Se discuten los desafíos encontrados y las soluciones adoptadas para garantizar la integridad y utilidad de los datos.

En el **capítulo 7**, titulado *Desarrollo del Proyecto*, se presentan los modelos de referencia que han sido tomados como punto de partida, así como las distintas redes neuronales implementadas. Se incluye una descripción detallada de la base teórica de cada modelo, junto con las razones para su selección y adaptación. Este capítulo

## 1.4. ORGANIZACIÓN DE CONTENIDOS

constituye el núcleo del trabajo, proporcionando una visión exhaustiva del proceso de desarrollo y los fundamentos científicos que lo sustentan.

El [capítulo 8](#) expone los *Resultados* obtenidos. En esta sección, se presentan y analizan los resultados derivados de las pruebas realizadas sobre los diferentes modelos de redes neuronales, tanto en la fase de entrenamiento como en la fase testeo. Se proporciona una discusión detallada de los resultados, evaluando su relevancia y precisión en relación con los objetivos del proyecto.

Finalmente, en el [capítulo 9](#), titulado *Conclusiones*, se realiza un análisis comparativo de los resultados obtenidos en relación con los sistemas de diagnóstico asistido por computadora (*CAD*<sup>3</sup>) mencionados en el capítulo 3. Además, se reflexiona sobre las posibles mejoras y direcciones futuras que podrían explorar en estudios posteriores, estableciendo una base para trabajos futuros.

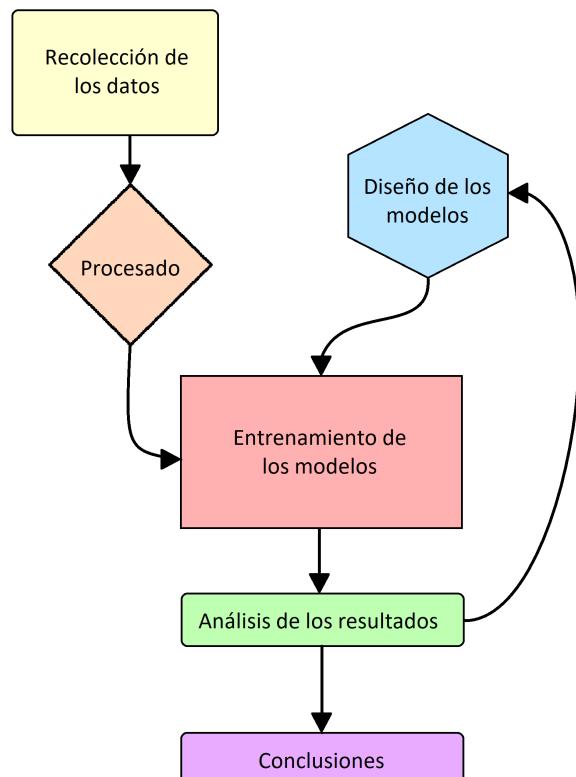


Figura 1.2: Flujo de trabajo con las etapas clave de cada uno de los modelos implementados.

<sup>3</sup>Sistemas de *Computer-Aided Diagnosis*

## CAPÍTULO 1. INTRODUCCIÓN

---

El flujo de trabajo con las etapas clave seguidas, esquematizadas en la figura 1.2, se describen como:

- **Recolección de los Datos.** Para llevar a cabo esta primera etapa, se realizó un largo proceso de selección e investigación de las diferentes opciones, el cual concluyó con la selección de los conjuntos de datos *Breast-Cancer-Screening-DBT* y CMMD.

Este proceso se detalla en el capítulo 6. Procesado de datos.

- **Procesado.** En esta fase se utilizó tanto código facilitado por uno de los conjuntos de datos como, sobretodo, código desarrollado en *Python*, siguiendo el entorno de trabajo descrito en el capítulo 4. Herramientas *software* y *hardware*. Es una de las fases más largas y complicadas del TFG, pues el formato y las características de los datos no estaban documentados, necesitándose una minuciosa tarea de análisis previa al posterior procesado. Además, es una etapa esencial, pues los datos deben ser limpiados y adaptados de forma correcta antes de ser utilizados en el entrenamiento por los diferentes modelos.

El proceso de análisis está documentado en el capítulo 5. Análisis de los datos, y el procesado de los datos en el capítulo 6. Procesado de datos.

- **Diseño de los modelos.** Si tenemos en cuenta la cronología mostrada en la sección 1.3 anterior, esta etapa ha tenido una duración de nueve meses desde que empezó la formación, con la lectura de los textos de Torres (2020) y Chollet, Francois (2021), indispensables para obtener las bases técnicas necesarias, y los 4 meses que se emplearon en desarrollar todos los modelo.

Esta etapa se documenta en el capítulo 7. Desarrollo del proyecto.

- **Entrenamiento de los modelos.** Durante esta fase se probaron todos los modelos diseñados en la fase anterior, la cual fue un proceso largo y complicado debido a la cantidad de problemas encontrados. Una vez los modelos fueron entrenados satisfactoriamente, se pasa a la siguiente etapa para el análisis de los resultados.

## 1.4. ORGANIZACIÓN DE CONTENIDOS

---

- **Análisis de los resultados.** Una vez entrenados los modelos, si los resultados son satisfactorios con el conjunto de datos de validación, se someten a un nuevo test con el conjunto de datos de prueba. Después, si los resultados no son aceptables, ya sea con el conjunto de validación o con el de prueba, se procede de nuevo e iterativamente a la fase de diseño del modelo para su mejora. Si, por el contrario, los resultados obtenidos son satisfactorios para el conjunto de datos de prueba, se pasa a la siguiente etapa dando por concluido el ciclo iterativo de los modelos.

Los resultados del entrenamiento están especificados en el capítulo [8. Resultados](#).

- **Conclusiones.** En esta etapa final, los resultados de cada modelo se comparan entre sí y con los ejemplos expuestos en el capítulo [3. Antecedentes](#). Llegando a la conclusión final de este trabajo de fin de grado y mostrando la viabilidad, o no, conseguidas con las técnicas *Deep Learning* empleadas para hacer frente a la tarea de clasificación de tumores y cáncer por medio de mamografías y tomosíntesis digitales de mama.

Las conclusiones y propuestas de mejora están recogidas en el capítulo [9. Conclusiones](#).

# **Capítulo 2**

## **Fundamentos médicos y tecnológicos del cáncer de mama**

Durante este capítulo se plantearán los fundamentos teóricos y el estado actual del cáncer, así como la definición de los conceptos de Sistemas de Detección Asistida por computadora, mamografía digital y tomosíntesis digital de mama.

### **2.1. Estado actual del cáncer**

Según el [Instituto Nacional de Estadística, INE \(2024\)](#), cada año mueren de cáncer alrededor de 231 personas por cada 100.000 habitantes, lo que deja la demoledora cifra anual de más de 110.000 fallecidos en España. En concreto, el cáncer de mama es la principal causa de muerte por cáncer entre las mujeres españolas, siendo un 14,2 % del total. De esta forma, este tipo de cáncer cuenta con un total de 6,75 muertes por cada 100.000 habitantes.

Según el comunicado de la [Organización Mundial de la Salud, OMS \(2024\)](#), en el año 2022 se registraron 2,3 millones de nuevos casos de cáncer de mama en mujeres, con 670.000 fallecimientos a causa de esta enfermedad. Es el cáncer más común en mujeres en 157 de 185 países estudiados, sin discriminar por nivel de desarrollo humano. Las desigualdades en la carga de morbilidad son evidentes, siendo más

## 2.2. ESTADO DE LA DETECCIÓN DEL CÁNCER

pronunciadas en países con un índice de desarrollo humano bajo.

La edad, antecedentes familiares, y factores de estilo de vida como la obesidad y el consumo de alcohol, también, pueden aumentar el riesgo. Es importante destacar que alrededor de la mitad de los casos no presentan factores de riesgo identificables aparte del género y la edad.

### 2.2. Estado de la detección del cáncer

En esta sección hablaremos sobre cómo se realiza la detección del cáncer, explicando como el método de detección es una variable pronóstica, los diferentes sesgos en exámenes de detección y las variables que influyen en la precisión de los exámenes.

Gran parte de la información de esta sección se basa en el resumen “*PDQ® sobre los exámenes de detección y la prevención*” realizado por [Instituto Nacional del Cáncer \(NCI\) \(2024\)](#). Este documento, dirigido a profesionales de la salud, ofrece una guía revisada por expertos sobre los exámenes de detección del cáncer de mama, proporcionando una referencia confiable y basada en evidencia.

El consejo editorial del PDQ, compuesto por una veintena de expertos independientes, revisa y actualiza periódicamente el resumen. Mensualmente, evalúan nuevos artículos científicos para decidir su inclusión y mantener el contenido actualizado <sup>1</sup>.

#### 2.2.1. Características del cáncer determinado mediante exámenes de detección

En un estudio longitudinal, realizado en Finlandia con 1.983 mujeres con cáncer de mama invasivo, se demostró que el método de detección es una variable pronóstica

<sup>1</sup>Información sobre el consejo editorial del PDQ: <https://www.cancer.gov/espanol/publicaciones/pdq/consejos-editoriales/deteccion-prevencion> (Accedido el 7 de junio de 2024)



## CAPÍTULO 2. FUNDAMENTOS MÉDICOS Y TECNOLÓGICOS DEL CÁNCER DE MAMA

independiente<sup>2</sup>. Tras ajustar por edad, compromiso ganglionar y tamaño tumoral, se observó que los cánceres detectados mediante exámenes de detección presentaban un menor riesgo de recaída y una mejor tasa supervivencia general. Para las mujeres cuyo cáncer no fue detectado mediante estos exámenes, el riesgo de muerte fue notablemente mayor, a pesar de haber recibido el mismo protocolo de tratamiento.

Otro análisis, que incluyó tres ensayos clínicos aleatorizados<sup>3</sup>, *Health Insurance Plan, National Breast Screening Study 1* y *National Breast Screening Study 2*, también reveló que las pacientes cuyos cánceres fueron detectados mediante exámenes de detección tenían un pronóstico más favorable.

En un tercer estudio, que evaluó a 5.604 mujeres inglesas diagnosticadas entre 1998 y 2003, también encontró que las mujeres con cánceres detectados mediante exámenes de detección tenían mejores desenlaces.

Estos hallazgos apoyan la evidencia de que los cánceres de mama detectados mediante exámenes de detección presentan un mejor pronóstico que aquellos no diagnosticados a través de estos métodos, independientemente del estado ganglionar y tamaño tumoral; lo que sugiere que son biológicamente menos agresivos, posiblemente de crecimiento más lento y con menor probabilidad de invasión o metástasis.

### 2.2.2. Concepto de sesgos en los exámenes de detección

La capacidad de la mamografía para diagnosticar cánceres de mama pequeños en estados tempranos se ha documentado en varios estudios no controlados y series retrospectivas<sup>4</sup>. Para entender por qué los casos de cánceres, detectados por exámenes de diagnóstico temprano, tienen menos mortalidad, debemos comprender los siguientes sesgos ([Instituto Nacional del Cáncer \(NCI\), 2024](#)):

1. **Sesgo de anticipación diagnóstica:** la detección temprana adelanta la fecha del diagnóstico sin alterar la progresión de la enfermedad, aumentando

<sup>2</sup>Variable pronóstica: Factor o característica que puede predecir el curso de una enfermedad.

<sup>3</sup>Ensayo controlado aleatorizado: estudio donde los participantes son asignados aleatoriamente a grupos de tratamiento o control para evaluar los efectos de una intervención.

<sup>4</sup>Series retrospectivas: estudios que analizan datos recolectados en el pasado para evaluar resultados.

## 2.2. ESTADO DE LA DETECCIÓN DEL CÁNCER

artificialmente la tasa de supervivencia a 5 años.

2. **Sesgo de duración:** la detección de cánceres de crecimiento lento mejora la tasa de supervivencia a 5 años, aunque no se altere la mortalidad global.
3. **Sesgo de sobrediagnóstico:** identificación de cánceres que nunca causarían síntomas o la muerte, aumentando las tasas de supervivencia sin cambiar la esperanza de vida.
4. **Sesgo de voluntarias sanas:** participación de mujeres, con buen estado de salud, en los exámenes de detección, lo que mejora los desenlaces observados.

La repercusión de estos sesgos no está clara. Un ensayo controlado aleatorizado con un criterio de valoración de mortalidad por causa específica sería necesario para determinar el beneficio de supervivencia y el impacto de estos sesgos. Sin embargo, la realización de tal estudio es poco viable por razones éticas y prácticas.

### 2.2.3. Variables relacionadas con la precisión

#### Características de la paciente

La precisión de las mamografías varía según las características de la paciente, como la edad, la densidad mamaria<sup>5</sup>, y si se trata de su primer examen o de uno posterior. Las mujeres jóvenes presentan una sensibilidad más baja y tasas de resultados positivos falsos más altas que las mujeres mayores.

En el *United Kingdom Age Trial*, según Instituto Nacional del Cáncer (NCI) (2024), se evaluó la eficacia de la detección mediante mamografía en mujeres menores de 50 años, sin encontrarse diferencias significativas en la mortalidad por cáncer de mama entre los grupos que comenzaron la detección a los 39-41 años y aquellos que lo hicieron a los 50-52 años.

<sup>5</sup>Densidad mamaria: proporción de tejido glandular y fibroso en comparación con el tejido graso en la mama.



## CAPÍTULO 2. FUNDAMENTOS MÉDICOS Y TECNOLÓGICOS DEL CÁNCER DE MAMA

También, se debe tener en cuenta que las mujeres obesas presentan un riesgo mayor de obtener falsos positivos. Aunque la sensibilidad no se vea afectada, ya que también disminuyen los falsos negativos.

### Densidad mamaria

Las mamas densas pueden oscurecer la detección de masas pequeñas en la mamografía, reduciendo su sensibilidad. La alta densidad mamaria se asocia con una reducción del 10% al 29% en la sensibilidad. Además, esto dé con un aumento moderado del riesgo de cáncer de mama, aunque no con una mayor mortalidad por dicho cáncer.

La densidad mamaria es un rasgo hereditario e influenciado por la edad, las hormonas, los moduladores selectivos del receptor de estrógeno y la alimentación. La hormonoterapia aumenta la densidad mamaria y la tasa de cánceres de intervalo.

De acuerdo a lo indicado por Instituto Nacional del Cáncer (NCI) (2024), el American College of Radiology's clasifica la densidad mamaria en cuatro categorías, siendo las dos últimas consideradas como tejido mamario denso, afectando a un 43% de las mujeres entre 40 y 74 años.

### Características tumorales

Continuando con lo indicado por Instituto Nacional del Cáncer (NCI) (2024), las mamografías detectan más fácilmente los cánceres de tipo mucinoso<sup>6</sup> y lobulillar<sup>7</sup>. Algunos cánceres de crecimiento rápido pueden confundirse con tejido mamario normal, como los carcinomas medulares, que aunque poco comunes, a menudo están asociados con la mutación BRCA1 y responden favorablemente al tratamiento.

<sup>6</sup>Cáncer mucinoso: tipo de cáncer que comienza en las células que revisten ciertos órganos internos y que producen mucina, el componente principal del moco (fuente: <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/carcinoma-mucinoso> [Accedido el 07 de junio de 2024]).

<sup>7</sup>Cáncer lobulillar: tipo de cáncer de mama invasivo que empieza en los lobulillos (glándulas lácteas) y se disemina al tejido normal cercano. A veces, también se disemina a otras partes del cuerpo a través de los sistemas sanguíneo y linfático (fuente: <https://www.cancer.gov/espanol/publicaciones/diccionarios/diccionario-cancer/def/carcinoma-lobulillar-invasivo> [Accedido el 07 de junio de 2024]).

### 2.3. MAMOGRAFÍA DIGITAL

Algunos cánceres relacionados con mutaciones BRCA1 y BRCA2<sup>8</sup> también pueden ser difíciles de detectar debido a su crecimiento lento.

#### Características de los médicos

Asimismo, en [Instituto Nacional del Cáncer \(NCI\) \(2024\)](#) se apunta que el desempeño de los radiólogos depende de dos factores fundamentales:

1. La experiencia cuantificada con la cantidad de mamografías que han interpretado. Los radiólogos en ámbitos académicos muestran un pronóstico positivo más alto en sus recomendaciones para biopsias. La formación especializada podría mejorar la detección en mamografías.
2. El centro médico. La detección es más precisa en centros que solo ofrecen exámenes de detección y que cuentan con un especialista permanente en oncología, con auditorías frecuentes de su desempeño. Las tasas de falsos positivos son más altas en establecimientos con mayores casos por negligencia médica y que atienden a mujeres vulnerables. Estas poblaciones presentan una mayor prevalencia de cáncer y carecen de seguimiento adecuado.

## 2.3. Mamografía digital

La mamografía digital es una técnica que utiliza radiación ionizante para obtener imágenes del tejido mamario mediante la compresión de la mama entre dos placas, lo cual despliega los tejidos superpuestos y reduce la cantidad de radiación necesaria. En los Estados Unidos, las mamografías de rutina se realizan en proyecciones oblicuas mediolaterales<sup>9</sup> y craneocaudales<sup>10</sup>, abarcando desde el pezón hasta el músculo pectoral. Los exámenes con dos proyecciones tienen una menor tasa de repetición y

<sup>8</sup>El BRCA1 (cáncer de seno 1) y el BRCA2 (cáncer de seno 2) son genes que producen proteínas que reparan daños en el ácido desoxirribonucleico (ADN).

<sup>9</sup>La proyección mediolateral oblicua es una vista en la que la imagen se toma en un ángulo oblicuo, permitiendo visualizar tanto el tejido central como lateral del seno.

<sup>10</sup>La proyección craneocaudal es una vista en la que la imagen se toma desde arriba del seno hacia abajo, abarcando desde el pezón hasta el músculo pectoral.



## CAPÍTULO 2. FUNDAMENTOS MÉDICOS Y TECNOLÓGICOS DEL CÁNCER DE MAMA

están asociados con una menor incidencia de cánceres de intervalos en comparación con los exámenes de una sola proyección ([Instituto Nacional del Cáncer \(NCI\), 2024](#)).

La mamografía digital representa una evolución significativa frente a las radiografías de mama convencionales, principalmente en términos de almacenamiento y tratamiento digital de las imágenes. Esta mejora es particularmente relevante para la integración y el uso efectivo de los sistemas de detección asistida por computadora (CAD), como se menciona en el capítulo [3. Antecedentes](#).

Sin embargo, diversos estudios han comparado directamente el uso de mamografía digital y películas radiográficas en términos de tasa de detección del cáncer, sensibilidad, especificidad y valor predictivo positivo (VPP), obteniendo resultados similares en la mayoría de los grupos de pacientes al ser usadas por radiólogos sin sistemas ([Instituto Nacional del Cáncer \(NCI\), 2024](#)).

### 2.4. Tomosíntesis digital de mama

La Tomosíntesis digital de mama, DBT<sup>11</sup>, es una modalidad avanzada de radiografía en 3 dimensiones que se desarrolló, específicamente, para superar los desafíos de la superposición de tejidos gracias a el avance de los detectores digitales, según [Niklason y cols. \(1997\)](#). Aunque el concepto de tomosíntesis para imágenes radiográficas se remonta a principios de los años 30 del siglo pasado, como [Des Plantes \(1932\)](#) indica, su aplicación en imágenes mamarias surgió en la década de 1990 ([Niklason y cols., 1999](#)).

Los sistemas clínicos y prototipos actuales de DBT varían significativamente en términos de geometría de las imágenes, rango angular del movimiento del tubo de rayos X, número y distribución de las proyecciones, duración del escaneo, método de adquisición y tecnología de detección. Estos sistemas también difieren en los algoritmos de reconstrucción utilizados.

<sup>11</sup>DBT: Digital Breast Tomosynthesis.

## 2.4. TOMOSÍNTESIS DIGITAL DE MAMA

Fabricante	<i>General Electric</i>	<i>Hologic</i>	<i>Internazionale Medico Scientifica</i>	<i>Siemens</i>
<b>Modelo o plataforma</b>	<i>SenoClaire/ Senograph Essential</i>	<i>Selenia Dimensions</i>	<i>Giotto Tomo</i>	<i>MAMMOMAT Inspiration</i>
<b>Distancia al detector (cm)</b>	66	70	68	65.5
<b>Distancia al centro de rotación (cm)</b>	62	70	66	60.8
<b>Distancia al soporte del seno (cm)</b>	63.8	67.5	65.8	63.8
<b>Movimiento del tubo de rayos X</b>	Paso y disparo	Continuo	Paso y disparo	Continuo
<b>Proyecciones</b>	9	15	13	25
<b>Tiempo de escaneo (seg)</b>	Típicamente <10	3.7	12	25
<b>Método de reconstrucción</b>	Iterativo (ASiR-DBT)	FBP/ contraste iterativo	Iterativo	FBP/ filtro de espesor de sección

Tabla 2.1: Comparación de diferentes sistemas de mamografía. Fuente: *Digital Breast Tomosynthesis: State of the Art* (Vedantham y cols., 2015).

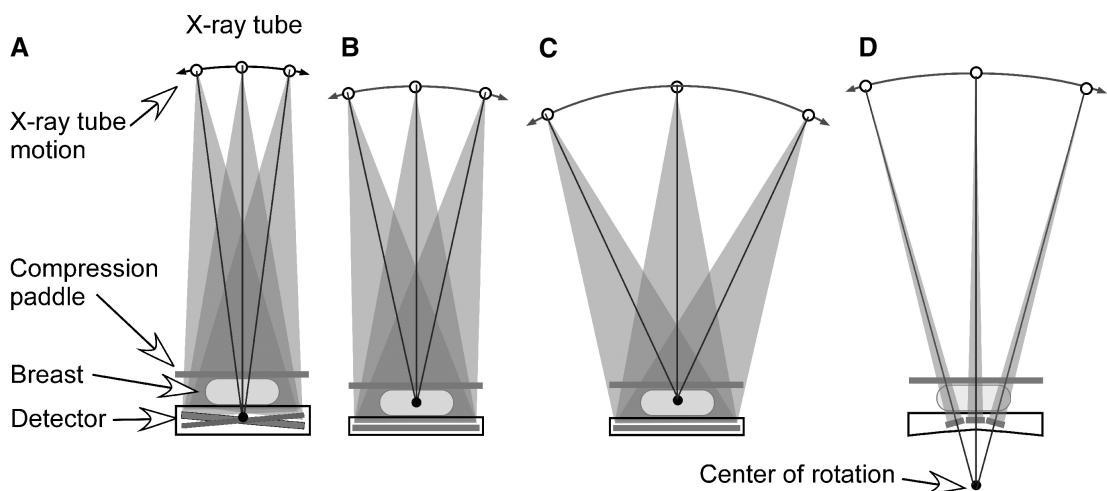


Figura 2.1: Diferentes geometrías de sistemas clínicos de DBT. Fuente: *Digital Breast Tomosynthesis: State of the Art* (Vedantham y cols., 2015).



## CAPÍTULO 2. FUNDAMENTOS MÉDICOS Y TECNOLÓGICOS DEL CÁNCER DE MAMA

En la figura 2.1 se presentan cuatro enfoques diferentes de geometría utilizados en los sistemas clínicos de DBT. Estos esquemas ilustran las distintas configuraciones del movimiento del tubo de rayos X y su impacto en la adquisición de imágenes.

La tabla 2.1 resume las especificaciones de algunos sistemas clínicos de DBT aprobados por las entidades regulatorias europeas.

### 2.4.1. Ejemplos de aplicación

En la figura 2.4 se muestra un caso de falso positivo que fue identificado mediante DBT. Se trata de una mujer de 41 años con mamas densas heterogéneas, donde las imágenes de DBT RCC<sup>12</sup> y RMLO<sup>13</sup> revelaron que la masa identificada (indicada por una flecha) probablemente correspondía a un grupo de ganglios linfáticos intramamarios con hilo graso. Esto fue confirmado mediante mamografía, permitiendo a la paciente someterse nuevamente a las pruebas anuales de detección sin preocupaciones. Ello demuestra que la DBT permite superar las limitaciones impuestas por la superposición de tejidos en las radiografías convencionales, proporcionando más información.

Para que el lector pueda comparar la diferencia, la figura 2.5 ilustra un caso de una paciente con un tumor, señalado con un cuadrado blanco. Las imágenes obtenidas mediante DBT han demostrado ser efectivas en la detección de tumores y malformaciones mamarias, siendo utilizadas tanto por radiólogos como por los sistemas CADe.

<sup>12</sup>Right Craniocaudal (RCC): craneocaudal derecho.

<sup>13</sup>Right Mediolateral Oblique (RMLO): oblicua medio-lateral derecha.

## 2.4. TOMOSÍNTESIS DIGITAL DE MAMA

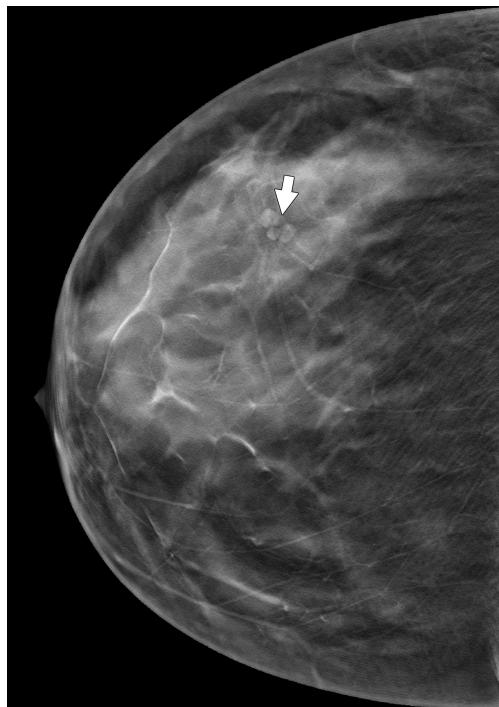


Figura 2.2: RCC

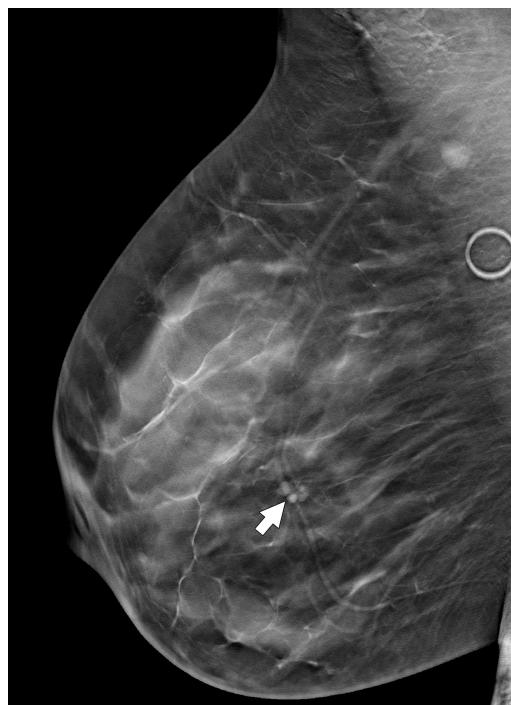


Figura 2.3: RMLO

Figura 2.4: Falso positivo identificado por *DBT - Breast Tomosynthesis: State of the Art* (Vedantham y cols., 2015).

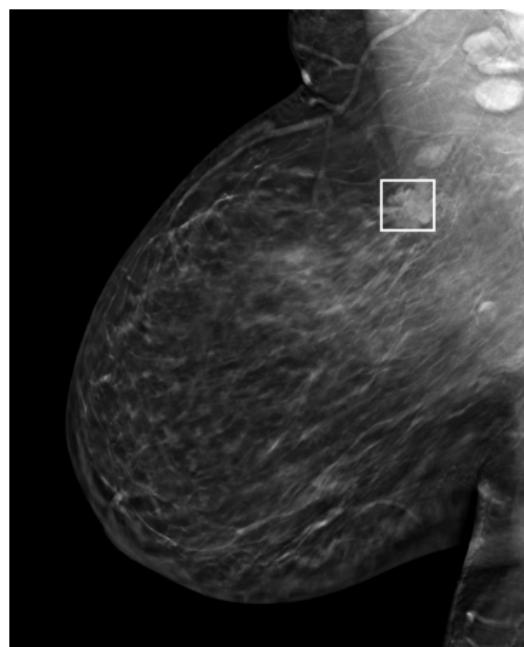


Figura 2.5: Tumor detectado por *DBT - Breast Cancer Screening* (Buda y cols., 2020).

# **Capítulo 3**

## **Antecedentes**

Antes de explicar como se ha llevado a cabo el proyecto, se exponen los trabajos encontrados como relevantes en el ámbito de la detección de cánceres de mama. Para ello, tras contextualizar el problema, nos detendremos en los sistemas CADe de detección asistida por computadora actuales, y veremos el modelo base que se propone en uno de los conjuntos de datos y desarrollaremos un modelo base para nuestro problema con nuestros datos.

### **3.1. Contextualización**

El aprendizaje profundo surgió principalmente debido al rápido aumento en el acceso a recursos computacionales y datos etiquetados a gran escala. La imagen médica es una aplicación natural de los algoritmos de *deep learning*. Sin embargo, los datos médicos de calidad son escasos, lo que plantea un desafío en el entrenamiento y la validación de modelos de aprendizaje profundo.

Por un lado, el número de imágenes médicas disponibles es mucho menor que el número de imágenes disponibles en otros ámbitos, como la fotografía general o las redes sociales. Particularmente, es un problema al investigar una condición con una prevalencia bastante baja, como el cáncer de mama en un entorno de detección (menos del 1 % de los exámenes de detección resultan en un diagnóstico de cáncer). Por otro

### 3.2. SISTEMAS CAD

lado, el acceso a los datos de imágenes médicas está guiado por una serie de políticas estrictas dado que contienen información médica de los pacientes. Compartir datos de imágenes médicas requiere un esfuerzo, a menudo no trivial, y que consume mucho tiempo, para despersonalizar los datos y garantizar el cumplimiento de los requisitos de la institución que comparte los datos y más allá. Además, la anotación de datos de imágenes médicas típicamente requiere el trabajo de radiólogos, que ya tienen altas demandas en su tiempo.

Como resultado, la cantidad de datos es un problema al entrenar modelos de aprendizaje profundo, pero también resulta en una falta de transparencia al evaluar el rendimiento del modelo.

La limitada reproducibilidad de los resultados ha sido particularmente visible en la investigación de mamográfica, probablemente la aplicación más común de inteligencia artificial (IA) en radiología en las últimas dos décadas. Los investigadores utilizan conjuntos de datos diferentes, a menudo no disponibles públicamente, y resuelven tareas relacionadas pero diferentes. Más aún, los estudios tienen estrategias de evaluación diferentes, lo que dificulta comparar de manera confiable métodos y resultados. Un sistema de IA debe estar ampliamente validado antes de su aplicación en la práctica clínica; por ejemplo, una deficiencia común en muchos estudios es que el conjunto de pruebas se obtuvo de una sola institución y un número limitado de dispositivos. Asimismo, algunos estudios excluyen datos, lo que oscurece aún más el verdadero rendimiento de los algoritmos.

## 3.2. Sistemas CAD

Los sistemas CAD<sup>1</sup>, llevan varios años usándose para apoyar a los radiólogos en la interpretación de mamografías. Para este propósito, existen dos tipos diferentes de sistemas CAD: los sistemas de detección asistida por computadora (CADe) y de diagnóstico asistido por computadora (CADx). El CADe se utiliza para identificar estructuras anormales en las mamografías, mientras que el CADx se enfoca en evaluar

<sup>1</sup>CAD: Computer-Aided Detection.

## CAPÍTULO 3. ANTECEDENTES

---

la relevancia y clasificación de dichas anomalías. Aunque estos sistemas han mostrado resultados prometedores, los actuales sistemas CADe presentan limitaciones debido a la alta tasa de falsos positivos, y los sistemas CADx para mamografía aún no están aprobados para uso clínico. Estos sistemas enfrentan grandes desafíos debido a que trabajan con datos médicos, los cuales, como ya hemos mencionado, cuentan con muchas restricciones. La mayoría han sido testeados con conjuntos de datos privados, por lo tanto, no reproducibles. También existe la problemática de que no hay un conjunto de datos que se utilice como prueba de referencia estandarizada para todos los sistemas, lo que hace que los porcentajes obtenidos no sean concluyentes ni comparables entre sí ([Lee y cols., 2017](#)).

Según un estudio publicado en la Revista Europea de Radiología por [Balleyguier y cols. \(2005\)](#), se evaluó el impacto de un sistema CAD en la capacidad de radiólogos junior y senior para detectar cánceres de mama en mamografías y determinar el potencial del CAD como herramienta de enseñanza en mamografía. Se encontró una mejora en la precisión de detección para ambos grupos. En el caso del radiólogo sénior, la precisión aumentó ligeramente del 76,9 % al 84,6 % tras el análisis con el sistema CAD, mientras que para el radiólogo junior la precisión mejoró significativamente del 61,9 % al 84,6 %. Estos hallazgos sugieren que los sistemas CAD pueden ser particularmente beneficiosos para los radiólogos menos experimentados, elevando su precisión a niveles comparables con los más experimentados. Una parte importante de los sistemas CAD, en la actualidad, es su papel en la enseñanza y formación de nuevos radiólogos.

A continuación, vamos a ver algunos de los sistemas CAD que se han desarrollado para su uso clínico.

Primeramente, la tabla [3.1](#) muestra las estadísticas de rendimiento de varios métodos CADe utilizados para la detección de anomalidades en mamografías. Por otro lado, la tabla [3.2](#) presenta las estadísticas de rendimiento de métodos CADx seleccionados para la clasificación de masas ya detectadas.

Cómo podemos ver entre los tipos de datos, la mayoría utiliza una fuente de datos

### 3.2. SISTEMAS CAD

privada y entre las fuentes de datos públicas encontramos *MIAS*<sup>2</sup> y *DDSM*<sup>3</sup>.

Autores	Casos	Tipo de datos	Precisión	Sensibilidad	Falsos positivos por imagen
Karssemeijer y te Brake (1996)	50	Público (MIAS)	NA	90 %	1
Mudigonda y cols. (2001)	56	Público (MIAS)	NA	81 %	2,2
Liu y cols. (2001)	38	Público (MIAS)	NA	90 %	1
Li y cols. (2002)	94	Privado	NA	91 %	3,21
Baum y cols. (2002)	63	Privado	NA	89 %	0,61
Kim y cols. (2006)	83	Privado	NA	96 %	0,2
Yang y cols. (2007)	203	Privado	96.1 %	95–98 %	1,8
The y cols. (2009)	123	Privado	NA	94 %	2,3 por caso
Sadaf y cols. (2011)	127	Privado	NA	91 %	NA
Chu y cols. (2015)	230	Público (DDSM)	NA	98.5 %	0.84

Tabla 3.1: Estadísticas de rendimiento de sistemas CADe seleccionados para la detección de anomalías (Lee y cols., 2017).

Las tablas 3.1 y 3.2 destacan el pequeño tamaño de los conjuntos de datos y las diferencias en las métricas de rendimiento reportadas, subrayando la necesidad de establecer pruebas de referencia estandarizadas para una evaluación comparativa fiable y reproducible. A pesar de esto, hay evidencias de que la detección del cáncer por parte de radiólogos utilizando estos sistemas está en torno al 85 % (Balleyguier y cols., 2005).

También hay que tener en cuenta que estos datos son del 2017, por lo que es probable que haya sistemas CAD más precisos en la actualidad.

<sup>2</sup>MIAS Mammographic Imaging Analysis Society.

<sup>3</sup>DDSM Digital Database for Screening Mammography.

## CAPÍTULO 3. ANTECEDENTES

---

<b>Autores</b>	<b>Casos</b>	<b>Tipo de datos</b>	<b>Precisión</b>
Brzakovic y cols. (1990)	25	Privado	85 %
Huo y cols. (1998)	65	Privado	NA
Rangayyan y cols. (2000)	54	Público (MIAS) y Privado	91 %
Mudigonda y cols. (2000)	39	Público (MIAS)	82,1 %
Sahiner y cols. (2001)	102	Privado	NA
Timp y cols. (2007)	465	Privado	NA
Ganesan y cols. (2014)	282	Privado	88,8 %
Görgel y cols. (2015)	78, 65	Privado, Público (MIAS)	91,4 %, 90,1 %
Qiu y cols. (2016)	560	Privado	77,14 %
Choi y cols. (2016)	600	Público (DDSM)	NA

Tabla 3.2: Estadísticas de rendimiento de sistemas CADx seleccionados para la clasificación de masas (Lee y cols., 2017).



### *3.2. SISTEMAS CAD*

---

# Capítulo 4

## Herramientas *software* y *hardware*

Para el desarrollo de este trabajo de fin de grado, se han utilizado diversas herramientas, tanto de *software* como de *hardware*, que han sido esenciales para la realización de los experimentos y la implementación de los modelos de aprendizaje profundo. A continuación, se describen detalladamente las herramientas utilizadas.

### 4.1. *Hardware*

El *hardware* empleado en este proyecto ha sido clave para satisfacer los requerimientos computacionales que implica el entrenamiento de modelos de *deep learning* con datos voluminosos y complejos como las mamografías. Las características del *hardware* utilizado son las siguientes:

#### 4.1.1. Unidad de Procesamiento Gráfico, *GPU*

Se ha utilizado la tarjeta gráfica *NVIDIA GeForce RTX 3070 Ti* de la marca *msi*, la cual se puede ver en la figura 4.1. Esta *GPU*<sup>1</sup>, es parte de la serie *RTX 30* de *NVIDIA*, basada en la arquitectura *Ampere*.

Sus especificaciones principales son:

- *CUDA cores*: 6.144.

---

<sup>1</sup>*GPU*: *Graphics Processing Unit*.

## 4.1. HARDWARE



Figura 4.1: NVIDIA GeForce RTX 3070 Ti ([Best Buy, 2024](#)).

- **Memoria:** 8 GB GDDR6X.
- **Ancho de banda de memoria:** 608 GB/s
- **Frecuencia de reloj:** 1.575 MHz (base) / 1.770 MHz (boost)

Aunque no es una GPU de las más potentes, y cuenta con limitaciones en la cantidad de memoria RAM disponible, su uso ha facilitado significativamente el proceso de entrenamiento de los modelos *Inception v3* y *ResNet*. Sin ella habría sido necesario el uso de alguna plataforma como *Google Colab*, la cual se intentó utilizar múltiples veces y dio muchos problemas provenientes del límite de tiempo que estas tienen. Haciendo más incómodo el proceso iterativo de diseño y entrenamiento de los modelos.

### 4.1.2. Almacenamiento

Para el almacenamiento de datos y modelos, se han utilizado dos tipos de dispositivos de almacenamiento:

- **Disco duro (HDD) de 5 Terabytes:** este dispositivo se ha utilizado principalmente para el almacenamiento de datos crudos, código y modelos entrenados. Su gran capacidad ha permitido almacenar grandes volúmenes de datos de mamografías, así como múltiples versiones de los modelos y sus respectivos *checkpoints*. Debido a que los datos crudos ocupan mas de 2

*Terabytes* el uso de este disco dura ha jugado una parte fundamental durante la obtención y procesado de los datos.

- **Unidad de estado sólido (SSD) de 500 Gigabytes:** este dispositivo se ha utilizado para el almacenamiento de datos que requieren un acceso rápido y frecuente durante el proceso de entrenamiento, así como para la instalación de *software* crítico. La alta velocidad de lectura y escritura del SSD ha mejorado considerablemente el rendimiento del sistema durante los procesos intensivos en la fase de entrenamiento.

## 4.2. *Software*

El *software* utilizado en este proyecto incluye varias librerías y entornos de desarrollo que han facilitado la implementación y el entrenamiento de los modelos de *deep learning*. A continuación se van a remarcar algunas de las herramientas *software* más importantes:

### 4.2.1. *Anaconda*

*Anaconda* es una distribución de *Python* y *R* para la ciencia de datos y el aprendizaje automático que simplifica la gestión de paquetes y entornos. En este proyecto, *Anaconda* ha sido utilizado para gestionar las dependencias y crear un entorno de trabajo aislado, asegurando la reproducibilidad y estabilidad del entorno de desarrollo. Las principales ventajas de utilizar *Anaconda* incluyen:

- **Gestión de dependencias:** facilitar la instalación y actualización de las librerías necesarias para el proyecto, como *TensorFlow*, *Numpy* y otras dependencias.
- **Entornos virtuales:** permite la creación de entornos virtuales específicos para el proyecto, evitando conflictos entre distintas versiones de librerías.

## 4.2. SOFTWARE

### 4.2.2. CUDA

*CUDA*<sup>2</sup> es una plataforma de computación paralela y una API de programación creada por *NVIDIA*. *CUDA* permite a los desarrolladores utilizar las *GPU* de *NVIDIA* para el procesamiento general en una amplia variedad de aplicaciones. En este proyecto, *CUDA* ha sido utilizado en combinación con *TensorFlow* para acelerar el entrenamiento de los modelos de aprendizaje profundo, aprovechando la capacidad de procesamiento paralelo de la *GPU*.

### 4.2.3. cuDNN

*cuDNN*<sup>3</sup> es una biblioteca *GPU-accelerated* de primitivas para *Deep Learning* desarrollada por *NVIDIA*. *cuDNN* proporciona implementaciones altamente optimizadas de rutinas para redes neuronales profundas, diseñadas para maximizar el rendimiento de los modelos en las *GPU NVIDIA*. En este proyecto, *cuDNN* ha sido utilizado junto con *TensorFlow* y *CUDA* para optimizar y acelerar el entrenamiento de los modelos de *Deep Learning*. Las principales características de *cuDNN* que han sido aprovechadas en este trabajo incluyen:

- **Optimización de la GPU:** proporciona implementaciones optimizadas de operaciones comunes en *Deep Learning*, como convoluciones y *pooling*, mejorando significativamente el rendimiento en las *GPU*.
- **Compatibilidad:** se integra fácilmente con *frameworks* de *Deep Learning* como *TensorFlow*, facilitando su uso y maximizando el rendimiento sin necesidad de modificaciones significativas en el código.

### 4.2.4. TensorFlow

Se ha utilizado la librería *TensorFlow* para la construcción y el entrenamiento de los modelos de aprendizaje profundo. *TensorFlow* es una librería de código abierto

<sup>2</sup>*CUDA: Compute Unified Device Architecture.*

<sup>3</sup>*cuDNN: CUDA Deep Neural Network library.*

desarrollada por *Google*, ampliamente utilizada en la comunidad de *Machine Learning* por su flexibilidad y capacidad para manejar modelos complejos. Las principales características de *TensorFlow* que han sido aprovechadas en este trabajo incluyen:

- **Compatibilidad con las GPU:** *TensorFlow* permite la utilización de las *GPU* para acelerar el entrenamiento de los modelos, lo que ha sido fundamental dada la gran cantidad de datos y la complejidad de los modelos utilizados.
- **Keras:** *Keras* es una biblioteca de alto nivel para la construcción y el entrenamiento de modelos de *Deep Learning*, que está integrada dentro de *TensorFlow*. *Keras* proporciona una interfaz simplificada y fácil de usar que permite definir y entrenar modelos complejos de manera rápida y eficiente. Entre las principales características de *Keras* se encuentran:
  - **Simplicidad y facilidad de uso:** *Keras* está diseñado para ser intuitivo y accesible, lo que permite a los desarrolladores construir y entrenar modelos de *Deep Learning* con pocas líneas de código.
  - **Modularidad:** *Keras* permite construir modelos de *Deep Learning* utilizando bloques modulares y reutilizables, facilitando la experimentación y la optimización del diseño de los modelos.
  - **Flexibilidad:** aunque *Keras* proporciona una interfaz de alto nivel, también permite acceder a funcionalidades de bajo nivel en *TensorFlow*, proporcionando la flexibilidad necesaria para realizar ajustes y optimizaciones detalladas en los modelos.



## *4.2. SOFTWARE*

---

# Capítulo 5

## Análisis de los datos

En este capítulo, se presentarán y analizarán los conjuntos de datos utilizados mas a delante para la detección de cáncer de mama mediante *deep learning*. Los datos han sido obtenidos de dos fuentes principales: la colección *Breast Cancer Screening-DBT* y la base de datos *CMMMD*. Se detallará la procedencia, estructura y características de cada conjunto de datos, así como el proceso de categorización y etiquetado realizado por expertos.

### 5.1. Fuentes de datos

Como se ha mencionado anteriormente los datos han sido obtenidos de la web *The Cancer Imaging Archive (TCIA)*<sup>1</sup>. En concreto se han utilizado dos colecciones de imágenes distintas que explicaremos a continuación, *Breast Cancer Screening (DBT)*, de Buda y cols. (2020), y *The Chinese Mammographies Database (CMMMD)*, de Sawyer-Lee y cols. (2016).

---

<sup>1</sup>*The Cancer Imaging Archive (TCIA)*: <https://www.cancerimagingarchive.net/>

## 5.1. FUENTES DE DATOS

### 5.1.1. Fuente de datos *Breast Cancer Screening (DBT)*

El conjunto de datos *DBT*<sup>2</sup>, utilizado en el estudio de *Breast Cancer Screening* Buda y cols. (2020), se basa en la técnica de tomosíntesis digital en lugar de imágenes de proyección tradicionales, como es el caso de la mamografía convencional. Tal y como se describió en la sección **Tomosíntesis digital de mama** del capítulo 2, la tomosíntesis digital es una técnica de imagen tridimensional que produce múltiples proyecciones desde diferentes ángulos durante un breve escaneo.

En este procedimiento, se toman varias radiografías de un pecho inmóvil y comprimido desde distintos ángulos, generando una serie de imágenes conocidas como “rebanadas”. Estas rebanadas pueden visualizarse individualmente o en secuencia, de manera similar a un video, permitiendo una inspección detallada de las estructuras internas del tejido mamario (Mariño, 2022).

Cada caso en el conjunto de datos incluye al menos uno de los cuatro tipos de volúmenes *DBT*: craneocaudal izquierdo (*LCC*), craneocaudal derecho (*RCC*), oblicuo mediolateral izquierdo (*LMLO*) y oblicuo mediolateral derecho (*RMLO*). Estos volúmenes corresponden a las diferentes vistas anatómicas obtenidas durante el procedimiento de tomosíntesis digital, proporcionando una cobertura completa del tejido mamario para una evaluación más precisa.

En la figura 5.1, obtenida de Buda y cols. (2020), podemos a ver un ejemplo de un volumen *DBT* para un caso *LCC*, el cual contiene 70 rodajas, con las que la técnica *DBT* consigue crear este efecto tridimensional.

Este conjunto de datos fue aprobado por la junta de revisión institucional del Sistema de Salud de la Universidad de Duke, y toda la información relativa al mismo se ha obtenido del artículo escrito en *JAMA NETWORK*, “*A Data Set and Deep Learning Algorithm for the Detection of Masses and Architectural Distortions in Digital Breast Tomosynthesis Images*” (Buda y cols., 2021).

Los volúmenes *DBT* de este conjunto de datos están identificadas mediante las fechas de estudio y los números de pacientes obtenidos de los informes radiológicos.

<sup>2</sup>*DBT: Digital Breast Tomosynthesis*

## CAPÍTULO 5. ANÁLISIS DE LOS DATOS

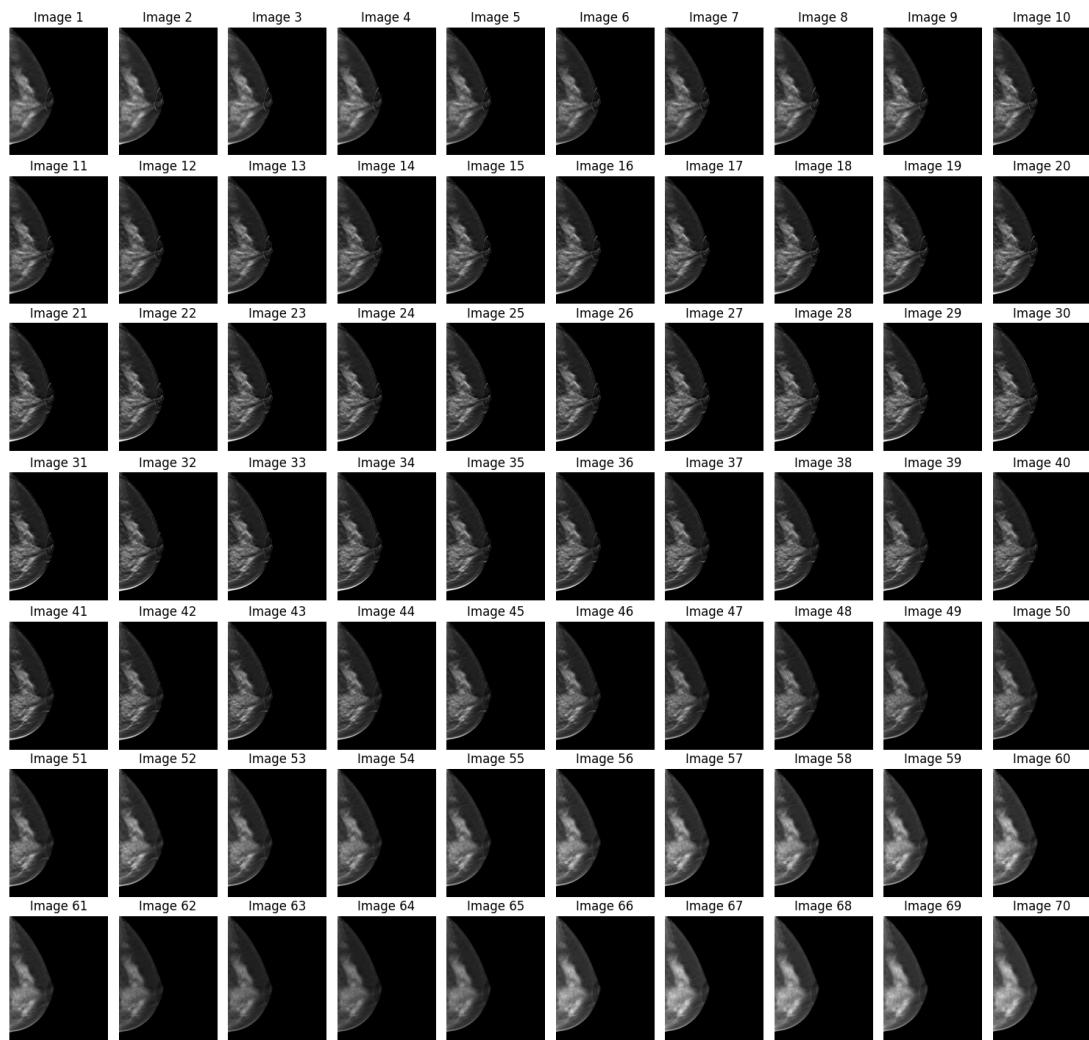


Figura 5.1: Volumen DBT-P00002 LCC, el cual cuenta con 70 rodajas (Buda y cols., 2020).

Este conjunto contiene más de 22.000 volúmenes *DBT* pertenecientes a 5.060 pacientes.

Este conjunto de datos cuenta con las siguientes cuatro categorías:

- *Normal*: con 5.129 casos representa el **91,42 %** de los datos.
- *Actionable*: con 280 casos representa el **5 %** de los datos.
- *Benign*: con 112 casos representa el **2 %** de los datos.
- *Cancer*: con 89 casos representa el **1,58 %** de los datos.

## 5.1. FUENTES DE DATOS

En la siguiente figura 5.2 podemos ver como los casos se están distribuidos en estas cuatro categorías de forma gráfica.

Se puede comprobar que la categoría *normal* cuenta con una gran sobre representación en comparación con el resto de categorías; esto es, algo ayudará a la hora de probar los diferentes modelos, pues en un aplicación real la mayoría de los pacientes son sanos. Sin embargo, el problema con esta colección es que contiene una cantidad muy escasa de casos de las catgorías *cancer* y *benign*, lo cual genera un problema que se solucionará, en parte, con el conjunto de datos *The Chinese Mammographies Database (CMMDB)* que se explicará después de éste.

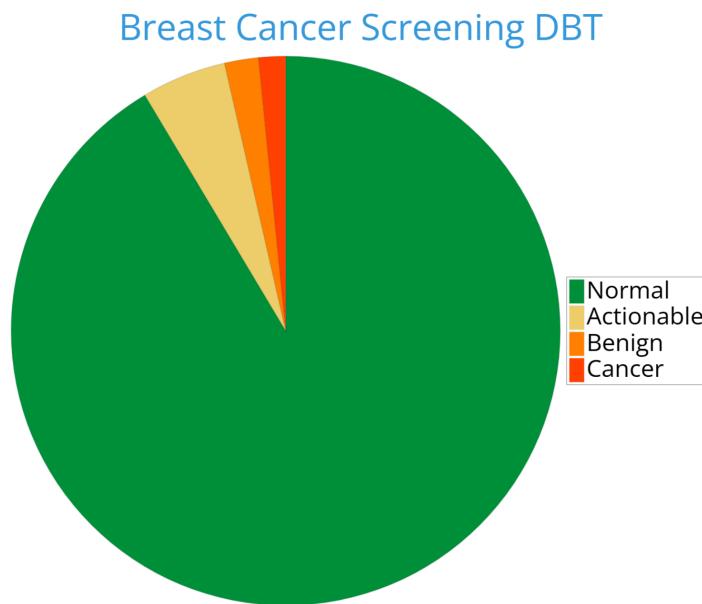


Figura 5.2: Diagrama circular con la distribución de casos para el conjunto *Breast Cancer Screening DBT*.

Los datos han sido categorizados mediante dos radiólogos de la universidad de Duke, Ruth Walsh y Sujata Ghate, con 18 y 25 años de experiencia respectivamente, los cuáles, contando con los informes radiológicos y patológicos correspondientes, identificaron los tumores con un cuadrado; para ello utilizaron un *software* personalizado desarrollado para cada caso de biopsia. En la figura 5.3 podemos ver un ejemplo de un cuadrado para uno de los volúmenes *DBT*.

Además, esta colección esta dividida en tres subcolecciones, *Training*, *Validation*

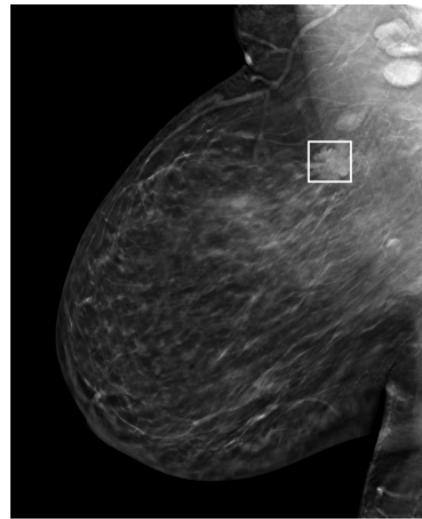


Figura 5.3: DBT-P00431 LCC Box ([Buda y cols., 2020](#)).

y *Test*. Las tres colecciones están etiquetadas mediante diferentes archivos CSV. A continuación, vamos a ver que información contiene cada uno. Como las tres subcolecciones contienen los mismos tipos de CSV vamos a explicar solo el formato de cada tipo de CSV.

#### Archivo *BCS-DBT-file-paths-subcolección-v2.csv*

Tiene la finalidad de identificar los volúmenes *DBT* con la ruta de descarga, permitiendo el uso de código para realizar el procesamiento de los mismos. La siguiente tabla 5.1 muestra las primeras cuatro filas que contiene este archivo CSV.

PatientID	StudyUID	View	Descriptive Path	Classic Path
DBT-P00013	DBT-S00163	rmlo	DBT...-32081/1-1.dcm	DBT...2081/1-1.dcm
DBT-P00023	DBT-S04378	lcc	DBT...-14072/1-1.dcm	DBT...4072/1-1.dcm
DBT-P00023	DBT-S04378	lmlo	DBT...-51654/1-1.dcm	DBT...1654/1-1.dcm
DBT-P00023	DBT-S04378	rcc	DBT...-09387/1-1.dcm	DBT...9387/1-1.dcm

Tabla 5.1: Primeras cuatro filas de *BCS-DBT-file-paths-subcolección-v2.csv* ([Buda y cols., 2020](#)).

A continuación, se explicarán las diferentes columnas de la tabla 5.1:

- *PatientID* y *StudyUID*: sirven para identificar el volumen *DBT* con el caso al

## 5.1. FUENTES DE DATOS

---

cual pertenece.

- *View*: tipo de vista que se utiliza en el volumen *DBT* (*LCC*<sup>3</sup>, *RCC*<sup>4</sup>, *LMLO*<sup>5</sup> o *RMLO*<sup>6</sup>).
- *Descriptive Path*: muestra la ruta donde se encuentra el volumen de tomosíntesis en caso de que haya sido descargado con la ruta descriptiva.
- *Classic Path*: muestra la ruta donde se encuentra el volumen de tomosíntesis en caso de que haya sido descargado con la ruta clásica.

### Archivo *BCS-DBT-labels-subcolección-v2.csv*

Identifica cada volumen *DBT* con la categoría al que pertenece, permitiendo también el uso de código para su procesamiento.

La tabla 5.2 muestra las primeras cinco filas que contiene este archivo CSV.

PatientID	StudyUID	View	Normal	Actionable	Benign	Cancer
DBT-P00013	DBT-S00163	rmlo	0	0	1	0
DBT-P00023	DBT-S04378	lcc	0	1	0	0
DBT-P00023	DBT-S04378	lmlo	0	1	0	0
DBT-P00023	DBT-S04378	rcc	0	1	0	0
DBT-P00023	DBT-S04378	rmlo	0	1	0	0

Tabla 5.2: Primeras cinco filas de *BCS-DBT-labels-subcolección-v2.csv* (Buda y cols., 2020).

A continuación, se explicarán las diferentes columnas de la tabla 5.2:

- *PatientID* y *StudyUID*: sirven para identificar el volumen *DBT* con el caso al cual pertenece.
- *View*: tipo de vista que se utiliza en el volumen *DBT* (*LCC*, *RCC*, *LMLO* o *RMLO*).
- *Normal*, *Actionable*, *Benign* y *Cancer*: Identifican la categoría a la que pertenece el volumen *DBT* siguiendo el formato *one-hot*<sup>7</sup>.

<sup>3</sup>*LCC*: Left Craniocaudal.

<sup>4</sup>*RCC*: Right Craniocaudal.

<sup>5</sup>*LMLO*: Left Mediолateral Oblique.

<sup>6</sup>*RMLO*: Right Mediолateral Oblique.

<sup>7</sup>El formato *one-hot* (también conocido como *one-hot encoding*) es una técnica de representación de

## CAPÍTULO 5. ANÁLISIS DE LOS DATOS

---

### Achivo ***BCS-DBT-boxes-train-v2.csv***

Los volúmenes *DBT* de las categorías *actionable*, *benign* y *bancer* tienen el tumor identificado mediante un cuadrado, para el cual se ha especificado la rodaja a la que pertenece dentro del volumen tridimensional y sus características. Toda la información sobre estos cuadrados, a demás de algunas columnas adicionales, se encuentra en este archivos.

Las tablas 5.3 y 5.4, muestran las primeras cinco filas que contiene este archivo CSV. Se ha divido en dos tablas para mejorar su visualización en el documento, para ello se ha identificado cada fila con *PatientID* y *StudyUID*.

PatientID	StudyUID	View	Slice	X	Y	Width	Height
DBT-P00013	DBT-S00163	rmlo	16	1116	1724	218	105
DBT-P00024	DBT-S03255	lcc	19	267	488	64	64
DBT-P00024	DBT-S03255	lcc	37	169	384	93	85
DBT-P00024	DBT-S03255	lmlo	11	471	1060	67	56
DBT-P00024	DBT-S03255	lmlo	12	640	673	93	80

Tabla 5.3: Primeras cinco filas del archivo CSV proporcionado - Parte 1 ([Buda y cols., 2020](#)).

PatientID	StudyUID	View	Class	VolumeSlices
DBT-P00013	DBT-S00163	rmlo	benign	49
DBT-P00024	DBT-S03255	lcc	benign	73
DBT-P00024	DBT-S03255	lcc	benign	73
DBT-P00024	DBT-S03255	lmlo	benign	81
DBT-P00024	DBT-S03255	lmlo	benign	81

Tabla 5.4: Primeras cinco filas del archivo CSV proporcionado - Parte 2 ([Buda y cols., 2020](#)).

A continuación, se explicarán las diferentes columnas de las dos tablas 5.3 y 5.4:

- *PatientID* y *StudyUID*: sirven para identificar el volumen *DBT* con el caso al cual pertenece.

datos categóricos como vectores binarios, donde cada categoría se representa con un vector de longitud igual al número total de categorías, con un único elemento en 1 y los demás en 0.

## 5.1. FUENTES DE DATOS

- *View*: tipo de vista que se utiliza en el volumen *DBT* (*LCC*, *RCC*, *MLLO* o *RLLO*).
- *Slice*: rodaja del volumen de tomosíntesis en donde los radiólogos colocaron la caja rectangular.
- *X* y *Y*: coordenadas de la caja rectangular dentro de la rodaja.
- *Width* y *Height*: dimensiones de la caja rectangular.
- *Class*: categoría del tumor *Actionable*, *Benign* o *Cancer*.
- *VolumeSlices*: número de rodajas total del volumen de tomosíntesis.

### 5.1.2. *The Chinese Mammographies Database (CMMMD)*

El conjunto de datos *CMMMD*<sup>8</sup> es el resultado de un proyecto desarrollado por la Escuela de Ciencias de la Computación e Ingeniería de la Universidad Tecnológica del Sur de China con el apoyo del área de Investigación y Desarrollo de la Provincia de Guangdong, la Fundación Nacional de Ciencias Naturales de China, la Fundación de Ciencias Naturales de Guangdong y el Proyecto de Innovación Colaborativa de Salud y Medicina de la Ciudad de Guangzhou. Contiene 1.775 pacientes de China con enfermedad mamaria que se sometieron a un examen de mamografía entre julio de 2012 y enero de 2016. La base de datos consta de 3.728 mamografías de estas 1.775 pacientes, con biopsia confirmada tipo de tumores benignos o malignos. Cuenta, por lo tanto, con dos etiquetas, *benign*, que hace referencia a un tumor benigno, y *malignant*, que hace referencia a un tumor maligno, el cual corresponde con un caso de cáncer. Además, en esta ocasión no se trata de imágenes tridimensionales, sino de mamografías digitales. Las mamografías digitales se explicaron en la sección **Mamografía digital** del capítulo **Fundamentos médicos y tecnológicos del cáncer de mama**.

En la figura 5.4 podemos observar la distribución de los datos, teniendo 495 casos etiquetados como *benign* y 1310 casos etiquetados como *malignant*. La categoría de *malignant* corresponde con la categoría de *cancer* de la anterior colección *Breast*

<sup>8</sup>CMMMD: *Chinese Mammographies Database*.

## CAPÍTULO 5. ANÁLISIS DE LOS DATOS

*Cancer Screening DBT.* Esto nos dará una mayor cantidad de estas dos categorías que complementarán los datos del conjunto de datos **Breast Cancer Screening DBT**.

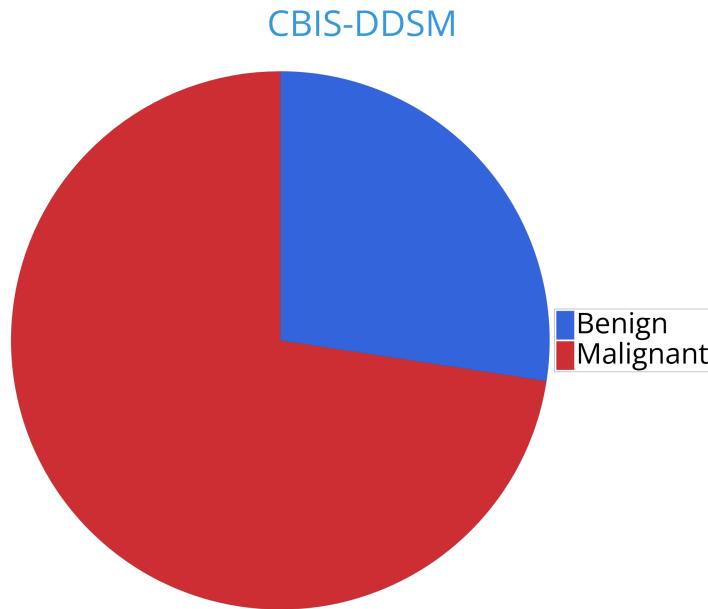


Figura 5.4: Distribución de casos en el conjunto **CMMD**.

A continuación, se explicará como la información relativa a los datos de este conjunto se encuentra contenida en un solo archivo.

### Archivo **CMMD\_clinicaldata\_revision.xlsx**

En este archivo se especifican las diferentes características de cada caso. En esta ocasión las diferentes vistas (*LCC*, *RCC*, *LMLO* o *RMLO*) están contenidas en un solo archivo y solo se utiliza un identificador para cada caso. Además, cada mamografía puede tener entre una y cuatro vistas, lo qué presenta una problemática al saber que vistas están contenidas en cada caso.

En la figura 5.5 podemos ver un ejemplo de un caso del conjunto de datos **CMMD**, en el cual, tenemos a todas las vistas posibles, *LCC*, *LMLO*, *RCC* y *RMLO*.

## 5.1. FUENTES DE DATOS

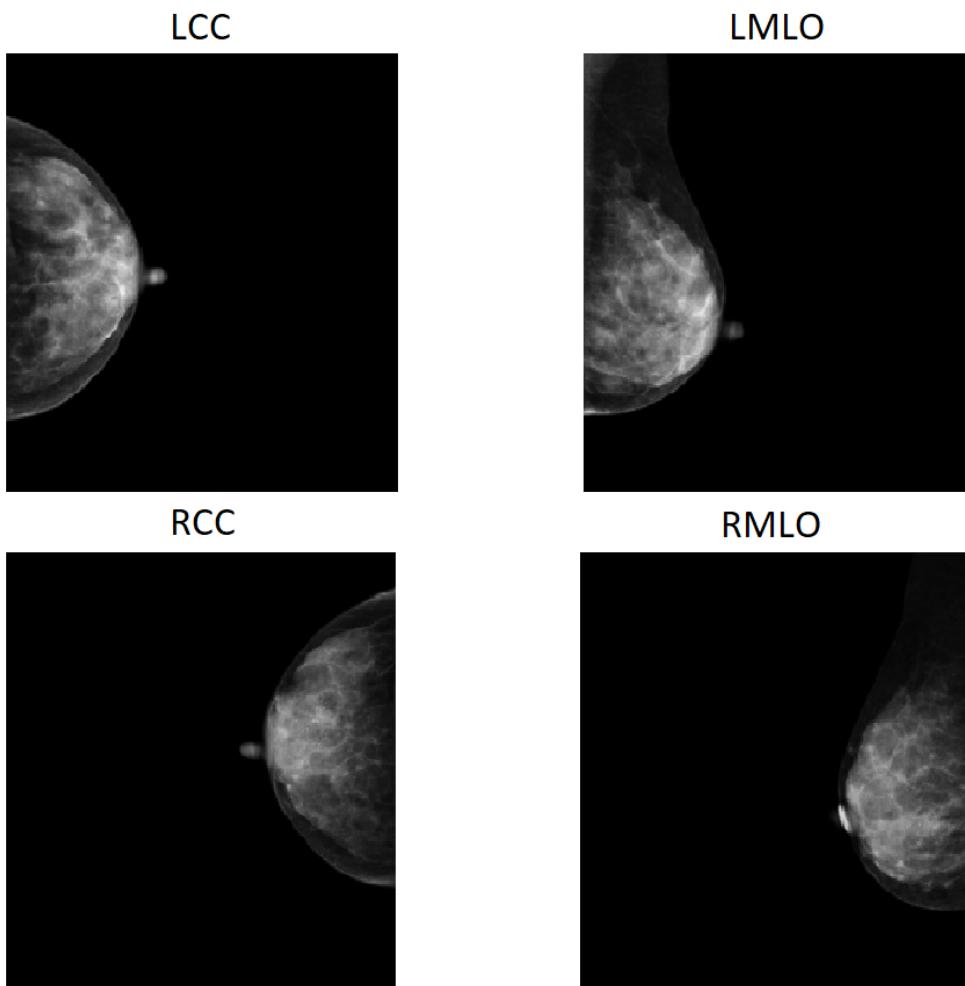


Figura 5.5: Caso *malignant* del conjunto *CMMD* ([Sawyer-Lee y cols., 2016](#)).

ID1	LeftRight	Age	number	abnormality	classification	subtype
D1-0001	R	44	2	calcification	Benign	
D1-0002	L	40	2	calcification	Benign	
D1-0003	L	39	2	calcification	Benign	
D1-0004	L	41	2	calcification	Benign	
D1-0005	R	42	2	calcification	Benign	

Tabla 5.5: Primeras cinco filas del archivo *XLSX* ([Sawyer-Lee y cols., 2016](#)).

La tabla 5.5 muestra las primeras cinco filas que contiene este archivo *XSLX*, donde sus diferentes elementos se explican como:

- *ID1*: identificador único del caso.
- *LeftRight*: indica si el tumor se encuentra en el seno izquierdo (L) o derecho (R).

## CAPÍTULO 5. ANÁLISIS DE LOS DATOS

---

- *Age*: edad de la paciente.
- *number*: número de anormalidades encontradas.
- *abnormality*: tipo de anormalidad detectada (por ejemplo, calcificación).
- *classification*: clasificación de la anormalidad (*Benign* o *Malignant*).
- *subtype*: subtipo de la anormalidad, si aplica.



### *5.1. FUENTES DE DATOS*

---

# Capítulo 6

## Procesado de datos

En este capítulo se mostrará como se ha realizado la obtención y preprocesado de los dos conjuntos de datos utilizados. Además, se abordará el proceso de análisis de datos, realizando todas las operaciones oportunas para realizar la unión de dos conjuntos de datos de diferentes fuentes y formato. Por último, se explicará los pasos y decisiones tomadas para la creación de los conjuntos de entrenamiento y test que han sido utilizados por los diferentes modelos.

### 6.1. Detalles sobre la recolección de datos

Los datos se han descargado mediante el *software NBIA Data Retriever* proporcionado por *TCIA*<sup>1</sup>. Este *software* es una herramienta desarrollada específicamente para facilitar la descarga de grandes volúmenes de datos de imágenes biomédicas almacenadas en la plataforma *TCIA*.

*NBIA Data Retriever* permite seleccionar y descargar de manera cómoda grandes conjuntos de datos médicos. Una de sus ventajas es su capacidad para gestionar descargas interrumpidas. Esto ha sido de gran ayuda para descargar unos conjuntos de datos con un tamaño enorme, de varios Tbytes, permitiendo reanudar la descarga si, por ejemplo, se ha perdido la conexión.

---

<sup>1</sup>TCIA: *The Cancer Imaging Archive*: <https://www.cancerimagingarchive.net/>.

## 6.1. DETALLES SOBRE LA RECOLECCIÓN DE DATOS

Descargar y almacenar los datos fue una tarea ardua y prolongada, debido a que el volumen total de datos ascendía aproximadamente a 1,5 *TB*. El proceso de descarga se extendió aproximadamente por dos semanas de ejecución, y hay que tener en cuenta que se hizo de manera interrumpida.

### 6.1.1. Formato de los datos

Ambos conjuntos de datos, **Breast Cancer Screening DBT** y **CMMD** vienen en un formato *DICOM*<sup>2</sup>.

Para el conjunto de datos *Breast Cancer Screening DBT*, se ha utilizado el código facilitado por la web para la lectura de datos en *Github duke-dbt-data*<sup>3</sup>. Este repositorio contiene, entre otras herramientas, el método *dcmread\_image*, que permite leer la imagen y convertirla en un *numpy array*<sup>4</sup>, facilitando así su uso. Este método modifica la imagen según el tipo de vista (*LCC*, *RCC*, *RLM*, *LLM*) para asegurar que esté en la orientación correcta.

En el caso de *CMMD*, se ha implementado una versión más simple del método *dcmread\_image* anterior, logrando de igual forma convertir un archivo *DICOM* a un *numpy array*.

Cambiar el formato de los datos fue fundamental. Son imágenes en escala de grises, pero, utilizan 8 canales de bits. El primer canal representa la escala de grises, el segundo canal indica el tipo de tejido detectado (0 = nada, 1 = piel, 2 = tejido mamario), mientras que los otros canales están en 0 y no representan ninguna información útil. El hecho de utilizar 8 canales más la elevada resolución de píxeles hacia imposible trabajar con los datos en su formato original. Por ejemplo, cada archivo tardaba aproximadamente un minuto en cargarse en formato *DICOM* y los datos ocupaban más de 1,5 *TB*. Los archivos *DICOM* se transformaron a archivos *numpy array* serializados con un solo canal, que representa la escala de grises.

<sup>2</sup>*DICOM: Imaging and Communications in Medicine*, estándar para el manejo, almacenamiento, impresión y transmisión de información en imágenes médicas de cáncer de mama.

<sup>3</sup>*duke-dbt-data*: <https://github.com/mazurowski-lab/duke-dbt-data/tree/master>.

<sup>4</sup>*Numpy array*: estructura de datos del paquete *NumPy* de Python utilizada para manejar grandes matrices y realizar operaciones matemáticas complejas.

Este proceso fue realizado mediante un largo y minucioso análisis de los datos, ya que no hay documentación disponible que explique el formato de los datos.

## 6.2. Preprocesamiento de datos

Una vez transformado los datos a un formato más manejable, se desarrolló código en *Python* para adaptar estos datos para ser procesados por los modelos de entrenamiento. Es necesario considerar varios aspectos importantes:

- Los conjuntos de datos *Breast Cancer Screening DBT* y *CMMMD* no están en el mismo formato de datos, ya que *Breast Cancer Screening DBT* utiliza un archivo para cada volumen tridimensional, el cual representa una vista<sup>5</sup> del caso al que pertenece. Cada archivo *CMMMD* contiene las diferentes vistas utilizadas en cada paciente.
- En el conjunto *Breast Cancer Screening DBT* los datos están mezclados, por lo que es necesario separarlos según las diferentes etiquetas: *actionable*, *benign*, *cancer*, *normal*.
- Se debe juntar los conjuntos *Breast Cancer Screening DBT* y *CMMMD*.

A continuación, detallaremos cómo se han realizado todos estos pasos.

### 6.2.1. *Breast Cancer Screening DBT*

Empezaremos con el conjunto de datos **Breast Cancer Screening DBT**. Analizando los datos, encontramos varios problemas particulares en este conjunto, tales como la resolución desigual de las imágenes y que los *numpy arrays* no contienen la misma cantidad de rodajas<sup>6</sup>, por lo que se debe establecer un estándar entre los diferentes volúmenes *DBT*.

Las imágenes presentan una resolución muy elevada y desigual. Por ejemplo, en el modelo base de *Breast Cancer Screening DBT*, desarrollado por [Buda y cols. \(2021\)](#),

<sup>5</sup>Vistas: *LCC*, *RCC*, *LMLO* o *RMLO*.

<sup>6</sup>Rodajas: número de imágenes para la representación *DBT* en 3 dimensiones.

## 6.2. PREPROCESAMIENTO DE DATOS

redimensionaron las imágenes a una escala de  $96 \times 96$ . En este caso, en términos de almacenamiento y uso, se decidió almacenar las imágenes en una escala cuadrada de  $720 \times 720$ , la cual proporciona suficiente información y permitiría reducir aún más la resolución si fuese necesario. Además, se comprobó que no había ninguna imagen con un ancho o largo superior a 720, lo que justificó esta decisión.

Para solucionar el problema de incompatibilidad, producido por el uso de volúmenes 3D en este conjunto de datos, se utilizó la información contenida en el archivo *BCS-DBT-boxes-train-v2.csv*<sup>7</sup> para seleccionar la rodaja donde se ha detectado el cáncer y desechar las otras. Además, se separaron los *numpy arrays* en diferentes directorios según sus respectivas categorías: *actionable*, *benign*, *cancer*, *normal*.

El resultado actual de los datos es que ahora están organizados y estandarizados en cuanto a resolución y cantidad de rodajas, facilitando su procesamiento y análisis posterior.

Después de esto, procedimos a juntar las imágenes por caso. Los *numpy arrays* están etiquetados e identificados por el número de paciente y el número de estudio, pero cada caso puede tener entre uno y cuatro *numpy arrays*, dependiendo de si se trata de la vista *LCC*, *RCC*, *LMLO* o *RLMO*. Por lo tanto, juntamos las rodajas individuales que ya tenemos por caso, creando un *array* de 4 imágenes, en el que cada imagen representa una de las vistas. En los casos donde se tienen menos de cuatro vistas, se llenó con imágenes de valor cero.

En la figura 6.1 se puede ver un ejemplo para un paciente con cáncer, en cuyo caso, solo cuenta con dos imágenes, la imagen 1 y la imagen 3 corresponden con las vistas *LMLO* y *LCC* respectivamente, y las imágenes 2 y 4 están en negro, es decir, rellenas con ceros.

Con esto, se completó el preprocesamiento específico para el conjunto de datos *DBT*. A continuación, se explicará el tratamiento del conjunto de datos *CMMMD* y la unión de los dos conjuntos.

<sup>7</sup>*BCS-DBT-boxes-train-v2.csv*: archivo que contiene información sobre las rodajas específicas donde se detectó el cáncer. Explicado en la sección 5.1.1.

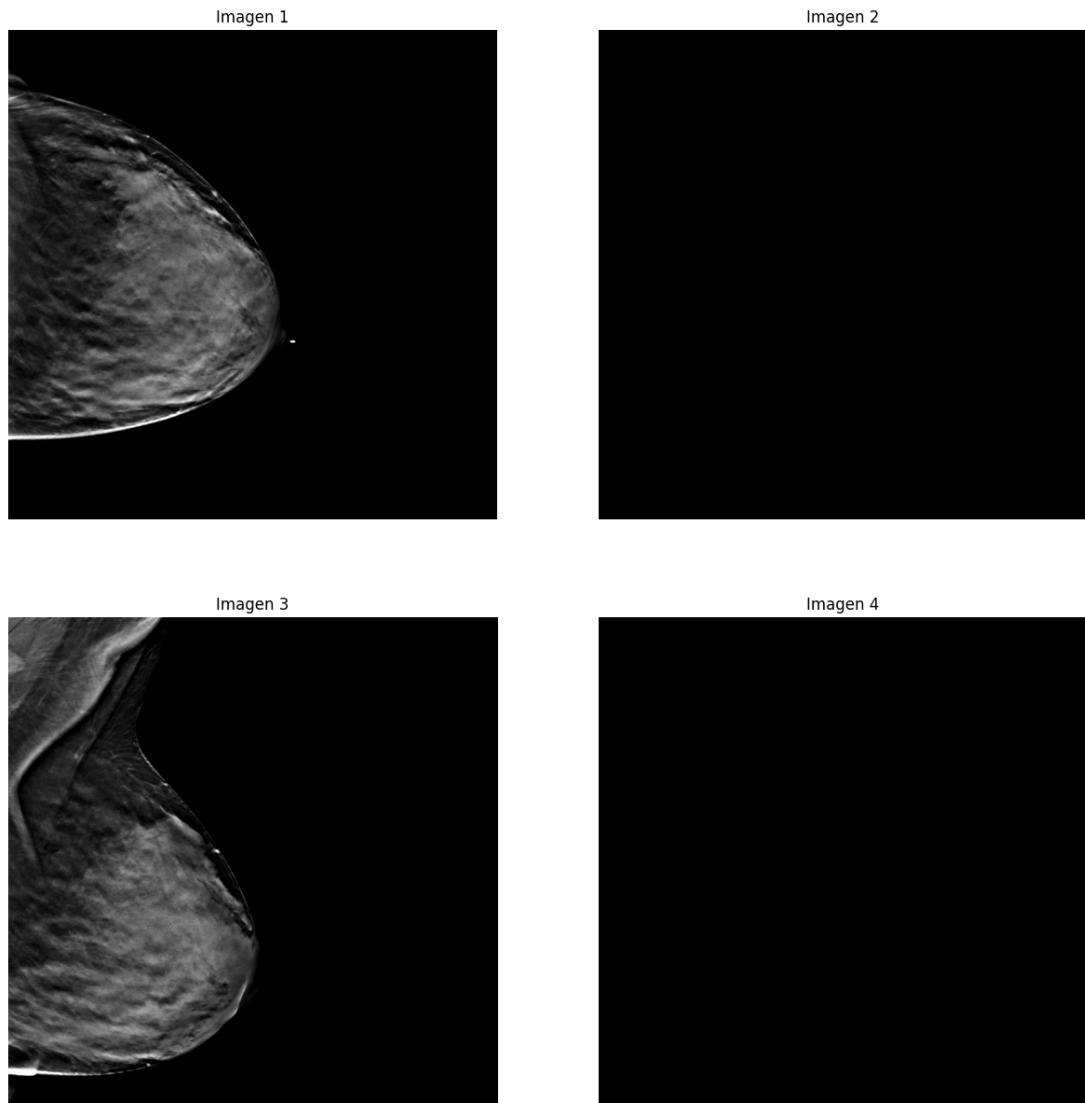


Figura 6.1: Ejemplo de un paciente con cáncer identificado con DBT-S00029\_DBT-P00659 ([Buda y cols., 2020](#)).

### 6.2.2. El conjunto de datos *CMMMD*

El conjunto de datos *CMMMD*<sup>8</sup> se diferencia del conjunto de datos *Breast Cancer Screening DBT* en que no incluye volúmenes 3D, sino únicamente imágenes 2D de mamografías. Este cambio de formato influye directamente en el proceso de preprocesamiento de datos.

En el caso del *CMMMD*, cada imagen de mamografía representa una vista en 2D

<sup>8</sup>*CMMMD*: *Chinese Mammography Database*, base de datos de mamografías chinas.

## 6.2. PREPROCESAMIENTO DE DATOS

donde se ha detectado el tumor. Por lo tanto, no es necesario realizar la selección de rodajas como en el caso del *DBT*. Sin embargo, todavía se requiere una organización y clasificación adecuadas de los datos para asegurar su utilidad en posteriores análisis y modelos de aprendizaje automático.

El primer paso en el preprocesamiento del conjunto de datos *CMMMD* fue convertir los archivos *numpy arrays* en una resolución cuadrada de  $720 \times 720$ , igual que en el caso *DBT* anterior. Esta estandarización de la resolución es crucial porque permite un manejo consistente de las imágenes durante el procesamiento y análisis. Además, la resolución  $720 \times 720$  es suficientemente alta para preservar los detalles importantes de las mamografías, facilitando tanto el almacenamiento eficiente como la posibilidad de reducir la resolución si es necesario para optimizar el rendimiento de los modelos de aprendizaje automático.

A continuación, las imágenes fueron divididas en dos grupos principales: *benign* y *malignant*. Esta clasificación se realizó utilizando la información contenida en el archivo *CMMMD\_clinicaldata\_revision.xlsx*<sup>9</sup>. Este archivo proporciona detalles clínicos esenciales sobre cada paciente, incluyendo la etiqueta de clasificación del tumor, lo que facilita la segregación de las imágenes en las categorías correctas. Las imágenes fueron organizadas en directorios separados según sus respectivas categorías: *benign* y *malignant*.

La figura 6.2 muestra un ejemplo de un paciente con cáncer del conjunto *CMMMD*. En esta figura se presentan cuatro imágenes correspondientes a diferentes vistas del paciente afectado, correspondiéndose las dos primeras a la mama izquierda y las dos últimas a la derecha. Estas vistas permiten una evaluación completa y detallada de la región sospechosa, lo que es crucial para un diagnóstico preciso.

El proceso de preprocesamiento del *CMMMD* no solo involucró la clasificación y conversión de imágenes, sino también la verificación de su calidad y consistencia. Se comprobó que todas las imágenes tuvieran una resolución uniforme y que los *numpy arrays* estuvieran correctamente formateados. Esto asegura que no haya

<sup>9</sup>*CMMMD\_clinicaldata\_revision.xlsx*: archivo que contiene información clínica detallada sobre cada caso, incluyendo la clasificación del tumor como benigno o maligno. Explicado en la sección 5.1.2.

## CAPÍTULO 6. PROCESADO DE DATOS

problemas técnicos durante el análisis y el desarrollo de modelos. En resumen, el preprocesamiento del conjunto de datos *CMMB* consistió en clasificar las imágenes en categorías bien definidas, convertirlas a un formato manejable (*numpy arrays*) y organizarlas de manera eficiente. Este conjunto de datos ahora está listo para ser utilizado en el desarrollo de modelos predictivos.

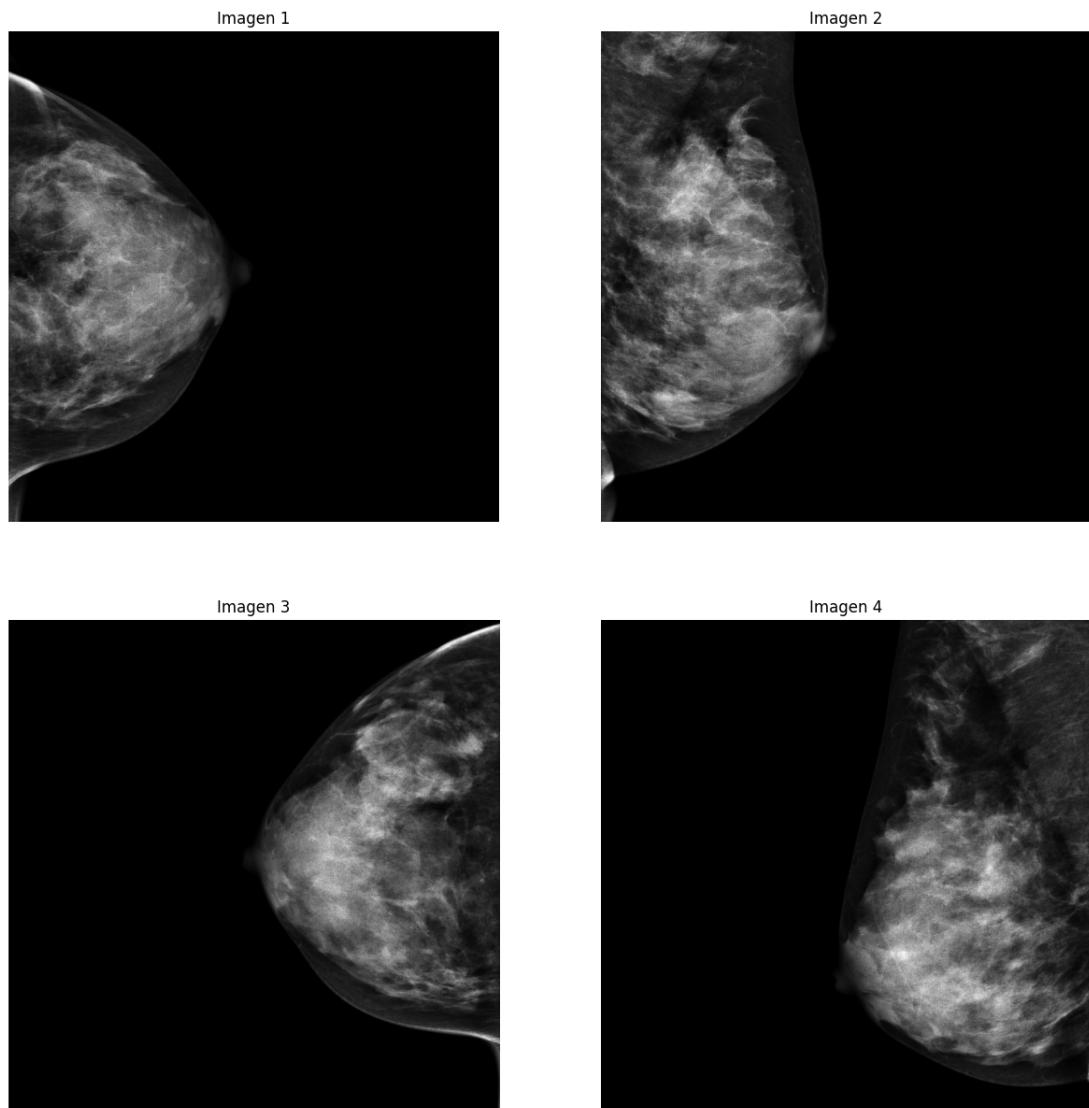


Figura 6.2: Ejemplo de un paciente con cáncer del conjunto *CMMB* ([Sawyer-Lee y cols., 2016](#)). La imagen muestra cuatro vistas diferentes del paciente afectado.

## 6.2. PREPROCESAMIENTO DE DATOS

### 6.2.3. Unión de los dos conjuntos de datos

Inicialmente, se escogió al conjunto de datos *Breast Cancer Screening DBT* para el entrenamiento de los modelos. Pero, debido a la necesidad de tener un conjunto de datos con una cantidad de casos suficiente en cada una de las categorías, se buscó otro conjunto de datos que fuese compatible. El conjunto *CMMMD* fue el escogido debido a que este muestra unas características similares. Por ejemplo, no tiene anotaciones en las mamografías, tiene una resolución parecida y se realizaron con el mismo formato de vistas<sup>10</sup>.

Los conjuntos de datos disponibles públicamente son muy escasos, además, suelen ser muy antiguos, teniendo la mayoría más de 20 años. Al ser la tomosíntesis digital de mama una tecnología moderna, es necesario buscar un conjunto compatible. Tras una larga investigación se ha llegado a la conclusión de que ambos tipos de datos son compatibles entre sí para su uso en un sistema *CAD*, tanto la mamografía digital, con datos en dos dimensiones, como la tomosíntesis digital de mama, con datos en tres dimensiones, siempre y cuando se realice un procesamiento a la tomosíntesis digital de mama para convertirla en datos de dos dimensiones.

Hay que tener en cuenta que es muy importante que sean conjuntos compatibles, pues si hubiese diferencias distinguibles entre ambos, los modelos podrían aprender a clasificar todos los datos del conjunto *CMMMD* en las categorías *cancer* o *benign*, y por lo tanto, no estarían detectando el cáncer, solo estarían aprendiendo a separar las fuentes de datos.

La hipótesis de que las dos fuentes de datos son compatibles, se constata en el capítulo 8 de **Resultados**, pues los modelos no categorizan los datos en función del conjunto del que provienen.

Para la unificación de los conjuntos de datos *CMMMD* y *Breast Cancer Screening DBT*, se procedió a combinar las categorías equivalentes de ambos conjuntos. En específico, se unieron los casos *benign* de *CMMMD* con los casos *benign* de *Breast Cancer Screening DBT* y los casos *malignant* de *CMMMD* con los casos *cancer* de

<sup>10</sup>Vistas: *LCC*, *RCC*, *LMLO*

### Breast Cancer Screening DBT.

Esta unificación permitió aumentar significativamente la cantidad de datos disponibles para cada categoría, los cuales resultaban insuficientes con solo el conjunto de datos *Breast Cancer Screening DBT*.

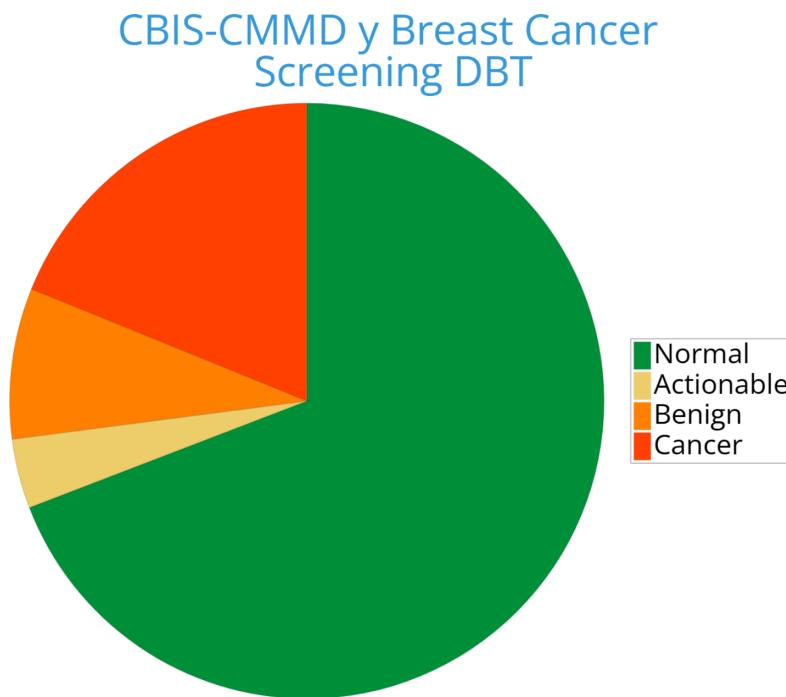


Figura 6.3: Casos combinados de *CBIS-CMMD* y *Breast Cancer Screening DBT*.

La figura 6.3 muestra la distribución de los casos tras la unificación de los dos conjuntos de datos. Esta integración ha permitido incrementar notablemente el número de casos en las categorías *Benign* y *Cancer*, logrando una distribución más equilibrada de los datos con un total de 7.415 casos diferentes:

- **Normal:** con 5.129 casos, representa el **69 %** de los datos.
- **Actionable:** con 280 casos, representa el **4 %** de los datos.
- **Benign:** con 607 casos, representa el **8 %** de los datos.
- **Cancer:** con 1.399 casos, representa el **19 %** de los datos.

Este incremento es particularmente beneficioso por las siguientes razones:

### 6.3. CONJUNTOS UTILIZADOS EN EL ENTRENAMIENTO DE MODELOS

1. Mejor Representación de Casos Diversos: al tener una mayor cantidad de datos en las categorías *benign* y *cancer*, los modelos de aprendizaje automático pueden aprender mejor las características distintivas de estas condiciones. Esto mejora la precisión y la capacidad del modelo para generalizar a nuevos datos.
2. Reducción de Desequilibrios: aunque la categoría *normal* sigue siendo predominante, la integración de más casos *benign* y *cancer* ayuda a reducir el desequilibrio entre clases. Un conjunto de datos más equilibrado contribuye a un mejor rendimiento del modelo, ya que evita que el algoritmo esté sesgado hacia la clase mayoritaria.
3. Variabilidad y Complejidad: la unificación de conjuntos de datos provenientes de diferentes fuentes y contextos introduce más variabilidad y complejidad en el conjunto de datos final. Esta diversidad es beneficiosa para desarrollar modelos que sean más resistentes a diferentes condiciones y variaciones presentes en datos del mundo real. Uno de los problemas más extendidos entre los sistemas *CAD* es que los datos suelen ser entrenados y probados con la misma fuente de datos. Esto hace que sus resultados no sean representativos del rendimiento que tendrían con otras fuentes.

El principal problema que se encuentra con este conjunto de datos es la escasa cantidad de casos para la categoría *actionable*. Además los tumores accionables son más pequeños y con formas más diversas que el resto, en las radiografías aparecen como pequeños puntos o hilos. Esto hace que esta categoría sea la más compleja y difícil de detectar para los modelos desarrollados.

## 6.3. Conjuntos utilizados en el entrenamiento de modelos

En esta sección se explicará la distribución de los datos y las técnicas utilizadas sobre los mismos para formar los conjuntos de datos finales. Estos serán los utilizados

por los diferentes modelos, tanto en la fase de entrenamiento como en la fase de testeo.

### 6.3.1. Conjunto de datos base

#### Selección de los datos

Para este conjunto se utilizaron 4.292 casos, de los cuales 280 pertenecen a la categoría *actionable*, 607 a la categoría *benign*, 1.399 a la categoría *cancer* y 2.006 a la categoría *normal*. Se ha recortado el número de casos normales para conseguir un mayor balance en el entrenamiento. La importancia de tener un balance en los datos de entrenamiento de un modelo radica en evitar el sesgo hacia la clase mayoritaria. Un modelo entrenado con un conjunto de datos desequilibrado puede aprender a predecir siempre la clase mayoritaria, ignorando las características de las clases minoritarias, lo que resulta en un desempeño pobre en generalización.

Al seleccionar 2.006 casos de la categoría *normal* se consigue que haya una cantidad de casos similar entre los pacientes sanos, correspondientes a las categorías *normal* o *actionable*, y los pacientes que pertenecen a las categorías *benign* o *cancer*. Con esta distribución se aumenta el numero de casos normales sin caer en un desbalance de clases demasiado elevado, manteniendo un equilibrio entre la diversidad de los datos y el desbalance de las clases, lo cual mejora su capacidad de detección y clasificación.

En la siguiente figura 6.4 se puede ver la distribución de estos datos, en comparación con la distribución observada anteriormente en la figura 6.3.

Para seleccionar los 2.006 casos de la categoría *normal*, se podrían haber realizado de forma aleatoria, pero, para asegurar la diversidad de los datos haciendo que el entrenamiento pueda generalizar mejor, se ha optado por seleccionar las mamografías con casos más diferentes.

Para medir la diferencia entre las distintas radiografías se ha utilizado el error absoluto medio (*MAE*) como métrica. Dadas dos imágenes la *MAE* cuantifica la diferencia promedio por píxel. El cálculo se realiza tomando el valor absoluto de la diferencia entre cada píxel correspondiente en dos imágenes y luego calculando el

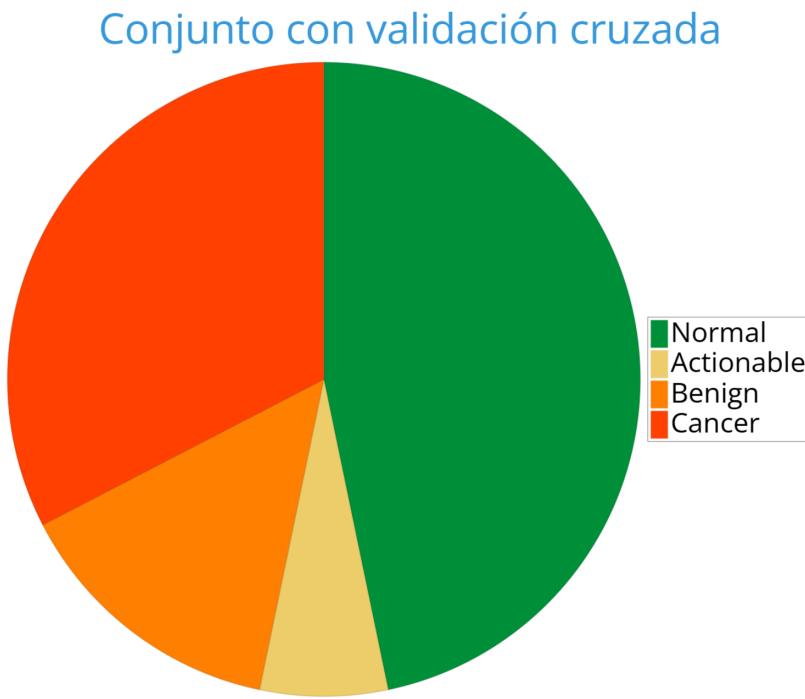


Figura 6.4: Distribución de los datos en el conjunto con validación cruzada.

promedio de todas estas diferencias absolutas, tal y como se puede ver en la siguiente fórmula:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |I_1(i) - I_2(i)|$$

donde:

- $N$  es el número total de píxeles en la imagen.
- $I_1(i)$  es el valor del  $i$ -ésimo píxel de la primera imagen.
- $I_2(i)$  es el valor del  $i$ -ésimo píxel de la segunda imagen.
- $|I_1(i) - I_2(i)|$  es el valor absoluto de la diferencia entre los valores del  $i$ -ésimo píxel de ambas imágenes.

Para realizar esta tarea se implementó un código en *Python* que calcula la *MAE* entre cada par de imágenes de la categoría *normal*. El cálculo de todas las imágenes es un proceso largo que llevó varios días de ejecución, por lo tanto los del cálculo del *MAE* se almacenaron en un archivo que contiene la lista serializada con todos los

resultados.

Una vez calculado el *MAE* de todas las imágenes, se ordena la lista con los resultados de mayor a menor y se seleccionaron las 2006 imágenes mas diferentes.

Este proceso es muy importante, debido a que cuanto mayor sea la diversidad de los casos, mayor será la generalización y capacidad de aprendizaje de los modelos. Especialmente si la cantidad de casos disponibles es reducida, como es en este caso. Características del paciente, como la densidad mamaria o el tamaño de la mama, pueden alterar mucho la sensibilidad, aumentando el numero de falsos positivos y bajando la precisión del modelo. Estas características están explicadas en el capítulo 2, **Fundamentos médicos y tecnológicos del cáncer de mama**.

## División

Con los datos ya seleccionados, se procedió a su división para su uso en las diferentes fases. Primero se dividieron entre datos usados para la fase de entrenamiento y datos usados para la fase de test. El conjunto de prueba cuenta con el 20% de los casos de cada categoría, sumando un total de 856 casos. Por lo tanto, se utilizará el 80% restante para el entrenamiento. La figura 6.5 muestra una representación gráfica de la división de ambos conjuntos.



Figura 6.5: División de los datos entre la fase de entrenamiento y de *test*.

Durante el entrenamiento se utilizará la técnica de validación cruzada. Es por esto que no se ha creado otro conjunto de validación. El proceso de validación cruzada se explicará mas adelante en el capítulo 7, **Desarrollo del proyecto**.

#### Formato

Para asegurar la calidad y consistencia de los datos, se realizó una verificación exhaustiva después de la división. Este proceso incluyó revisar que no hubiera fugas de datos<sup>11</sup> entre los subconjuntos, garantizando que los datos estuvieran correctamente etiquetados y formateados. Este paso es crucial para mantener la integridad de los resultados y asegurar que el modelo no esté sesgado por información filtrada inadvertidamente entre los diferentes conjuntos de datos.

Se ha realizado una normalización de los datos. Para ello, primero se cambió el tipo de los datos de *Unit8* a *Float32* para conseguir una mayor precisión, y después, se dividió entre 255 para conseguir que todos los valores estén en el rango comprendido entre 0 y 1.

Además de lo anterior, cada uno de los modelos implementados tienen características particulares que hacen necesaria una adaptación específica. La arquitectura de estos modelos se verá más adelante en el capítulo [7. Desarrollo del proyecto](#). Por ahora solo se explicará el formato de los datos de entrada.

#### 6.3.2. Conjunto de datos con aumento de datos

Este conjunto es muy similar al anterior. Se han duplicado la cantidad de datos *benign* mediante técnicas de aumento de datos<sup>12</sup>. Esto se debe a que la categoría *benign* tiene aproximadamente la mitad de casos en comparación con la categoría *cancer*.

#### Aumento de datos

Aunque la cantidad de datos puede parecer considerable, no es suficiente; un modelo de inteligencia artificial, enfocado en *Computer Vision*<sup>13</sup>, necesita miles

<sup>11</sup>Fugas de datos (*Data leakage*): situación donde la información del conjunto de prueba se filtra al conjunto de entrenamiento, afectando la evaluación del modelo.

<sup>12</sup>Aumento de datos: conjunto de técnicas utilizadas para incrementar la cantidad de datos de entrenamiento mediante la creación de nuevas muestras a partir de las existentes.

<sup>13</sup>*computer vision*: campo de la inteligencia artificial que permite a las computadoras interpretar y comprender el mundo visual a través de imágenes o videos. Utiliza técnicas de aprendizaje automático y procesamiento de imágenes para tareas como el reconocimiento de objetos, la detección de caras, y la segmentación de imágenes.

## CAPÍTULO 6. PROCESADO DE DATOS

---

de imágenes de cada clase para poder clasificarlas con un grado de incertidumbre aceptable. Para intentar solucionar este problema, se generarán más datos sintéticos mediante operaciones sencillas como rotaciones y volteos, tanto verticales como horizontales de la imagen.

Además de estas técnicas, se consideró el aumento de datos mediante una *GAN*<sup>14</sup>. Sin embargo, tras unas pruebas iniciales, se concluyó que la potencia del *hardware* disponible no era suficiente para obtener un grado de detalle adecuado. También se planteó la incertidumbre sobre si los casos generados mediante este método serían representativos de un tumor real, esto es, porque, al no contar con un radiólogo profesional, probar la calidad de los datos generados se presenta como una tarea imposible.

Para aumentar los datos de manera factible, se probaron múltiples combinaciones de operaciones sobre los *numpy arrays*. Las posibles operaciones incluyen rotaciones de 90, 180 y 270 grados, así como volteos horizontales y verticales, sumando un total de 16 combinaciones posibles. Sin embargo, solo se tuvieron en cuenta las 11 combinaciones siguientes:

- Rotación de 90 grados.
- Rotación de 180 grados.
- Rotación de 270 grados.
- Volteo horizontal.
- Volteo vertical.
- Rotación de 90 grados + Volteo horizontal.
- Rotación de 90 grados + Volteo vertical.
- Rotación de 180 grados + Volteo horizontal.
- Rotación de 180 grados + Volteo vertical.
- Rotación de 270 grados + Volteo horizontal.
- Rotación de 270 grados + Volteo vertical.

Tras probar todas estas combinaciones, se seleccionó únicamente una, aquella

---

<sup>14</sup>GAN: Redes Generativas Antagónicas, por sus siglas en inglés *Generative Adversarial Networks*.

### 6.3. CONJUNTOS UTILIZADOS EN EL ENTRENAMIENTO DE MODELOS

que mejor se ajustaba al formato de los datos originales. Esto se hizo porque, solo se busca equilibrar los casos *benign* y *cancer*, por lo que con duplicar los casos *benign* es suficiente para obtener este equilibrio y, también, al utilizar una técnica de generación de datos sobre una categoría específica, es esencial asegurarse de que el modelo no pueda distinguir entre los *numpy arrays* originales y los generados, por lo que las transformaciones no pueden deformar la orientación de las vistas de forma visible. Garantizar que sean indistinguibles es crucial para evitar sesgos en el modelo, que podría aprender a identificar características artificiales en lugar de las verdaderas señales presentes en las imágenes de tumores.

Esta técnica de aumento de datos no solo puede expandir el conjunto de datos disponible, sino que también mejorar el rendimiento del modelo al exponerlo a una mayor variabilidad de las imágenes, lo que puede permitirle generalizar mejor a nuevos datos no vistos.

Sin embargo, este enfoque también presenta algunos inconvenientes. Más datos implican que el modelo necesitará más tiempo para entrenarse. Además, aunque las transformaciones generan datos adicionales, estos datos son variantes de las imágenes originales, lo que puede llevar a una redundancia que puede no agregar mucho valor al modelo. Por esta razón, se decidió seleccionar menos combinaciones de las posibles, priorizando aquella que ofrece una mayor variabilidad y representatividad sin generar un volumen de datos excesivamente grande.

Después de realizar el aumento de datos, se utilizaron 5.533 casos, de los cuales 280 pertenecen a la categoría *actionable*, 1.214 a la categoría *benign*, 1.399 a la categoría *cancer* y 2.613 a la categoría *normal*.

Balancear más la categoría *benign* permite que el modelo aprenda mejor a identificar las características específicas de los tumores benignos.

Al aumentar la cantidad de casos *benign*, también se incluyeron más casos de la categoría *normal*. Esto se hizo para mantener el balance entre las dos categorías *benign* y *cancer* con respecto a la categoría *normal*. Esto ayuda a no disminuir la capacidad de los modelos para distinguir entre imágenes con y sin tumores.

## CAPÍTULO 6. PROCESADO DE DATOS

---

Después de esto, se realizó la división y el formato de los datos siguiendo la misma metodología explicada anteriormente en la sección [6.3.1](#).



### 6.3. CONJUNTOS UTILIZADOS EN EL ENTRENAMIENTO DE MODELOS

# Capítulo 7

## Desarrollo del proyecto

En este capítulo se presenta el desarrollo de las diferentes redes neuronales de inteligencia artificial empleadas para la clasificación de imágenes médicas en el marco de este Trabajo de Fin de Grado. Se abordarán cuatro redes, las cuales trabajan bajo arquitecturas específicas: dos redes *U-Net*, en los apartados 7.3 y 7.4; *Inceptionv3-Paralela*, en el apartado 7.5; y *ResNetv2-Paralela*, en el apartado 7.6. Cada uno de estos modelos será aplicado y evaluado para la misma tarea de clasificación de imágenes médicas. Pero primeramente, en el apartado 7.1, se presentarán dos modelos de referencia, *DBT* e *Inception V3* inicial, que se utilizarán como base de desarrollo y modelos a batir por las cuatro redes propuestas.

### 7.1. Modelos de referencia iniciales

Antes de explicar el diseño de los diferentes modelos es conveniente declarar un modelo de referencia el cual tener como punto de partida. A continuación se expondrá el modelo de referencia presentado por *Breast Cancer Screening (DBT)*, en el apartado 7.1.1, y el modelo de referencia que se ha desarrollado para este proyecto, en el apartado 7.1.2.

## 7.1. MODELOS DE REFERENCIA INICIALES

### 7.1.1. Modelo de referencia *Breast Cancer Screening (DBT)*

Este modelo, que usaremos como una de las bases o referencia, fue realizado en el mismo artículo que presentó el conjunto de datos *Breast Cancer Screening (DBT): A Data Set and Deep Learning Algorithm for the Detection of Masses and Architectural Distortions in Digital Breast Tomosynthesis Images*, realizado por [Buda y cols. \(2021\)](#), de la universidad de Duke.

La idea es procesar cada volumen de imágenes 3D con imágenes en 2D de forma independiente. Para detectar lesiones, se utilizó una red neuronal convolucional completamente conectada de una sola fase para la detección de objetos en 2D con una arquitectura *DenseNet*. El modelo procesa cada corte 2D de entrada y genera predicciones en una cuadrícula con celdas de  $96 \times 96$  píxeles. Cada celda proporciona una puntuación de confianza y las dimensiones de un cuadrado que localiza la lesión.

El modelo se optimizó con el algoritmo *Adam* y una tasa de aprendizaje inicial de 0,001, entrenándose durante 100 *epochs* con *early stopping*<sup>1</sup>. Para el entrenamiento, se utilizaron los cortes 2D que contenían un tumor positivos.

En la fase de diseño del modelo, se probaron diferentes tamaños de la red convolucional y funciones de pérdida para abordar el desequilibrio de clases. Las funciones de pérdida probadas fueron: entropía cruzada binaria, entropía cruzada binaria ponderada, pérdida focal y pérdida focal reducida. En total, se entrenaron 768 modelos y el mejor modelo, entrenado con pérdida focal, se evaluó en el conjunto de test.

En el test, el modelo final logró una sensibilidad del 42 % (*IC* del 95 %, 35 %-50 %) a dos falsos positivos por volumen de *DBT*. El intervalo de confianza del 95 % (*IC* del 95 %) indica que se tiene un 95 % de certeza de que la verdadera sensibilidad del modelo se encuentra entre el 35 % y el 50 %. Además, se evaluó la precisión utilizando *FROC* basada en el seno, logrando una sensibilidad del 67 % (*IC* del 95 %, 53 %-80 %) para casos de cáncer.

<sup>1</sup>*Early stopping*: técnica utilizada para evitar el sobreentrenamiento del modelo, deteniendo el entrenamiento cuando el rendimiento en un conjunto de validación deja de mejorar después de un número predefinido de épocas.

## CAPÍTULO 7. DESARROLLO DEL PROYECTO

---

En definitiva, el modelo se encarga de dibujar como máximo un cuadrado donde creyese que había un tumor. Obtuvieron una precisión muy baja, del 42 %. Además, hicieron uso de algunas “malas prácticas”, como eliminar los casos en los que el área del cuadrado estuviese mas de un 50 % fuera de la mama y el uso de un conjunto de datos no debidamente equilibrados, donde más del 93 % de los datos no tienen tumor, y por tanto no tienen cuadrado. De todas formas, este modelo fue presentado para poner un límite a futuros modelos desarrollados por terceros y muestra la dificultad que se encuentra al afrontar una tarea de clasificación de imágenes médicas.

### 7.1.2. Modelo de referencia *Inception V3* inicial

Además del anterior modelo presentado por la universidad de Duke, se ha utilizado el modelo *Inception V3* de *Google* como modelo de referencia y punto de partida del presente proyecto. Este modelo se ha adaptado específicamente para trabajar con nuestros datos. *Inception V3* es una arquitectura de red neuronal convolucional profunda que ha demostrado un rendimiento sobresaliente en la clasificación de imágenes. Esta misma arquitectura será explicada en detalle más adelante dentro de este mismo capítulo.

Los datos utilizados para entrenar y evaluar el modelo son *arrays* de tamaño  $1440 \times 1440$ . Estos *arrays* representan a una imagen que contiene las cuatro vistas para cada caso. Cada vista tiene una resolución de  $720 \times 720$  píxeles. Toda la información referente a los datos se encuentra en el capítulo 5, Análisis de los datos , y en el capítulo 6, Procesado de datos .

El modelo se inicializó con los siguientes parámetros:

- *weights='imagenet'*: se utilizan los pesos preentrenados en el dataset *ImageNet*, lo que permite que el modelo comience con características previamente aprendidas.
- *include\_top=False*: esta opción excluye las capas de clasificación final del modelo preentrenado, permitiéndonos añadir nuestras propias capas de

## 7.1. MODELOS DE REFERENCIA INICIALES

---

clasificación adaptadas a nuestro problema específico.

- *input\_shape=(1440, 1440, 3)*: define la forma de la entrada del modelo como imágenes de  $720 \times 720$  píxeles con tres canales de color (*RGB*).

Además, fue compilado usando la función de pérdida *sparse\_categorical\_crossentropy* y el optimizador *Adam*, con *accuracy* como métrica:

- *Sparse Categorical Crossentropy*: se utiliza esta función de pérdida porque estamos trabajando con una clasificación de múltiples clases, donde cada imagen pertenece a una sola clase y las etiquetas son enteros en lugar de vectores *one-hot*.
- *Adam Optimizer*: este optimizador se elige por su eficiencia y capacidad de adaptarse a diferentes problemas, ajustando automáticamente la tasa de aprendizaje. Además, de que es uno de los optimizadores más populares y utilizados.
- *Accuracy*: esta métrica proporciona una medida directa de la cantidad de predicciones correctas que realiza el modelo, lo que es esencial para evaluar su rendimiento en un problema de clasificación.

El modelo de referencia *Breast Cancer Screening (DBT)*, del apartado 7.1.2 anterior, aporta sus propios resultados en el artículo de [Buda y cols. \(2021\)](#). Sin embargo, al carecer de resultados para este modelo generalizado de *Google* con la fuente de datos utilizada en este proyecto, lo primero que se debía realizar es capturar el comportamiento del modelo con los datos de cáncer de mama con los que se va a trabajar. La siguiente sección presenta estos resultados base que, junto con los del modelo *DBT* anterior, son los resultados a batir con las propuestas ideadas en la presente memoria; resultados que se presentarán en el Capítulo 8, **Resultados**.



Figura 7.1: Valores de la pérdida (*validation loss*) y *accuracy* (*validation accuracy*) obtenidos durante el entrenamiento para el modelo *Inception V3*.

### Evaluación del modelo

La figura 7.1 muestra las curvas de pérdida y precisión tanto para el entrenamiento como para la validación a lo largo de 50 épocas. A continuación se presenta un análisis de la figura.

Dentro de esta figura 7.1, en la gráfica de la izquierda, observamos la pérdida de entrenamiento y validación. La pérdida de entrenamiento disminuye consistentemente, mientras que la pérdida de validación muestra fluctuaciones, indicando posibles problemas de sobreajuste. En la gráfica de la derecha, el *accuracy* de entrenamiento aumenta constantemente, pero la precisión de validación tiene un comportamiento más variable, lo que sugiere que el modelo podría estar ajustándose demasiado a los datos de entrenamiento y no generalizando bien.

En la figura 7.2 se revela la matriz de confusión obtenida al evaluar el modelo con los datos de *test*. Este modelo obtiene una pérdida de 0,7103 y una precisión de 60,83 % en el conjunto de datos de prueba. La matriz de confusión expone que el modelo tiene una precisión limitada y comete errores en todas las categorías. La mayoría de las predicciones se agrupan en la categoría *normal*, lo que indica que el modelo tiende

## 7.1. MODELOS DE REFERENCIA INICIALES

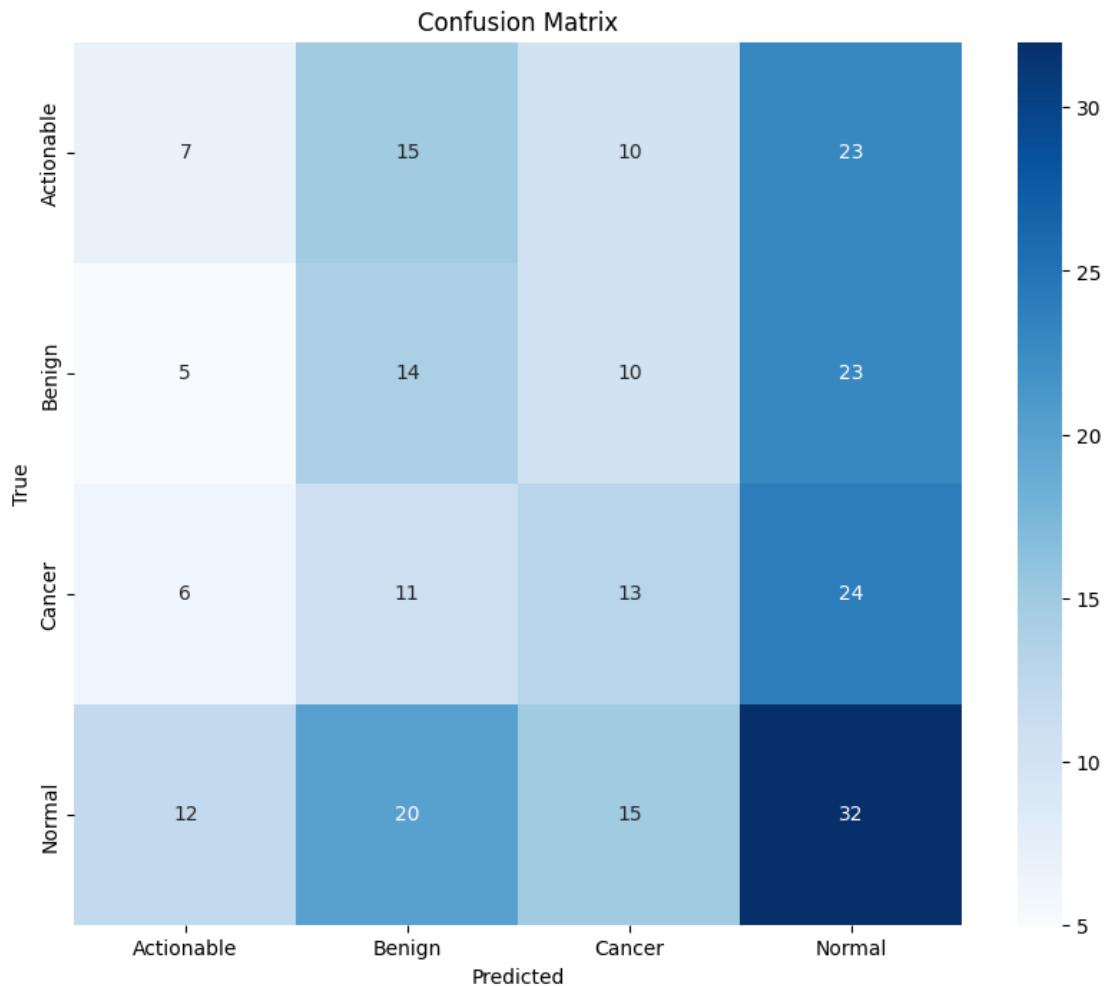


Figura 7.2: Matriz de confusión con los datos de test para el modelo *Inception V3*.

a clasificar la mayoría de las imágenes como normales debido a un desequilibrio de clases en el conjunto de datos.

Analizando la matriz de confusión más detalladamente, observamos lo siguiente:

- La clase *actionable* tiene 7 predicciones correctas, pero también hay un alto número de falsos negativos (15 como *benign*, 10 como *cancer* y 23 como *Normal*).
- La clase *benign* tiene 14 predicciones correctas, con errores dispersos en las demás categorías.
- La clase *cancer* tiene 13 predicciones correctas, pero también se confunde

significativamente con *benign* y *normal*.

- La clase *Normal* tiene el mayor número de predicciones correctas (32), pero aún así presenta confusiones con todas las otras clases, especialmente *benign*.

Estos resultados indican que el modelo no está diferenciando bien entre las diferentes clases, con una tendencia notable a clasificar incorrectamente muchas instancias como *normal*. Nos sugiere, en definitiva, que el modelo podría estar sesgado debido a un desequilibrio de clases en el conjunto de datos de entrenamiento y podría necesitar mejoras adicionales en su arquitectura o proceso de entrenamiento para manejar mejor las diferencias entre las clases.

Este modelo se utiliza como punto de partida debido a su arquitectura avanzada y su capacidad demostrada para manejar tareas de clasificación de imágenes complejas. Aunque el rendimiento inicial del modelo no es óptimo, sirve como una base sólida sobre la cual se pueden realizar mejoras adicionales.

## 7.2. Aspectos comunes de los modelos implementados

Antes de proceder a la descripción detallada de cada uno de los modelos implementados, *UNet3D-Dense*, *UNet-Paralela*, *InceptionV3-Paralela* y *ResNetV2-Paralela*, se presentará una explicación previa de las técnicas comunes que han sido empleadas en las cuatro redes. De este modo, se evita la repetición innecesaria de conceptos a lo largo del documento, permitiendo al lector consultar estas técnicas de manera centralizada y cómoda, facilitando así la comprensión de los modelos en su conjunto.

### 7.2.1. Función de pérdida ponderada

A continuación se va a explicar la función de pérdida utilizada en este trabajo que se presenta en todos los modelos implementados.

Una función de pérdida es una medida de la discrepancia entre las predicciones del modelo y los valores reales. Su objetivo principal es guiar el proceso de entrenamiento,

## 7.2. ASPECTOS COMUNES DE LOS MODELOS IMPLEMENTADOS

proporcionando una señal de error que el modelo utiliza para ajustar sus pesos y minimizar discrepancias. En términos simples, la función de pérdida cuantifica qué tan bien o mal está funcionando un modelo durante su entrenamiento.

*Weighted categorical crossentropy* (entropía cruzada categórica ponderada) aplica un peso a cada clase, lo cual es especialmente útil para manejar el desequilibrio de clases en un problema de clasificación. Al asignar diferentes pesos a las clases en función de sus frecuencias, se puede equilibrar la influencia de cada clase en el proceso de entrenamiento, evitando que las clases más frecuentes dominen el aprendizaje del modelo.

Al asignar mayor peso a las clases menos representadas, el modelo puede aprender de manera más equilibrada, mejorando su capacidad para identificar correctamente tanto las clases mayoritarias como las minoritarias. Esto resulta en un rendimiento más robusto y equitativo del modelo en datos de prueba desbalanceados como el que se está utilizando, donde la precisión en la predicción de todas las clases es crucial.

Para comprender mejor el funcionamiento de esta función, el lector puede acceder al apartado B.1, *Weighted categorical crossentropy*, perteneciente al anexo. El desarrollo de esta función de pérdida ha sido algo imprescindible para poder aprovechar los pocos datos de los que se dispone al máximo, sin necesidad de desechar datos en la mayoría de categorías.

### 7.2.2. Técnica de validación cruzada

La validación cruzada es una técnica crucial en el desarrollo de modelos de aprendizaje automático que cuentan con una cantidad de datos limitados, tal y como es este caso. Cuando no es necesaria esta técnica, los datos se suelen dividir en tres conjuntos: el conjunto de entrenamiento, el conjunto de validación y el conjunto de *test* o prueba. En la validación cruzada, los datos que se utilizarían normalmente para el entrenamiento y la validación, se dividen en múltiples subconjuntos llamados *folds*, o fases. Además, el proceso de entrenamiento se divide en un número de iteraciones tantas como fases se hayan realizado. En cada iteración se utiliza un *fold* distinto como



Figura 7.3: Validación cruzada en  $K = 5$  fases ([Martínez, 2017](#)).

datos de validación y el resto de *folds* como datos entrenamiento. Este proceso lo podemos visualizar en la figura 7.3.

El sobreajuste u *overfitting*<sup>2</sup> es uno de los mayores problemas a los que se enfrentan los modelos de *deep learning*, y aumentar los datos es la mejor manera de luchar contra esta problemática. La validación cruzada asegura que cada subconjunto sea utilizado tanto para entrenamiento como para validación, por lo que se maximiza el uso de datos disponibles de una forma robusta.

Sin embargo, la validación cruzada presenta ciertas desventajas. Una de ellas es el aumento del tiempo de computación, ya que el proceso implica entrenar y evaluar el modelo múltiples veces. Además, está el problema de utilizar los mismos datos para el entrenamiento y la validación, lo que puede llevar a un sobreajuste detectable solo al usar los datos de prueba. Estos datos deberían mantenerse al margen de cualquier decisión de diseño del modelo, lo cual puede resultar complicado si se presenta este sobreajuste. En este caso, se ha empleado un *early stopping* con los datos de prueba. Aunque el *early stopping* puede reducir hacer mas difícil definir el rendimiento del

<sup>2</sup>Sobreajuste (*Overfitting*): ocurre cuando un modelo de *deep learning* aprende demasiado bien los detalles y el ruido del conjunto de datos de entrenamiento, resultando en un rendimiento pobre al generalizar a nuevos datos no vistos.

### 7.3. RED UNET3D-DENSA

---

modelo en la práctica, también puede mejorarlo significativamente.

La técnica de validación cruzada se ha aplicado durante décadas en inteligencia artificial, ofreciendo buenos resultados y contribuyendo a mejorar modelos, tanto en *Deep Learning* como en otras áreas de la ciencia de datos. Es fundamental equilibrar las ventajas y desventajas de la validación cruzada, considerándola una herramienta útil, aunque no infalible, en el desarrollo de modelos de aprendizaje automático.

Igual que en la figura 7.3, la validación cruzada implementada en este proyecto consta de 5 *folds* o fases, donde los casos se rotan entre los conjuntos de validación y entrenamiento en cada una de las 5 iteraciones.

En el apartado B.2, **Validación cruzada**, perteneciente al anexo podrá comprobar con más detalle el código *Python* que se ha desarrollado para esta tarea, además de la explicación del mismo.

Una vez que ya se tienen seleccionados los datos y agrupados por cinco fases, o *folds*, se realiza un preprocesamiento para adaptarlos a las diferentes dimensiones y formatos de los diferentes modelos utilizados.

A continuación siguen los apartados con los diseños de los diferentes modelos propuestos: *UNet3D-Densa*, en el apartado 7.3; *UNet-Paralela*, en el apartado 7.4; *Inception V3* finalmente desarrollado a partir del inicial, en el apartado 7.5; y *ResNet V2*, en el apartado 7.6;

## 7.3. Red **UNet3D-Densa**

Vamos a empezar con la explicación de la primera arquitectura desarrollada, la *U-Net*. En el presente apartado, se justificará cada una de sus capas, así como todas las decisiones de diseño tomadas en la misma.

### 7.3.1. Aspectos teóricos de las redes *U-Net*

Las redes *U-Net* surgieron a partir de las redes neuronales convolucionales (*Convolutional Neural Networks*, CNNs)<sup>3</sup>, específicamente diseñadas para la segmentación de imágenes biomédicas. La primera *U-Net* fue introducida en el año 2015 en la conferencia *U-Net: Convolutional Networks for Biomedical Image Segmentation* (Ronneberger y cols., 2015), basada en la red de redes neuronales completamente convolucionales (*Fully Convolutional Network*, FCN)<sup>4</sup>. Este modelo se destacó por incorporar conexiones de salto (*skip connections*) para concatenar mapas de características de baja y alta resolución, obteniendo así una segmentación detallada y precisa.

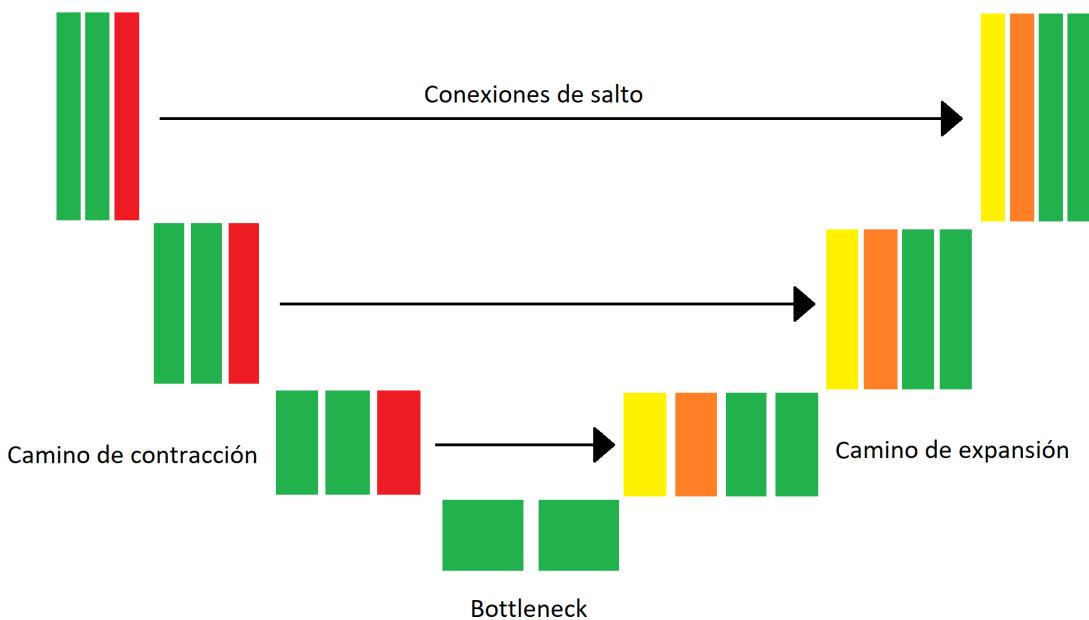


Figura 7.4: Esquema de una arquitectura *U-Net*.

El diseño de la *U-Net* permitió superar las limitaciones de los modelos previos en la segmentación de imágenes médicas, donde es crucial clasificar cada píxel de la imagen, en lugar de etiquetar la imagen completa. Sus resultados se pueden corroborar con el artículo *Using U-Net network for efficient brain tumor segmentation in MRI*

<sup>3</sup>CNN: Redes Neuronales Convolucionales, por sus siglas en inglés.

<sup>4</sup>FCN: Red de Redes Neuronales Completamente Convolucionales, por sus siglas en inglés.

### 7.3. RED UNET3D-DENSA

*images Walsh y cols. (2022)*, donde se utilizó para la detección de tumores cerebrales.

Como se puede observar en la figura 7.4 la *U-Net* se caracteriza por su arquitectura en forma de “U”, que incluye un camino de contracción (camino de codificación) y un camino de expansión (camino de decodificación), lo cual facilita la captura de contextos tanto globales como locales de la imagen.

Una *U-Net* sencilla sigue el formato *encoder-bottleneck-decoder*<sup>5</sup> con conexiones de salto. A continuación, se detallan cada una de estas partes.

#### Codificador

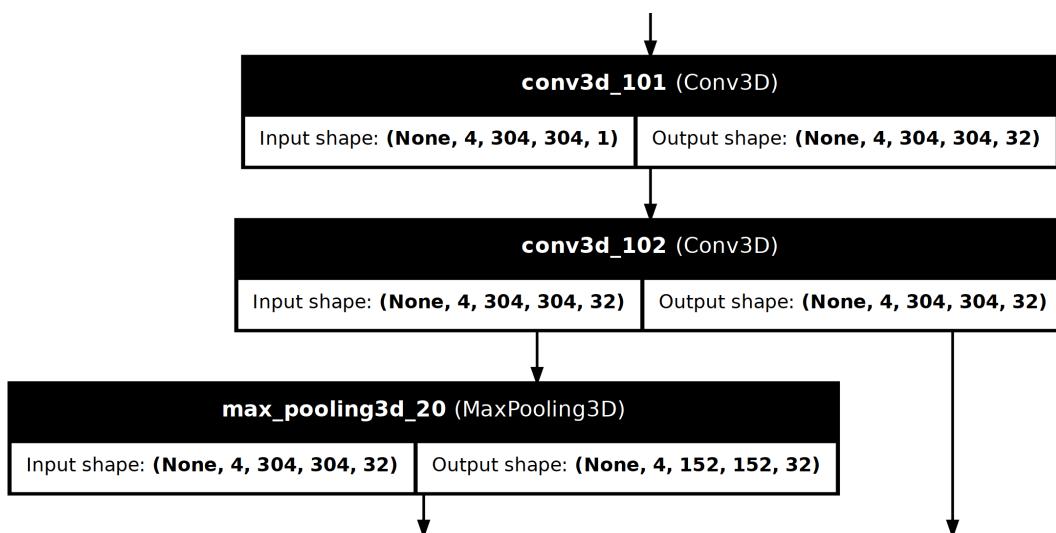


Figura 7.5: Ejemplo de codificador en una *U-Net*.

Las capas del codificador incluyen varios filtros convolucionales para extraer características de bajo nivel, con activación *ReLU* y *padding* adecuado, y filtros *Maxpooling* para reducir la resolución espacial. Esto permite que el modelo aprenda representaciones cada vez más abstractas de los datos.

El codificador pertenece a el camino de contracción y su función es disminuir las dimensiones de la imagen para generalizar y abstraer los conceptos. Para explicarlo con un ejemplo, si tenemos un modelo *U-Net*, el cual se encarga de detectar un gato dentro

<sup>5</sup> Arquitectura de codificador-*Bottleneck*-decodificador: estructura que incluye una parte de reducción (codificador), una parte de cuello de botella (*bottleneck*) y una parte de expansión (decodificador) de los datos.

de una imagen. El primer codificador cogería todos los píxeles y crearía una matriz de características, la cual define espacialmente que características ha encontrado, estas podrían ser: bigote, ojo, orejas, cola, etc. El segundo codificador cogería esta información y abstraería aún más, por ejemplo: cabeza, torso, cola, patas. Y el último se encargaría de definir el gato completo. Podemos observar un ejemplo de un módulo codificador en la figura 7.5

Hay que tener en cuenta que la matriz va disminuyendo en dimensiones, pero aumentando la cantidad de filtros para cada dimensión, por lo que cada posición de la matriz de salida de un codificador agrupa más características que la posición de entrada, para que la información pueda fluir a lo largo de las capas y poder matizar estas características.

### Bottleneck

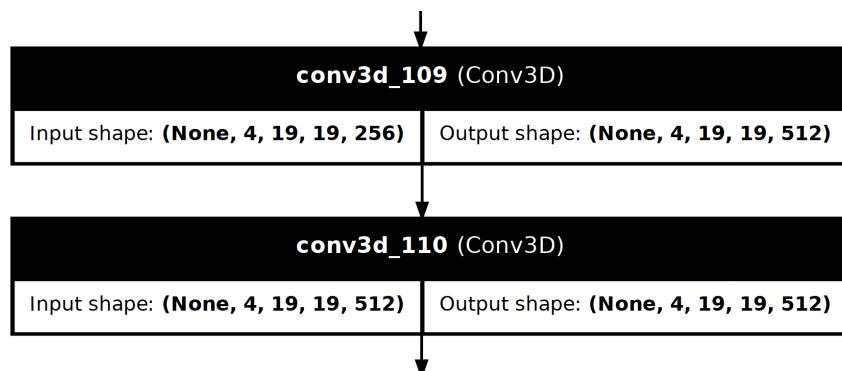


Figura 7.6: Ejemplo de *bottleneck* en una *U-Net*.

El *bottleneck* es la parte intermedia que conecta al camino de contracción con el camino de expansión y se encarga de procesar la representación más comprimida de los datos. Esta sección contiene capas convolucionales con un número elevado de filtros (por ejemplo, 1024), permitiendo capturar las características más abstractas de la imagen. En esta parte, se aplica convolución con filtros grandes y funciones de activación *ReLU*, manteniendo el *padding* adecuado para preservar las dimensiones espaciales. En la figura 7.6 podemos ver un ejemplo de un *bottleneck*.

### 7.3. RED UNET3D-DENSA

## Decodificador

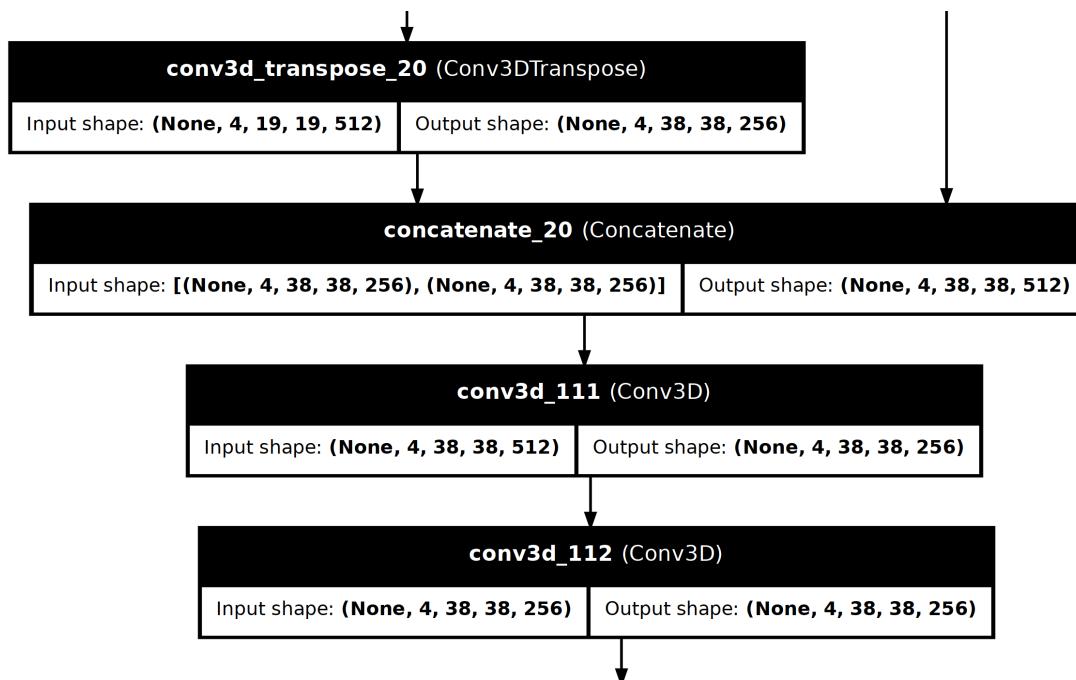


Figura 7.7: Ejemplo de decodificador en una *U-Net*.

El decodificador, ilustrado en la figura 7.7, se encarga de restaurar la resolución espacial de la imagen a través de operaciones de convolución transpuesta (*transposed convolutions*)<sup>6</sup> o *upsampling*, combinadas con las características capturadas en el codificador mediante las conexiones de salto. Esto permite a la *U-Net* generar una segmentación precisa, combinando la información de alta resolución con la de baja resolución. En este proceso, se utilizan convoluciones transpuestas y concatenaciones de características, aplicando múltiples capas convolucionales con activaciones *ReLU* y *padding* adecuado.

Esta estructura permite que la *U-Net* capture y combine características a diferentes niveles de resolución, facilitando una segmentación precisa de cada píxel en la imagen original. La principal ventaja de la *U-Net* radica en su capacidad para entrenarse con un número reducido de muestras anotadas, gracias a técnicas como *data augmentation*<sup>7</sup>,

<sup>6</sup>Convoluciones transpuestas: operación utilizada para aumentar la resolución espacial de los mapas de características.

*<sup>7</sup>Data augmentation:* técnicas utilizadas para aumentar artificialmente el tamaño del conjunto de datos de entrenamiento mediante transformaciones aleatorias de las imágenes.

que mejoran la robustez y la precisión del modelo (Roy y cols., 2023).

### **7.3.2. Aplicación de *U-Net* en la clasificación de radiografías de mama**

La arquitectura *U-Net* ha demostrado ser eficaz en la clasificación y detección de tumores a partir de radiografías. La capacidad del modelo para segmentar de manera precisa tejidos y estructuras específicas en imágenes médicas es fundamental para el diagnóstico temprano y preciso del cáncer. Al aplicar *U-Net* a la clasificación de radiografías de mama, se pueden identificar áreas sospechosas con mayor detalle y exactitud, mejorando así la capacidad de los radiólogos para detectar tumores malignos en etapas tempranas (Walsh y cols., 2022; Roy y cols., 2023).

Una de las principales ventajas de utilizar *U-Net* en este contexto es su capacidad para lograr una segmentación detallada de las imágenes. A diferencia de las técnicas que veremos más adelante con las arquitecturas *Inception*, la estructura de la *U-Net* permite identificar características sutiles que pueden pasar desapercibidas para otros métodos. Esto es crucial para detectar pequeños tumores o anomalías en las radiografías de mama.

Además, la *U-Net* es capaz de entrenarse eficientemente con un número reducido de muestras anotadas, gracias a la augmentación de datos, lo cual es particularmente valioso en el campo de la medicina, donde los datos etiquetados son escasos. Este enfoque permite mejorar la robustez y precisión del modelo, haciendo posible su aplicación en situaciones donde la disponibilidad de datos es limitada.

Otro aspecto destacado de la *U-Net* es su relativa facilidad de implementación y ajuste. Esta simplicidad facilita su integración en sistemas de diagnóstico automático y asistencia a radiólogos. Sin embargo, es importante tener en cuenta que, a pesar de su sencillez, la *U-Net* puede requerir muchos recursos *hardware* dependiendo del problema, lo que puede ser un gran limitante. La necesidad de recursos *hardware* robustos para manejar la complejidad del modelo y los datos ha sido un desafío en este proyecto al contar con recursos limitados.

### 7.3.3. Implementación del modelo *UNet3D-Densa*

Las redes *U-Net* fueron inicialmente diseñadas para trabajar con datos bidimensionales, es decir, imágenes. Sin embargo, en el contexto de este trabajo, se ha adaptado la estructura de la *U-Net* para trabajar con datos tridimensionales, dado que se trabaja con *arrays* de imágenes. Esta adaptación permite manejar volumetrías de datos, lo cual es crucial en la segmentación y análisis de imágenes médicas en 3D, como las tomografías computarizadas o resonancias magnéticas.

La implementación llevada a cabo sigue la arquitectura explicada anteriormente. A continuación, se detallan las partes principales de la implementación.

#### Camino de codificación

Como se puede ver en la figura 7.8, durante el camino de codificación, el modelo aplica una serie de capas convolucionales 3D con función de activación ReLU y un tamaño de *kernel* de 3. Cada capa convolucional incrementa la profundidad de las características extraídas, comenzando con 8 filtros en la primera capa y duplicando progresivamente el número de filtros en las capas subsiguientes, siguiendo la progresión geométrica:

$$a_n = 2^{n+3} \text{ para } n = 0, 1, 2, 3$$

Cada capa convolucional en este modelo utiliza un *kernel* cúbico de  $3 \times 3 \times 3$ , permitiendo la detección de características tridimensionales en la entrada. Estas convoluciones capturan patrones locales y refinan la información a medida que avanzamos en la red.

Después de cada bloque de capas convolucionales, se aplica una capa de *MaxPooling3D* con un tamaño de ventana (1,2,2), que reduce a la mitad las dimensiones espaciales de la entrada, permitiendo al modelo enfocarse en características más globales sin aumentar el tamaño de los filtros. Esto es evidente comparando las dimensiones de las primeras capas con la última, donde las

## CAPÍTULO 7. DESARROLLO DEL PROYECTO

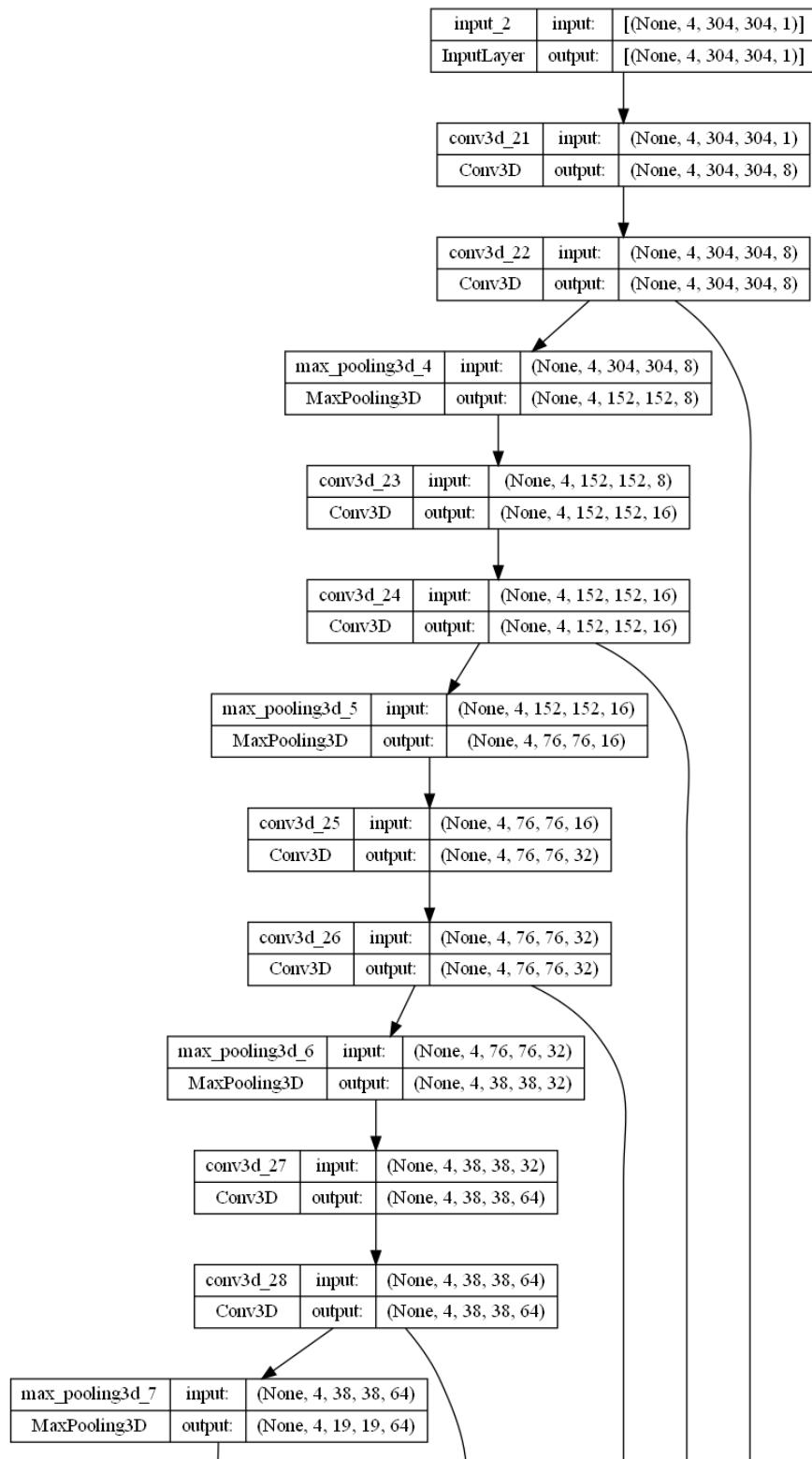


Figura 7.8: Camino de codificación en el modelo *U-Net* implementado.

### 7.3. RED UNET3D-DENSA

dimensiones de la entrada se reducen progresivamente de  $304 \times 304$  hasta  $19 \times 19$  al final del cuarto bloque de codificación.

#### **Bottleneck**

Como explicamos antes, el *bottleneck* es la parte intermedia de la *U-Net* que conecta el codificador con el decodificador y se encarga de procesar la representación más comprimida de los datos, capturando las características más abstractas y esenciales de la imagen.

Como se muestra en la figura 7.9 esto se implementa mediante dos capas convolucionales adicionales con 128 filtros cada una, utilizando un *kernel* de  $3 \times 3 \times 3$  y activación ReLU.

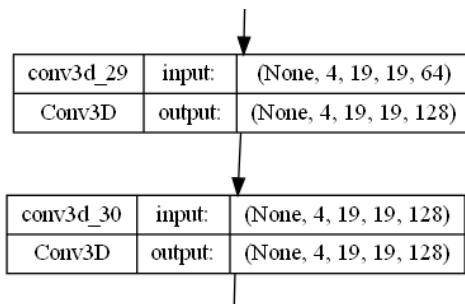


Figura 7.9: *Bottleneck* en el modelo *U-Net* implementado.

#### **Camino de expansión**

El camino de expansión invierte el proceso de codificación para reconstruir la imagen segmentada con las dimensiones originales. Como se muestra en la figura 7.10, el proceso comienza con una capa de *Conv3DTranspose*, que aumenta las dimensiones espaciales.

Después, las características extraídas en las capas de codificación correspondientes se concatenan con las obtenidas durante la expansión para refinar la reconstrucción. Este proceso se repite hasta que se alcanzan las dimensiones originales de la imagen de entrada. Finalmente, se aplica una última capa convolucional con cuatro filtros, uno por cada clase que se busca clasificar, y una función de activación *sigmoid* para generar

## CAPÍTULO 7. DESARROLLO DEL PROYECTO

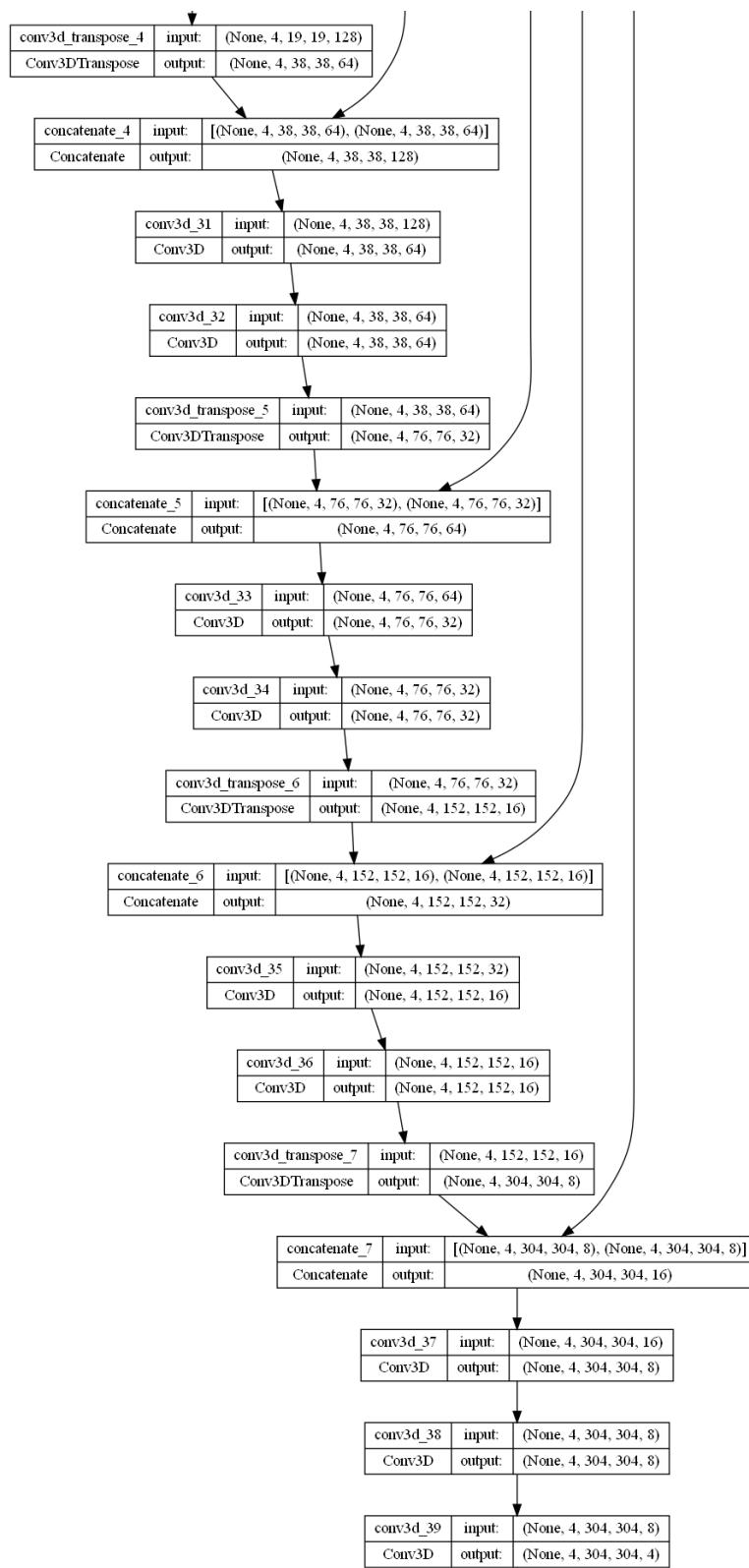


Figura 7.10: Camino de expansión en el modelo *U-Net* implementado.

### 7.3. RED UNET3D-DENSA

---

la salida segmentada. Con esto el modelo genera una cadena de cuatro imágenes con las dimensiones originales  $4 \times 304 \times 304$  y cuatro canales.

Una vez realizado esto, habríamos acabado con la arquitectura *U-Net*, pero debemos recordar que la *U-Net* tiene la finalidad de hacer segmentación de imágenes, y lo que se busca es la clasificación. Por esto, debemos añadir más capas que nos ayuden a clasificar las imágenes segmentadas.

#### Capas de clasificación

En la figura 7.11 se muestran las capas implementadas para la clasificación de las imágenes segmentadas generadas por la *U-Net*. A continuación, se explicarán detalladamente cada una de las capas y su finalidad dentro del modelo.

Primero, se aplica una capa de *BatchNormalization*, la cual normaliza las activaciones de la capa previa. Esto se hace ajustando y escalando los valores de activación para que sigan una distribución normal con media cero y desviación estándar uno. La normalización de lotes es crucial para mejorar la estabilidad y velocidad del entrenamiento, ya que reduce la sensibilidad a la inicialización de pesos y acelera la convergencia. En otras palabras, esta capa ayuda a mantener los valores de activación dentro de un rango deseado, evitando valores extremos que puedan dificultar el aprendizaje del modelo.

A continuación, se utilizan dos capas de *Conv3D* con un filtro de tamaño  $3 \times 3 \times 3$  y función de activación RELU. La función de activación RELU introduce no linealidad en el modelo, permitiendo que este aprenda relaciones más complejas en los datos. La primera capa *Conv3D* toma la salida normalizada y aplica convoluciones tridimensionales con un *stride* de 2 en cada dimensión, lo que reduce la dimensionalidad espacial de las características extraídas y, al mismo tiempo, preserva la información relevante para la tarea de clasificación. Este proceso se repite en la segunda capa *Conv3D*, donde se continúa reduciendo la dimensionalidad y refinando las características. La finalidad de estas capas convolucionales es extraer patrones complejos y detectar características significativas en las imágenes tridimensionales.

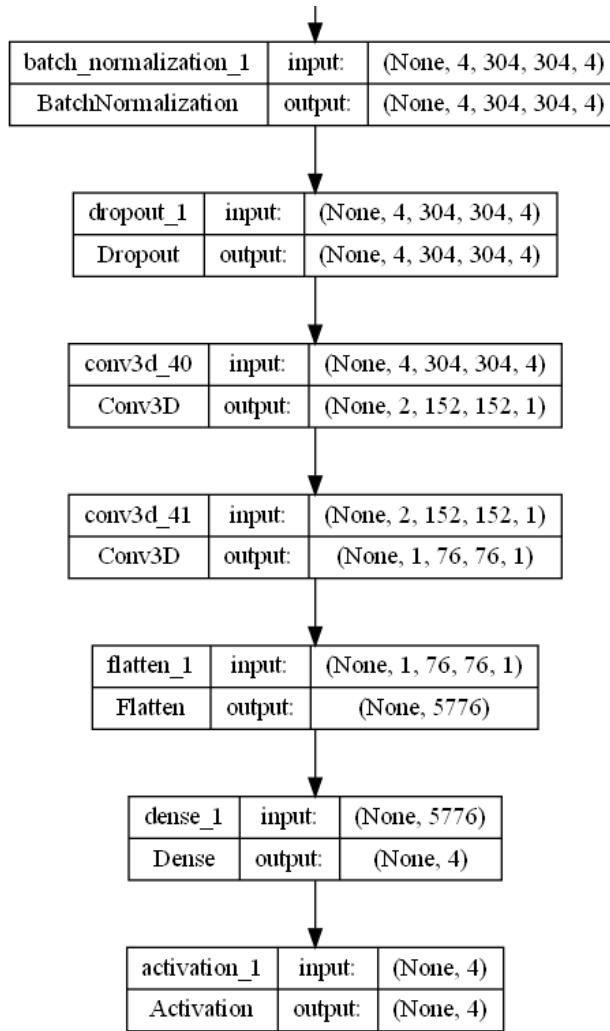


Figura 7.11: Capas para clasificar los resultados obtenidos en la *U-Net* implementada.

segmentadas, capturando información de diferentes niveles de abstracción.

Después de las convoluciones, se utiliza una capa *Flatten* que aplana el tensor de entrada, convirtiendo los datos tridimensionales en un vector unidimensional. Este paso es necesario para preparar los datos para la capa densa subsecuente, ya que las capas densas operan sobre vectores en lugar de tensores multidimensionales. La capa *Flatten* esencialmente reorganiza los datos de manera que puedan ser procesados por la siguiente etapa del modelo.

Finalmente, se aplica una capa *Dense* que realiza una transformación lineal sobre el vector aplanado, generando una salida de cuatro dimensiones, correspondiente a las cuatro clases que se buscan clasificar. Esta capa densa actúa como un clasificador,

## 7.4. RED UNET-PARALELA

tomando las características extraídas y aprendidas por las capas previas y mapeándolas a una probabilidad para cada clase. Para convertir estas salidas en probabilidades, se utiliza una capa de activación *Softmax*, que asegura que la suma de las probabilidades de todas las clases sea igual a uno. La *Softmax* permite interpretar las salidas del modelo como probabilidades, facilitando la tarea de clasificación multiclas al identificar la clase más probable para cada entrada.

## 7.4. Red *UNet-Paralela*

A diferencia de la anterior red neuronal, que utilizaba una sola red que analizaba los datos en tres dimensiones, en este caso vamos a utilizar una red neuronal *U-Net* por cada mamografía, en concreto 4 redes *U-Net*, para luego juntar sus resultados y analizarlos para la clasificación de los diferentes casos.

Estas cuatro redes son idénticas entre sí, por lo que vamos ver como se dividen los datos, además de los caminos de codificación, *bottleneck* y codificación que las componen. Después de esto se explicará las capas encargadas de analizar las salidas de estas cuatro redes y realizar la clasificación.

### División de las mamografías

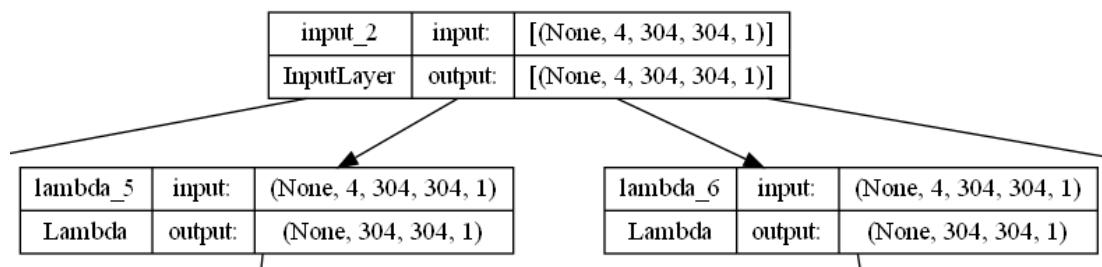


Figura 7.12: Capas encargadas de la obtención y división de los diferentes *arrays*.

Las dos primeras capas de esta arquitectura se encargan de obtener y dividir cada mamografía contenida en el *array* y redirigirlas a cada una de las *U-Net*. Tal y como se puede observar en la figura 7.12, *InputLayer* representa la entrada principal que

## CAPÍTULO 7. DESARROLLO DEL PROYECTO

contiene todas las vistas de las mamografías. La división se consigue mediante el uso de cuatro capas Lambda. Estas capas permiten extraer individualmente las mamografías de cada una de las cuatro vistas contenidas en la entrada, para luego ser procesadas por las U-Net correspondientes.

### Camino de codificación

Como es lógico, vamos a empezar con la explicación de la implantación de las diferentes *U-Net* por el camino de codificación, este es el encargado de extraer características relevantes de la imagen de entrada, reduciendo progresivamente sus dimensiones mientras incrementa la profundidad de las representaciones. Este proceso se lleva a cabo mediante una serie de bloques secuenciales que combinan varias operaciones clave, permitiendo al modelo capturar y comprender patrones complejos dentro de los datos. En la figura 7.13 podemos ver una representación de los mismos.

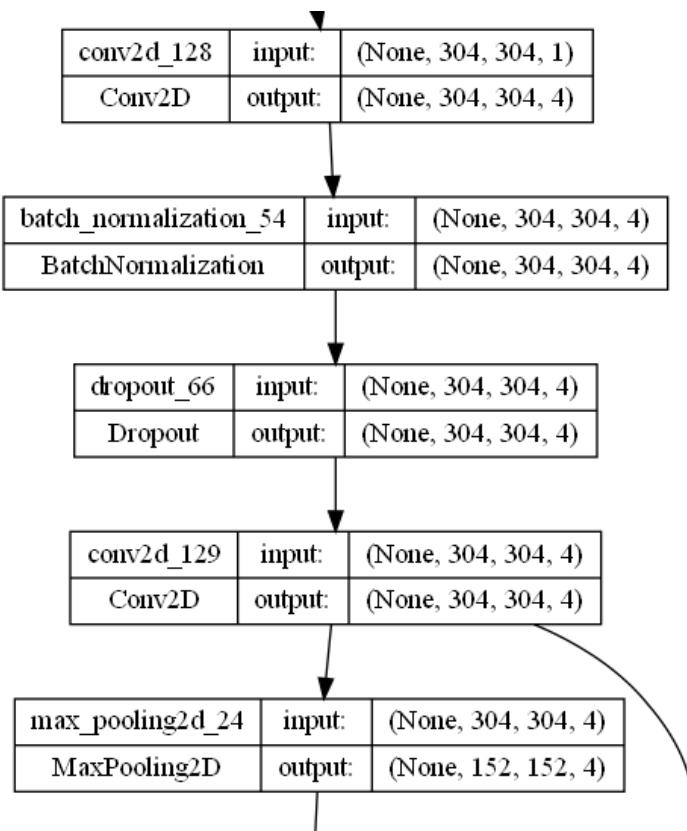


Figura 7.13: Capas que componen un módulo del camino de codificación de cada una de las cuatro redes *U-Net* del modelo *UNet-Paralela*.

## 7.4. RED UNET-PARALELA

---

Cada bloque de codificación comienza con la aplicación de capas convolucionales, donde se emplean filtros de tamaño  $3 \times 3$ . Estas convoluciones están diseñadas para extraer características locales de la imagen, como bordes y texturas, y están seguidas por una función de activación ReLU. La activación ReLU introduce no linealidad en el modelo, permitiendo que la red aprenda relaciones complejas en los datos. A medida que avanzamos a través de los bloques, la cantidad de filtros en las capas convolucionales aumenta, lo que incrementa la capacidad del modelo para capturar características más abstractas y de alto nivel. Además, cada una de las capas convolucionales aplica un regularizador  $L_2$  con un factor de 0,01, lo que ayuda a prevenir el sobreajuste al penalizar grandes valores en los pesos de la red.

Posteriormente, se aplica una capa de normalización por lotes, o *Batch Normalization*, que ajusta y escala las activaciones de las capas previas para estabilizar y acelerar el proceso de entrenamiento. Esta normalización es crucial para mantener el flujo de información a lo largo de la red, reduciendo la sensibilidad a la inicialización de pesos y mejorando la eficiencia del aprendizaje.

Para evitar el sobreajuste durante el entrenamiento, cada bloque incorpora también una capa de *Dropout*. Esta capa apaga aleatoriamente una fracción de las neuronas, lo que fuerza al modelo a desarrollar representaciones más robustas y evita que dependa demasiado de ciertas rutas internas.

Finalmente, cada bloque de codificación concluye con una operación de *MaxPooling2D*, que reduce las dimensiones espaciales de la imagen procesada. Esta reducción de dimensiones permite condensar la información más relevante, preservando los aspectos críticos mientras se descartan los detalles menos importantes. La operación de *MaxPooling* no solo ayuda a reducir la carga computacional, sino que también permite que la red capture el contexto global de la imagen a medida que avanza hacia capas más profundas.

Este proceso secuencial de convolución, normalización, *Dropout* y reducción de dimensiones se repite en varios bloques a lo largo del camino de codificación, con un número creciente de filtros en cada capa convolucional. De esta manera, el modelo

logra construir una representación jerárquica de la imagen, capturando características desde los niveles más bajos hasta los más abstractos y asegurando que la información esencial se conserve para las etapas posteriores de la red.

### Bottleneck

El *bottleneck* se encarga de procesar las características más profundas y abstractas de una red *U-Net* antes de transferirlas a el camino de expansión.

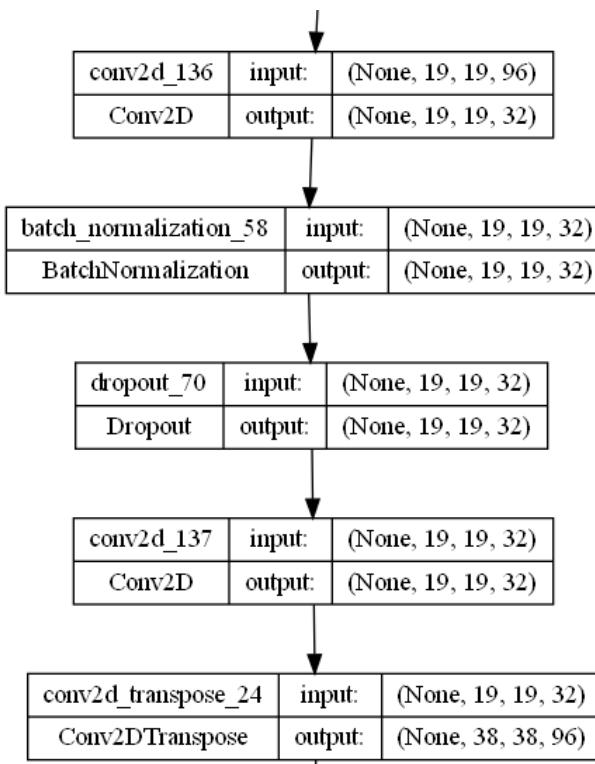


Figura 7.14: Capas que componen el *bottleneck* de cada una de las cuatro redes *U-Net* del modelo *UNet-Paralela*.

Como vemos en la figura 7.14, el *bottleneck* se implementa utilizando una combinación de capas de convolución, normalización por lotes y *dropout*. Primero, se aplica una convolución 2D con un filtro de tamaño  $3 \times 3$  y una profundidad de 32 filtros. Al igual que el resto de capas convolucionales, la función de activación utilizada es ReLU y se aplica un regularizador  $L_2$  con un factor de 0,01.

Después de la primera convolución, se aplica una normalización por lotes, conocida

## 7.4. RED UNET-PARALELA

como *Batch Normalization*, que estabiliza y acelera el proceso de entrenamiento ajustando y escalando la salida de la capa convolucional. Este proceso es seguido por la incorporación de *dropout*, con una tasa de eliminación de neuronas del 60 %, lo que contribuye a la reducción del sobreajuste al desactivar aleatoriamente una fracción de las neuronas durante el entrenamiento. Finalmente, se realiza una segunda operación de convolución con los mismos parámetros que la primera, reforzando la capacidad de la red para capturar características más complejas de la entrada.

### Camino de expansión

El camino de expansión, o decodificador, es la fase de la red *U-Net* que se encarga de reconstruir la imagen a partir de las características procesadas en el *bottleneck*, aumentando progresivamente la resolución espacial mientras combina estas características con las provenientes del camino de contracción.

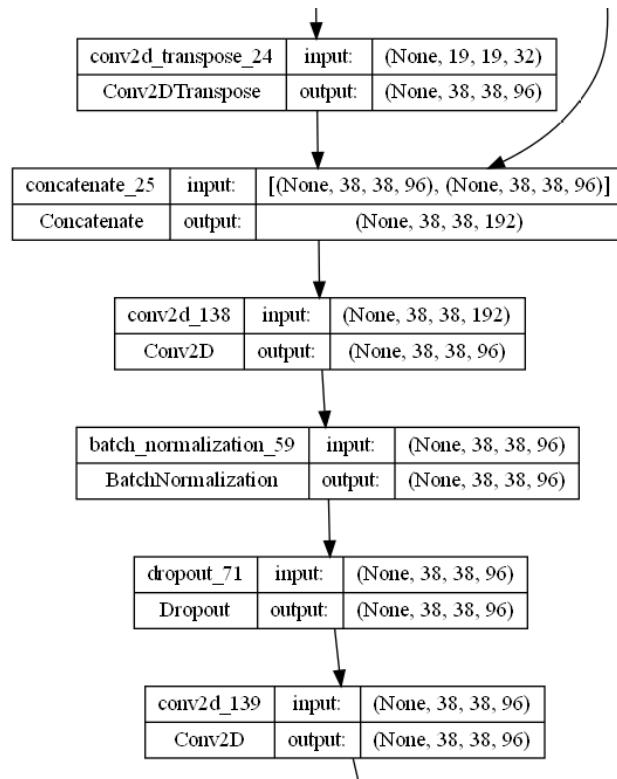


Figura 7.15: Capas que componen un módulo del camino de expansión de cada una de las cuatro redes *U-Net* del modelo *UNet-Paralela*.

## CAPÍTULO 7. DESARROLLO DEL PROYECTO

Los bloques que componen este camino están formados por las capas representadas en la figura 7.15 y que se explicarán a continuación.

Primero, se utiliza una capa de convolución transpuesta (*Conv2DTranspose*) para duplicar la resolución de las características extraídas en el *bottleneck*, generando así un mapa de características más grande. A continuación, este mapa de características se concatena con las características correspondientes de la fase de contracción, asegurando que la información espacial fina capturada en las primeras capas no se pierda durante la reconstrucción.

Después de la concatenación, se aplican una capa de convolución (*Conv2D*) utilizando una cantidad de filtros que se va decrementando con el mismo valor que en el camino de codificación, esta capa está seguida por una normalización de *batch* (*BatchNormalization*) para estabilizar y acelerar el entrenamiento, y una capa de *Dropout* para prevenir el sobreajuste. Por último, se utiliza otra capa de convolución. Las convoluciones utilizan una función de activación *ReLU* para mantener la no linealidad y se aplica un regularizador  $L_2$  con un factor de 0,01. Esta secuencia se repite varias veces, aumentando progresivamente la resolución de las características, mientras se combina la información con las capas de contracción correspondientes.

Finalmente, se utiliza una última capa de convolución que reduce el número de filtros a solo uno, produciendo un mapa de salida que representa la segmentación final.

### Capas de reducción de la resolución

Tal y como se puede ver en la figura 7.16, después del camino de expansión se utilizan una serie de capas convolucionales adicionales para refinar aún más la salida de la red y reducir la resolución del mapa de características resultante del camino de expansión de cada red *U-Net*. Esto se hace por motivos de limitaciones hardware, al exceder la red la cantidad de RAM disponible.

Primero, se aplica una capa de *Dropout* para continuar mitigando el riesgo de sobreajuste. Luego, se utiliza una capa de convolución (*Conv2D*) con un único filtro y

## 7.4. RED UNET-PARALELA

un tamaño de kernel de 3x3, que no solo reduce la dimensionalidad espacial mediante el uso de un *stride* de 2x2, sino que también refina las características extraídas. La función de activación *ReLU* sigue siendo utilizada para mantener la no linealidad en la predicción.

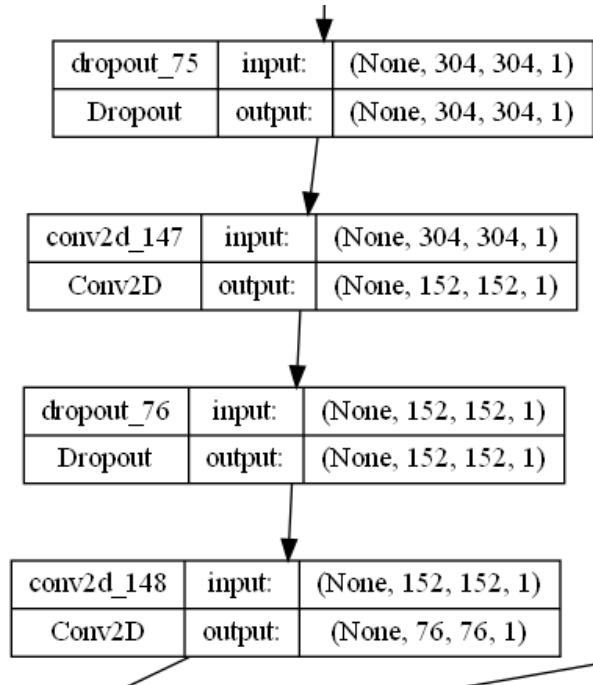


Figura 7.16: Capas que se encargan de reducir las dimensiones del mapa de características de las cuatro redes *U-Net* del modelo *UNet-Paralela*.

Este proceso de convolución, normalización mediante *Dropout*, y reducción de la resolución se repite, consolidando la predicción final en un mapa de características que representa la segmentación o clasificación de la imagen procesada. De esta manera, las capas de clasificación proporcionan una salida final precisa y lista para ser interpretada o utilizada en aplicaciones posteriores.

### Capas de clasificación

La salida producida por las cuatro redes *U-Net* individuales se combina en una etapa final de clasificación que integra la información procesada por cada una de ellas. En la figura 7.17 se muestra la representación de estas capas. En primer lugar, las salidas de las cuatro redes *U-Net* se concatenan a lo largo del eje de

## CAPÍTULO 7. DESARROLLO DEL PROYECTO

las características, formando un mapa de características compuesto que contiene la información combinada de todas las redes.

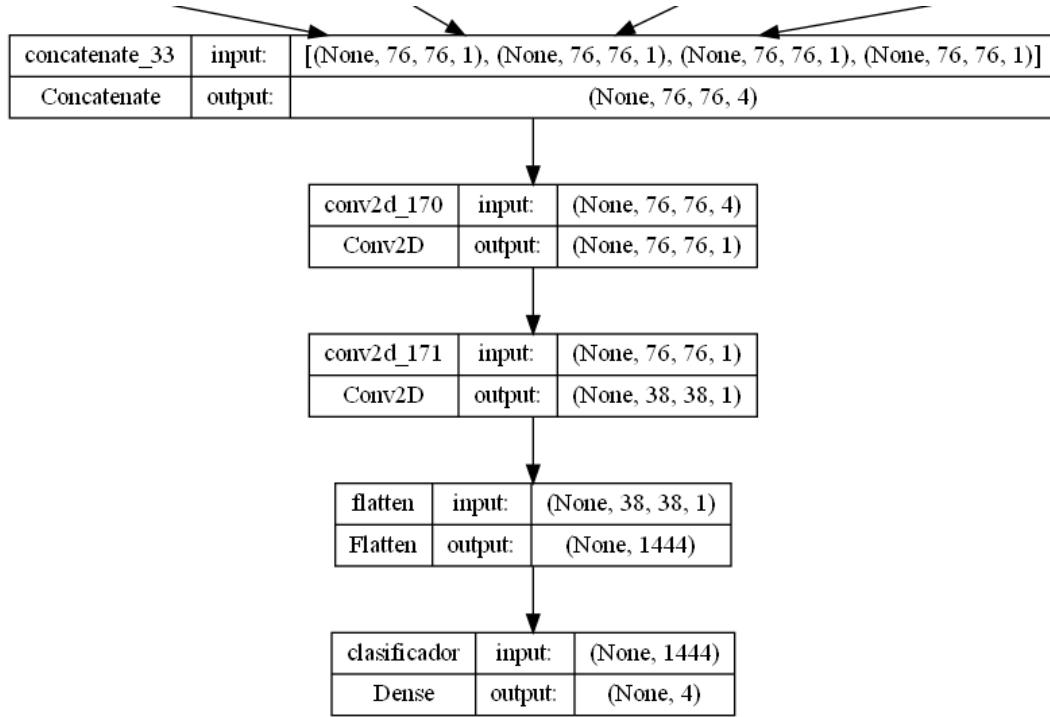


Figura 7.17: Capas que se encargan de combinar la salida de las cuatro redes *U-Net* y realizar la clasificación de los distintos casos del modelo *UNet-Paralela*.

A continuación, este mapa concatenado se somete a un par de capas convolucionales adicionales (*Conv2D*) con un tamaño de kernel de 2x2. La primera de estas capas realiza una operación de convolución simple con una función de activación *ReLU* para refinar aún más las características combinadas. Luego, se aplica una segunda capa de convolución con un *stride* de 2x2, lo que reduce la resolución espacial del mapa de características resultante, compactando la información para la etapa de clasificación final.

Después de las operaciones de convolución, se incluye una capa de *Flatten*, que convierte el mapa de características bidimensional en un vector unidimensional, preparado para ser procesado por la última capa densa (*Dense*). Esta capa densa, que utiliza una función de activación *softmax*, produce la salida final del modelo, proporcionando probabilidades asociadas a cada una de las clases en el problema

## 7.5. RED INCEPTIONV3-PARALELA

de clasificación. Así, se completa el proceso de integración y clasificación de las características extraídas por las redes UNet, generando una predicción final precisa.

### 7.5. Red *Inceptionv3-Paralela*

En esta sección veremos el origen de las redes *Inception*, así como los detalles técnicos más reseñables de la red *Inception V3*. Después, conoceremos cómo se ha implementado este modelo para nuestro problema particular y nuestro tipo de datos. Para conseguir los mejores resultados posibles se ha optado por un enfoque de modelos *Inception* en paralelo mas una clasificación final. Por último, se comentará el proceso de entrenamiento.

Las radiografías de mama suelen contener detalles finos y variaciones en las texturas y formas que son cruciales para el diagnóstico. Veamos el rendimiento de *Inception V3* con su capacidad para capturar información a diferentes escalas y su eficiencia en el procesamiento de imágenes complejas.

#### 7.5.1. Origen y evolución de *Inception*

La red *Inception* fue un hito importante en el desarrollo de los clasificadores basados en Redes Neuronales Convolucionales, *CNN*<sup>8</sup>. Antes, las *CNN* más populares simplemente apilaban capas de convolución cada vez más profundas, con la esperanza de obtener un mejor rendimiento. La red *Inception*, por otro lado, era compleja y altamente diseñada, utilizando diferentes técnicas para impulsar el rendimiento, tanto en términos de velocidad como de precisión. Su constante evolución llevó a la creación de varias versiones de la red, incluyendo *Inception V1, V2, V3, V4* e *Inception-ResNet*, según nos indica el artículo '*Rethinking the Inception Architecture for Computer Vision*', de Szegedy y cols. (2016).

<sup>8</sup>*CNN: Convolutional Neural Networks*, por sus siglas en inglés.

### 7.5.2. Detalles técnicos de *Inception V3*

#### Convoluciones factorizadas

Una de las innovaciones clave en *Inception V3* es la factorización de convoluciones más grandes en una serie de convoluciones más pequeñas y manejables. Por ejemplo, una convolución de  $5 \times 5$  puede ser factorizada en dos convoluciones de  $3 \times 3$ . Esto no solo reduce el número de parámetros, sino que también mejora la eficiencia computacional. En la figura 7.18 podemos observar un ejemplo de lo dicho.

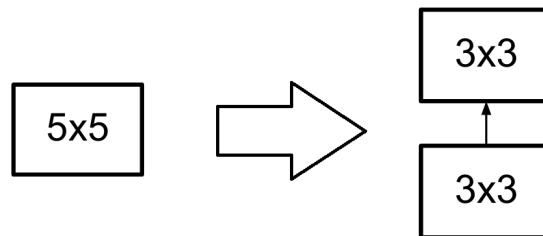


Figura 7.18: Ejemplo de factorización de una convolución de  $5 \times 5$  en dos convoluciones de  $3 \times 3$  (Szegedy y cols., 2016).

#### Módulos *Inception* mejorados

Los módulos *Inception* en V3 han sido optimizados para gestionar de manera más eficiente las variaciones en la escala de las características de entrada. Cada módulo combina convoluciones de múltiples tamaños ( $1 \times 1$ ,  $3 \times 3$ ,  $5 \times 5$ ) y *pooling*<sup>9</sup> en paralelo, permitiendo que la red capture información a diferentes escalas de manera simultánea.

En la figura 7.19 podemos vislumbrar un ejemplo de cómo se organizan estos módulos en paralelo.

---

<sup>9</sup>*Pooling*: proceso de reducción de la dimensionalidad de las representaciones.

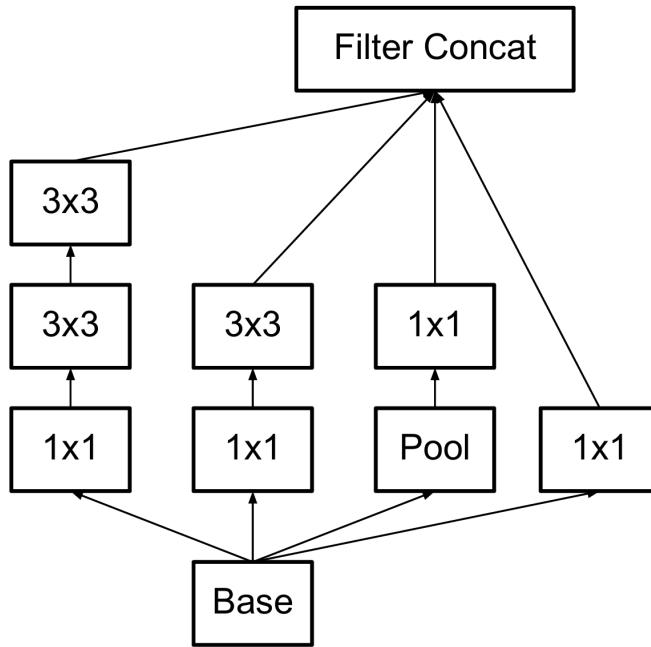


Figura 7.19: Estructura de un módulo *Inception* con diferentes tamaños de filtros operando en paralelo (Szegedy y cols., 2016).

### Uso de *RMSProp* y suavizado de etiquetas

*Inception V3* incorpora el optimizador *RMSProp*<sup>10</sup> y técnicas de suavizado de etiquetas<sup>11</sup>. El uso de *RMSProp* mejora la estabilidad y velocidad del entrenamiento, mientras que el suavizado de etiquetas actúa como un regularizador, mejorando la generalización del modelo.

### 7.5.3. Aplicación de *Inception V3* en la clasificación de radiografías de mama

La arquitectura *Inception V3* ha demostrado ser robusta frente a variaciones en los datos y puede adaptarse bien a conjuntos de datos médicos diversificados, lo cual es fundamental dado que las imágenes médicas pueden variar en términos de calidad, resolución y condiciones de captura. En este proyecto, se ha empleado *Inception*

<sup>10</sup>*RMSProp*: Root Mean Square Propagation, un método de optimización.

<sup>11</sup>Suavizado de etiquetas (*label smoothing*): técnica que evita el sobreajuste asignando probabilidades no cero a clases incorrectas.

V3 para la clasificación de radiografías de mama, aprovechando su capacidad para capturar detalles finos y variaciones de textura en las imágenes, lo cual es crucial para el diagnóstico precoz de cáncer y otras anomalías.

#### 7.5.4. Implementación del modelo *Inceptionv3-Paralela*

La implementación se llevó a cabo en función de la naturaleza de los datos, buscando la eficiencia computacional sin perder rendimiento.

Para comenzar, recordemos que cada caso consta de un máximo de cuatro radiografías, en las que no todas son representativas del caso que se pretende clasificar. Es decir, puede haber un caso con un tumor benigno en la mama derecha, pero las mamografías de la mama izquierda pueden ser completamente sanas.

Debido a lo anterior, no se puede analizar cada mamografía por separado. Por ello, se implementó un modelo que combina cuatro modelos *Inception V3* en paralelo, cada uno sin la capa superior (*top*) y con los pesos preentrenados en *ImageNet*<sup>12</sup>. Para hacer esto, primero se declaran cuatro capas *Lambda*, las cuales se encargan de dividir cada una de las mamografías y pasárlas a los diferentes modelos *Inception*, tal como se puede ver en la figura 7.20.

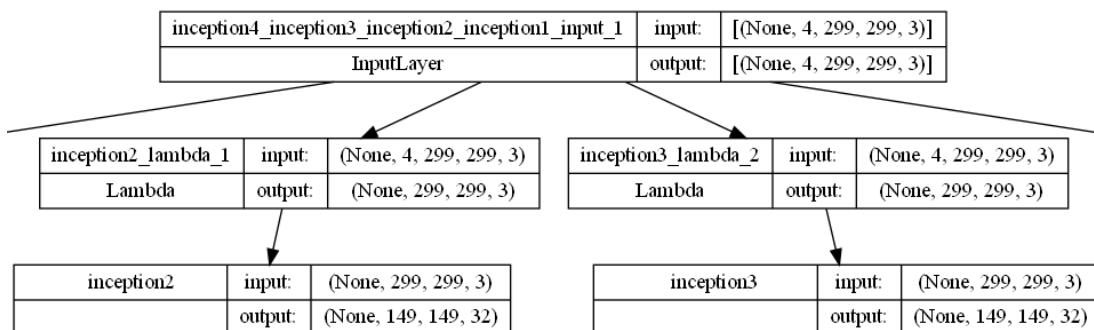


Figura 7.20: Capas *Lambda* en el modelo *Inceptionv3-Paralela* implementado.

La utilización de *feature extraction* con la arquitectura *Inception V3* en este modelo tiene un gran potencial para esta tarea. Mantiene un coste computacional asequible,

<sup>12</sup>*ImageNet*: un conjunto de datos amplio utilizado para preentrenar modelos de visión por computadora.

## 7.5. RED INCEPTIONV3-PARALELA

lo que la hace idónea para nuestro *hardware* descrito en la sección 4.1, y, a su vez, la combinación de múltiples modelos *Inception V3* permite una gran capacidad para capturar características complejas y detalles finos en las imágenes de mamografías.

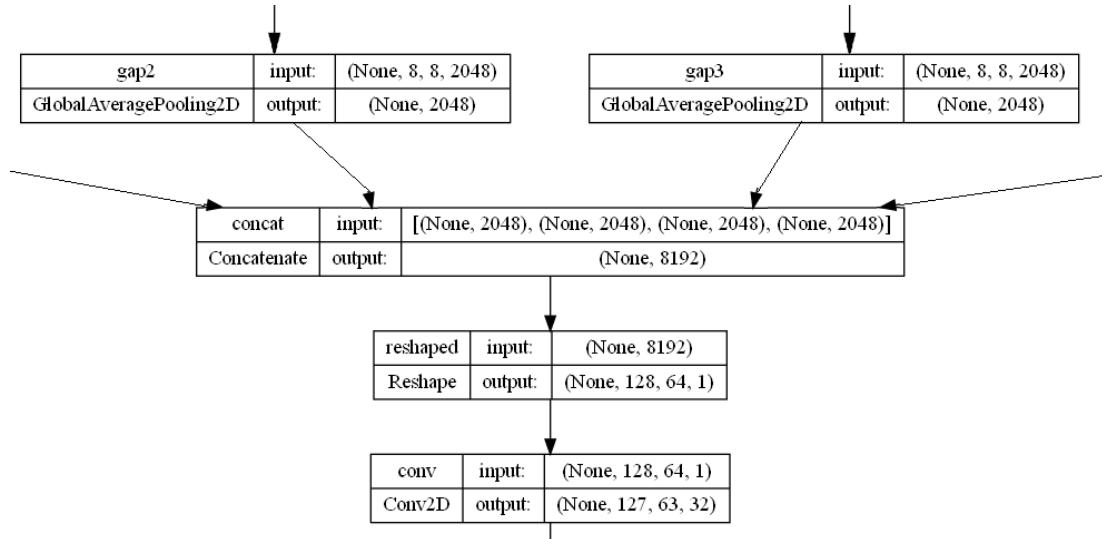


Figura 7.21: Capas GlobalAveragePooling2D, Concatenate, Reshape y Conv2D en el modelo *Inceptionv3-Paralela* implementado.

Como podemos observar en la figura 7.21, después de la arquitectura principal de cada modelo *Inception V3*, se añadió una capa GlobalAveragePooling2D. Esta capa reduce las dimensiones de las características extraídas por los modelos *Inception v3* previos. Específicamente, realiza una operación de *pooling* global que convierte una entrada de varias dimensiones (una matriz 2D por canal) en un único valor promedio por canal. Esta técnica ayuda a reducir el número de parámetros y a prevenir el sobreajuste, además de mantener la información más importante.

Las salidas resultantes de las capas GlobalAveragePooling2D se concatenan en un solo tensor mediante la capa Concatenate, cuya salida tiene una dimensión de 8192 filtros. Después de esto, el tensor concatenado se remodela en una imagen 2D con dimensiones  $128 \times 64 \times 1$  mediante la capa Reshape. Se eligen estas dimensiones ya que son la combinación que, al multiplicarse, dan como resultado los 8.192 filtros de la capa Concatenate.

La importancia de crear una matriz de datos bidimensional radica en la capacidad

## CAPÍTULO 7. DESARROLLO DEL PROYECTO

de esta estructura para capturar patrones espaciales complejos que son característicos de las imágenes. Permite que la siguiente capa convolucional identifique y aprenda estos patrones de manera más efectiva, mejorando el rendimiento del modelo en la tarea de clasificación de imágenes.

Por lo tanto, se aplica una capa Conv2D con 32 filtros, un tamaño de kernel de  $2 \times 2$  para extraer características adicionales de la imagen generada, y una función de activación ReLU para introducir no linealidad en el modelo.

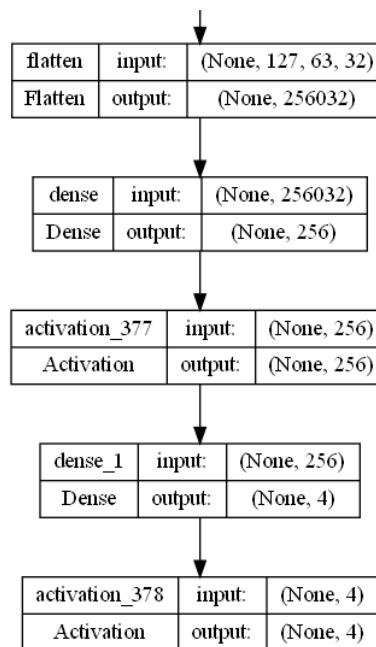


Figura 7.22: Últimas capas del modelo *Inceptionv3-Paralela* implementado.

Tal y como se muestra en la figura 7.22, la salida de la capa convolucional se aplana (Flatten) y luego se pasa a una capa completamente conectada (Dense) con 256 neuronas. Se utiliza la función de activación Mish para esta capa, que ha demostrado ser efectiva en varias tareas de aprendizaje profundo, haciendo que el entrenamiento sea mas estable y reduciendo el sobreajuste u *overfitting*.

Finalmente, la capa de salida (Dense) produce una predicción para cada clase posible, utilizando una activación Softmax. La función de activación Softmax convierte las salidas del modelo en probabilidades, asignando un valor entre 0 y 1

## 7.5. RED INCEPTIONV3-PARALELA

a cada clase de manera que la suma de todas las probabilidades sea 1. Esto es útil en tareas de clasificación, ya que proporciona una medida de confianza sobre a qué clase pertenece la entrada, facilitando la interpretación de los resultados del modelo.

### Resumen de la arquitectura *Inceptionv3-Paralela* implementada

A continuación, se presenta un resumen de las capas del modelo *Inceptionv3-Paralela* finalmente implementado:

- *Lambda* (4 capas): selección de arrays para cada modelo *Inception V3*.
- *InceptionV3* (4 modelos): modelos preentrenados sin la capa superior.
- *GlobalAveragePooling2D* (4 capas): reducción de mapas de características bidimensionales.
- *Concatenate*: concatenación de las salidas de las capas *GlobalAveragePooling2D*.
- *Reshape*: remodelación del tensor concatenado en una imagen 2D.
- *Conv2D*: extracción de características adicionales con 32 filtros y kernel de  $2 \times 2$ .
- *Flatten*: aplanamiento de la salida de la capa *Conv2D*.
- *Dense* (256 neuronas): capa completamente conectada con activación *Mish*.
- *Dense* (capa de salida): predicción para cada clase con activación *Softmax*.

#### 7.5.5. Compilación del modelo *Inceptionv3-Paralela*

Para compilar el modelo se siguieron varios pasos, incluyendo la configuración del optimizador, la especificación de la función de pérdida y la selección de las métricas de evaluación.

El modelo se compiló utilizando el optimizador *Adam*, con un *learning rate* de 0,001. Este optimizador es uno de los más populares y efectivos debido a su adaptabilidad del ritmo de aprendizaje.

La función de pérdida utilizada es la misma que en el resto de modelos, la cual fue explicada anteriormente en este mismo capítulo en la sección [7.2.1. Función de pérdida ponderada](#).

La métrica que utilizaremos para observar el rendimiento del modelo durante el entrenamiento será la exactitud, o accuracy, la métrica más común por medir el porcentaje de predicciones correctas del modelo.

## 7.6. Red *ResNetv2-Paralela*

A continuación veremos el origen de la arquitectura *ResNet V2*, así como sus detalles técnicos. Después, nos adentraremos la implementación realizada mediante *transfer learning* para nuestro problema de clasificación y, por último, se explicará el proceso de entrenamiento establecido.

Después de aplicar el modelo *Inception V3*, se ha utilizado un modelo más moderno, como es el modelo *Inception-ResNet V2*, para intentar conseguir una mejora en la precisión. En este apartado veremos cómo se ha realizado la adaptación.

### 7.6.1. Origen y evolución de *ResNet V2*

Según el artículo redactado por los desarrolladores de *Google* titulado '*Inception-V4, inception-resnet and the impact of residual connections on learning*', realizado por [Szegedy y cols. \(2017\)](#), la arquitectura *ResNet* significó un avance destacable en el campo del aprendizaje profundo, abordando el problema de la desaparición del gradiente que limitaba el rendimiento de las redes muy profundas. La innovación clave fue la introducción de bloques residuales, que permiten el paso directo de la información a través de capas. Esta técnica ha permitido entrenar redes mucho más profundas de manera efectiva.

*ResNet V2*, o *Residual Network V2*, es una evolución de la arquitectura original de *ResNet* propuesta por [He y cols. \(2016\)](#). Esta versión mejora sobre la arquitectura de *ResNet* original al introducir modificaciones clave que optimizan el rendimiento y la estabilidad del entrenamiento.

La idea principal detrás de *ResNet* es la incorporación de conexiones residuales, las cuales permiten que la información fluya más fácilmente a través de la red. Esto es

## 7.6. RED RESNETV2-PARALELA

particularmente útil en redes profundas donde los gradientes tienden a desvanecerse, dificultando el entrenamiento. En *ResNet V2*, estas conexiones se han refinado para mejorar aún más la capacidad de la red de aprender representaciones complejas sin comprometer la estabilidad del entrenamiento.

Una de las mayores ventajas de la arquitectura *ResNet V2* es su capacidad para mejorar la eficiencia del modelo, reducir el tiempo de entrenamiento y asegurar una mayor estabilidad en el proceso de aprendizaje. Esto resulta en una red que puede alcanzar resultados comparables a los de la arquitectura *Inception V3* en un tiempo significativamente menor.

Es importante destacar que uno de los mayores limitantes en los modelos de inteligencia artificial es la capacidad computacional disponible. Por lo tanto, una red que puede lograr los mismos resultados con menos procesamiento es una mejora significativa en este campo. La eficiencia computacional de *ResNet V2* no solo optimiza el uso de recursos, sino que también permite el desarrollo de modelos más robustos y precisos, incluso cuando se dispone de recursos computacionales limitados. Esto representa un avance crucial en el desarrollo y aplicación de redes neuronales profundas, facilitando su implementación en una variedad de entornos y aplicaciones donde la potencia computacional es una restricción crítica, como es el de este caso.

### 7.6.2. Detalles técnicos de *Inception-ResNet V2*

Esta arquitectura de *Inception-ResNet V2* está diseñada para combinar las ventajas de las conexiones residuales (que facilitan el entrenamiento de redes profundas) con la capacidad de los módulos *Inception* para capturar características a múltiples escalas.

A continuación, vamos a describir cómo se comporta la arquitectura *ResNet V2*.

#### Arquitectura de *ResNet V2*

La figura 7.23 representa el diagrama de la arquitectura de una variante de *ResNet* llamada *Inception-ResNet V2*. La arquitectura combina los bloques residuales con los módulos *Inception*, lo que permite aprovechar las ventajas de ambos enfoques.

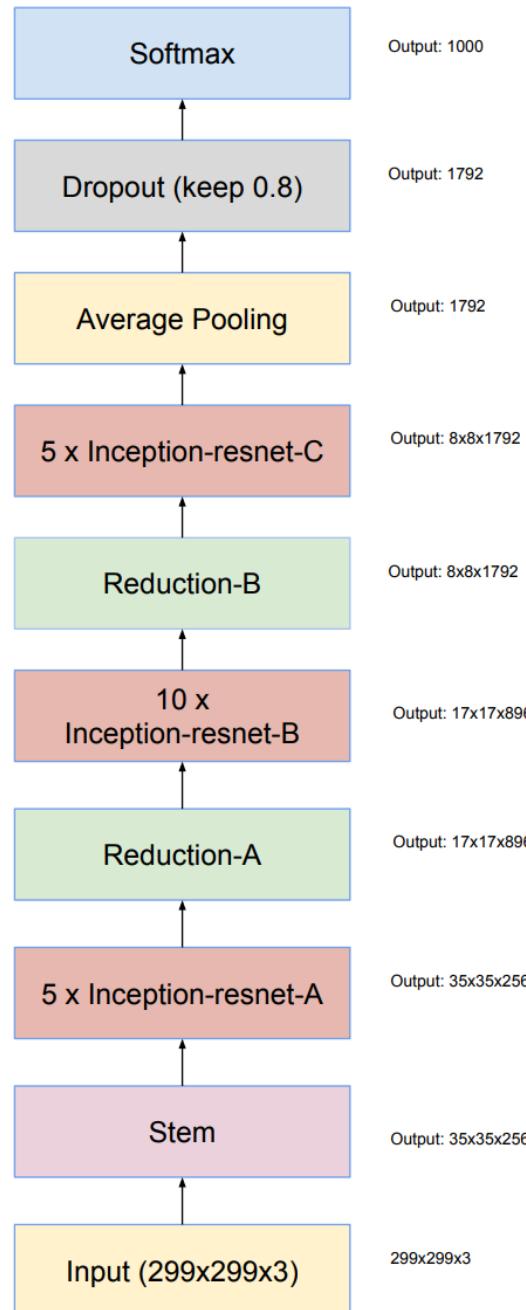


Figura 7.23: Estructura de la arquitectura *Inception-ResNet V2* (Szegedy y cols., 2017).

Continuando con la figura 7.23, los módulos *Inception* capturan información en múltiples escalas mediante el uso de convoluciones de diferentes tamaños en paralelo, mientras que las conexiones residuales facilitan el entrenamiento de redes más profundas.

## 7.6. RED RESNETV2-PARALELA

A continuación, se explica detalladamente cada componente de la arquitectura de Szegedy y cols. (2017):

- *Input* ( $299 \times 299 \times 3$ ): la entrada a la red es una imagen de tamaño  $299 \times 299$  píxeles con 3 canales de color (*RGB*).
- *Stem*: esta etapa inicial de la red se encarga de realizar una serie de operaciones convolucionales y de *pooling* para reducir la dimensión espacial de la imagen y extraer características iniciales. La salida de esta etapa es de tamaño  $35 \times 35 \times 256$ .
- $5 \times Inception\text{-}ResNet\text{-}A$ : esta sección consiste en 5 bloques *Inception-ResNet-A*, que son módulos residuales diseñados para captar diferentes escalas de características dentro de la imagen. Cada módulo combina múltiples convoluciones ( $1 \times 1, 3 \times 3, 5 \times 5$ ) y luego suma la entrada original a la salida del módulo. La salida de esta sección permanece en  $35 \times 35 \times 256$ .
- *Reduction-A*: este bloque realiza una reducción en la dimensión espacial, permitiendo que la red profundice y capture características más abstractas. Esto se logra a través de convoluciones y *pooling*. La salida es reducida a  $17 \times 17 \times 896$ .
- $10 \times Inception\text{-}ResNet\text{-}B$ : similar a los bloques *Inception-ResNet-A*, esta sección contiene 10 módulos *Inception-ResNet-B*. La salida de esta sección permanece en  $17 \times 17 \times 896$ .
- *Reduction-B*: otra etapa de reducción de dimensiones que prepara los datos para los siguientes bloques de *Inception-ResNet*. La salida es de  $8 \times 8 \times 1792$ .
- $5 \times Inception\text{-}ResNet\text{-}C$ : esta sección contiene 5 módulos *Inception-ResNet-C*, que funcionan de manera similar a los anteriores. La salida es de  $8 \times 8 \times 1792$ .
- *Average Pooling*: esta capa realiza un promedio global sobre las características, reduciendo la dimensión espacial a un vector de características de longitud 1.792. Por lo tanto la salida es de tamaño 1.792.

- *Dropout (keep 0.8)*: una técnica de regularización que ayuda a prevenir el sobreajuste, manteniendo el 80% de las unidades activas durante el entrenamiento y apagando el 20% restante de manera aleatoria. La salida sigue siendo de tamaño 1.792.
- *Softmax*: la capa final de la red, que aplica la función *Softmax* para convertir las características finales en probabilidades de pertenencia a cada una de las clases objetivo. La salida es un vector de tamaño 1.000, correspondiente a las 1.000 clases del conjunto de datos *ImageNet*.

### Estructura de bloques residuales

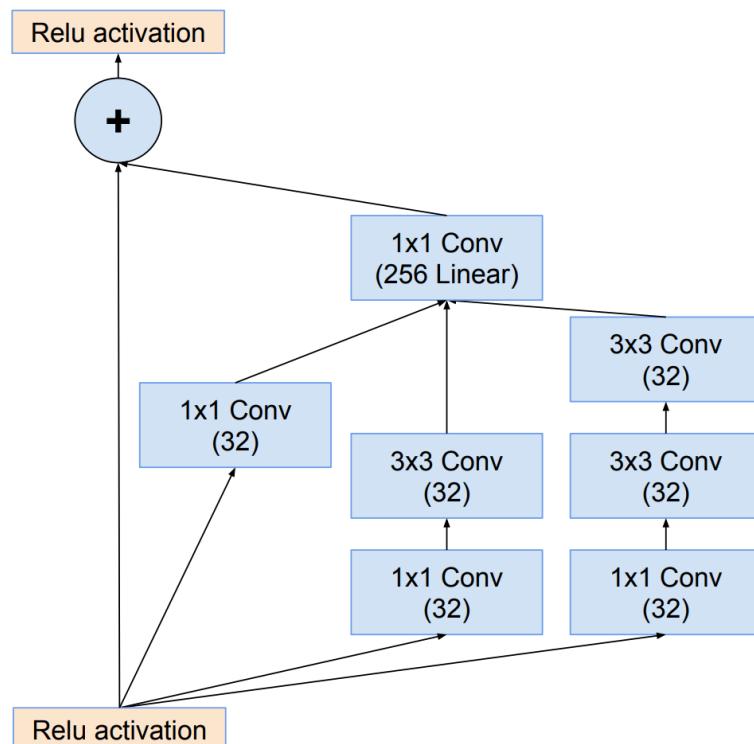


Figura 7.24: Estructura de la capa *Inception-ResNet-A* ([Szegedy y cols., 2017](#)).

Una de las características distintivas de *ResNet V2* es el uso de bloques residuales. Cada bloque incluye una conexión directa que permite que el gradiente fluya sin obstáculos a través de la red, facilitando el entrenamiento de redes profundas. En *ResNet V2* la normalización por lotes (*batch normalization*) y la activación (*ReLU*)

## 7.6. RED RESNETV2-PARALELA

se aplican antes de las convoluciones, lo cual representa una modificación respecto a la versión original *Inception*.

### 7.6.3. Implementación del modelo *ResNetv2-Paralela*

Tal y como se explica en la sección 7.5.4, inicialmente se utilizó el modelo *Inception v3*; sin embargo, en esta fase del proyecto se decidió emplear *ResNet V2* para mejorar los resultados. El cuerpo de este modelo tiene prácticamente la misma estructura que el empleado con el modelo *Inception V3*, por lo que en aras de no repetir los mismos conceptos de nuevo se procederá a explicar solo las diferencias.

Debido a que los modelos *ResNet V2* aceptan las mismas dimensiones que los modelos *Inception V3*, las primeras capas se encuentran inalteradas. La mayor diferencia radica en la salida, tal y como se puede ver en la figura 7.25, pues el modelo *ResNet v2* tiene una salida de (tamaño de *batch*, 8, 8, 1534) mientras que el modelo *InceptionV3* tenía una salida de (tamaño de *batch*, 8, 8, 2048). Tiene como resultado que todas las capas siguientes tengan que adaptarse a estas nuevas dimensiones, por lo que las capas *GlobalAveragePooling2D* ahora tienen unas dimensiones de 1.534 de salida. Esto a su vez hace que al concatenar estas capas dé como resultado una salida de 6.144 en la capa *Concatenate* ( $1534 \times 4 = 6144$ ) y, por tanto, las dimensiones de la imagen formada por la capa *Reshape* tengan las dimensiones de  $96 \times 64 \times 1$ .

Al tener una salida más pequeña la capa *Conv2D* y *Dense* pueden tener más dificultades para identificar las características que definen a los tumores benignos o al cáncer. Esto es debido a que la salida de menor tamaño significa que la información que llega a las capas convolucionales está más comprimida, lo que puede llevar a una pérdida de detalles que son cruciales para la correcta identificación y clasificación de las imágenes médicas. Esta compresión de información puede hacer que la red necesite más ajustes para poder aprender a extraer y utilizar las características relevantes de manera efectiva.

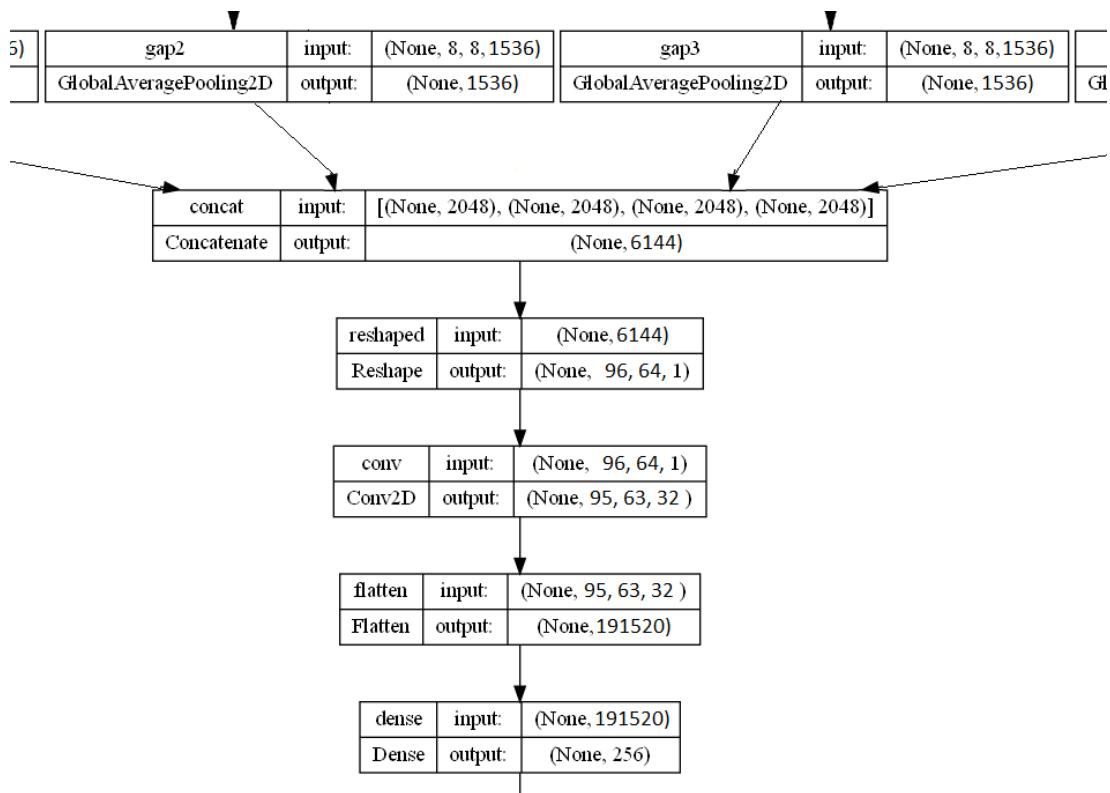


Figura 7.25: Capas modificadas en el modelo *ResNetv2-Paralela* implementado con respecto al modelo *Inception V3*.

### Resumen de la arquitectura *ResNetv2-Paralela* implementada

A continuación, se presenta un resumen de las capas del modelo final:

- *Lambda* (4 capas): selección de *array* para cada modelo *ResNet V2*.
- *ResNet V2* (4 modelos): modelos preentrenados sin la capa superior.
- *GlobalAveragePooling2D* (4 capas): reducción de mapas de características bidimensionales.
- *Concatenate*: concatenación de las salidas anteriores, provenientes de las capas *GlobalAveragePooling2D*.
- *Reshape*: remodelación del tensor concatenado en una imagen 2D.
- *Conv2D*: extracción de características adicionales con 32 filtros y kernel de  $2 \times 2$ .
- *Flatten*: aplanamiento de la salida de la capa *Conv2D*.
- *Dense* (256 neuronas): capa completamente conectada con activación *Mish*.
- *Dense* (capa de salida): predicción para cada clase con activación *Softmax*.



## *7.6. RED RESNETV2-PARALELA*

---

# Capítulo 8

## Resultados

En este capítulo se van a ver los resultados obtenidos en cada uno de los modelos. Se mostrarán tanto las métricas obtenidas durante el entrenamiento como las diferentes pruebas que se harán a cada una de las diferentes redes. Estas pruebas se harán con dos conjuntos diferentes, el conjunto de datos base y el conjunto de datos con aumento de datos. recordar que ambos conjuntos fueron explicados en el capítulo 6, concretamente en la sección [Conjuntos utilizados en el entrenamiento de modelos](#).

Para analizar el comportamiento de los modelos con el conjunto de datos de prueba se van a utilizar matrices de confusión. Es por esto que, antes de ver los resultados, conviene ver como funcionan estas matrices. La explicación de estas matrices se encuentra en el anexo de este documento, en la sección [Matrices de confusión](#).

### 8.1. Red UNet3D-Dense

Vamos a comenzar exponiendo los resultados obtenidos con la red UNet3D-Dense, la cual ha sido explicada en la sección [Red UNet3D-Densa](#) del capítulo 7.

Para realizar el entrenamiento y la fase de prueba se han utilizado los conjuntos de datos definidos en la sección [Conjuntos utilizados en el entrenamiento de modelos](#) del capítulo 6, [procesado de datos](#). Cada modelo tiene un formato adaptado para su caso específico. En este caso, los datos se han normalizados a valores entre 0 y 1 con el

## 8.1. RED UNET3D-DENSE

formato *float32*, la normalización de datos es una práctica habitual que suele mejorar la potencia de los modelos. Además, después de múltiples pruebas se optó por una resolución de  $304 \times 304$  para cada una de las mamografías. Esta resolución permite al modelo obtener suficientes características como para diferenciar las distintas categorías y, a su vez, no es muy demandante en recursos, tanto en memoria RAM como en tiempo de procesamiento.

### 8.1.1. Conjunto de datos base

El entrenamiento de esta red con este conjunto de datos llevó más de 4 horas, en las que se realizaron 5 *folds* de 20 *epochs* cada uno. En total se realizaron 100 *epochs*. A continuación vamos a ver las diferentes métricas que guiaron este proceso.

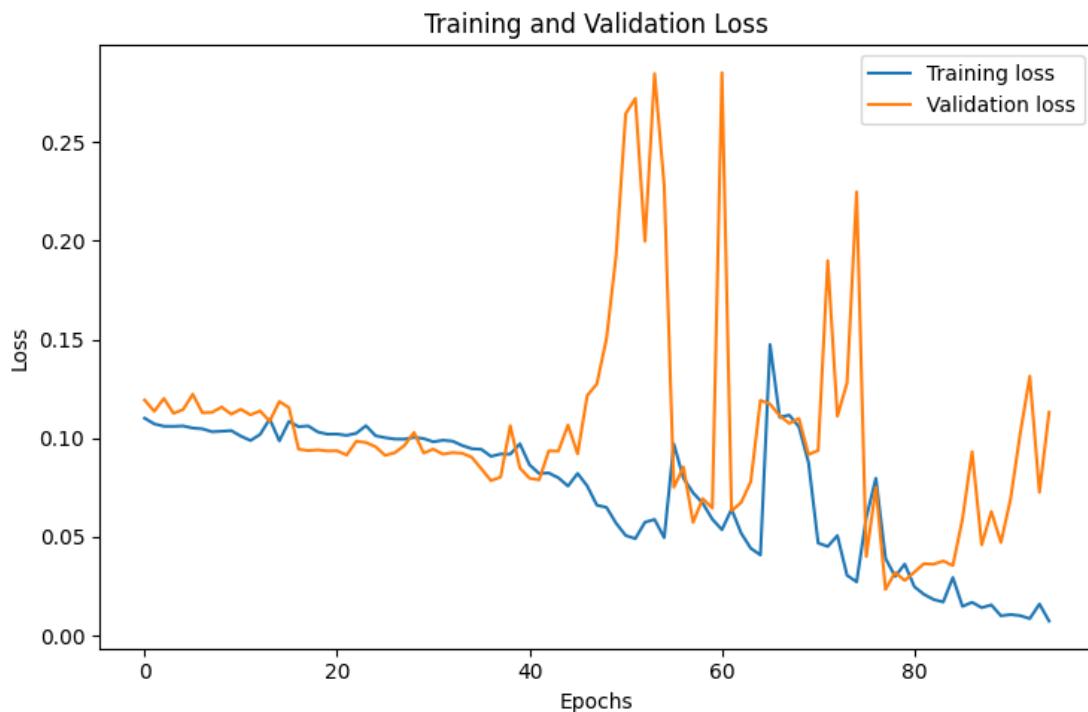


Figura 8.1: Pérdida durante el entrenamiento.

En primer lugar, la figura 8.1 muestra la evolución de la pérdida durante este proceso de entrenamiento. Podemos ver como ya en el primer *epoch* ya muestra un rendimiento bueno, esto indica que ha aprendido bien desde el principio. Después, las pérdidas continúan bajando progresivamente, la pérdida de entrenamiento

## CAPÍTULO 8. RESULTADOS

de forma estable y la pérdida de validación con inestabilidad a partir del *epoch* número 40. Esta inestabilidad puede ser producida debido a el cambio de datos que se produce en cada *fold*, obligando al modelo a aprender casos distintos.

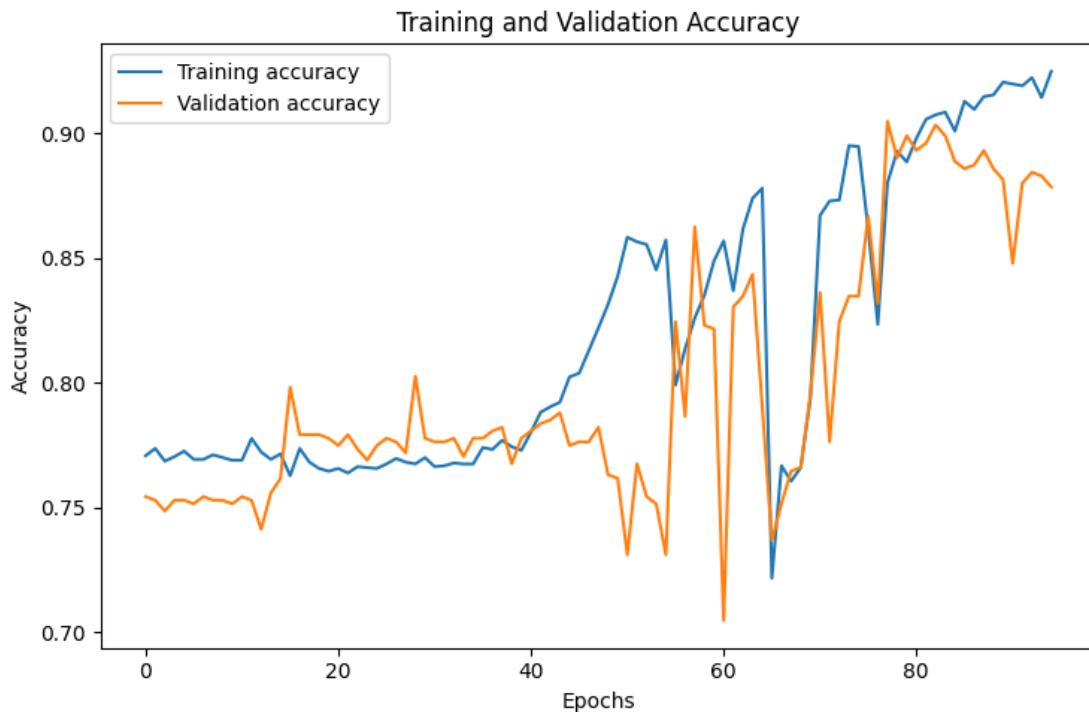


Figura 8.2: *Accuracy* durante el entrenamiento.

En segundo lugar, la figura 8.2 muestra la evolución del *accuracy*. Podemos ver como tanto el *accuracy* de validación se mantiene parejo a el de entrenamiento durante todo el proceso, por lo que no hay muestras de sobreajuste. En un principio empieza con un valores superiores al 75 %, a partir de entonces, empieza a aumentar de manera progresiva y constante hasta encontrarse en valores cercanos al 90 %.

Aunque las gráficas no nos muestran signos de sobreajuste, si es posible, esto es debido a que al utilizar validación cruzada se están reciclando datos. La existencia de sobre ajuste se comprobará al realizar las pruebas con nuevos datos.

### Pruebas del modelo

Para comprobar el rendimiento de una red neuronal entrenada con validación cruzada se debe de realizar la media aritmética de los resultados obtenidos en cada

## 8.1. RED UNET3D-DENSE

iteración de datos. Por esto, se han guardado dos modelos por cada *fold*, uno que representa al modelo y que más *accuracy* a conseguido con el conjunto de validación, y el otro, que representa el modelo al final del *fold*.

Los resultados que nos han dado estos modelos con los datos de prueba se pueden ver en las figuras 8.3 y 8.4, de forma gráfica, y en la tabla 8.1, expresado de forma numérica.

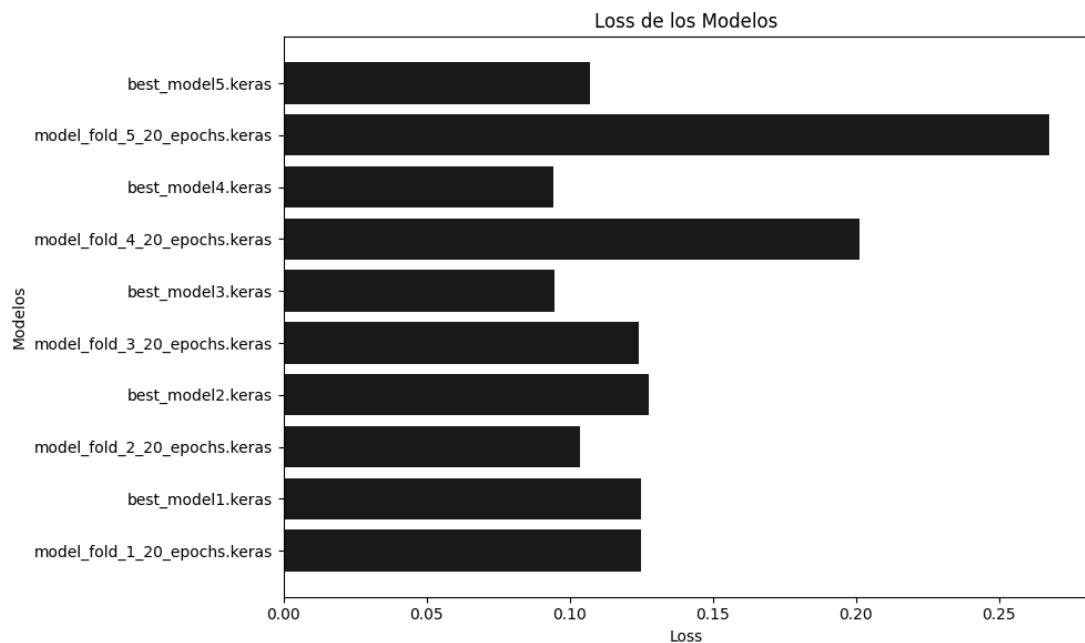


Figura 8.3: Pérdida de los diferentes modelos.

En la figura 8.3 podemos ver como la pérdida de se mantiene en valores cercanos al 0,1 a excepción de dos picos en los modelos “model\_fold\_4\_20\_epochs.keras” y “model\_fold\_5\_20\_epochs.keras”. Estos picos indican cierta inestabilidad en el entrenamiento. Además, la pérdida obtenida es relativamente alta, lo que indica que los diferentes modelos no muestran una alta seguridad a la hora de categorizar los diferentes datos.

En la figura 8.4 se muestra el *accuracy* obtenido. El resultado es favorable, pues ha ido aumentando conforme se han realizado los diferentes *folds*. Es por esto, que la validación cruzada ha ayudado ha entrenar el modelo detectando las diferentes características que definen cada categoría.

## CAPÍTULO 8. RESULTADOS

---

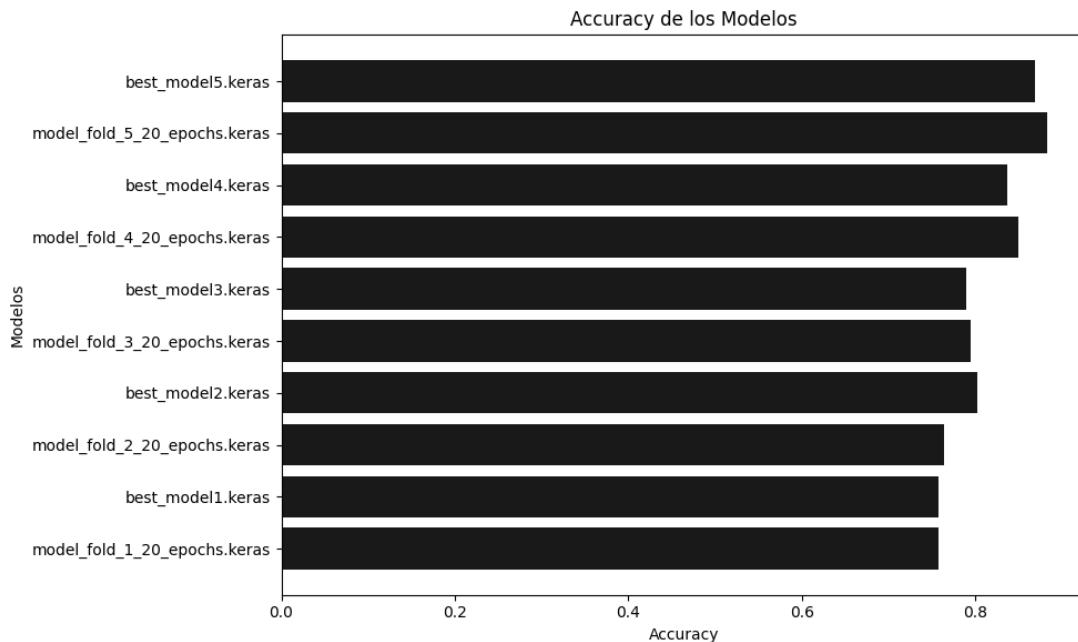


Figura 8.4: *Accuracy* de los diferentes modelos.

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,1250	0,7573
best_model1.keras	0,1250	0,7573
model_fold_2_20_epochs.keras	0,1038	0,7643
best_model2.keras	0,1276	0,8028
model_fold_3_20_epochs.keras	0,1241	0,7946
best_model3.keras	0,0946	0,7888
model_fold_4_20_epochs.keras	0,2014	0,8495
best_model4.keras	0,0942	0,8366
model_fold_5_20_epochs.keras	0,2678	0,8833
best_model5.keras	0,1072	0,8681
<b>Media aritmética</b>	<b>0,1371</b>	<b>0,8103</b>

Tabla 8.1: Resultados de los modelos evaluados en el conjunto de datos de prueba.

Por último, en la tabla 8.1 vemos que la media de pérdida obtenida es de 0,1371 y el *accuracy* medio es del 81,03 %. Debido a que el modelo ha ido mejorando progresivamente, se obtiene una perdida de 0,1072 y un *accuracy* del 86,81 % al final del entrenamiento. Aunque la media no muestra unos resultados muy satisfactorios, en el último fold si que se encuentran unos resultados prometedores. También hay que añadir que estas pruebas se han realizado teniendo en cuenta la categoría *actionable* y

## 8.1. RED UNET3D-DENSE

con unos datos balanceados. Debido al escaso numero de casos de esta categoría y su dificultad para ser detectada solo se han tenido en cuenta para realizar el proceso de entrenamiento.

Para ver de forma mas detallada el funcionamiento del modelo vamos a ver el rendimiento del modelo “best\_model5.keras” con cada categoría. Esta información se muestra en la matriz de confusión representada en la figura 8.5.

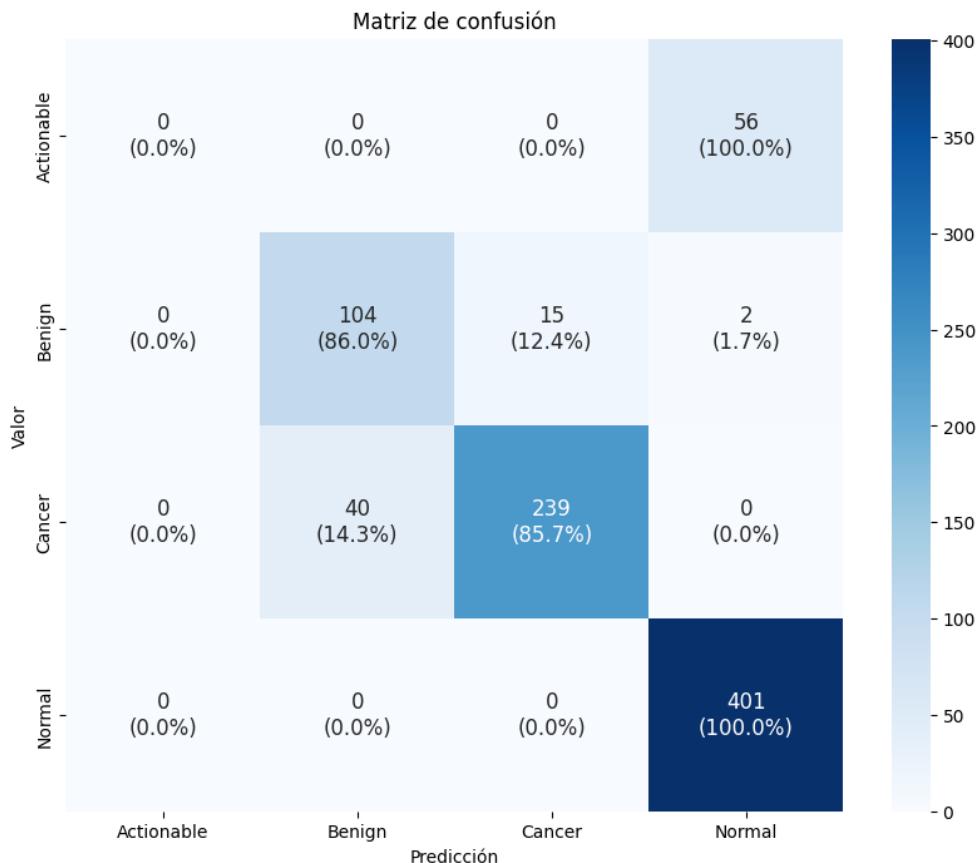


Figura 8.5: Matriz de confusión del modelo “best\_model5.keras”.

Lo primero que podemos observar es que los casos *actionable* se han clasificado todos como casos de la categoría *normal*. Esto nos indica que estos casos son más parecidos a pacientes sanos que a pacientes con un tumor o con cáncer, al menos en visualmente en un mamografía. Esto refuerza la idea de ser casos más difíciles de detectar.

Lo segundo que vemos es que logra diferenciar de manera excelente los casos

## CAPÍTULO 8. RESULTADOS

---

de pacientes sanos de los de las categorías *benign* o *cancer*. Categorizando de forma correcta todos los casos *normal* y solo 2 casos *benign* como *normal*.

En relación con los casos *benign* y *cancer*, vemos que es donde más dificultades encuentra, obteniendo un 86% y un 85,7% respectivamente. Estos resultados no son del todo malos, sobretodo teniendo en cuenta el escaso número de casos disponibles y las limitaciones hardware existentes.

En general, este modelo logra un buen rendimiento con estas pruebas, mostrando que es capaz de diferenciar en gran medida las tres categorías que nos importan, *normal*, *benign* y *cancer*.

### Pruebas sin la clase *actionable*

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,1337	0,8102
best_model1.keras	0,1337	0,8102
model_fold_2_20_epochs.keras	0,1110	0,8177
best_model2.keras	0,1365	0,8589
model_fold_3_20_epochs.keras	0,1328	0,8502
best_model3.keras	0,1013	0,8439
model_fold_4_20_epochs.keras	0,2155	0,9089
best_model4.keras	0,1008	0,8951
model_fold_5_20_epochs.keras	0,2865	0,9451
best_model5.keras	0,1147	0,9288
<b>Media aritmética</b>	<b>0,1467</b>	<b>0,8669</b>

Tabla 8.2: Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase *actionable*.

Los rendimientos anteriores ayudan a hacernos una idea de como rinde el modelo, pero, para ver un rendimiento más real vamos a correr estas pruebas sin tener en cuenta la clase *actionable*. El resultado de estas pruebas se muestra en la tabla 8.2, la cual nos da un rendimiento medio de 0,1467 de pérdida y del 92,88% de accuracy. En esta red no podemos obviar el rendimiento obtenido en el último epoch ya que ha ido mejorando los resultados de forma progresiva. Esto es algo que se puede ver tanto en las pruebas, como en las métricas de entrenamiento. Es por esto que cabe mencionar

## 8.1. RED UNET3D-DENSE

---

los resultados de su mejor modelo, el modelo “best\_model5.keras”, el cual obtiene una pérdida de 0,1147 y un *accuracy* del 92,88 %.

En todos los casos suben tanto el *accuracy* como la pérdida. El *accuracy* es natural que suba puesto que fallaba en todos los casos, sin embargo, la subida de la pérdida indica una perdida en el grado de confianza con la que el modelo categoriza cada clase. Esto indica que los casos de tipo *actionable* eran categorizados como casos de tipo *normal* con un grado de confianza mayor que la media de el resto de casos.

### Pruebas con casos normales ampliados

Para obtener un rendimiento más realista vamos a romper el desbalance de los datos añadiendo todos los casos de la categoría *normal* que fueron desechados en la división de datos y, por tanto, no se han utilizado en el proceso de entrenamiento. Esto nos da un total de 3.524 de casos *normal*.

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,0314	0,9580
best_model1.keras	0,0314	0,9580
model_fold_2_20_epochs.keras	0,0248	0,9615
best_model2.keras	0,0316	0,9679
model_fold_3_20_epochs.keras	0,0320	0,9646
best_model3.keras	0,0239	0,9666
model_fold_4_20_epochs.keras	0,0509	0,9796
best_model4.keras	0,0233	0,9771
model_fold_5_20_epochs.keras	0,0695	0,9850
best_model5.keras	0,0262	0,9832
<b>Media aritmética</b>	<b>0,0345</b>	<b>0,9702</b>

Tabla 8.3: Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase *actionable* y con datos ampliados para la clase *normal*.

En la tabla 8.3 se muestran los resultados con los casos de la categoría *normal* ampliados. Debido a que el modelo muestra mucha seguridad y un alto elevado de aciertos en esta categoría la perdida ha bajado y el *accuracy* ha subido, obteniendo un 0,0345 de pérdida y un 97,02 % de *accuracy*.

Para ver el rendimiento mejor se facilita la matriz de confusión del modelo

## CAPÍTULO 8. RESULTADOS

“best\_model5.keras” en la figura 8.6. Podemos ver un rendimiento más realista pues ahora si hemos conseguido que falle en los casos del tipo *normal*, en concreto falla en nueve casos, categorizados erróneamente como *cancer*. Esto nos indica que la categoría *benign* es la más difícil de diferenciar de las otras, y la *normal* es la más fácil.

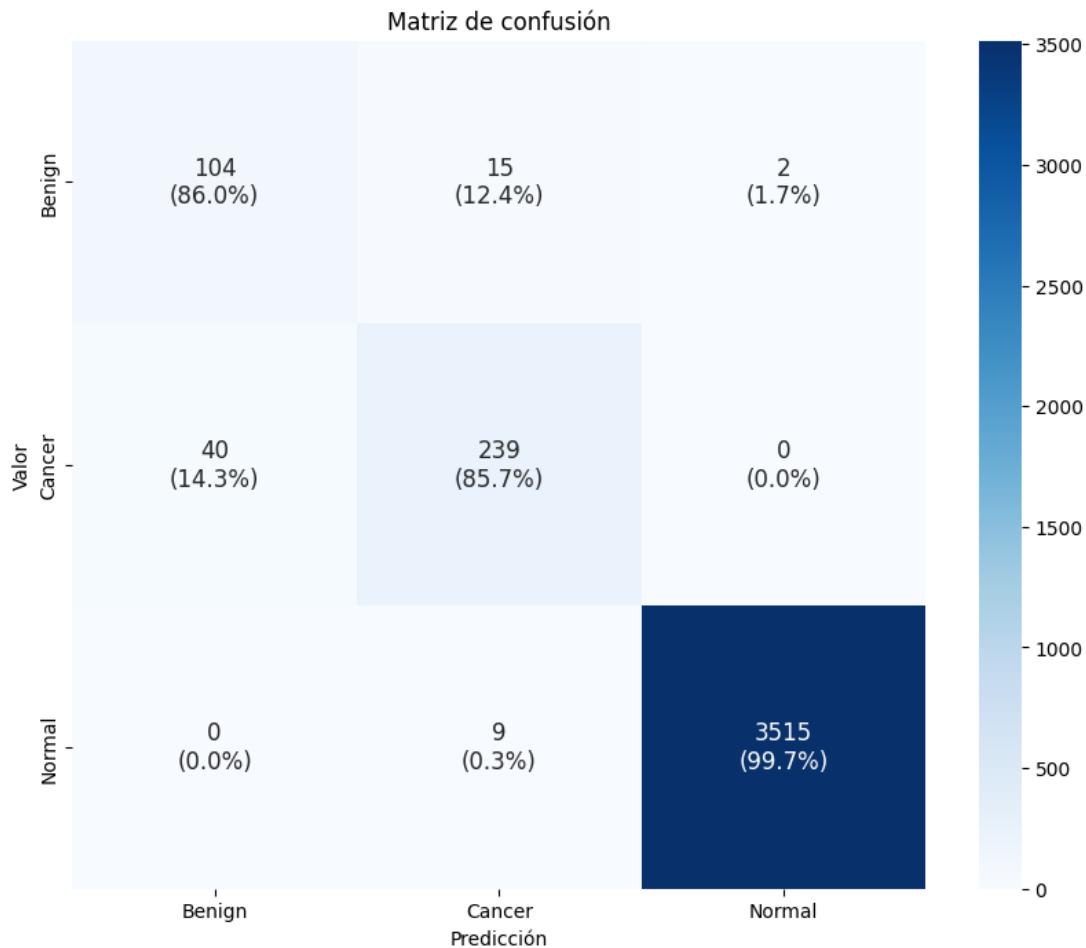


Figura 8.6: Matriz de confusión del modelo “best\_model5.keras” sin la categoría *actionable* y con datos ampliados para la clase *normal*.

### 8.1.2. Conjunto de datos con aumento de datos

Ya hemos visto como ha rendido esta red neuronal con los datos base. Ahora, vamos a ver como rinde con el conjunto de datos que contiene datos generados por *data augmentation*. Recordemos que este conjunto de datos tiene el doble de casos de

## 8.1. RED UNET3D-DENSE

---

la categoría *benign*.

Debido a que se manejan más datos, la duración del entrenamiento asciende a más de cinco horas y media. Nuevamente se han realizado 100 *epochs* repartidos en bloques de 20 en cinco *folds*.

Comencemos analizando las métricas obtenidas durante este proceso de entrenamiento. La figura 8.7 nos muestra la evolución de la pérdida. Podemos ver como la pérdida de entrenamiento, representada en azul, se ha ido reduciendo de manera progresiva y estable con el paso de los *epochs*, empezando en 0,15 hasta valores muy bajos, de entorno a 0,01. También se pueden notar de forma muy clara el cambio entre *folds* cada 20 *epochs*, lo que genera una serie de picos que rápidamente se estabilizan. Por parte de la pérdida de validación vemos como empieza en en cada *fold* a la par que la entrenamiento, pero aumenta distanciándose bastante. Este comportamiento puede indicar cierto grado de sobreajuste a los datos de entrenamiento, el cual, si existe, se hará más visible cuando se ejecuten las pruebas.

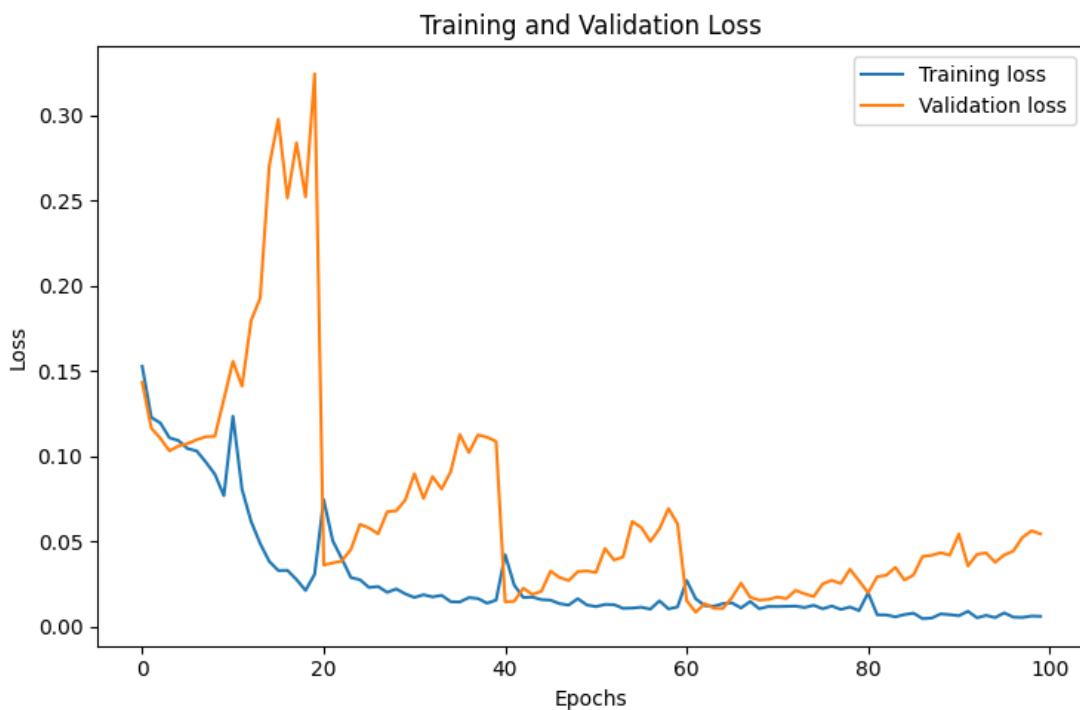


Figura 8.7: Pérdida durante el entrenamiento.

Por otro lado la figura 8.8 muestra el comportamiento del *accuracy*. El *accuracy* de

## CAPÍTULO 8. RESULTADOS

entrenamiento va aumentando progresivamente desde el 75 % hasta alcanzar el 94 %. El *accuracy* de validación sigue un comportamiento similar a la pérdida de validación, al principio del *fold* se encuentra cercano a el del entrenamiento pero se va despegando progresivamente. Este comportamiento vuelve a ser una señal de sobre ajuste con los datos utilizados en la validación cruzada.

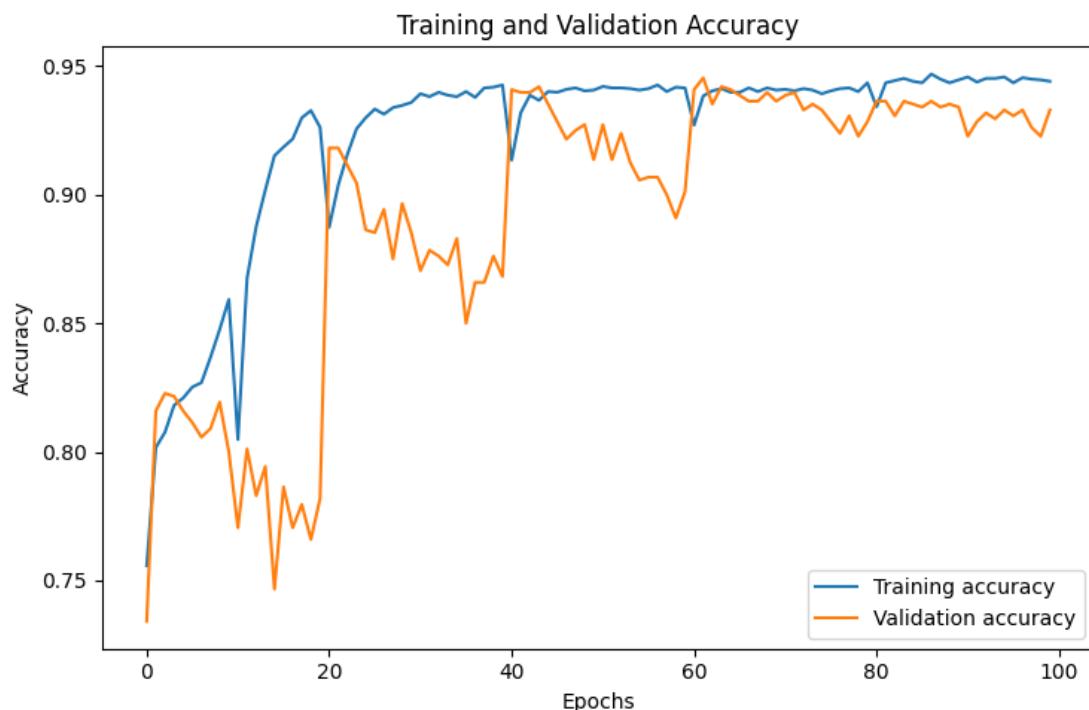


Figura 8.8: *Accuracy* durante el entrenamiento.

### Pruebas del modelo

Igual que en el conjunto de datos base se han guardado dos modelos por cada *fold*. Se han guardado por cada *fold* el modelo que mayor *accuracy* de validación ha conseguido y el del último *epoch*. A continuación, vamos a analizar los resultados obtenidos de la ejecución de las pruebas sobre cada uno de estos modelos.

En la figura 8.9 podemos ver la pérdida obtenida en las pruebas de forma gráfica. Analizándola vemos que la pérdida es mayor y más inestable que en el anterior conjunto de datos. Además, esta va aumentando conforme pasan los *epochs*, esto nos da señales de sobreajuste.

## 8.1. RED UNET3D-DENSE

---

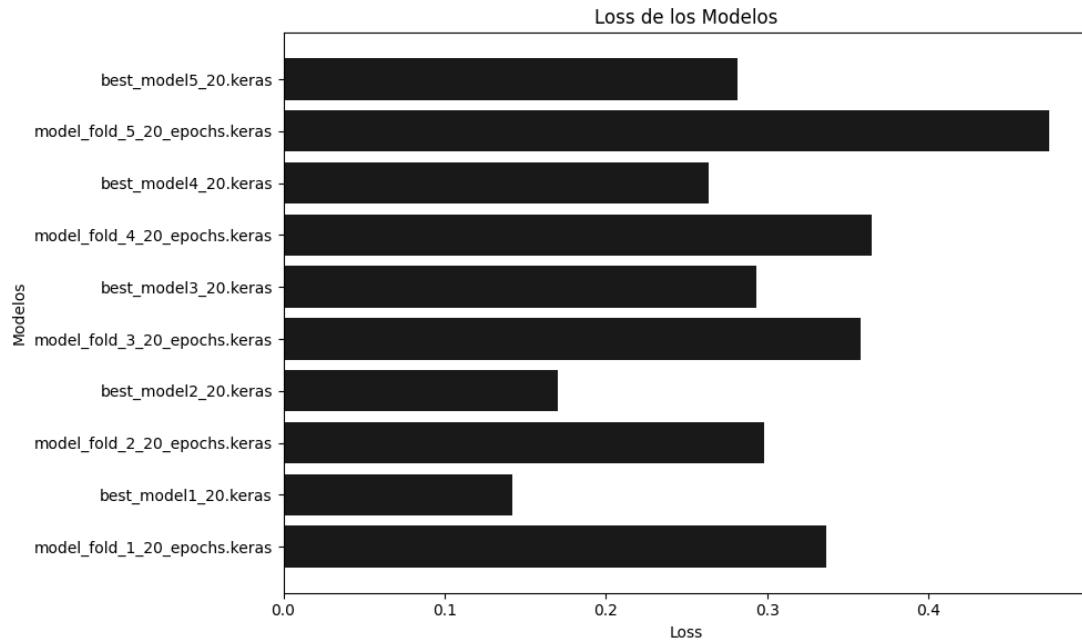


Figura 8.9: Pérdida de los diferentes modelos.

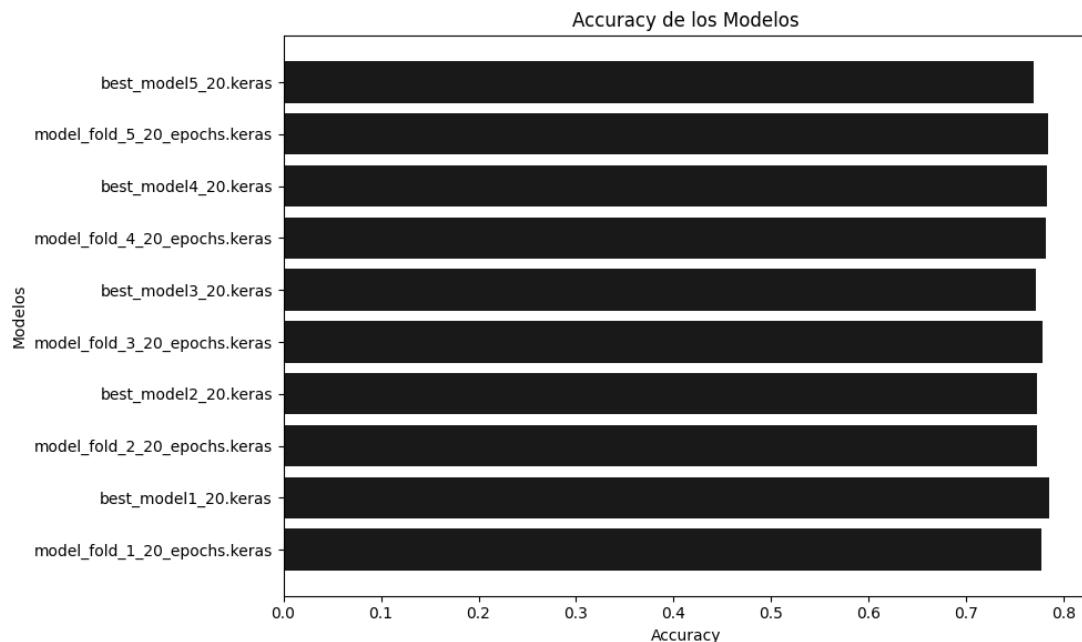


Figura 8.10: Accuracy de los diferentes modelos.

Por otro lado la figura 8.10 muestra un rendimiento de la métrica *accuracy* muy estable, sin picos ni apenas cambios. Esto nos indica que el sobreajuste detectado en la

## CAPÍTULO 8. RESULTADOS

---

pérdida no esta haciendo que el modelo empeore en su predicciones, pero tampoco le está permitiendo mejorar a lo largo del entrenamiento.

Por último la tabla 8.4 nos muestra los mismos datos pero de forma numérica. Podemos ver que la red consigue un rendimiento medio de 0,2984 de pérdida y del 77,73 % de *accuracy*.

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0.3364	0.7771
best_model1_20.keras	0.1419	0.7853
model_fold_2_20_epochs.keras	0.2978	0.7725
best_model2_20.keras	0.1699	0.7725
model_fold_3_20_epochs.keras	0.3580	0.7780
best_model3_20.keras	0.2936	0.7707
model_fold_4_20_epochs.keras	0.3650	0.7816
best_model4_20.keras	0.2639	0.7825
model_fold_5_20_epochs.keras	0.4753	0.7834
best_model5_20.keras	0.2819	0.7689
<b>Media aritmética</b>	<b>0.2984</b>	<b>0.7773</b>

Tabla 8.4: Resultados de los modelos DA evaluados en el conjunto de datos de prueba.

Para ver mas detenidamente el funcionamiento del modelo vamos a analizar la matriz de confusión del modelo “best\_model1\_20.keras” representado en la figura 8.11.

Primero que todo, podemos ver nuevamente que el modelo clasifica todos los casos de la categoría *actionable* como *normal*. Esto era algo de esperar y nos refuerza la teoria de que esta clase es más parecida a casos de pacientes sanos que a los casos *benign* o *cancer*, aunque solo sea en las mamografías.

Además, en la categoría *normal*, aunque da un rendimiento peor que en el anterior conjunto de datos, consigue unos muy buenos resultados con un 99,6 %. Solo falla en dos casos que los categoriza como *benign* o *cancer*.

Por otro lado vemos un empeoramiento del rendimiento más severo en los casos *benign* y *cancer*, dando un rendimiento del 66,9 % y 64,9 % respectivamente. Con estos resultados si podemos concluir que no es capaz de diferenciar correctamente estas dos clases. Dado que este conjunto de datos ha aumentado el número de casos *benign*, este resultado es muy contraproducente.

## 8.1. RED UNET3D-DENSE

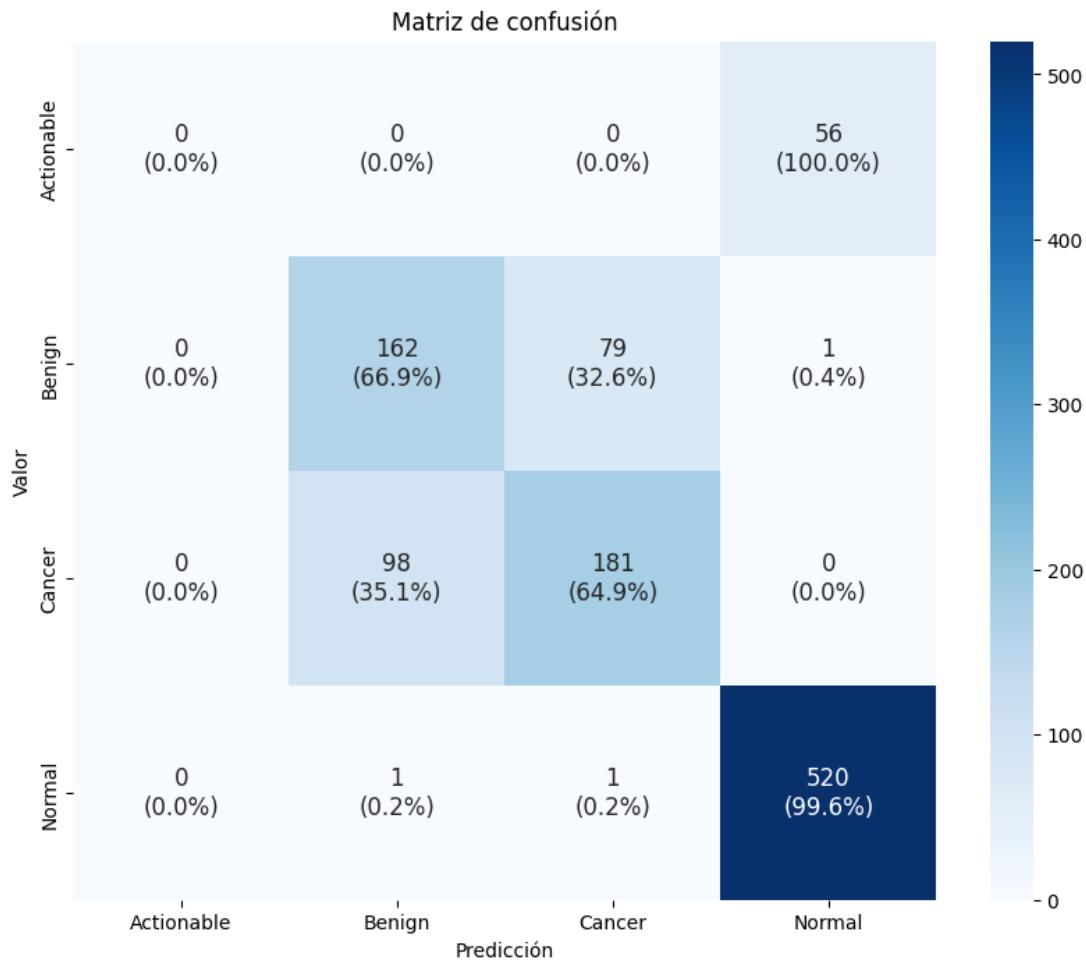


Figura 8.11: Matriz de confusión del modelo “best\_model1\_20.keras”.

### Pruebas sin la clase *Actionable*

A pesar de los resultados insatisfactorios obtenidos en las pruebas iniciales, se ha decidido realizar pruebas adicionales excluyendo los casos correspondientes a la categoría *Actionable*. El objetivo de esta prueba es comparar los resultados obtenidos sin esta clase con aquellos obtenidos por el resto de los modelos.

Los resultados de esta prueba se presentan en la tabla 8.5. Como se puede observar, los valores de pérdida y *accuracy* han aumentado, alcanzando 0,3144 y 81,90%, respectivamente. La mejora en el *accuracy* se debe a que ahora no está fallando en todos los casos *actionable*, y la subida en la pérdida muestra que la clasificación de los casos *actionable* mostraban un grado de certeza superior a la media.

## CAPÍTULO 8. RESULTADOS

---

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0.3544	0.8188
best_model1_20.keras	0.1495	0.8274
model_fold_2_20_epochs.keras	0.3138	0.8140
best_model2_20.keras	0.1790	0.8140
model_fold_3_20_epochs.keras	0.3772	0.8198
best_model3_20.keras	0.3094	0.8121
model_fold_4_20_epochs.keras	0.3846	0.8236
best_model4_20.keras	0.2781	0.8245
model_fold_5_20_epochs.keras	0.5008	0.8255
best_model5_20.keras	0.2970	0.8102
<b>Media aritmética</b>	<b>0.3144</b>	<b>0.8190</b>

Tabla 8.5: Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase *actionable*.

### Pruebas con casos normales ampliados

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,1055	0,9458
best_model1_20.keras	0,0460	0,9475
model_fold_2_20_epochs.keras	0,0936	0,9444
best_model2_20.keras	0,0546	0,9424
model_fold_3_20_epochs.keras	0,1136	0,9441
best_model3_20.keras	0,0934	0,9418
model_fold_4_20_epochs.keras	0,1142	0,9469
best_model4_20.keras	0,0837	0,9455
model_fold_5_20_epochs.keras	0,1486	0,9480
best_model5_20.keras	0,0920	0,9362
<b>Media aritmética</b>	<b>0,0945</b>	<b>0,9443</b>

Tabla 8.6: Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase *actionable* y con datos ampliados para la clase *normal*.

También, vamos a correr las pruebas añadiendo todos los casos normales con la finalidad de comparar esta red neuronal con el resto. En la tabla 8.6 podemos ver el rendimiento de cada modelo guardado. Con estos datos alcanza una pérdida media de 0,0945 y un *accuracy* medio del 94,43 %.

Vamos a ver también como se comporta para cada una de las categorías con los casos normales ampliados, para ello hacemos uso de la matriz de confusión

## 8.1. RED UNET3D-DENSE

representada en la figura 8.12. Como vemos, ahora comete nueve errores en total con los casos de la clase *normal*, siete los clasifica como *benign* y dos como *cancer*. Por tanto, obtiene una precisión del 99,7% con esta clase. El resto de clases se mantienen igual, pues son los mismos datos.

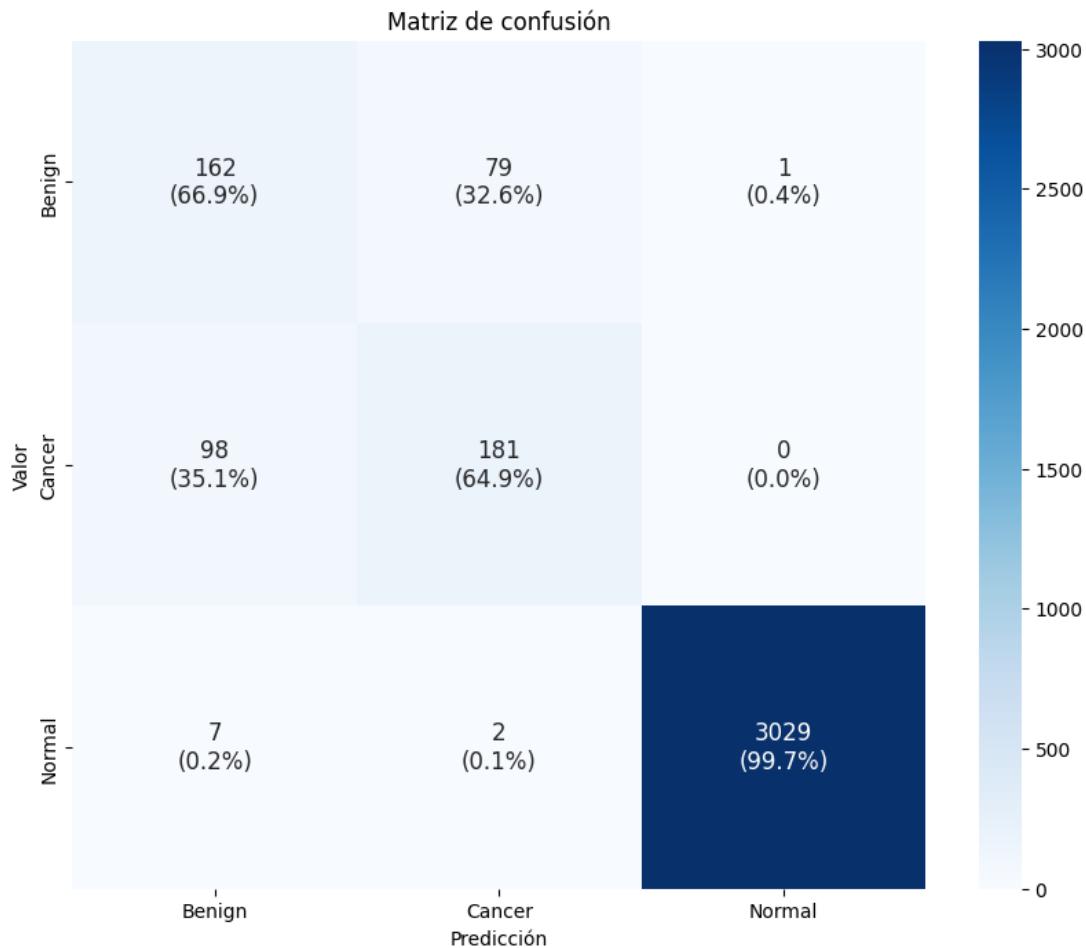


Figura 8.12: Matriz de confusión del modelo “model\_fold\_2\_20\_epochs.keras” sin la categoría *actionable* y con datos ampliados para la clase *normal*.

Con ya hemos acabado de analizar los resultados obtenidos con la red neuronal UNet3D-Dense. Esta red nos ha dado un gran rendimiento con el conjunto de datos base, en contraposición con los mediocres resultados obtenidos del conjunto de datos ampliados. Estos resultados se compararán con los obtenidos en las otras redes neuronales en el capítulo 9, [conclusiones](#).

## 8.2. Red UNet-Paralela

Una vez visto los resultados de la arquitectura UNet3D-Dense, vamos a ver los resultados en la red neuronal UNet-Paralela. Durante su entrenamiento se hará uso de los mismos datos utilizados en la anterior red, los cuales están adaptados a una resolución  $304 \times 304$  y han sido normalizados, con valores entre cero y uno, en formato *float32*.

Primero, veremos los resultados del entrenamiento de la red UNet-Paralela con el conjunto de datos base y luego con el conjunto aumentado.

### 8.2.1. Conjunto de datos base

El entrenamiento de este conjunto de datos llevó alrededor de 2 horas y 15 minutos. Durante este tiempo, se ejecutaron 20 *epochs* para cada uno de los 5 *folds* de validación cruzada, acumulando un total de 100 *epochs*. A continuación, vamos a ver el resultado que nos dan las métricas utilizadas durante este proceso.

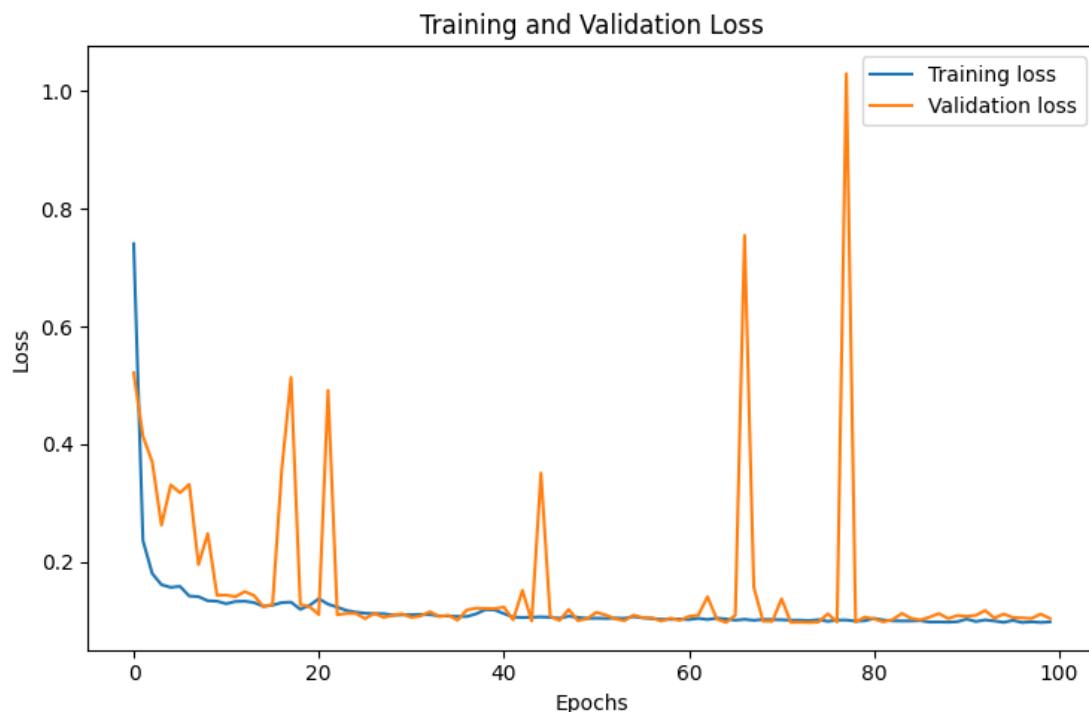


Figura 8.13: Pérdida durante el entrenamiento

## 8.2. RED UNET-PARALELA

En la figura 8.13 podemos observar la evolución de la pérdida durante el entrenamiento. Tanto la pérdida de entrenamiento como la de validación se mantienen cercanas entre sí, viendo una alta disminución en los primeros *epochs* hasta estabilizarse en torno a 0,1 en ambas pérdidas. Esto nos muestra un rendimiento favorable y sin muestras de sobreajuste. También cabe resaltar los picos en la perdida de validación, los cuales, a pesar de ser escasos, son muy elevados. Si fuesen más frecuentes mostrarían un entrenamiento menos consistente, pero debido a su número no son demasiado relevantes.

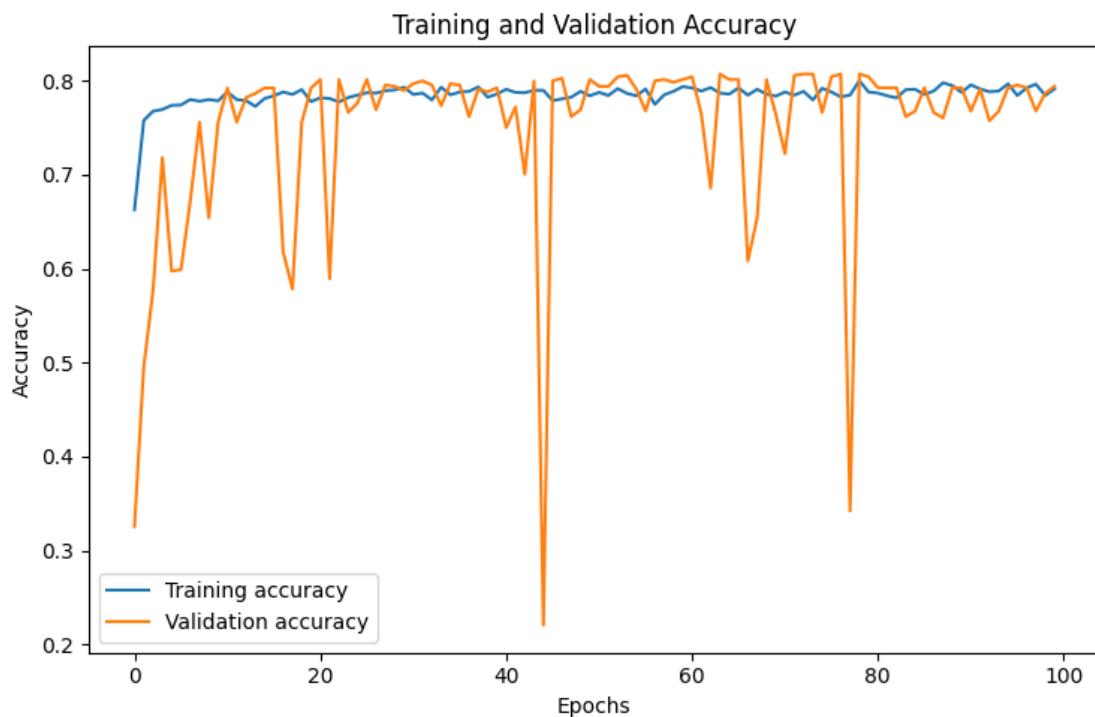


Figura 8.14: *Accuracy* durante el entrenamiento

Por otro lado, la figura 8.14 muestra un gráfico con la evolución del *accuracy* tanto del conjunto de entrenamiento como del conjunto de validación. Vemos que ambos alcanzan valores un poco inferiores a el 80% en los primeros *epochs*. Después de esto, se mantiene estable en esos valores, con una ligera subida después de el *epoch* 20, probablemente producida por el cambio de *fold* y el uso de nuevos datos para el entrenamiento. También, se pueden apreciar los diferentes picos que encontrábamos en el gráfico de la pérdida relativos al conjunto de validación.

## CAPÍTULO 8. RESULTADOS

### Pruebas del modelo

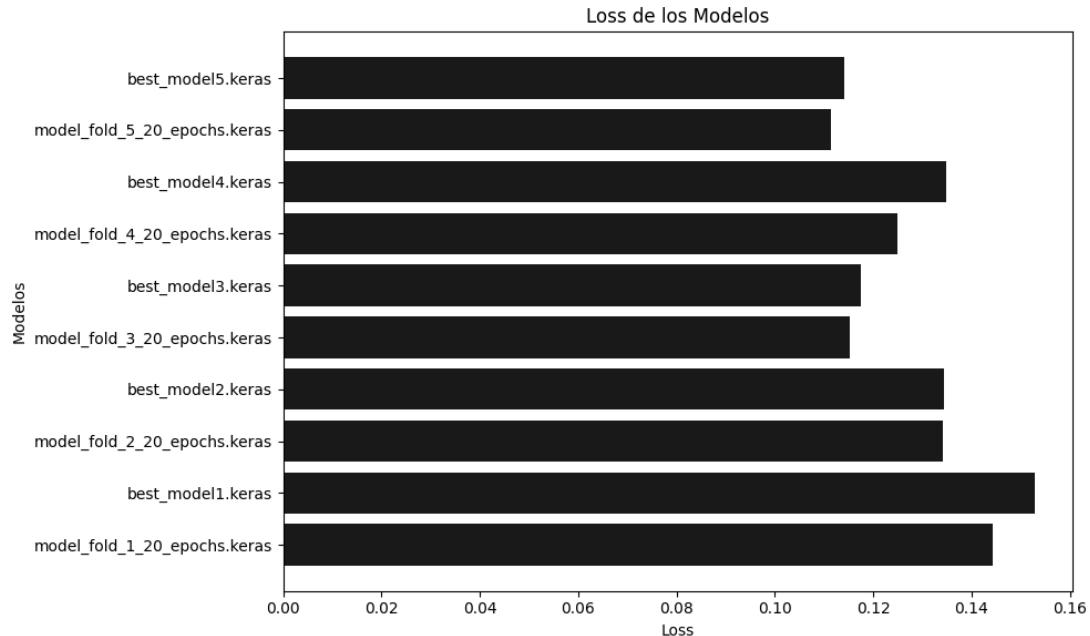


Figura 8.15: Pérdida de los diferentes modelos.

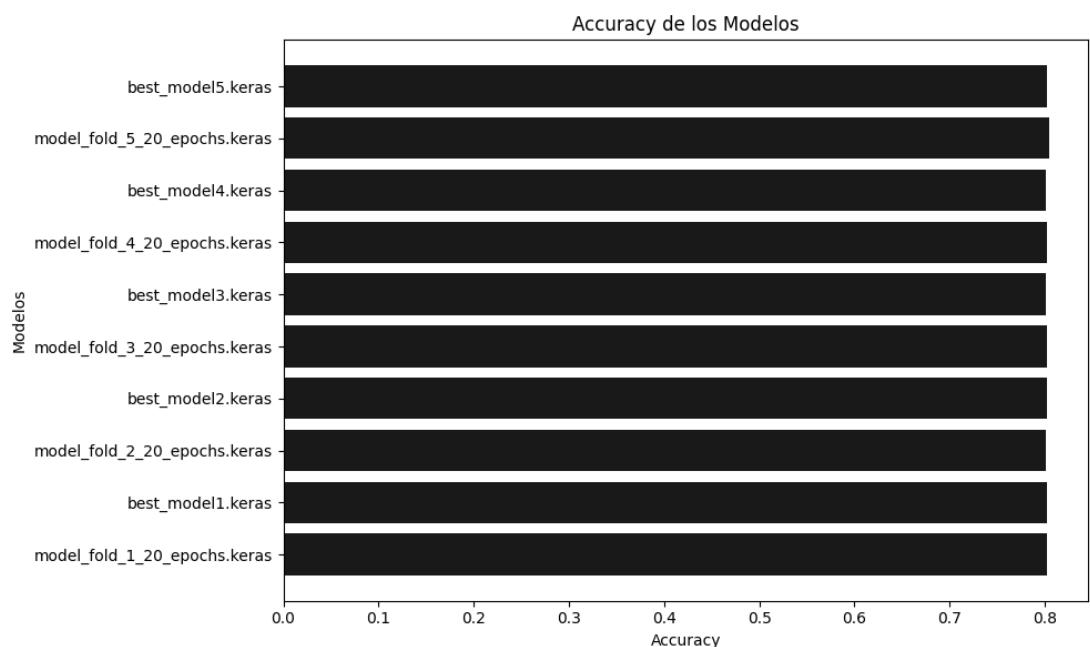


Figura 8.16: Accuracy de los diferentes modelos.

## 8.2. RED UNET-PARALELA

---

Para ver el rendimiento del modelo, vamos a probar los modelos que se han guardado durante el proceso de validación cruzada con el conjunto de datos de prueba. En la validación cruzada el rendimiento se obtiene mediante la media aritmética de las métricas de cada *fold*. Durante el entrenamiento se han guardado dos modelos por cada *fold*, uno que representa el final del *fold* y otro que representa el modelo con mayor *accuracy* dentro del *fold*. Estos serán los modelos que se probarán.

En primer lugar, la figura 8.15 nos muestra la pérdida obtenida por cada uno de los modelos guardados. Como se puede apreciar esta va disminuyendo de forma progresiva pasando de entorno al 0,15 en el primer *fold* a 0,11 en el último. Esta pérdida sigue siendo algo elevada, pero muestra una buena evolución.

En segundo lugar, la figura 8.16 muestra el *accuracy* obtenido. En este caso el *accuracy* se muestra muy estable, sin ninguna mejora fundamental en el transcurso de los *folds*. Esto indica que el modelo ya alcanzó su máximo *accuracy* al completar el primer *fold* y no ha sido capaz de generalizar más.

Además, en la tabla 8.7 se nos muestran los valores numéricos de los gráficos anteriores. Además aparece calculado el rendimiento del modelo, con una pérdida media de 0,1284 y un *accuracy* medio del 80,27%.

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,1443	0,8028
best_model1.keras	0,1528	0,8028
model_fold_2_20_epochs.keras	0,1342	0,8016
best_model2.keras	0,1345	0,8028
model_fold_3_20_epochs.keras	0,1152	0,8028
best_model3.keras	0,1176	0,8016
model_fold_4_20_epochs.keras	0,1248	0,8028
best_model4.keras	0,1349	0,8016
model_fold_5_20_epochs.keras	0,1115	0,8051
best_model5.keras	0,1140	0,8028
<b>Media aritmética</b>	<b>0,1284</b>	<b>0,8027</b>

Tabla 8.7: Resultados de los modelos evaluados en el conjunto de datos de prueba.

Para analizar más detalladamente el rendimiento, podemos probar uno de los modelos, en concreto el modelo “model\_fold\_5\_20\_epochs.keras”, y comprobar las

## CAPÍTULO 8. RESULTADOS

predicciones en cada una de las categorías de datos. Esto lo podemos ver en la matriz de confusión representada en la figura 8.17.

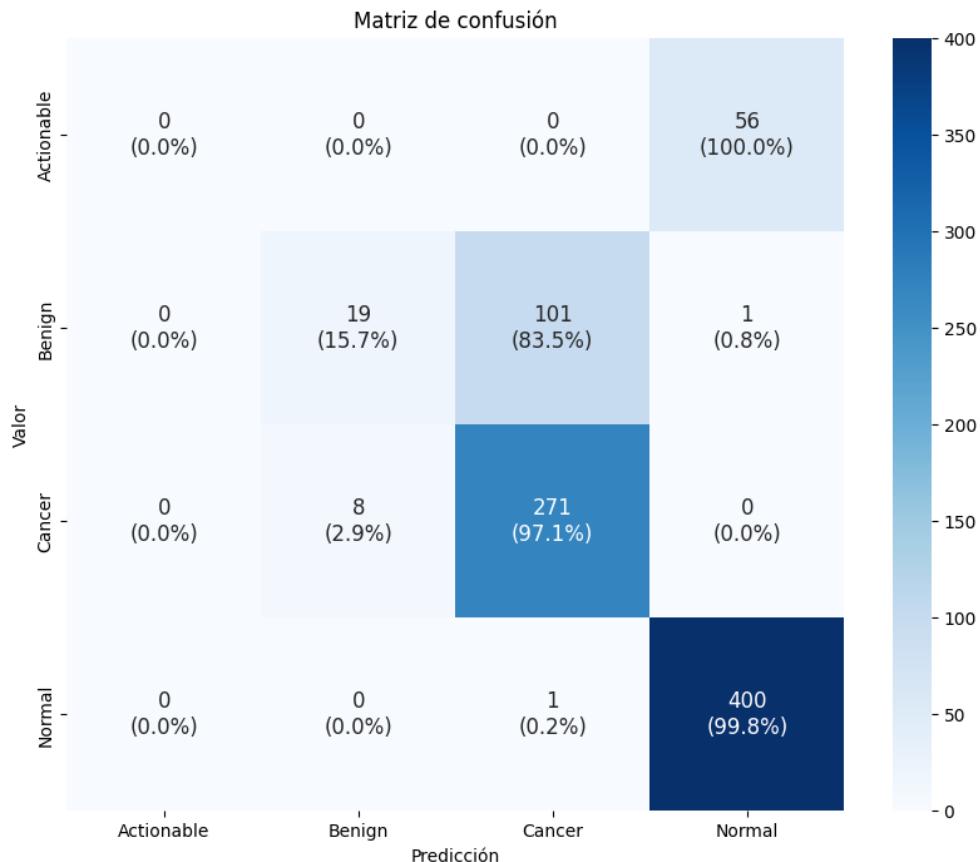


Figura 8.17: Matriz de confusión del modelo “model\_fold\_5\_20\_epochs.keras”.

Lo primero que nos muestra esta matriz de confusión, es que los casos de la clase *actionable* han sido clasificados como casos de la clase *normal*. Como ya se ha explicado esto es debido a que esta clase se ha utilizado en el entrenamiento para ver como el modelo se comporta con la misma, pero el modelo no aprende de esta porque tiene un peso cero en la función de perdida del entrenamiento de forma deliberada.

Con respecto a las otras clases, el modelo logra diferenciar los casos sanos, pertenecientes a la clase *normal*, de aquellos que tienen un tumor o un cáncer, pertenecientes a las clases *benign* y *cancer* respectivamente. En este aspecto solo ha cometido dos errores de 801 casos, un caso *normal* erróneamente clasificado como un caso de *cancer* y un caso *benign* clasificado como *normal*. Esto muestra una precisión

## 8.2. RED UNET-PARALELA

---

entorno al 99,8 % en esta tarea.

Además, con respecto a las clases *benign* y *cancer*, podemos ver que el modelo sobre representa a la clase *cancer*, teniendo una buena precisión con los casos de esta categoría, pero una pésima con los casos de la categoría *benign*. Esto hace que el modelo no sea apto para la diferenciación entre estos dos tipos de casos. Más adelante, veremos si este problema se soluciona con el conjunto de datos que utiliza el aumento de datos, añadiendo más casos a la categoría *benign*.

### Pruebas sin la clase *actionable*

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,1536	0,8589
best_model1.keras	0,1623	0,8589
model_fold_2_20_epochs.keras	0,1431	0,8577
best_model2.keras	0,1432	0,8589
model_fold_3_20_epochs.keras	0,1232	0,8589
best_model3.keras	0,1257	0,8577
model_fold_4_20_epochs.keras	0,1334	0,8589
best_model4.keras	0,1443	0,8577
model_fold_5_20_epochs.keras	0,1192	0,8614
best_model5.keras	0,1218	0,8589
<b>Media aritmética</b>	<b>0,1370</b>	<b>0,8588</b>

Tabla 8.8: Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase *actionable*.

Los rendimientos obtenidos son útiles para hacernos una idea de que está sucediendo con las predicciones, pero la clase *actionable* nos impide sacar su rendimiento real. Para ello vamos a correr las pruebas de los mismos modelos guardados sin esta categoría. El resultado se puede ver en la tabla 8.8, en la cual podemos ver que se ha obtenido una pérdida y un *accuracy* más altos, con unos valores de 0,1370 y 85,88 % respectivamente.

### Pruebas con casos normales ampliados

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,0423	0,9674
best_model1.keras	0,0645	0,9674
model_fold_2_20_epochs.keras	0,0554	0,9620
best_model2.keras	0,0445	0,9651
model_fold_3_20_epochs.keras	0,0299	0,9676
best_model3.keras	0,0303	0,9684
model_fold_4_20_epochs.keras	0,0306	0,9692
best_model4.keras	0,0310	0,9699
model_fold_5_20_epochs.keras	0,0280	0,9697
best_model5.keras	0,0285	0,9694
<b>Media aritmética</b>	<b>0,0385</b>	<b>0,9676</b>

Tabla 8.9: Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase *actionable* y con datos ampliados para la clase *normal*.

La mayor parte de los pacientes que se someten a exámenes de detección son pacientes sanos, por ello, para ver un rendimiento que coincida más con la realidad, se han añadido todos los casos de la categoría *normal* que se desecharon y no se utilizaron durante el entrenamiento. De esta forma se conseguirá una precisión más precisa manteniendo la independencia entre los datos usados en el entrenamiento y las pruebas.

En la tabla 8.9 podemos ver los resultados de estas últimas pruebas, en las cuales se ha conseguido un rendimiento bastante bueno con una pérdida media de 0,0385 y un *accuracy* del 96,76%.

Además, vamos a analizar estos resultados con la matriz de confusión representada en la figura 8.18 en la que se muestran los resultados con el modelo “model\_fold\_5\_20\_epochs.keras”. En esta matriz vemos que ha fallado nueve de los 3.524 casos de la clase *normal*, 1 clasificado erróneamente como *benign* y los otros ocho como *cancer*, constatando la preponderancia que tiene el modelo a clasificar los casos de *cancer* frente a los *benign*.

Añadir que, a pesar de los buenos números en las métricas de este modelo, tal y como se puede ver en la matriz de confusión, no es capaz de diferenciar los casos

## 8.2. RED UNET-PARALELA

*benign* de los casos *cancer* de forma satisfactoria, haciendo que este modelo no sea apto para esta tarea. Vamos a ver a continuación si esto cambia gracias a la técnica de aumento de datos.

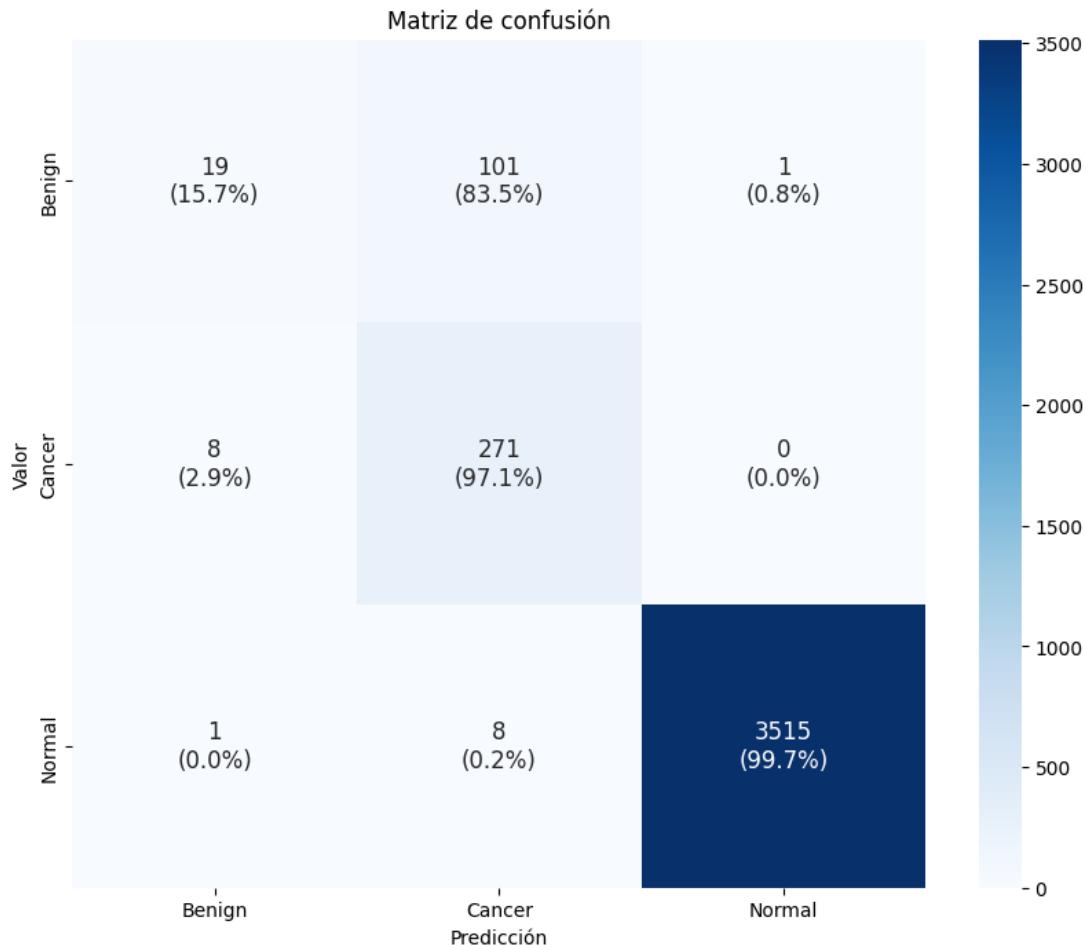


Figura 8.18: Matriz de confusión del modelo “model\_fold\_5\_20\_epochs.keras” sin la categoría *actionable* y con datos ampliados para la clase *normal*.

### 8.2.2. Conjunto de datos con aumento de datos

En esta sección se presentan los resultados obtenidos al entrenar el modelo utilizando el conjunto de datos que incorpora la técnica de *data augmentation* (DA). Es importante recordar que, en este conjunto, se han duplicado las instancias de la categoría *benign* con el fin de mejorar la precisión en esta clase.

El proceso de entrenamiento de este conjunto de datos tuvo una duración

## CAPÍTULO 8. RESULTADOS

aproximada de 2 horas y media. Durante este tiempo, al igual que en el entrenamiento con el conjunto de datos previo, se ejecutaron 20 *epochs* para cada uno de los 5 *folds* usados en el proceso de validación cruzada, acumulando un total de 100 *epochs*. El mayor tiempo requerido para este entrenamiento se debe al incremento en la cantidad de datos correspondientes a la clase *benign*.

A continuación, se presentarán las métricas obtenidas a lo largo de este proceso. La figura 8.19 muestra la evolución de la pérdida durante el entrenamiento, tanto para los datos de entrenamiento como para los de validación.



Figura 8.19: Pérdida durante el entrenamiento.

De manera similar a lo observado en el conjunto de datos anterior, la pérdida en el entrenamiento es elevada al comienzo, pero disminuye rápidamente en las primeras épocas, estabilizándose en valores muy bajos. Esto indica que el modelo está aprendiendo a reducir los errores de predicción. La pérdida en la validación sigue una tendencia parecida a la de entrenamiento, lo que, en un principio, sugiere que el modelo no está sobreajustando.

La figura 8.20 representa el *accuracy* tanto del conjunto de datos de entrenamiento

## 8.2. RED UNET-PARALELA

como de validación. Como podemos observar el *accuracy* de validación es más inestable que el de entrenamiento, con picos y caídas elevadas. Aún así, ambos siguen una tendencia similar, en la que según van pasando los *folds* van aumentando de valor. En general, muestra un tendencia estable, situándose en valores circundantes al 80% y sin signos de sobreajuste observables.

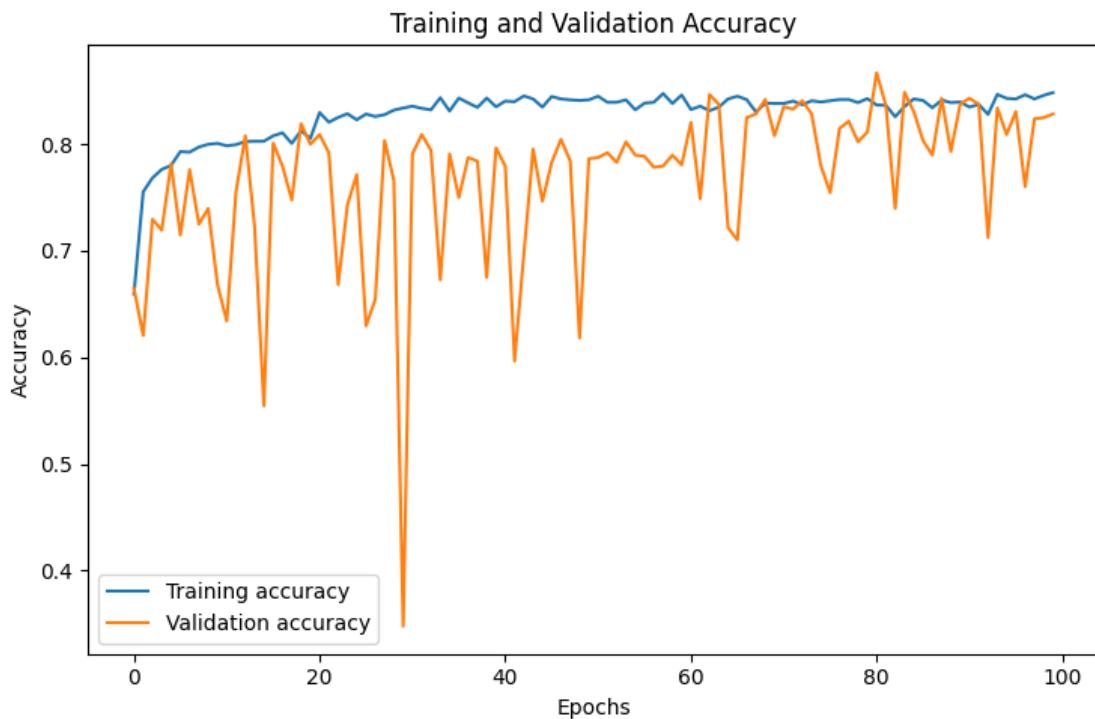


Figura 8.20: *Accuracy* durante el entrenamiento.

En este caso, el modelo alcanza una mayor precisión y exactitud debido a dos factores principales. En primer lugar, el incremento en la cantidad de datos dentro de la categoría *benign* contribuye a mejorar el rendimiento en la clasificación de esta clase. En segundo lugar, el aumento de datos en la categoría *benign* reduce la representatividad de la categoría *actionable* en el conjunto total, lo que permite al modelo concentrarse más eficazmente en las clasificaciones de *benign* y *cancer*, optimizando su rendimiento global.

## CAPÍTULO 8. RESULTADOS

---

### Pruebas del modelo

Al igual que antes, durante el entrenamiento se han guardado dos versiones del modelo por cada *fold*. Uno de estos modelos corresponde al punto en que se alcanzó la mayor exactitud en los datos de validación, mientras que el otro se guarda al finalizar las 20 *epochs*. Esta metodología permite realizar una comparación entre el rendimiento óptimo del modelo y su rendimiento al final del proceso de entrenamiento en cada *fold*.

A continuación, se analiza el comportamiento de los modelos guardados al ser evaluados con el conjunto de datos de prueba. La figura 8.21 ilustra la evolución de la pérdida en los diferentes modelos guardados, mientras que la figura 8.22 muestra el *accuracy* alcanzado por estos modelos. Los valores numéricos correspondientes se encuentran en la tabla 8.10.

Podemos observar unos valores de pérdida que van mejorando de forma constante según pasan los distintos *folds*. Paralelo a esto, el *accuracy* se mantiene constante. Esto nos indica que el modelo está cogiendo más confianza en sus predicciones, pero, no están mejorando a lo largo del entrenamiento.

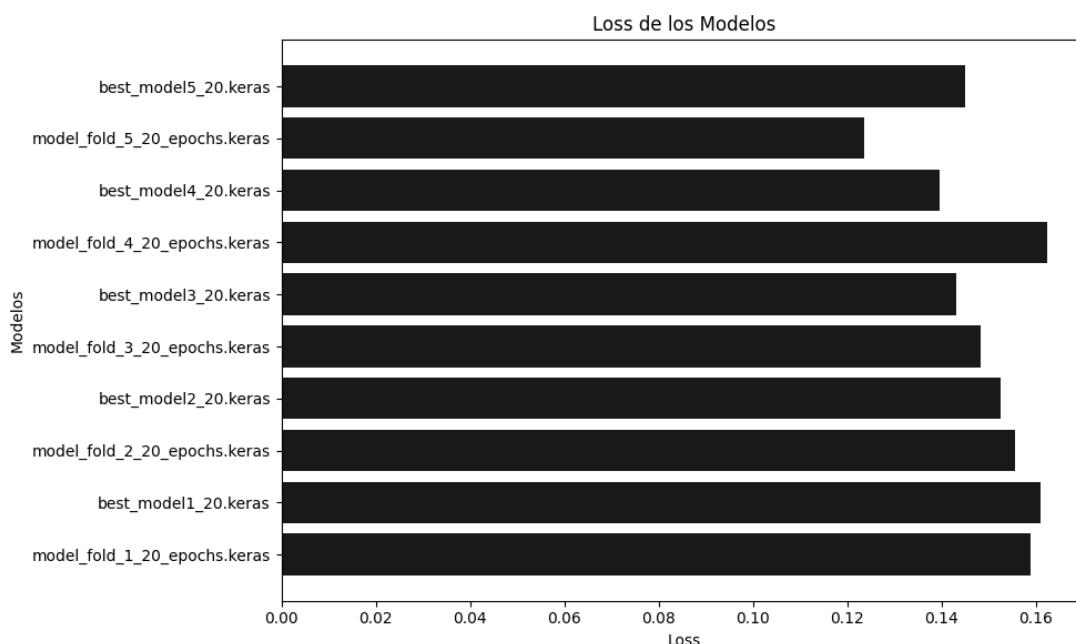


Figura 8.21: Pérdida de los diferentes modelos.

## 8.2. RED UNET-PARALELA

---

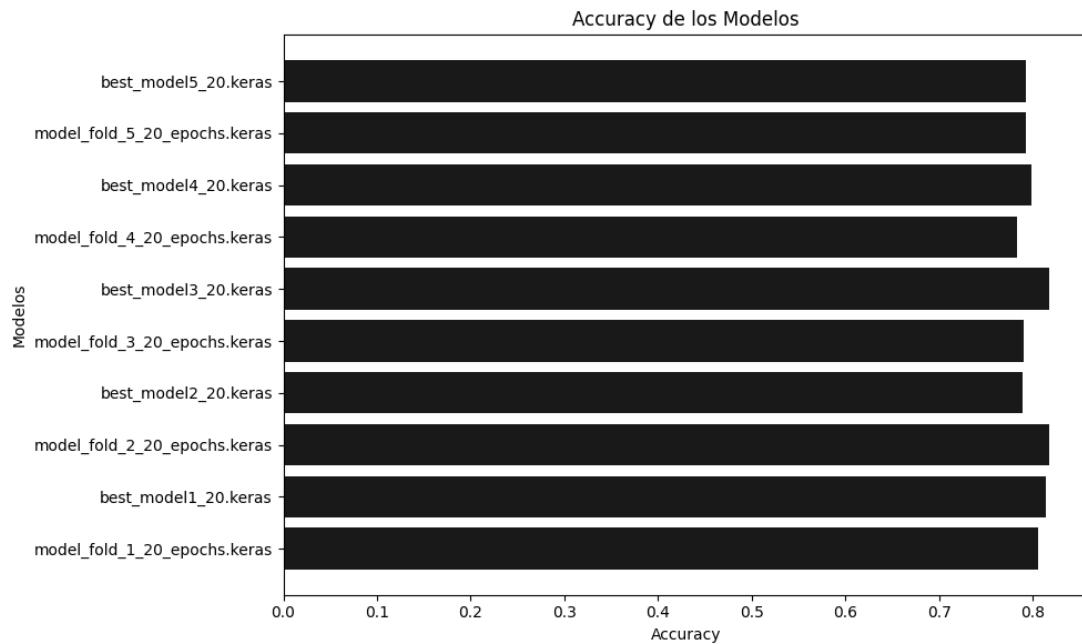


Figura 8.22: *Accuracy* de los diferentes modelos.

Para medir el rendimiento del modelo se debe de realizar la media aritmética, la cual, según estos resultados, nos da un 80,02 % de *accuracy* y 0,1490 de pérdida. Para poder ver mejor el rendimiento de estos resultados vamos a analizar la matriz de confusión representada en la figura 8.23. La cual, se ha realizado con el modelo “model\_fold\_2\_20\_epochs.keras”, que es el con el que mejor rendimiento se ha obtenido.

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,1589	0,8062
best_model1_20.keras	0,1610	0,8144
model_fold_2_20_epochs.keras	0,1555	0,8180
best_model2_20.keras	0,1525	0,7889
model_fold_3_20_epochs.keras	0,1484	0,7898
best_model3_20.keras	0,1432	0,8171
model_fold_4_20_epochs.keras	0,1625	0,7834
best_model4_20.keras	0,1395	0,7989
model_fold_5_20_epochs.keras	0,1236	0,7925
best_model5_20.keras	0,1450	0,7925
<b>Media aritmética</b>	<b>0,1490</b>	<b>0,8002</b>

Tabla 8.10: Resultados de los modelos DA evaluados en el conjunto de datos de prueba.

## CAPÍTULO 8. RESULTADOS

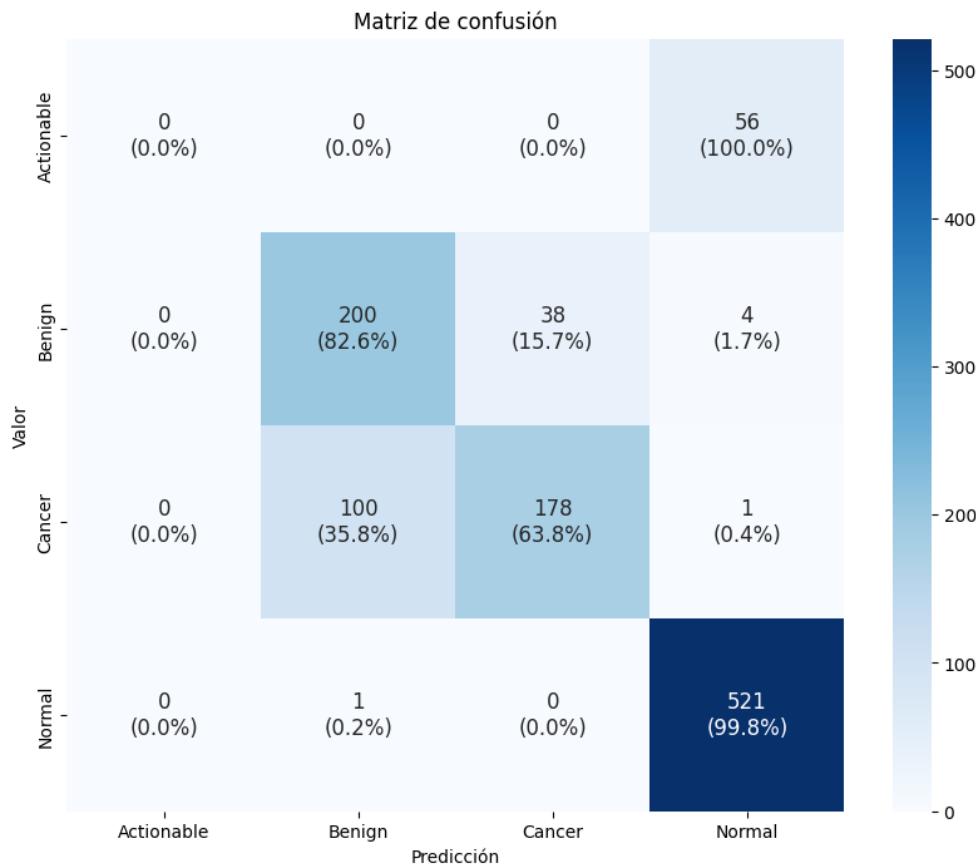


Figura 8.23: Matriz de confusión del modelo “model\_fold\_2\_20\_epochs.keras”.

En primer lugar, los casos de la clase *actionable* son categorizados como *normal*. Esto lo vemos en todos los modelos, mostrando evidencia de que las mamografías esta clase son más parecidas a la categoría *normal* que al resto.

En segundo lugar, se obtiene un resultado sobresaliente al diferenciar a los pacientes sanos del resto, falla 6 de los 1.043 casos de las categorías *normal*, *benign* y *cancer* para esta tarea, obteniendo una precisión del 99,6% en los casos de la clase *normal*.

Por otro lado, el modelo presenta dificultades a la hora de discernir entre los casos de las clases *benign* y *cancer*, aunque es mejor que los resultados obtenidos con el anterior conjunto de datos. El modelo consigue una precisión del 82,6% con la clase *benign* y un 63,8% con la clase *cancer*, cometiendo un menor número de errores con la clase *benign*.

## 8.2. RED UNET-PARALELA

---

### Pruebas sin la clase *actionable*

Los resultados anteriores incluyen la clase *actionable*, la cual no se utilizará para medir el rendimiento del modelo. Es por esto que en la tabla 8.11 podemos ver el rendimiento sin tener en cuenta los casos *actionables*. En esta tabla, podemos observar como el *accuracy* medio ha aumentado hasta el 84,32 % y la pérdida media hasta el 0,1568.

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,1672	0,8495
best_model1_20.keras	0,1695	0,8581
model_fold_2_20_epochs.keras	0,1637	0,8619
best_model2_20.keras	0,1605	0,8313
model_fold_3_20_epochs.keras	0,1561	0,8322
best_model3_20.keras	0,1507	0,8610
model_fold_4_20_epochs.keras	0,1710	0,8255
best_model4_20.keras	0,1469	0,8418
model_fold_5_20_epochs.keras	0,1300	0,8351
best_model5_20.keras	0,1568	0,8432
<b>Media aritmética</b>	<b>0,1568</b>	<b>0,8432</b>

Tabla 8.11: Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase *actionable*.

### Pruebas con casos normales ampliados

Modelo	Pérdida	Accuracy
model_fold_1_20_epochs.keras	0,1081	0,9351
best_model1_20.keras	0,0931	0,9548
model_fold_2_20_epochs.keras	0,0541	0,9562
best_model2_20.keras	0,0598	0,9416
model_fold_3_20_epochs.keras	0,0672	0,9281
best_model3_20.keras	0,0522	0,9550
model_fold_4_20_epochs.keras	0,0557	0,9449
best_model4_20.keras	0,0521	0,9466
model_fold_5_20_epochs.keras	0,0449	0,9480
best_model5_20.keras	0,0536	0,9461
<b>Media aritmética</b>	<b>0,0641</b>	<b>0,9456</b>

Tabla 8.12: Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase *actionable* y con datos ampliados para la clase *normal*.

## CAPÍTULO 8. RESULTADOS

Para medir el rendimiento final de esta red neuronal UNet-Paralela, se va a realizar una prueba más, incluyendo todos los casos de la categoría *normal* que no han sido utilizados en durante el entrenamiento y que fueron desechados para conseguir un mejor balance durante el entrenamiento. De esta manera, conseguiremos una proporción más parecida a la presentada en los exámenes de detección reales. Los resultados obtenidos de esta prueba se muestran en la tabla 8.12, donde se consigue una *accuracy* medio del 94,56 % y una pérdida media de 0,0641.

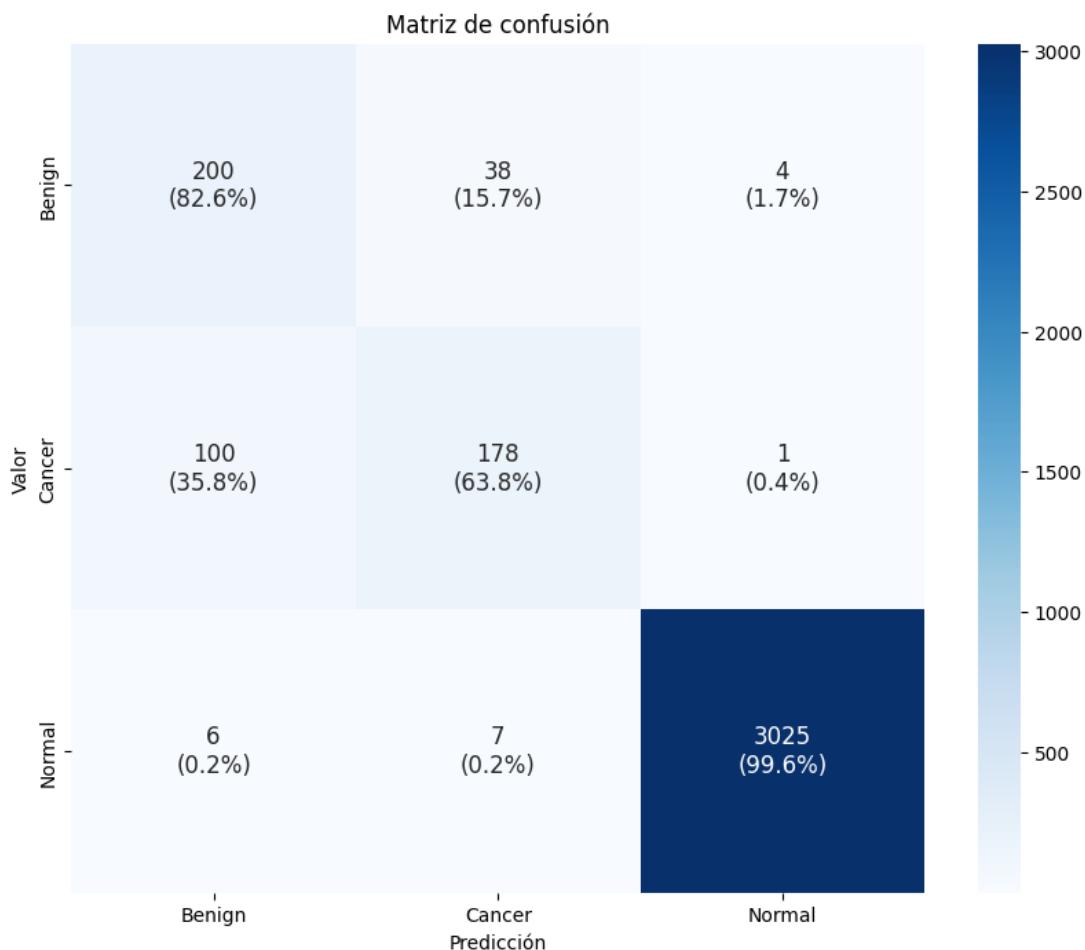


Figura 8.24: Matriz de confusión del modelo “model\_fold\_2\_20\_epochs.keras” sin la categoría *actionable* y con datos ampliados para la clase *normal*.

Para ver estos resultados más en detalle, vamos a utilizar la matriz de confusión representada en la figura 8.24, en la cual se ha vuelto a utilizar el modelo “model\_fold\_2\_20\_epochs.keras”. Como vemos, la clasificación de los casos *benign*

### 8.3. RED INCEPTIONV3-PARALELA

y *cancer* no ha cambiado, pues no han cambiado sus datos. Lo que sí hemos obtenido es una precisión más exacta de este modelo con la clase *normal*. Para esta clase se obtiene un 99,6% fallando 6 casos como *benign* y 7 como *cancer*.

Debido a el número de errores relacionados con la clase *cancer*, podemos llegar a la conclusión de que el modelo tiene dificultades para diferenciar esta clase del resto, obteniendo un mejor resultado en la clase *benign*, aunque estos datos son mejores que los del conjunto de datos anterior, todavía no se pueden considerar como unos resultados que puedan ser utilizados para un sistema de detección real.

Después de esto, ya habríamos visto los resultados obtenidos de las dos redes neuronales que están basados en la arquitectura U-Net. A continuación, veremos las redes faltantes basadas en la tecnología Inception.

## 8.3. Red Inceptionv3-Paralela

En esta sección vamos a analizar los resultados obtenidos para el modelo que utiliza la arquitectura Inception v3 explicada en la sección 7.5.

Igual que en el resto de casos se han utilizado los conjuntos de datos definidos en la sección 6.3 del capítulo 6. Cada modelo tiene un formato adaptado para su caso específico. En este caso, después de muchas pruebas e iteraciones en el entrenamiento se optó por una resolución de  $299 \times 299$  para cada mamografía. Utilizando datos normalizados se obtuvo unos resultados con un grado muy elevado de sobreajuste, por lo que se desecharon la opción de realizar una normalización. En su lugar los datos están en formato *unit8* con valores que van desde 0 a 255. Al hacer esto se logró combatir el sobreajuste y obtener unos resultados más elevados.

### 8.3.1. Conjunto de datos base

El entrenamiento de este modelo llevó alrededor de 4 horas y media. Durante este tiempo, se ejecutaron 50 *epochs* para cada uno de los 5 *folds* de validación cruzada, acumulando un total de 250 *epochs*. Debido a una interrupción, el proceso de

## CAPÍTULO 8. RESULTADOS

entrenamiento se llevó a cabo en dos partes: la primera con los dos primeros pliegues y la segunda con los tres pliegues restantes. A continuación veremos las distintas métricas utilizadas durante este proceso.

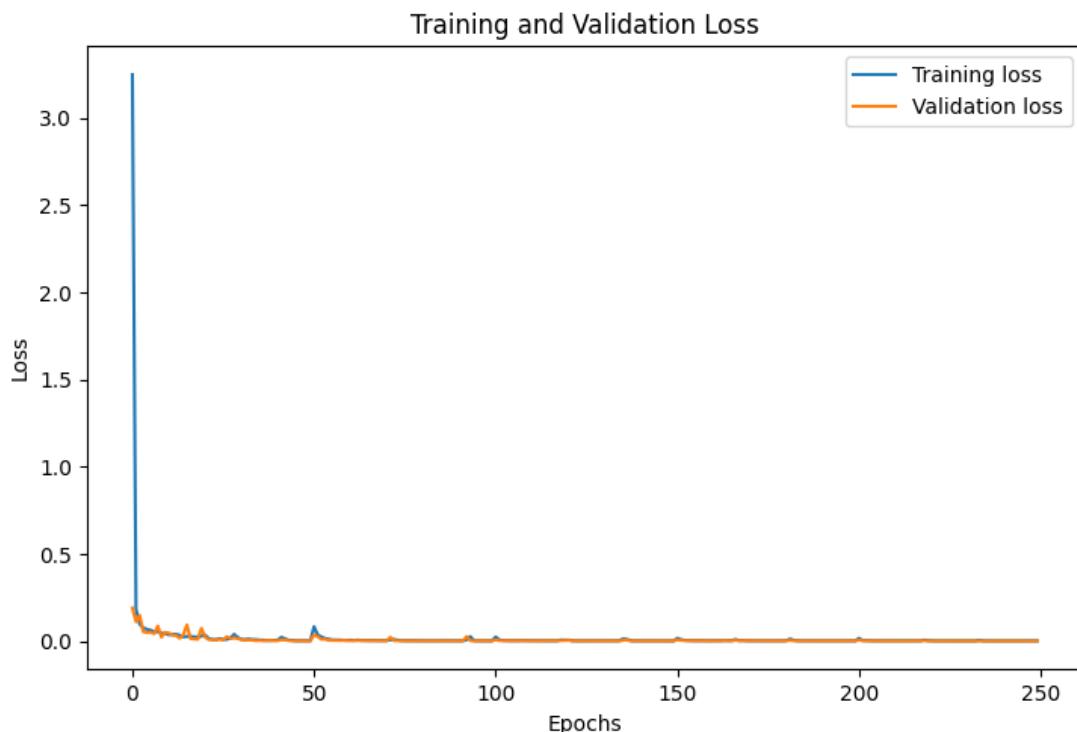


Figura 8.25: Pérdida durante el entrenamiento.

En primer lugar, la figura 8.25 muestra un gráfico con la evolución de la pérdida durante el entrenamiento. Inicialmente, la pérdida es alta, lo cual es esperado, ya que el modelo recién comienza a aprender. Sin embargo, se observa una rápida disminución de la pérdida en los primeros *epochs*, estabilizándose alrededor del valor 0,002, lo que indica que el modelo está aprendiendo a minimizar los errores de predicción. La pérdida con los datos de validación sigue una tendencia similar a la de los datos de entrenamiento, manteniéndose cercana a la pérdida de entrenamiento, lo que sugiere que el modelo no se está sobreajustando.

### 8.3. RED INCEPTIONV3-PARALELA

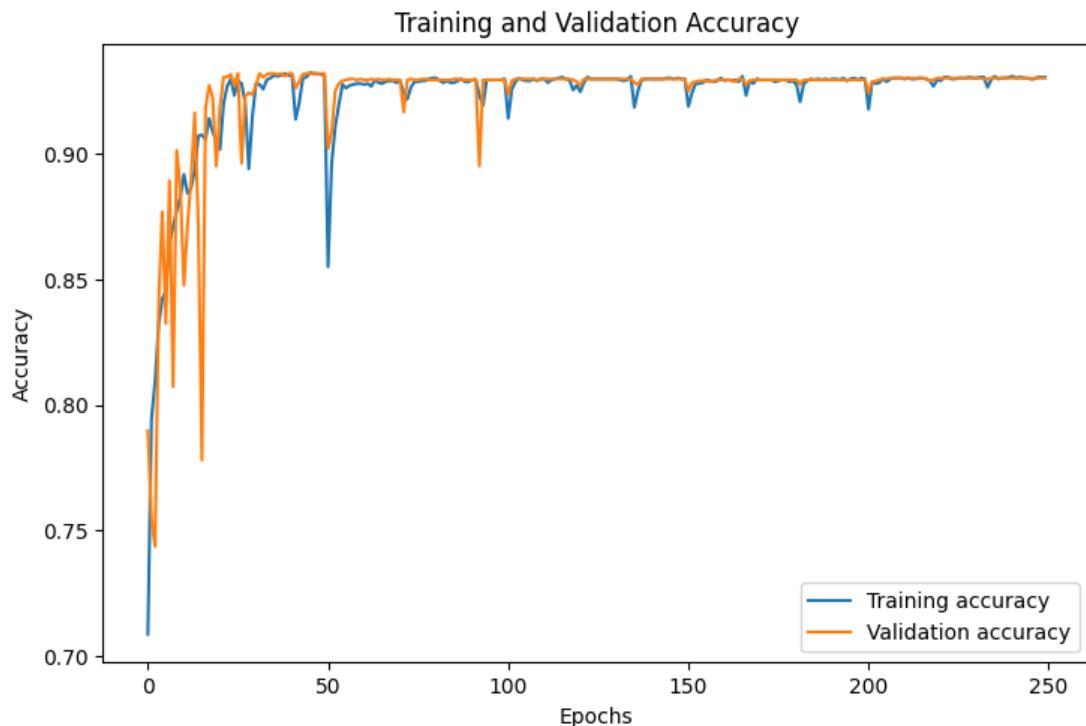


Figura 8.26: *Accuracy* durante el entrenamiento.

En segundo lugar, la figura 8.26 presenta la *accuracy* tanto del conjunto de entrenamiento como del conjunto de validación. A medida que los *epochs* avanzan, la *accuracy* aumenta constantemente, alcanzando valores cercanos al 93 % hacia el final del entrenamiento. Los valores de *accuracy* de entrenamiento y validación se mantienen próximos entre sí, lo cual es indicativo de un buen desempeño del modelo en ambos conjuntos de datos sin dar muestras de sobreajuste.

Finalmente, la precisión<sup>1</sup> esta representada en dos partes, la figura 8.27 representa los dos primeros *folds* y la figura 8.28 los últimos tres. Similar a los valores de *accuracy*, se observa un incremento sostenido en la precisión de clasificación durante los *epochs*, siendo en los últimos *epochs* cercano al 93 %. La cercanía entre las curvas de entrenamiento y validación en este gráfico refuerza la idea de que el modelo mantiene una buena capacidad de generalización sin caer en el sobreajuste.

<sup>1</sup>Precisión (precision): Métrica que indica la proporción de verdaderos positivos entre todas las predicciones positivas.

## CAPÍTULO 8. RESULTADOS



Figura 8.27: Precisión durante el entrenamiento (primeros dos *folds*).



Figura 8.28: Precisión durante el entrenamiento (últimos tres *folds*).

### 8.3. RED INCEPTIONV3-PARALELA

#### Pruebas del modelo

Durante el proceso de entrenamiento, se han guardado dos modelos por cada *fold*. Uno representa el modelo con mayor *accuracy* en los datos de validación, mientras que el otro es el modelo al final de los 50 *epochs*. Esto permite comparar el rendimiento del modelo en su punto óptimo contra su rendimiento al final del entrenamiento.

Vamos a ver los resultados obtenidos con el conjunto de datos de prueba de los diferentes modelos guardados en forma de gráfico de barras. En las figuras 8.29 y 8.30 se muestran la pérdida y el *accuracy* respectivamente. Además, en la tabla 8.13 se muestran estos mismos datos de forma numérica. Veamos cómo se comporta cada uno de los modelos.

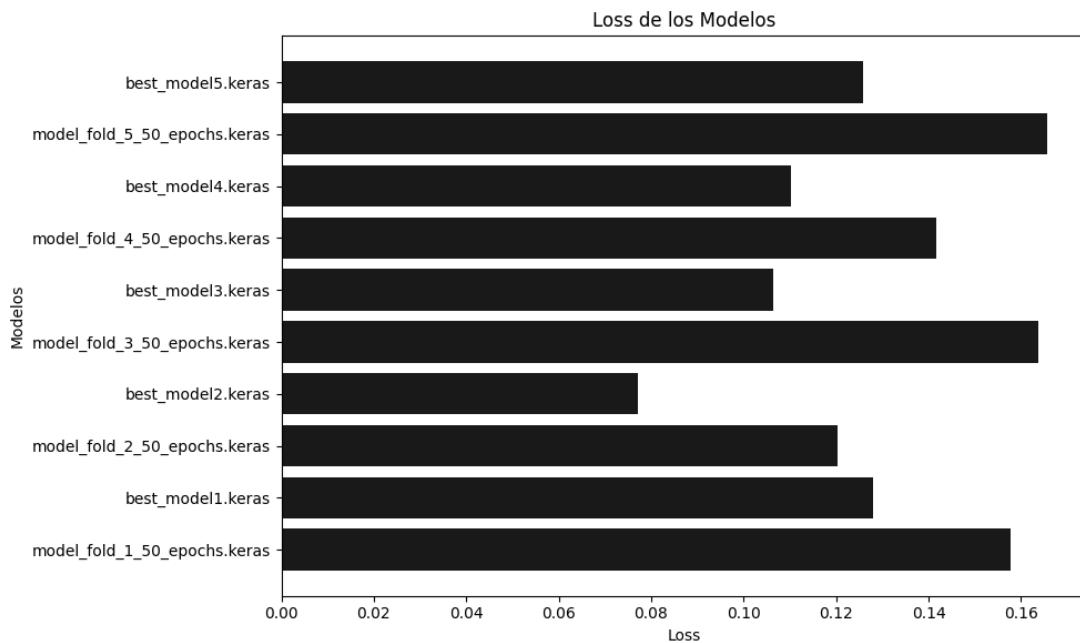


Figura 8.29: Pérdida de los diferentes modelos.

Podemos ver que la pérdida es muy inestable, consiguiendo valores muy bajos dentro de cada *fold* pero valores considerablemente mas altos al final de cada uno. Esto crea una discrepancia elevada entre los modelos que representan el mejor modelo de cada *fold* con los que se guardaron al final.

## CAPÍTULO 8. RESULTADOS

---

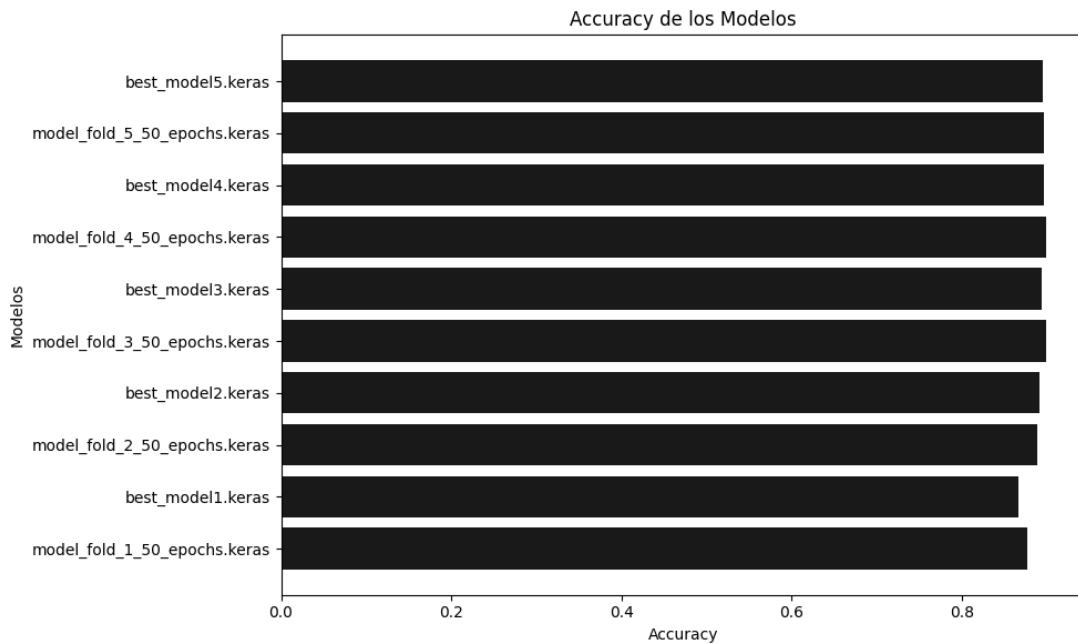


Figura 8.30: *Accuracy* de los diferentes modelos.

Por otro lado, el *accuracy* es muy estable. Este se mantiene relativamente estable, con una mejora leve en los cuatro primeros *folds*. Esto nos da mucha confianza en el resultado obtenido.

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,1564	0,8681
best_model11.keras	0,1270	0,8658
model_fold_2_50_epochs.keras	0,1194	0,8880
best_model12.keras	0,0766	0,8915
model_fold_3_50_epochs.keras	0,1627	0,8868
best_model13.keras	0,1056	0,8961
model_fold_4_50_epochs.keras	0,1406	0,8891
best_model14.keras	0,1095	0,8856
model_fold_5_50_epochs.keras	0,1647	0,8868
best_model15.keras	0,1250	0,8938
<b>Media aritmética</b>	<b>0,1288</b>	<b>0,8852</b>

Tabla 8.13: Resultados de los modelos evaluados en el conjunto de datos de prueba.

Basándonos en estos resultados podemos calcular el rendimiento del modelo. En la validación cruzada, se debe de realizar la media aritmética de sus métricas. En este caso se obtiene un pérdida media del 0,1299 y un *accuracy* medio del 88,30 %. Estos

### 8.3. RED INCEPTIONV3-PARALELA

muestran como es capaz de realizar una diferenciación de cada una de las clases con cierta precisión, pero debemos de tener en cuenta que los modelos están entrenados para no clasificar una de las cuatro clases, la clase *actionable*. La inclusión de esta categoría se hizo para poder extraer conclusiones de las características de la clase *actionable* durante el entrenamiento.

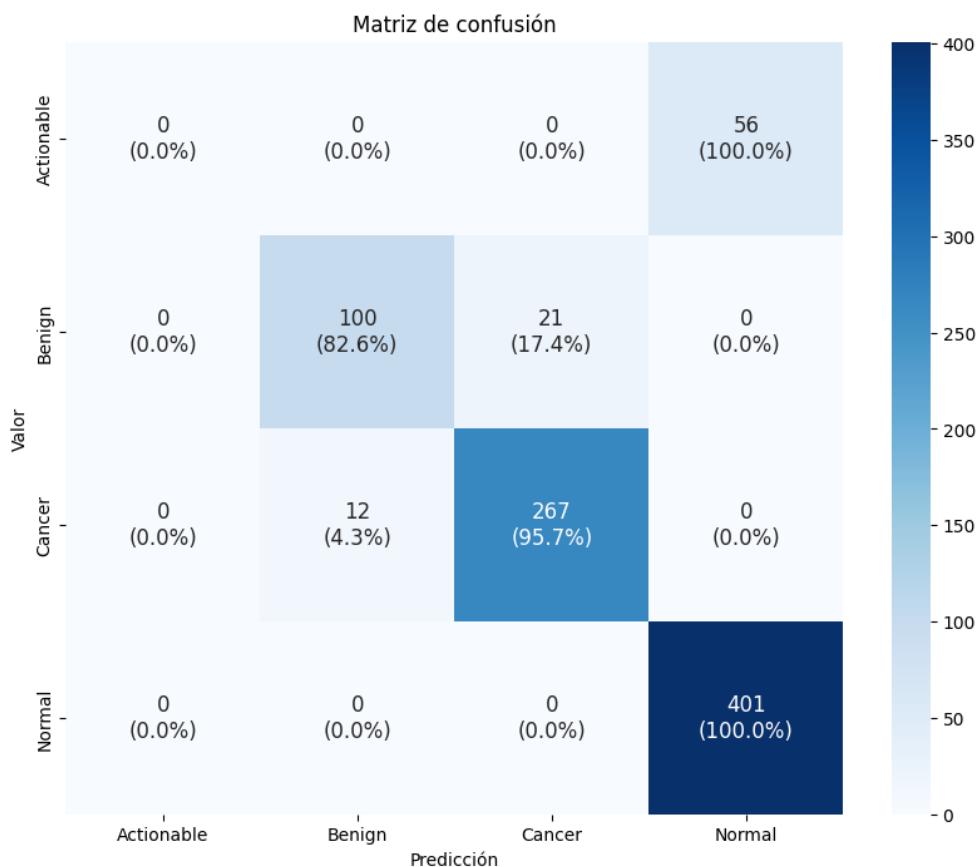


Figura 8.31: Matriz de confusión del modelo “best\_model3.keras”.

Para poder ver que es lo que está haciendo el modelo de una forma más concreta, vamos a ver la matriz de confusión del modelo “best\_model3.keras” en la figura 8.31, el cual es el que obtiene mayor *accuracy* de todos.

En esta matriz, lo primero que observamos es como la clase *actionable* se clasifica siempre como *normal*. Debido a que en la función de pérdida se especificó a esta clase con un peso cero, es normal que no acierte. Lo que sacamos en claro con esto es que la clase *actionable* se parece mas a un caso *normal* que a un caso con un tumor, ya sea

## CAPÍTULO 8. RESULTADOS

---

*benign* o *cancer*.

En segundo lugar, destaca la capacidad del modelo para diferenciar los casos de pacientes sanos de aquellos que tienen un tumor benigno o un cáncer, logrando un resultado del 100 %. No obstante, el problema de tener datos tan limitados implica que no podamos medir completamente las capacidades del modelo.

En último lugar, muestra un resultado notable sobre las clases *benign* y *cancer*, teniendo una precisión del 95,7 % para la clase *cancer* y del 82,6 % para la clase *benign*. Era de esperar un resultado menor en la clase *benign* debido a que esta cuenta con un 57 % menos de casos en comparación con la clase *cancer*. Aun así el modelo muestra un excelente resultado clasificando los casos de todas las categorías.

### Pruebas sin la clase *actionable*

La siguiente tabla 8.14 muestra cual es el rendimiento del modelo al correr las mismas pruebas sin los casos de la clase *actionable*.

Al quitar los casos de la categoría *actionable* han subido tanto la pérdida como el *accuracy*. La pérdida ha subido porque los modelos categorizaban los casos eliminados con mucha confianza, y el *accuracy* debido a que ahora ya no está fallando en los 56 casos *actionable*. Con esto conseguimos un rendimiento con una pérdida media de 0,1377 y un *accuracy* medio del 94,71 %.

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,1673	0,9288
best_model1.keras	0,1359	0,9263
model_fold_2_50_epochs.keras	0,1277	0,9501
best_model2.keras	0,0820	0,9538
model_fold_3_50_epochs.keras	0,1740	0,9488
best_model3.keras	0,1130	0,9588
model_fold_4_50_epochs.keras	0,1504	0,9513
best_model4.keras	0,1171	0,9476
model_fold_5_50_epochs.keras	0,1762	0,9488
best_model5.keras	0,1338	0,9563
<b>Media aritmética</b>	<b>0,1377</b>	<b>0,9471</b>

Tabla 8.14: resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase *actionable*.

### 8.3. RED INCEPTIONV3-PARALELA

---

#### Pruebas con casos normales ampliados

Dado que el modelo ha cometido no ha cometido errores al clasificar los casos normales en el conjunto de datos de prueba original se ha aumentado el número de casos de la categoría *normal* para una evaluación más precisa. Para lograr esto, se han añadido todos los casos *normales* que fueron desechados y no utilizados durante el proceso de validación cruzada. Manteniendo así la independencia fundamental de los datos.

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,0344	0,9852
best_model1.keras	0,0283	0,9845
model_fold_2_50_epochs.keras	0,0263	0,9896
best_model2.keras	0,0169	0,9903
model_fold_3_50_epochs.keras	0,0359	0,9893
best_model3.keras	0,0233	0,9913
model_fold_4_50_epochs.keras	0,0312	0,9893
best_model4.keras	0,0245	0,9883
model_fold_5_50_epochs.keras	0,0366	0,9888
best_model5.keras	0,0289	0,9901
<b>Media aritmética</b>	<b>0,0286</b>	<b>0,9887</b>

Tabla 8.15: resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase *actionable* y con datos ampliados para la clase *normal*.

Además, al aumentar los casos de esta categoría, se está consiguiendo un rendimiento mucho más fiel a la realidad. Esto es debido a que la mayor parte de los pacientes que se someten a exámenes de detección están completamente sanos.

En la tabla 8.15 podemos ver los resultados obtenidos, viendo como la pérdida ha bajado hasta una media de 0,0286 y el *accuracy* ha aumentado hasta un valor medio del 98,87 %.

La matriz de confusión de la figura 8.32 nos muestra los resultados finales obtenidos con el conjunto de datos base. En esta matriz los casos de las categorías *benign* y *cancer* mantienen se mantienen igual al ser los mismos datos. La diferencia reside en la categoría *normal*, la cual sigue mostrando un resultado sobresaliente, con un valor aproximado del 100 %. Solamente uno de los casos de esta categoría fue

## CAPÍTULO 8. RESULTADOS

erróneamente predicho como un caso de cáncer.

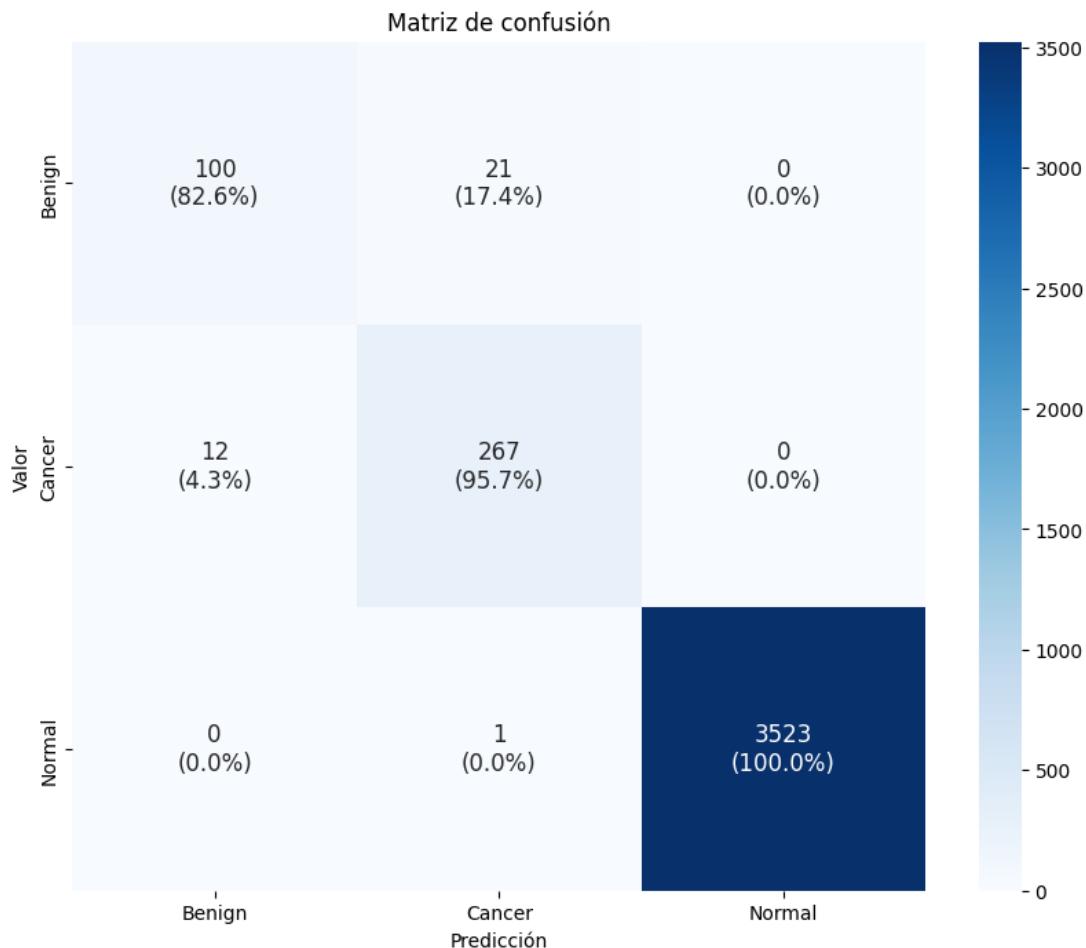


Figura 8.32: matriz de confusión del modelo “best\_model3.keras” sin la categoría *actionable* y con datos ampliados para la clase *normal*.

Con esto, ya habríamos visto los resultados de esta red neuronal para el conjunto de datos base, en los cuales observamos un resultado prácticamente perfecto para los casos *normales*, pero, mejorables para el resto de categorías.

### 8.3.2. Conjunto de datos con aumento de datos

A continuación, se mostrarán los resultados del entrenamiento con el conjunto de datos que hace uso de la técnica *data augmentation* (DA). Recordemos que en este conjunto se han duplicado los casos de la categoría *benign* buscando una mejora en la

### 8.3. RED INCEPTIONV3-PARALELA

precisión de esta clase.

El entrenamiento de este conjunto de datos tomó aproximadamente 5 horas y media. Durante este periodo, al igual que con el conjunto de datos anterior, se ejecutaron 50 épocas para cada uno de los 5 *folds* de la validación cruzada, acumulando un total de 250 *epochs*. Este entrenamiento tomó más tiempo debido al aumento de datos en la clase *benign*.

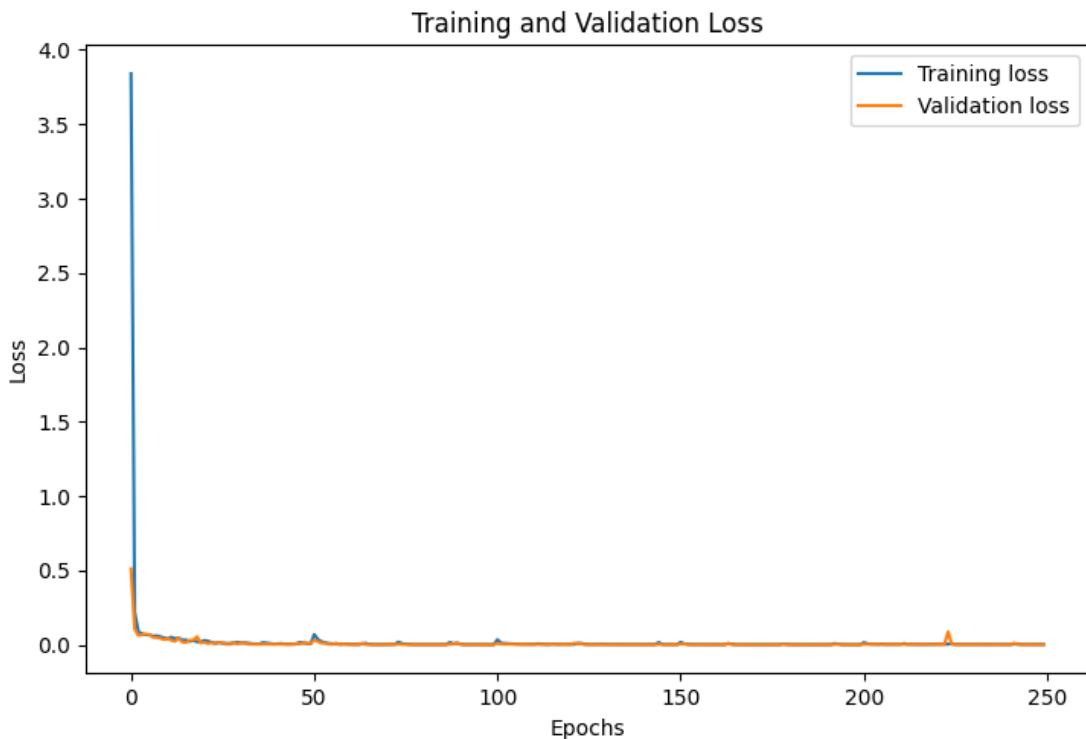


Figura 8.33: Pérdida durante el entrenamiento.

Primero vamos a ver las métricas utilizadas durante este proceso. La figura 8.33 muestra un gráfico con la evolución de la pérdida durante el entrenamiento, tanto con los datos usados para el entrenamiento como para la validación. Similar al comportamiento observado con el conjunto de datos anterior, la pérdida de entrenamiento es alta al inicio, pero se reduce rápidamente en las primeras épocas, estabilizándose en valores muy bajos. Esto indica que el modelo está aprendiendo a minimizar los errores de predicción. La pérdida de validación sigue una tendencia similar a la de entrenamiento, lo que sugiere que el modelo no se está sobreajustando. Al comparar este conjunto de datos con el anterior, observamos que se obtienen

## CAPÍTULO 8. RESULTADOS

resultados muy similares, lo cual es una buena señal ya que es un valor muy bajo.

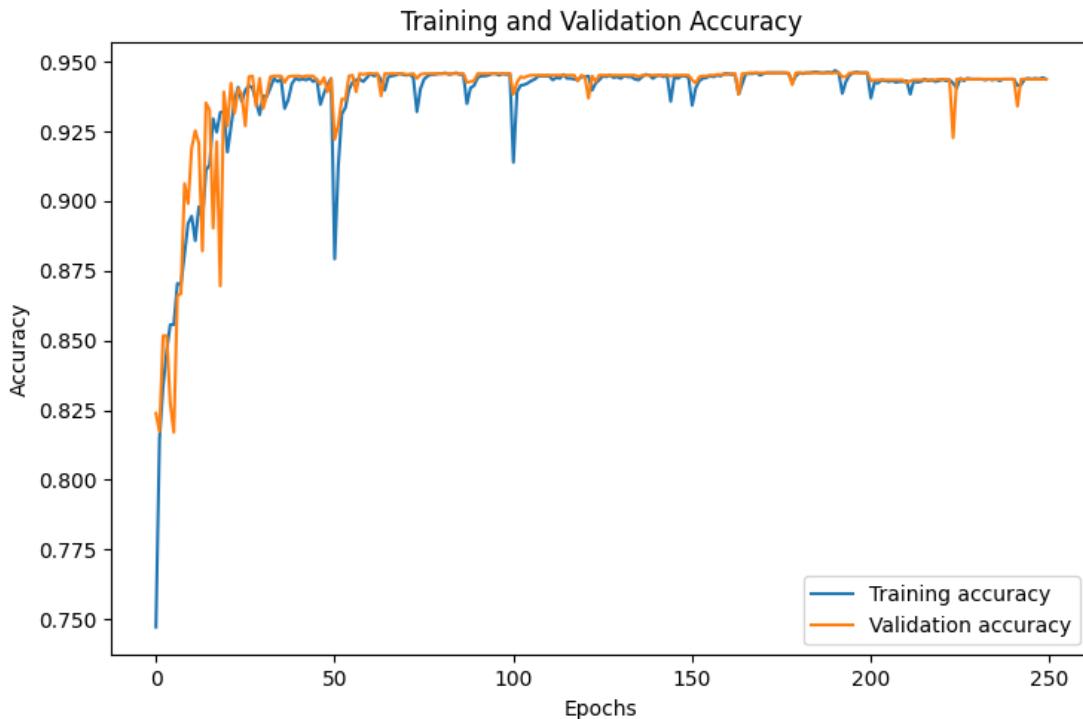


Figura 8.34: *Accuracy* durante el entrenamiento.

La figura 8.34 representa el *accuracy* tanto del conjunto de datos de entrenamiento como de validación. El *accuracy* se estabiliza durante el segundo *fold*, alcanzando un valor que ronda el 94.5% hasta el final del entrenamiento. El *accuracy* del entrenamiento y de la validación se mantienen cercanas entre sí, lo cual indica un buen desempeño del modelo en ambos conjuntos de datos, sin mostrar sobreajuste observable.

La evolución de la precisión se muestra en la figura 8.35. Al igual que la gráfica de *accuracy*, se observa una estabilización sostenida a lo largo de las épocas, alcanzando un valor cercano al 94.5%.

En este caso, el modelo logra una mayor precisión y exactitud por dos posibles razones: primero, hay un mayor número de datos en la categoría *benign*; y segundo, debido a este mismo aumento de datos, la categoría *actionable* se vuelve menos representativa en el conjunto de datos total. Esto permite al modelo enfocarse mejor en las clasificaciones de *benign* y *cancer*, mejorando su rendimiento general.

### 8.3. RED INCEPTIONV3-PARALELA

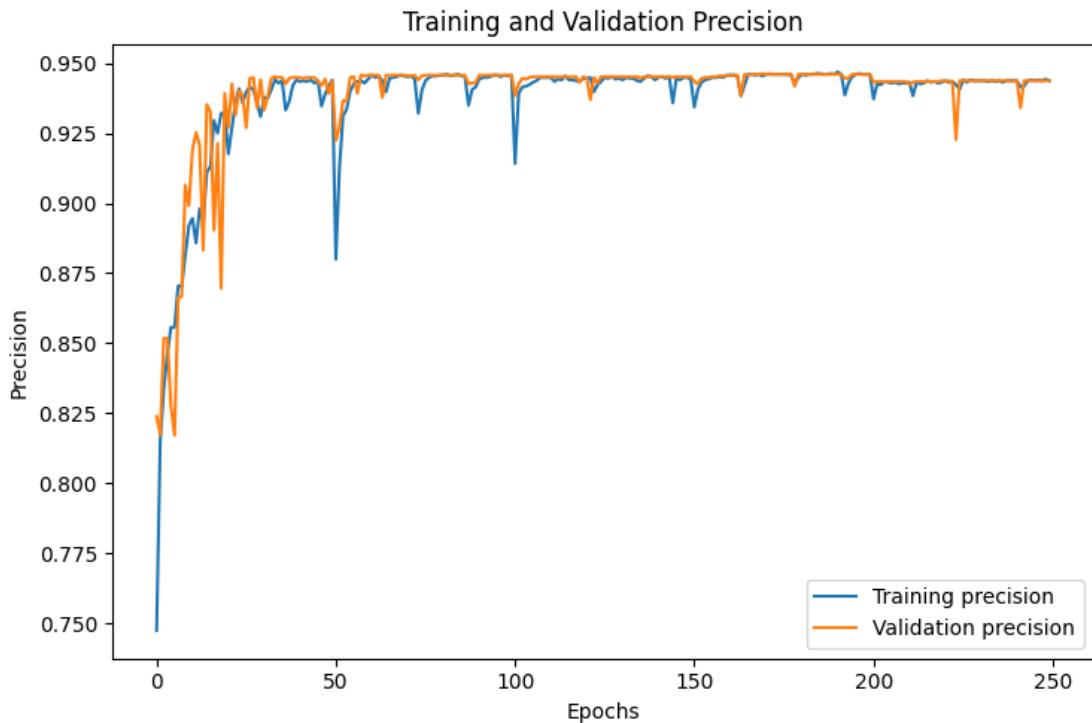


Figura 8.35: Precisión durante el entrenamiento.

## Pruebas del modelo

Al igual que con el conjunto de datos anterior, durante el proceso de entrenamiento se han guardado dos modelos por cada *fold*. Uno de estos modelos representa el modelo con la mayor exactitud en los datos de validación, mientras que el otro es el modelo al final de las 50 *epochs*.

Vamos a analizar cómo se comporta cada uno de los modelos guardados con el conjunto de datos de prueba. En la figura 8.36 se muestra la pérdida de los diferentes modelos guardados y a figura 8.37 presenta el *accuracy* de estos modelos. Además, en la tabla 8.16 podemos ver los valores numéricos.

La pérdida disminuye de manera significativa hasta el cuarto *fold*, momento en el cual se observa un aumento. Este incremento puede atribuirse, con alta probabilidad, a un fenómeno de sobreajuste en los datos utilizados para entrenar el modelo. Es importante destacar que, dado que estamos empleando la técnica de validación cruzada, este sobreajuste resulta difícil de detectar durante el proceso de

## CAPÍTULO 8. RESULTADOS

entrenamiento. Sin embargo, en el quinto *fold*, la pérdida vuelve a disminuir, lo que sugiere una ligera mitigación del sobreajuste.

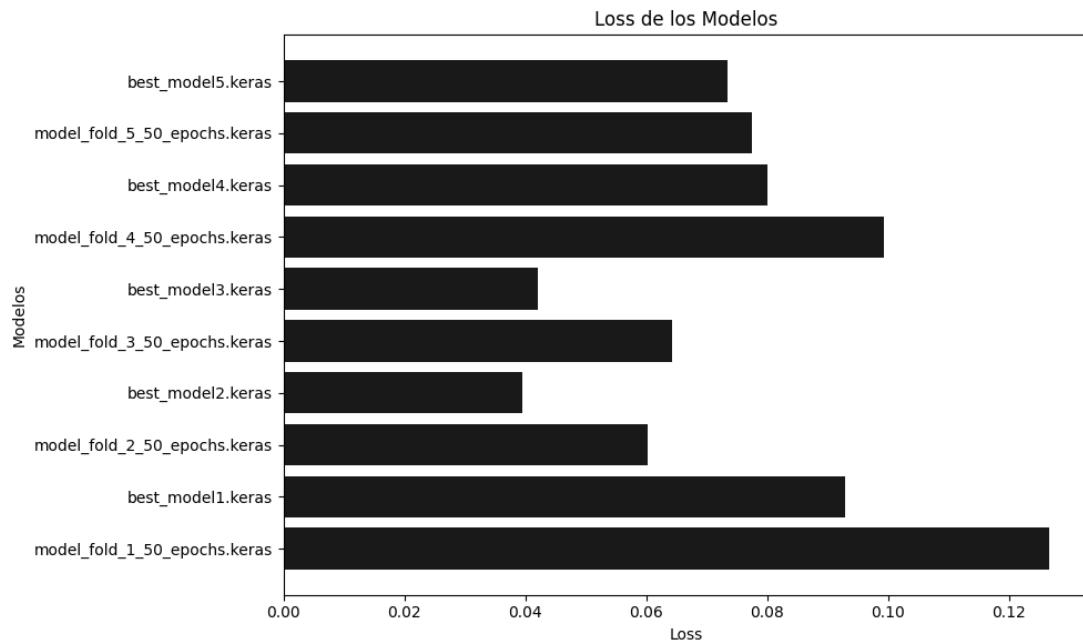


Figura 8.36: Pérdida de los diferentes modelos.

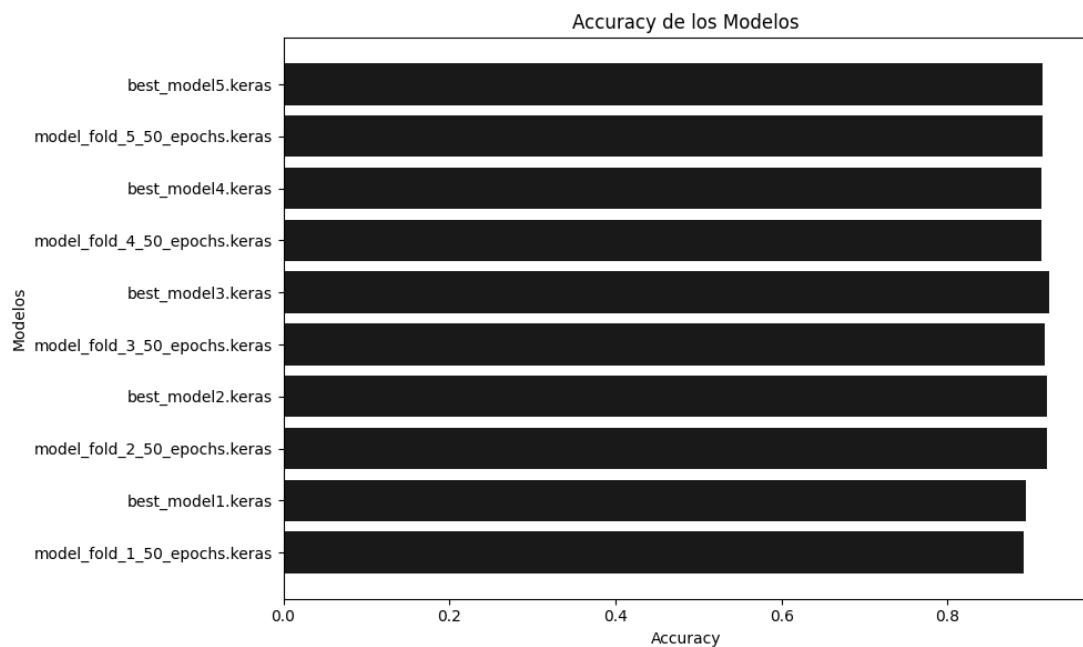


Figura 8.37: Accuracy de los diferentes modelos.

### 8.3. RED INCEPTIONV3-PARALELA

---

Con respecto al *accuracy* obtenido de los diferentes modelos, este muestra un aumento entre los *folds* uno y dos. Después se mantiene muy estable, alcanzando su pico máximo en el *fold* tres.

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,1258	0,8926
best_model1.keras	0,0921	0,8954
model_fold_2_50_epochs.keras	0,0597	0,9208
best_model2.keras	0,0392	0,9199
model_fold_3_50_epochs.keras	0,0638	0,9172
best_model3.keras	0,0417	0,9236
model_fold_4_50_epochs.keras	0,0985	0,9136
best_model4.keras	0,0795	0,9136
model_fold_5_50_epochs.keras	0,0770	0,9145
best_model5.keras	0,0730	0,9154
<b>Media aritmética</b>	<b>0,0750</b>	<b>0,9127</b>

Tabla 8.16: Resultados de los modelos DA evaluados en el conjunto de datos de prueba.

Teniendo en cuenta estos resultados podemos calcular el rendimiento de esta red neuronal con el uso de validación cruzada. Se obtiene una media del 91,27% en el *accuracy* y de 0,0750 en la pérdida. En ambas métricas hay una mejoría notable con respecto de el anterior conjunto de datos, obteniendo un resultado sobresaliente. Además, todos los modelos tienen un *accuracy* muy similar, aunque difieren más en el valor de pérdida, en el cual podemos ver como disminuye hasta el *fold* 4, donde vuelve a aumentar. El aumento de la pérdida puede que se produzca como resultado de un ligero sobreajuste con los datos utilizados en la validación cruzada.

Vamos a ver cómo está clasificando cada uno de los casos gracias a la matriz de confusión que se presenta en la tabla 8.38. El modelo que más *accuracy* tiene es el “best\_model3.keras”, el cual además cuenta con la segunda pérdida mas baja. En concreto, tiene 92,36 % de *accuracy* y una pérdida de 0,0417.

En primer lugar, como era de esperar, falla en todos los casos de la categoría *actionable* clasificándolos nuevamente como *normal*. Por otro lado, vuelve a tener un desempeño excelente al diferenciar la categoría *Normal* con el resto, fallando únicamente con caso *benign*. La mayor diferencia con respecto al conjunto de datos

## CAPÍTULO 8. RESULTADOS

base se encuentra en los casos de las categorías *cancer* y *benign*, en los cuales obtiene una mucho mayor precisión, con un 95 % y un 94,2 % respectivamente. Esta mejoría es el resultado de tener un mayor balance entre los casos de estas dos categorías.

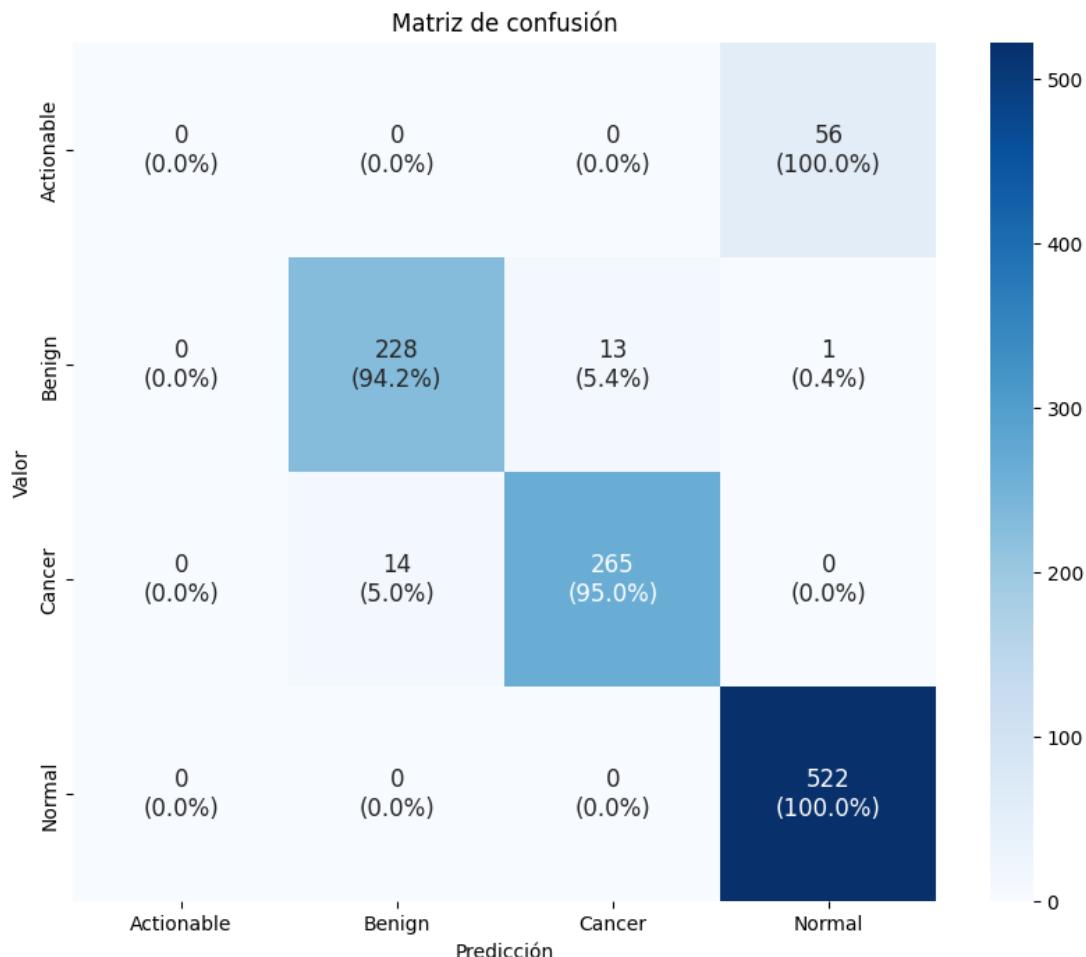


Figura 8.38: Matriz de confusión del modelo “best\_model3.keras”.

### Pruebas sin la clase *actionable*

Debemos volver a tener en cuenta que los modelos están entrenados para no clasificar la clase *actionable*. En la tabla 8.17 vamos a ver el desempeño de estos modelos quitando los casos de esta clase. De esta forma nos acercaremos más al rendimiento real.

### 8.3. RED INCEPTIONV3-PARALELA

---

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,1325	0,9406
best_model1.keras	0,0971	0,9434
model_fold_2_50_epochs.keras	0,0629	0,9703
best_model2.keras	0,0413	0,9693
model_fold_3_50_epochs.keras	0,0672	0,9664
best_model3.keras	0,0440	0,9732
model_fold_4_50_epochs.keras	0,1038	0,9626
best_model4.keras	0,0837	0,9626
model_fold_5_50_epochs.keras	0,0811	0,9636
best_model5.keras	0,0769	0,9645
<b>Media aritmética</b>	<b>0,0791</b>	<b>0,9617</b>

Tabla 8.17: Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase *actionable*.

Nuevamente, sin los casos *actionable*, ha subido el valor medio de la pérdida y de el *accuracy* consiguiendo un 0,0791 y un 96,17% respectivamente. Estos valores son muy buenos y el modelo “best\_model3.keras” vuelve a ser el modelo que da un mejor resultado.

### Pruebas con casos normales ampliados

Finalmente, para ver el rendimiento final de la red con Inception v3 implementada, vamos a aumentar los casos de la categoría *normal* nuevamente.

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,0396	0,9820
best_model1.keras	0,0290	0,9829
model_fold_2_50_epochs.keras	0,0189	0,9910
best_model2.keras	0,0125	0,9907
model_fold_3_50_epochs.keras	0,0201	0,9899
best_model3.keras	0,0135	0,9913
model_fold_4_50_epochs.keras	0,0313	0,9885
best_model4.keras	0,0249	0,9888
model_fold_5_50_epochs.keras	0,0245	0,9890
best_model5.keras	0,0232	0,9893
<b>Media aritmética</b>	<b>0,0238</b>	<b>0,9883</b>

Tabla 8.18: Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase *actionable* y con datos ampliados para la clase *normal*.

## CAPÍTULO 8. RESULTADOS

En la tabla 8.18 podemos ver los resultados obtenidos, viendo como la pérdida ha bajado hasta una media de 0,0238 y el *accuracy* ha aumentado hasta un valor medio del 98,83 %.

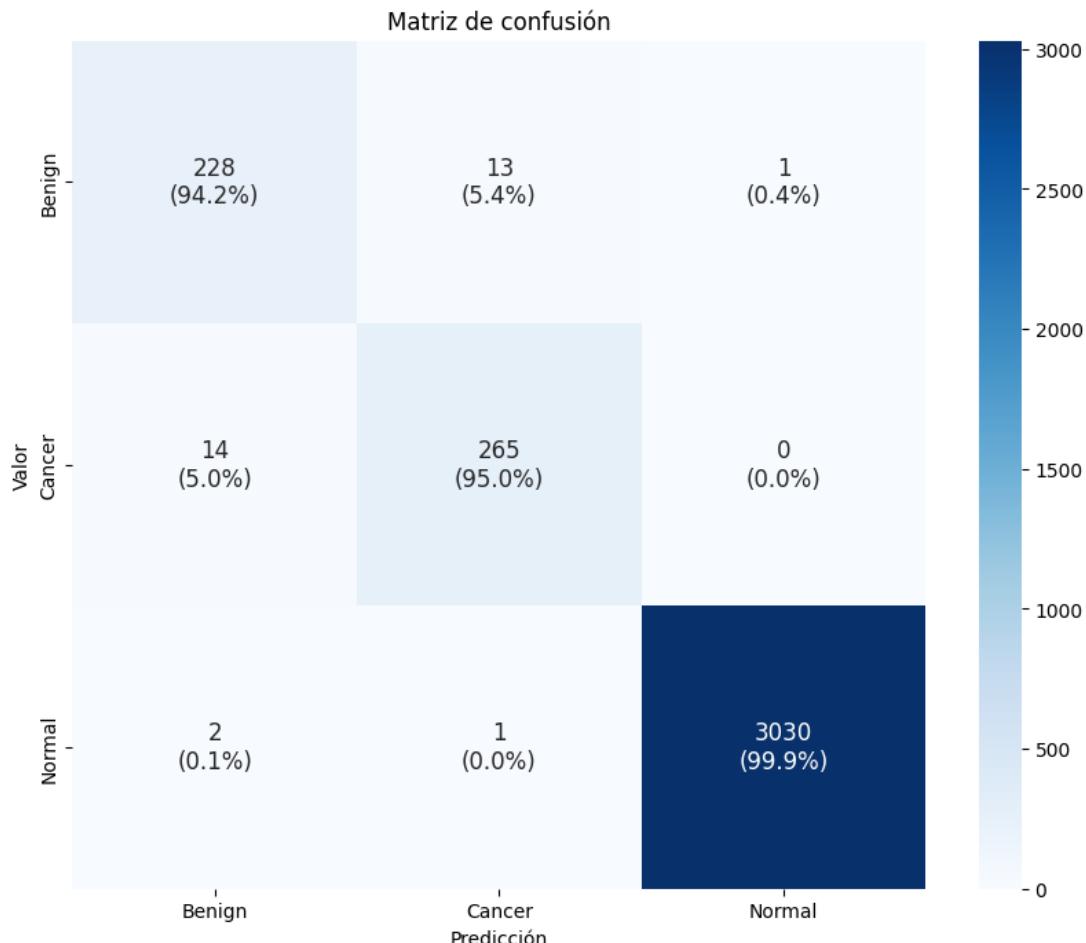


Figura 8.39: Matriz de confusión del modelo “best\_model3.keras” sin la categoría *actionable* y con datos ampliados para la clase *normal*.

La matriz de confusión de la figura 8.39 muestra los resultados finales obtenidos con el conjunto de datos con aumento de datos. Los casos de las categorías *benign* y *cancer* mantienen se mantienen igual al ser los mismos datos. La diferencia reside en la categoría *normal*, la cual sigue mostrando un resultado sobresaliente, con un valor aproximado del 100 %. Solamente 3 casos de esta categoría fueron erróneamente predichos, dos casos como *benign* y un caso como *cancer*.

Con esto, ya habríamos visto el rendimiento de esta red neuronal con sus dos

## 8.4. RED RESNETV2-PARALELA

conjuntos de datos, en el capítulo [9, Conclusiones](#), se discutirán estos resultados comparándolos con los obtenidos con el resto de modelos y los antecedentes.

## 8.4. Red ResNetv2-Paralela

A continuación, vamos a ver los resultados del último modelo desarrollado. Nuevamente, se analizarán los dos conjuntos de datos definidos en la sección [Conjuntos utilizados en el entrenamiento de modelos](#) del capítulo [6, procesado de datos](#).

Igual que para el anterior modelo Inception, se han adaptado los datos con una resolución de  $299 \times 299$  para cada mamografía, sin normalización de datos y con formato *Unit8*. Las causas de estas decisiones son similares a las explicadas con el modelo *Inception v3*. Al utilizar datos normalizados se obtuvo unos resultados con un grado muy elevado de sobreajuste y muy poco rendimiento en la fase de prueba, por lo que se deseó esta opción. Por esta razón, los valores van desde 0 a 255. Al hacer esto se logró combatir el sobreajuste y obtener unos resultados muy superiores.

Además, el entrenamiento del modelo se ha realizado nuevamente con los dos conjuntos y utilizando la técnica de validación cruzada.

### 8.4.1. Conjunto de datos base

El entrenamiento de este conjunto de datos requirió aproximadamente 8 horas de procesamiento. Durante este período, se ejecutaron 50 *epochs* para cada uno de los 5 *folds* de la validación cruzada, lo que resultó en un total de 250 *epochs*. Este enfoque permitió mitigar problemas como el sobreajuste, al aumentar la cantidad de datos utilizados en el entrenamiento.

La figura [8.40](#) muestra un gráfico con la evolución de la pérdida durante el entrenamiento. Inicialmente, la pérdida del entrenamiento es alta, lo cual es esperado, ya que el modelo recién comienza a aprender. Sin embargo, se observa una rápida disminución de la pérdida en los primeros *epochs*, estabilizándose en valores muy

## CAPÍTULO 8. RESULTADOS

bajos, lo que indica que el modelo está aprendiendo a minimizar los errores de predicción. La pérdida de validación sigue una tendencia similar, manteniéndose cercana a la pérdida de entrenamiento, lo que sugiere que el modelo no está sobreajustando los datos de entrenamiento frente a los de validación.

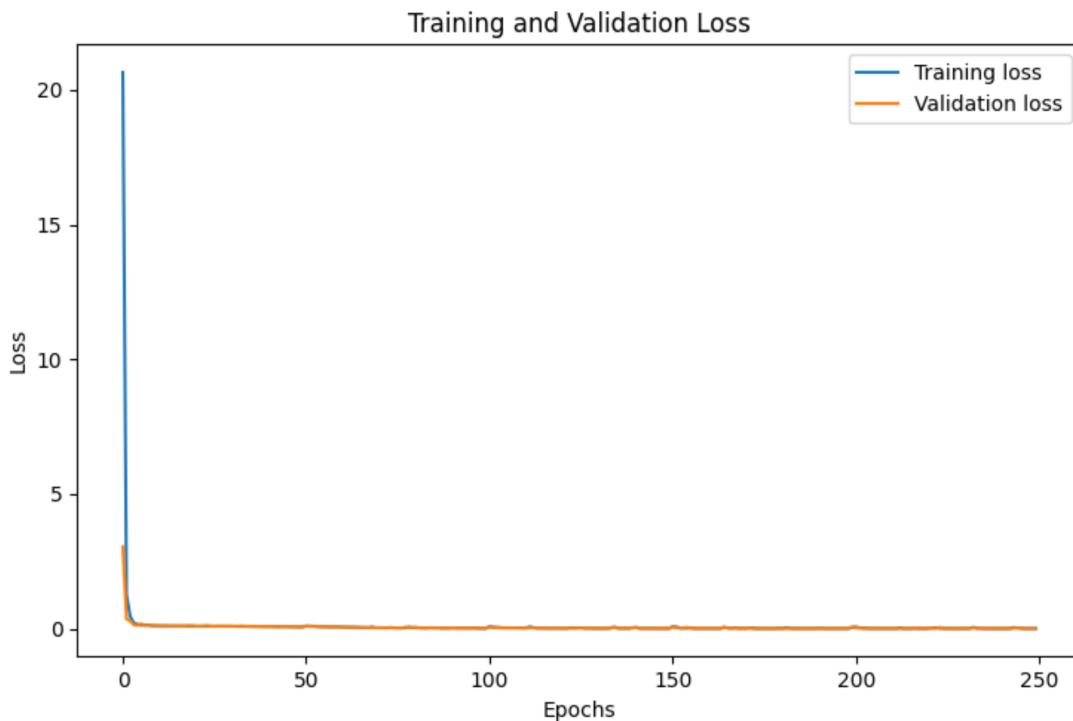


Figura 8.40: Pérdida durante el entrenamiento

La figura 8.41 presenta el *accuracy* tanto del conjunto de entrenamiento como del conjunto de validación. A medida que los *epochs* avanzan, el *accuracy* aumenta constantemente, alcanzando valores cercanos al 90 % hacia el final del entrenamiento. Las curvas de exactitud de entrenamiento y validación se mantienen próximas entre sí, lo cual es indicativo de un buen desempeño del modelo en ambos conjuntos de datos.

También, puede observarse una bajada en el *accuracy* cada 50 *epochs*. Esta bajada es producida por el cambio de *fold*, debido a que es el momento en el que el modelo cambia los datos de entrenamiento y validación, por lo tanto, es el momento en el que vuelve a aprender de datos diferentes. Estas bajadas reducen su caída a medida que pasan los distintos *folds*, probablemente debido a que está aprendiendo a generalizar mejor el total de datos o se está produciendo un sobreajuste entre los datos utilizados

#### 8.4. RED RESNETV2-PARALELA

en la validación cruzada.



Figura 8.41: Accuracy durante el entrenamiento

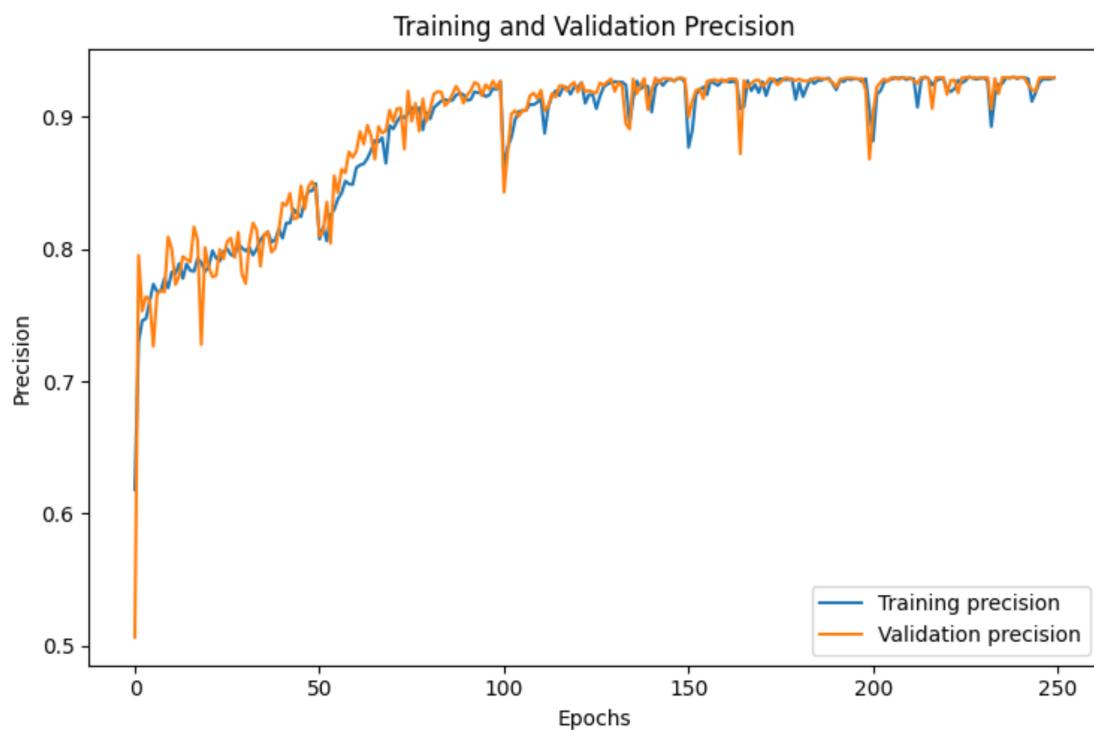


Figura 8.42: Precisión durante el entrenamiento

## CAPÍTULO 8. RESULTADOS

Finalmente, la figura 8.42 muestra la evolución de la precisión en términos de clasificación correcta de las diferentes clases. Similar al *accuracy*, se observa un incremento sostenido en la precisión de clasificación durante el paso de los *epochs* y las bajadas relativas a los cambios de *fold*. La cercanía entre las curvas de entrenamiento y validación en este gráfico refuerza la idea de que el modelo mantiene una buena capacidad de generalización.

### Pruebas del modelo

Durante el proceso de entrenamiento, se han guardado dos modelos por cada *fold*. Uno representa el modelo con mayor *accuracy* en los datos de validación, mientras que el otro es el modelo al final de cada *fold*. Esto permite comparar el rendimiento del modelo en su punto óptimo contra su rendimiento al final del entrenamiento.

En la figura 8.43 se muestra la pérdida de los diferentes modelos guardados y en la figura 8.44 presenta el *accuracy* de estos modelos. Además podemos ver los resultados numéricos de estas pruebas en la tabla 8.19. Con estos resultados podemos analizar cómo se comporta cada uno de los modelos con el conjunto de datos de prueba.

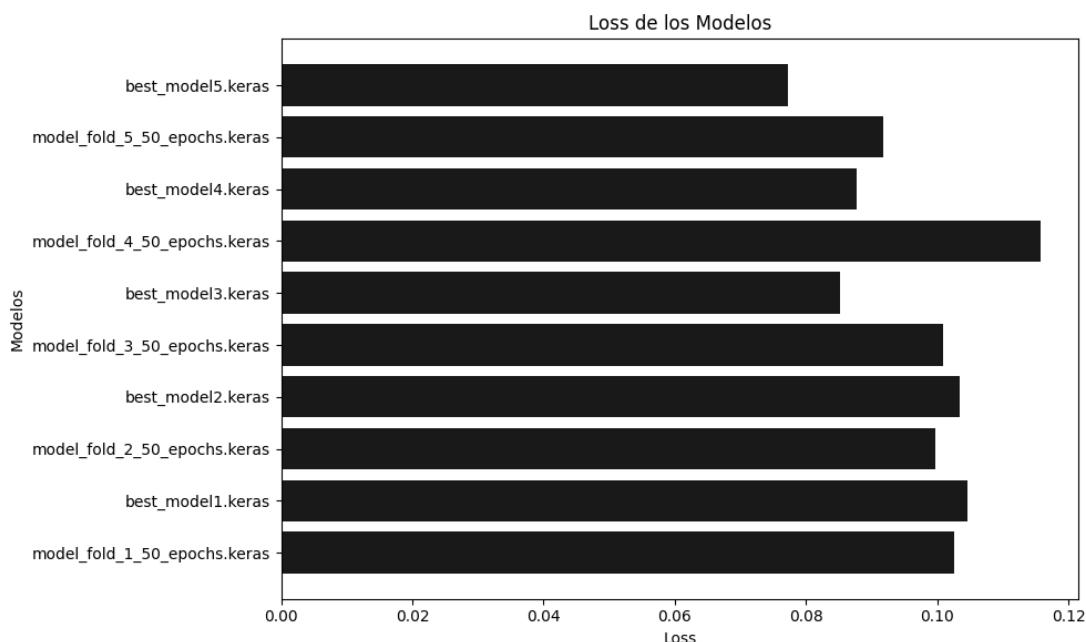


Figura 8.43: Pérdida de los diferentes modelos

#### 8.4. RED RESNETV2-PARALELA

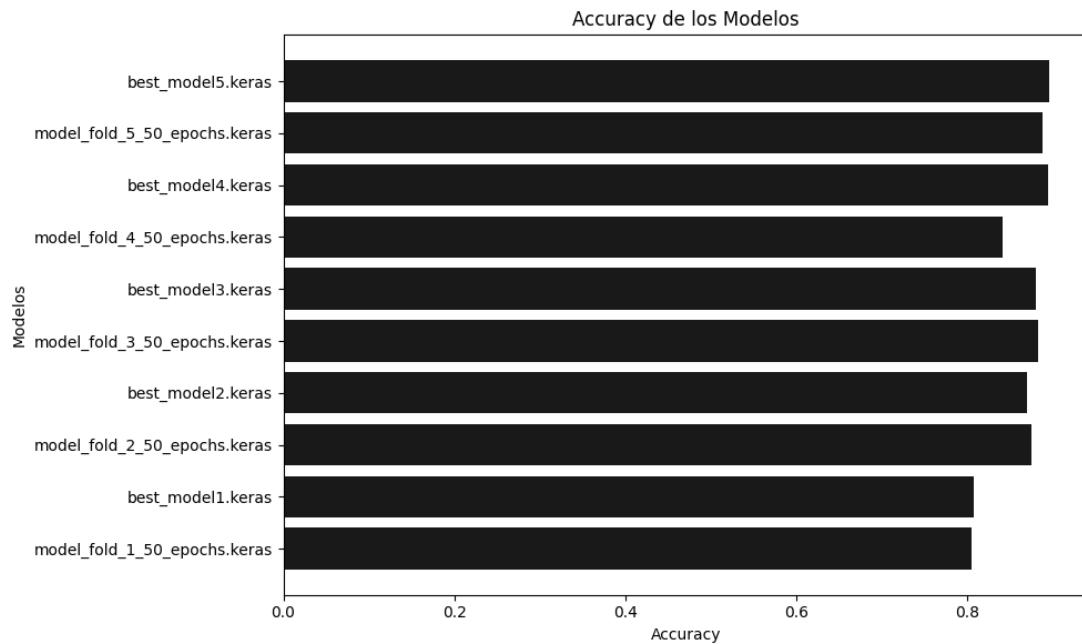


Figura 8.44: Accuracy de los diferentes modelos

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,0873	0,7981
best_model1.keras	0,0856	0,8133
model_fold_2_50_epochs.keras	0,1401	0,8705
best_model2.keras	0,1352	0,8751
model_fold_3_50_epochs.keras	0,1185	0,8810
best_model3.keras	0,1160	0,8798
model_fold_4_50_epochs.keras	0,1355	0,8506
best_model4.keras	0,1188	0,8845
model_fold_5_50_epochs.keras	0,1217	0,8915
best_model5.keras	0,1123	0,8938
<b>Media aritmética</b>	<b>0,1171</b>	<b>0,8638</b>

Tabla 8.19: Resultados de los modelos evaluados en el conjunto de datos de prueba.

Para calcular el rendimiento del modelo con validación cruzada, debemos realizar la media aritmética de la pérdida y el *accuracy* de los diferentes modelos. Como resultado obtenemos una pérdida media de 0,1171 y un *accuracy* del 86,38 %. Este rendimiento nos permite observar que es capaz de categorizar relativamente las diferentes categorías, pero debemos de tener en cuenta que las pruebas se están realizando con los casos de la categoría *actionable*. Esta categoría solo se incluyó para

## CAPÍTULO 8. RESULTADOS

observar el comportamiento del modelo con la misma y, por tanto, baja y empeora el *accuracy*.

Para ver el rendimiento del modelo de una forma más detallada, vamos a ver los resultados que muestra para cada caso. Esto lo podemos hacer mediante la matriz de confusión representada en la figura 8.45. Esta matriz muestra los resultados obtenidos con el modelo “best\_model5.keras”. Se ha elegido este modelo debido a que es el que muestra menor pérdida y mayor *accuracy* muestra entre el resto.

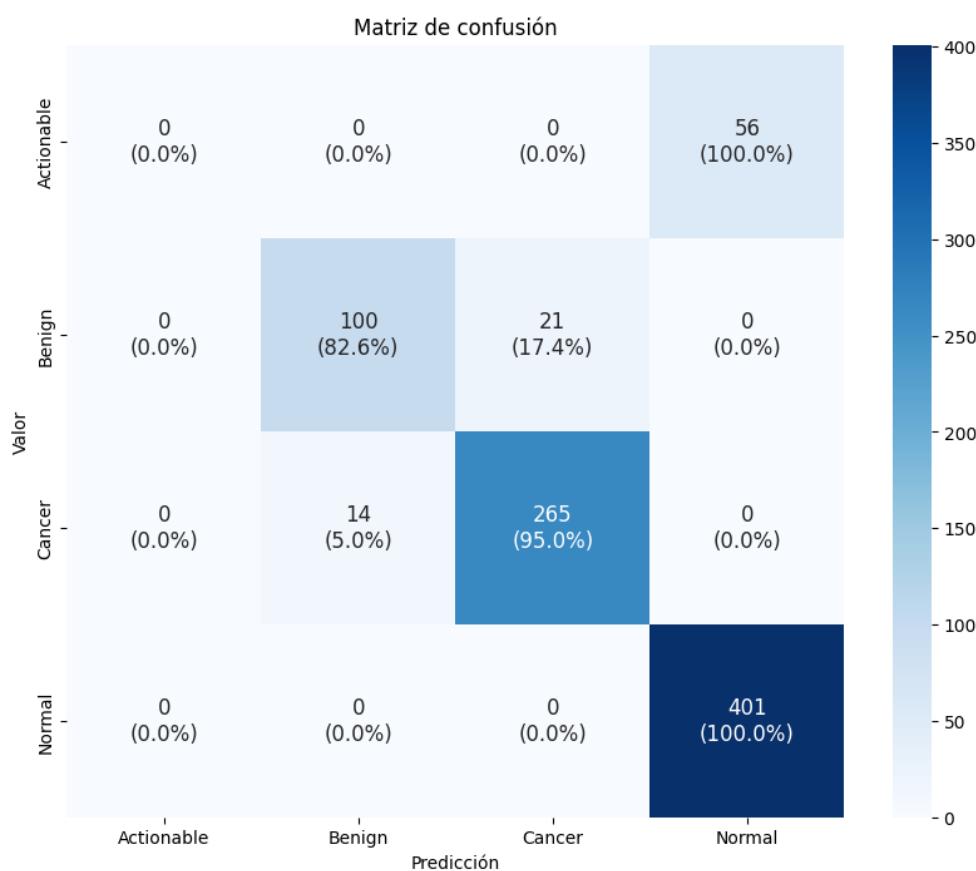


Figura 8.45: Matriz de confusión del modelo “best\_model5.keras”.

En primer lugar, todos los casos *actionables* se clasifican como *normal*. Esto es algo normal y que ya hemos visto en el modelo Inception v3 y que en la función de pérdida esta categoría contaba con un peso cero. El hecho de que se clasifiquen como *normal* nos muestra que estos casos son más parecidos a un paciente sano que a un paciente con un tumor benigno o con cáncer, al menos en las mamografías.

#### 8.4. RED RESNETV2-PARALELA

---

Lo siguiente que se puede observar es la capacidad del modelo en diferenciar los casos de pacientes sanos de aquellos que no lo están. Esto lo vemos debido a que todos los casos de la categoría *normal* se han clasificado correctamente y que ninguno de los casos *benign* o *cancer* se han clasificado como normal.

Por último, las categorías *benign* y *cancer* muestran un rendimiento notable, con una precisión del 82,6% y 95,0% respectivamente. La categoría *benign* muestra un peor resultado, probablemente debido a que cuenta con un la mitad de casos que la categoría *cancer*. Aun así el resultado es notablemente es bueno para todas las categorías.

A continuación, vamos a ver el rendimiento del modelo sin la categoría *actionable*. Cómo hemos mencionado antes esta categoría solo se dejó para poder ver la interpretación que hacían los diferentes modelos durante el entrenamiento. Tal y como vemos en la tabla 8.20, al quitar los estos casos ha subido la pérdida y *accuracy*. La subida de la pérdida indica que los casos *actionables* eran categorizados con un porcentaje de confianza superior a la media. El *accuracy* sube debido a que ahora no está fallando en estos casos. Con todo esto el rendimiento del modelo muestra una pérdida media de 12,53 y un *accuracy* media del 92,42%.

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,0934	0,8539
best_model1.keras	0,0916	0,8702
model_fold_2_50_epochs.keras	0,1499	0,9313
best_model2.keras	0,1447	0,9363
model_fold_3_50_epochs.keras	0,1268	0,9426
best_model3.keras	0,1241	0,9413
model_fold_4_50_epochs.keras	0,1449	0,9101
best_model4.keras	0,1271	0,9463
model_fold_5_50_epochs.keras	0,1302	0,9538
best_model5.keras	0,1202	0,9563
<b>Media aritmética</b>	<b>0,1253</b>	<b>0,9242</b>

Tabla 8.20: Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase *actionable*.

### Pruebas con casos normales ampliados

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,0201	0,9687
best_model1.keras	0,0193	0,9730
model_fold_2_50_epochs.keras	0,0315	0,9852
best_model2.keras	0,0306	0,9860
model_fold_3_50_epochs.keras	0,0271	0,9875
best_model3.keras	0,0270	0,9865
model_fold_4_50_epochs.keras	0,0333	0,9773
best_model4.keras	0,0290	0,9880
model_fold_5_50_epochs.keras	0,0285	0,9898
best_model5.keras	0,0255	0,9906
<b>Media aritmética</b>	<b>0,0272</b>	<b>0,9833</b>

Tabla 8.21: Resultados de los modelos evaluados en el conjunto de datos de prueba sin la clase *actionable* y con datos ampliados para la clase *normal*.

No se han cometido errores al diferenciar entre los casos de la categoría *normal* y el resto, pero un rendimiento del 100% no es creíble. Además, en un uso real de exámenes periódicos, la cantidad de casos de pacientes sanos es mucho mayor que de pacientes con un tumor o un cáncer. Es por estas dos razones que vamos a añadir todos los casos de la categoría normal que se desecharon y no fueron utilizados en el entrenamiento. Con esto conseguiremos medir de una forma mucho más fiel el rendimiento del modelo.

Los resultados obtenidos en las pruebas se muestran en la tabla 8.21. Se puede observar que la pérdida ha bajado hasta una media de 0,0272 y el *accuracy* ha aumentado hasta un valor medio del 98,33 %.

De forma mas detallada, la matriz de confusión representada en la figura 8.46 muestra como el modelo “best\_model5.keras” comete dos errores para los casos *normal*, bajando el *accuracy* a 99,9 %. El resto da categorías no cambian, debido a que no se han modificado sus datos. Con esto, obtenemos el rendimiento final de esta arquitectura con el conjunto de datos base.

## 8.4. RED RESNETV2-PARALELA

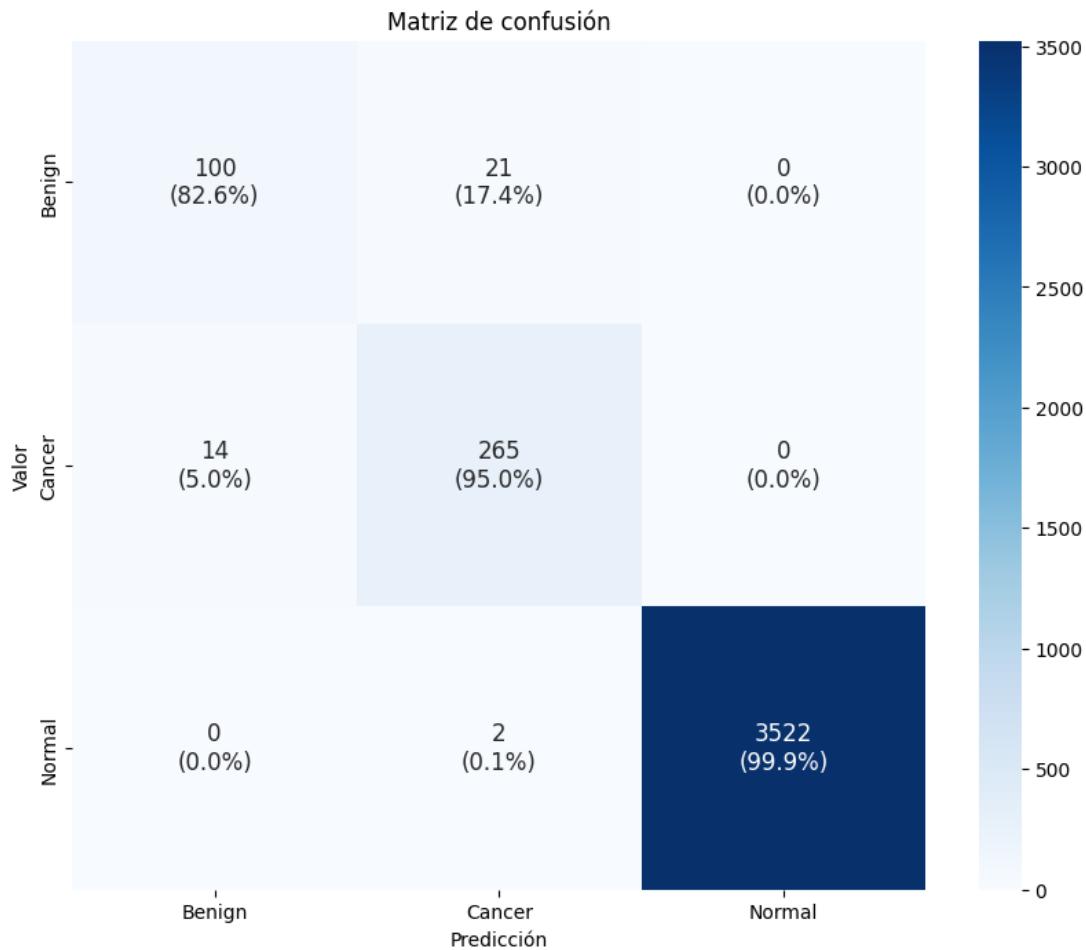


Figura 8.46: Matriz de confusión del modelo “best\_model5.keras” sin la categoría *actionable* y con datos ampliados para la clase *normal*.

### 8.4.2. Conjunto de datos con aumento de datos

Vamos a presentar los resultados del entrenamiento utilizando el conjunto de datos que emplea la técnica de aumento de datos (DA). Cabe recordar que en este conjunto se han duplicado los casos de la categoría “benign” con el objetivo de mejorar la precisión en esta clase. El entrenamiento tomó alrededor de 5 horas y media, en las cuales se ejecutaron cinco *folds* con 50 *epochs* cada uno. Este incremento en la duración del entrenamiento se debe al aumento de datos en la clase “benign”.

Primero revisaremos las métricas utilizadas durante este proceso. En el gráfico representado por la figura 8.47 se muestra la evolución de la pérdida durante el

## CAPÍTULO 8. RESULTADOS

entrenamiento. Como podemos observar, la pérdida baja muy rápidamente a valores bajos, después de esto continua descendiendo de forma muy leve y estable. La pérdida de validación se mantiene cercana a la de entrenamiento, mostrando que el modelo no se está sobreajustando. Igual que en el anterior conjunto de datos, el modelo muestra unos niveles de pérdida que demuestran una alta confianza en sus predicciones.

De igual forma, la figura 8.48 representa la evolución del *accuracy*. Se manifiesta un aumento progresivo de esta métrica hasta el segundo *fold*, donde se estabiliza en valores comprendidos entre el 90 % y el 95 %. No hay signos de sobre ajuste, debido a que los valores de *accuracy* con los datos de entrenamiento se mantienen cercanos a los obtenidos con los datos de validación.

Finalmente, la evolución de la precisión se contempla en la figura 8.49. La evolución de la precisión sigue una representación y características similares la evolución del *accuracy*, es por esto que tampoco hay muestras de sobreajuste y logra dar unos resultados bastante buenos.

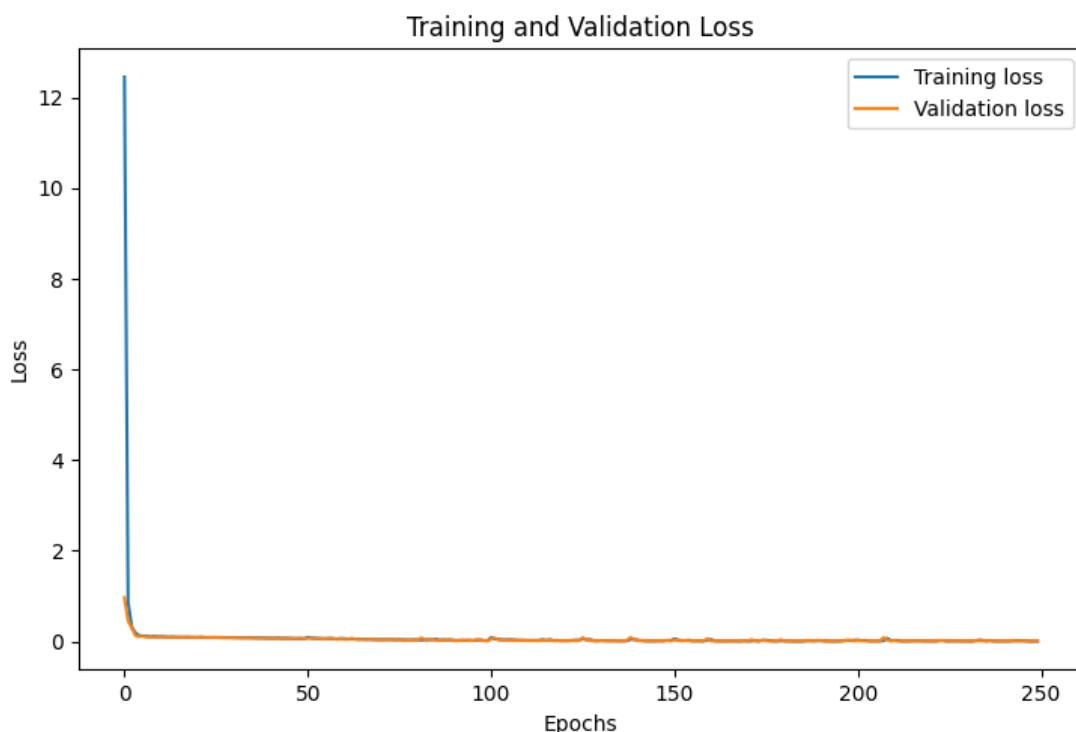


Figura 8.47: Pérdida durante el entrenamiento.

#### 8.4. RED RESNETV2-PARALELA

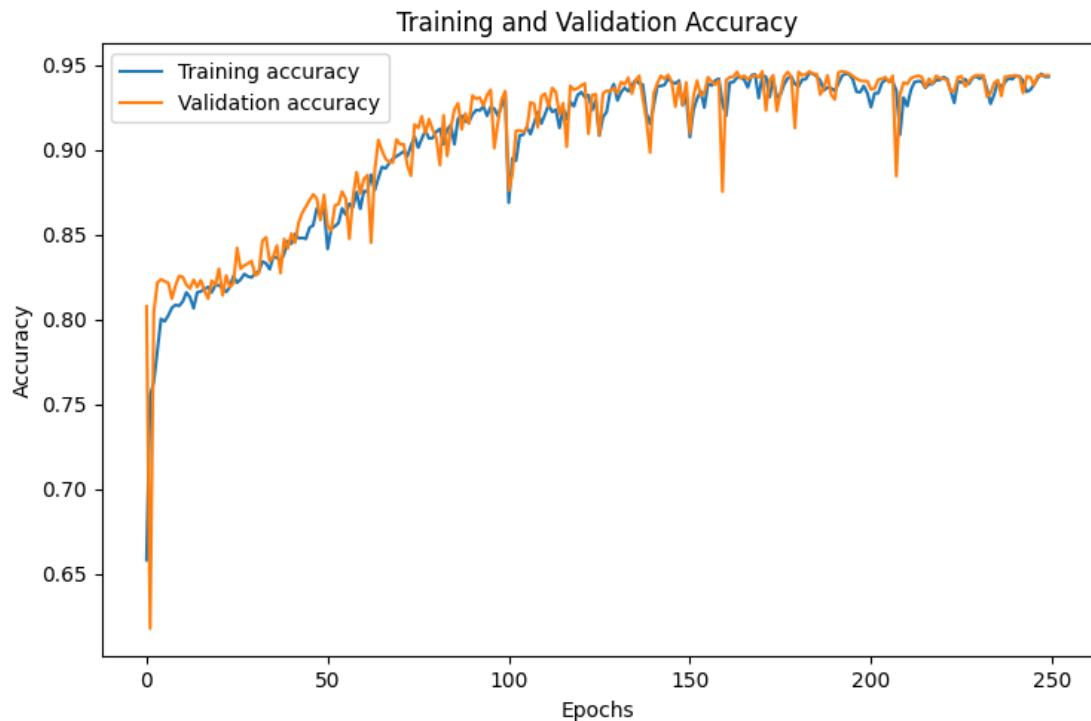


Figura 8.48: *Accuracy* durante el entrenamiento.

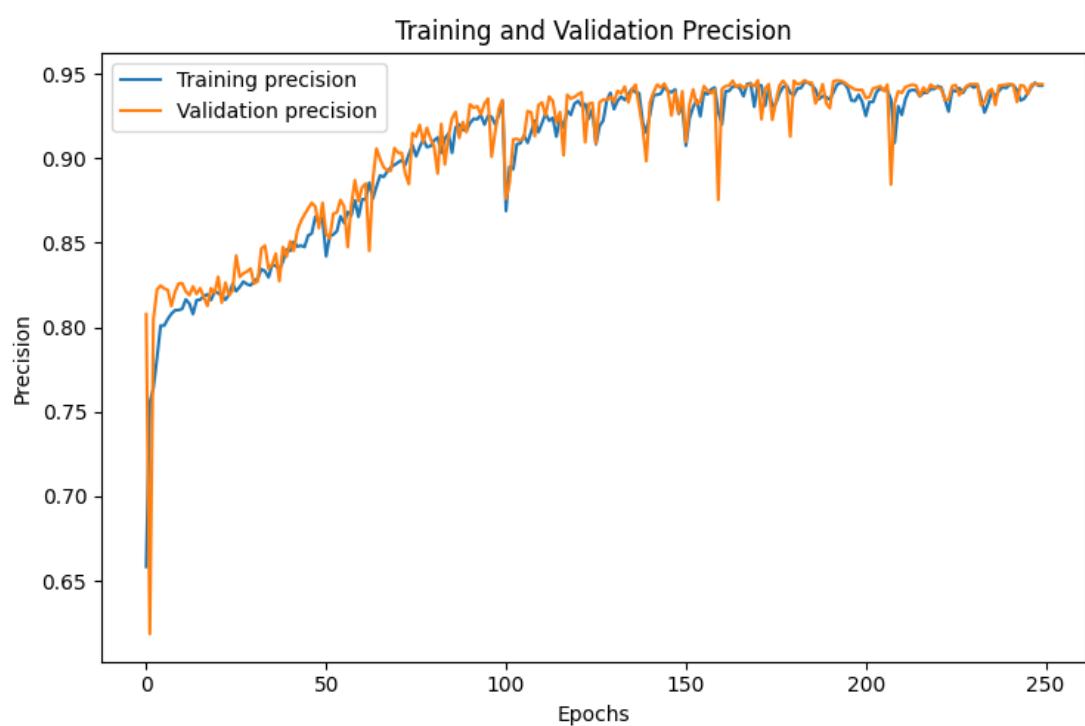


Figura 8.49: Precisión durante el entrenamiento.

## CAPÍTULO 8. RESULTADOS

A primera vista, los resultados son mejores a los obtenidos en el anterior conjunto de datos. Probablemente, incluir mas datos de la categoría *benign* haya dado resultados beneficiosos, tal y como dio con la anterior red Inception v3. Esto lo constataremos en la siguiente sección.

### Pruebas del modelo

Igual que en el anterior conjunto, para poder medir el rendimiento de un modelo con validación cruzada, se han guardado dos modelos por cada *fold*. Donde uno representa el modelo con mayor *accuracy* en los datos de validación, mientras que el otro es el modelo al final de cada *fold*.

En la figura 8.50 se muestra la pérdida de los diferentes modelos guardados y en la figura 8.44 presenta el *accuracy* de estos modelos. Además podemos ver los resultados numéricos de estas pruebas en la tabla 8.22. Con estos resultados podemos analizar cómo se comporta cada uno de los modelos con el conjunto de datos de prueba y calcular el rendimiento obtenido.

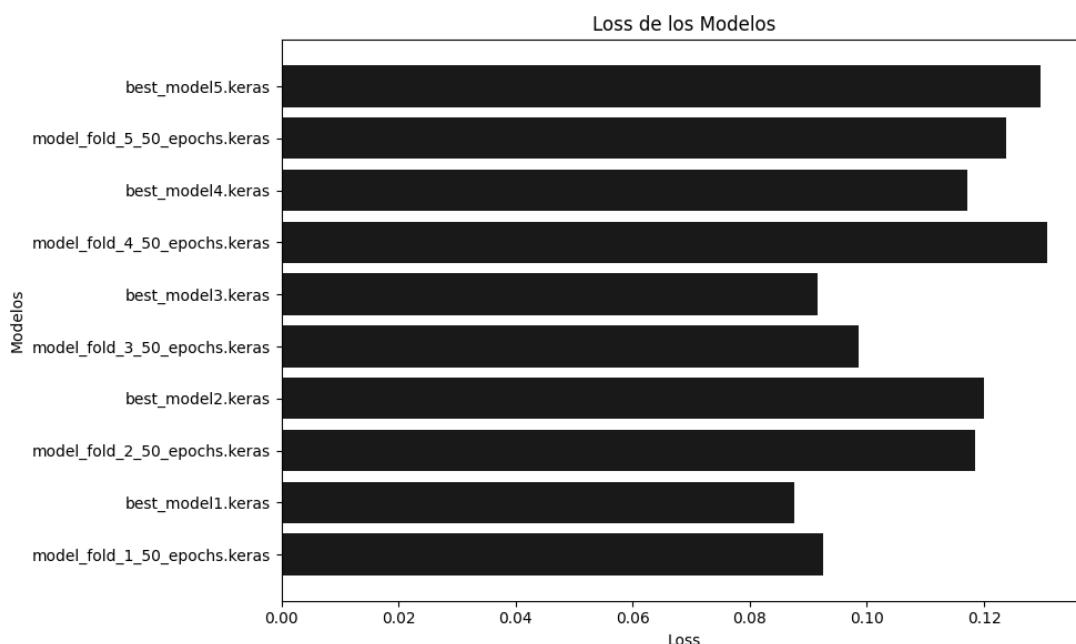


Figura 8.50: Pérdida de los diferentes modelos

#### 8.4. RED RESNETV2-PARALELA

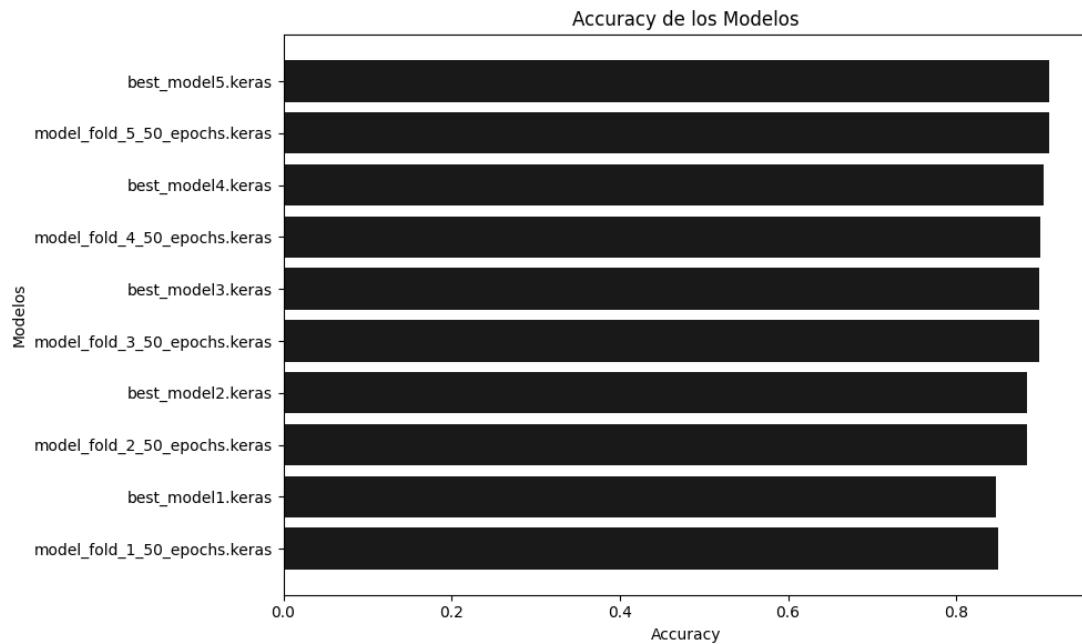


Figura 8.51: *Accuracy* de los diferentes modelos

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,0918	0,8508
best_model1.keras	0,0869	0,8480
model_fold_2_50_epochs.keras	0,1175	0,8844
best_model2.keras	0,1190	0,8844
model_fold_3_50_epochs.keras	0,0979	0,8999
best_model3.keras	0,0908	0,8990
model_fold_4_50_epochs.keras	0,1299	0,9008
best_model4.keras	0,1162	0,9045
model_fold_5_50_epochs.keras	0,1229	0,9117
best_model5.keras	0,1287	0,9117
<b>Media aritmética</b>	<b>0,1101</b>	<b>0,8895</b>

Tabla 8.22: Resultados de los modelos evaluados en el conjunto de datos de prueba.

Los resultados obtenidos muestran una pérdida media de 0,1101 y un *accuracy* del 88,95 %. Por tanto, este conjunto de datos ha dado un resultado superior al conjunto de datos base, obteniendo alrededor de un 3 % más de *accuracy*. Además, se consiguen resultados muy estables con un *accuracy* que aumenta de forma lenta pero constante. En el caso de la pérdida, esta se mantiene por encima de 0,08 y por debajo de 0,13, teniendo sus máximos valores después del cuarto *fold*.

## CAPÍTULO 8. RESULTADOS

Con la matriz de confusión mostrada en la figura 8.52 podemos ver el rendimiento de forma más detallada. Esta matriz de confusión vislumbra los resultados del modelo “model\_fold\_5\_50\_epochs.keras” con los datos de *test*. Se eligió este modelo al contar con un *accuracy* empatado al modelo “best\_model5.keras” pero con menor pérdida.

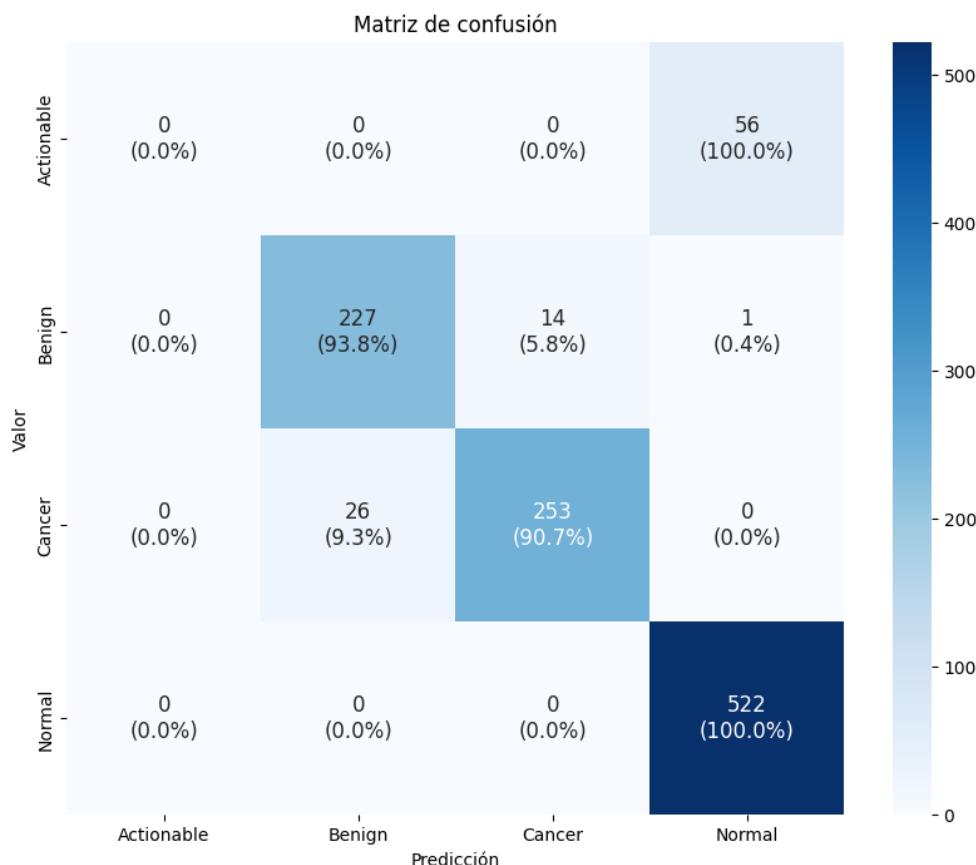


Figura 8.52: Matriz de confusión del modelo “model\_fold\_5\_50\_epochs.keras”.

Lo primero que podemos ver es que vuelve a fallar en todos los casos *actionable*, clasificándolos como *normal*, lo cual ya era de esperar en vista del rendimiento de los resultados obtenidos en los modelos anteriores.

Lo segundo que más destaca es la precisión del 100% al clasificar los casos de la categoría *normal*. El modelo es perfectamente capaz de detectar a los pacientes sanos y diferenciarlos de aquellos que se encuentran en las categorías *benign* o *cancer*. En este enfoque solo comete un error, con un caso *benign* que es clasificado erróneamente como *normal*.

#### 8.4. RED RESNETV2-PARALELA

---

Finalmente las categorías *benign* y *cancer* consiguen un alto rendimiento, del 93,4% y el 91% respectivamente. Para este conjunto de datos, la precisión en los casos *benign* llega incluso a superar a la de los casos *cancer*. Esto constata que el aumento de datos está ayudando de forma considerable a clasificar los distintos casos de forma óptima.

Para ver un rendimiento más realista vamos a volver a correr las pruebas sin los casos *actionable*. Como hemos hecho anteriormente esta categoría solo estaba incluida para sacar conclusiones en la fase de entrenamiento. Estos resultados se observan en la tabla 8.23, en la cual, sin los casos *actionable*, ha subido el valor medio de la pérdida y de el *accuracy* consiguiendo un 0,1160 y un 93,73% respectivamente. La subida de la pérdida, aunque es leve, indica que los casos *actionables* eran categorizados con un porcentaje de confianza superior a la media. El *accuracy* sube debido a que ahora no está fallando en estos casos.

<b>Modelo</b>	<b>Pérdida</b>	<b>Accuracy</b>
model_fold_1_50_epochs.keras	0,0968	0,8965
best_model1.keras	0,0915	0,8936
model_fold_2_50_epochs.keras	0,1238	0,9319
best_model2.keras	0,1253	0,9319
model_fold_3_50_epochs.keras	0,1031	0,9482
best_model3.keras	0,0956	0,9473
model_fold_4_50_epochs.keras	0,1367	0,9492
best_model4.keras	0,1224	0,9530
model_fold_5_50_epochs.keras	0,1294	0,9607
best_model5.keras	0,1356	0,9607
<b>Media aritmética</b>	<b>0,1160</b>	<b>0,9373</b>

Tabla 8.23: Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase *actionable*.

#### Pruebas con casos normales ampliados

Para ver el rendimiento final de esta red ResNet v2, vamos a incluir todos los casos de la categoría *normal* que no han sido utilizados en durante el entrenamiento. De esta manera, conseguiremos una proporción más parecida a la presentada en los exámenes de detección.

## CAPÍTULO 8. RESULTADOS

---

Modelo	Pérdida	Accuracy
model_fold_1_50_epochs.keras	0,0289	0,9688
best_model1.keras	0,0276	0,9677
model_fold_2_50_epochs.keras	0,0378	0,9784
best_model2.keras	0,0372	0,9795
model_fold_3_50_epochs.keras	0,0309	0,9843
best_model3.keras	0,0289	0,9840
model_fold_4_50_epochs.keras	0,0407	0,9845
best_model4.keras	0,0373	0,9854
model_fold_5_50_epochs.keras	0,0394	0,9874
best_model5.keras	0,0412	0,9876
<b>Media aritmética</b>	<b>0,0350</b>	<b>0,9808</b>

Tabla 8.24: Resultados de los modelos evaluados en el conjunto de datos de prueba con DA sin la clase *actionable* y con datos ampliados para la clase *normal*.

En la tabla 8.24 se presentan los resultados obtenidos, donde se observa una disminución significativa en la pérdida, alcanzando un valor promedio de 0,0350. Simultáneamente, la precisión (*accuracy*) ha mejorado, llegando a un valor medio del 98,08 %.

Para profundizar en los resultados finales, se realizará un análisis detallado de las predicciones generadas por el modelo `model_fold_5_50_epochs.keras` utilizando este conjunto de datos de prueba. Los resultados de este análisis se ilustran en la matriz de confusión que se muestra en la figura 8.53.

Como podemos observar, el modelo ha cometido errores en cuatro casos de la categoría *normal*, clasificándolos incorrectamente como *cancer*. Como resultado, la precisión para la categoría *normal* se sitúa ahora en un 99,9 %, lo que proporciona una medida más precisa y realista de su rendimiento.

En cuanto a las otras categorías, *benign* y *cancer*, sus resultados se mantienen consistentes con los obtenidos en las pruebas anteriores, dado que los datos correspondientes a estas clases no han sufrido modificaciones.

Con este análisis, hemos concluido la evaluación del rendimiento de la última red neuronal empleada en este estudio, considerando sus dos conjuntos de datos. Los resultados obtenidos proporcionan una visión clara de cómo este modelo, en

## 8.5. RESUMEN DE RESULTADOS

particular, se comporta bajo diferentes condiciones de entrenamiento y validación, permitiéndonos entender mejor sus fortalezas y limitaciones en comparación con otros enfoques.

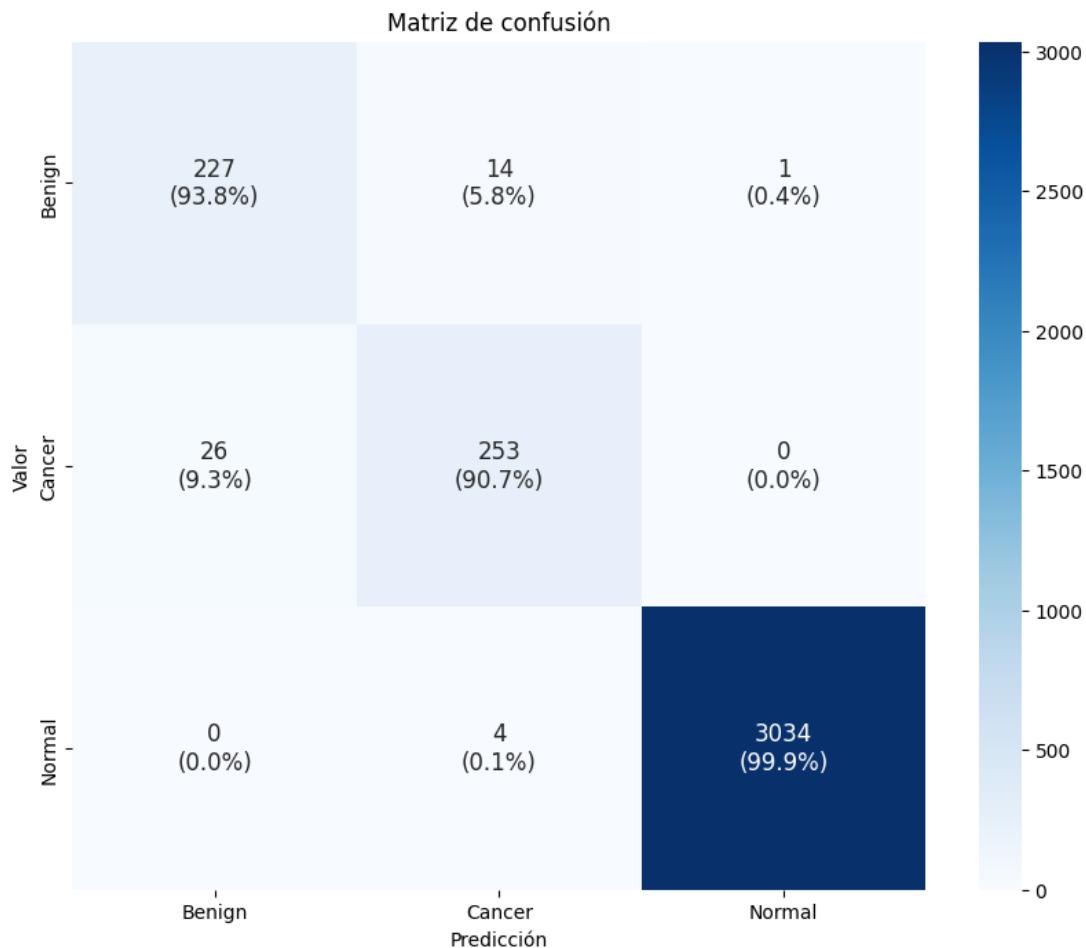


Figura 8.53: Matriz de confusión del modelo “model\_fold\_5\_50\_epochs.keras” sin la categoría *actionable* y con datos ampliados para la clase *normal*.

## 8.5. Resumen de resultados

Para facilitar la comparación de los resultados obtenidos de las diferentes redes neuronales implementadas, se presenta una síntesis de los mismos en la tabla 8.25. Esta tabla ofrece un resumen detallado de los desempeños alcanzados por cada modelo. En particular, la columna titulada “Precisión *normal (%)*” muestra el

## CAPÍTULO 8. RESULTADOS

---

porcentaje de precisión de cada modelo en la identificación de casos que pertenecen a la clase *normal*. Es importante destacar que en esta métrica se incluye tanto los verdaderos positivos como aquellos casos de las categorías *benign* y *cancer* que fueron incorrectamente clasificados como *normal*. Por otro lado, la columna “Precisión *benign* y *cancer* (%)” presenta la precisión lograda por cada modelo al identificar correctamente los casos pertenecientes a las clases *benign* y *cancer*.

Red Neuronal	Datos base	Aumento de datos	Precisión <i>normal</i> (%)	Precisión <i>benign</i> y <i>cancer</i> (%)
<b>UNet3D Densa</b>	Pérdida: 0,0345 Accuracy: 97,02	Pérdida: 0,0945 Accuracy: 94,43	Base: 99,69 DA: 99,67	Base: 86,18 DA: 65,96
<b>UNet Paralela</b>	Pérdida: 0,0385 Accuracy: 96,76	Pérdida: 0,0641 Accuracy: 94,56	Base: 99,72 DA: 99,44	Base: 72,68 DA: 73,26
<b>InceptionV3 Paralela</b>	Pérdida: 0,0286 Accuracy: 98,87	Pérdida: 0,0238 Accuracy: 98,83	Base: 99,97 DA: 99,87	Base: 91,75 DA: 94,81
<b>ResNetV2 Paralela</b>	Pérdida: 0,0272 Accuracy: 98,33	Pérdida: 0,0350 Accuracy: 98,08	Base: 99,94 DA: 99,87	Base: 91,25 DA: 92,31

Tabla 8.25: Comparación de desempeño de las diferentes redes neuronales con los dos conjuntos de datos, con la precisión de la clase *normal* y la precisión en las clases *benign* y *cancer*.

Vamos a empezar por analizar las dos primeras columnas de la tabla, “Datos base” y “Aumento de datos”, las cuales representan las métricas obtenidas por las diferentes redes neuronales en los dos conjuntos de datos utilizados durante la fase de prueba.

A primera vista, la red *InceptionV3-Paralela* demuestra un rendimiento superior en comparación con las otras arquitecturas evaluadas, alcanzando una pérdida de 0,0286 y una precisión (*accuracy*) del 98,87% en el conjunto de datos base. Este desempeño se mantiene robusto incluso cuando se utiliza un conjunto de datos con técnicas de aumento de datos, lo que sugiere que esta red es altamente eficaz en la detección de tumores bajo diversas condiciones de entrenamiento, destacando por su capacidad de generalización.

En segundo lugar se encuentra la red *ResNetV2-Paralela*, que aunque ligeramente

## 8.5. RESUMEN DE RESULTADOS

inferior a *InceptionV3*, ofrece un desempeño igualmente notable con una pérdida de 0,0272 y una precisión del 98,33% en el conjunto de datos base. La pequeña disminución en el rendimiento observada al aplicar técnicas de aumento de datos es un aspecto digno de mayor investigación, y será comentada en el apartado de desarrollos futuros.

Las redes basadas en la arquitectura *UNet*, como *UNet3D-Densa* y *UNet-Paralela*, muestran un rendimiento competitivo, lo cual es particularmente destacable dado que no emplearon técnicas de *transfer learning*<sup>2</sup>. Aunque no alcanzaron los niveles de precisión obtenidos por *InceptionV3* y *ResNetV2*, el hecho de que *UNet3D-Densa*, por ejemplo, haya logrado una precisión del 97,02% con una arquitectura con menos parámetros y datos de entrenamiento, subraya su potencial en aplicaciones donde los recursos computacionales son limitados.

Al examinar las dos últimas columnas, se confirma el análisis previo, con una observación adicional. La red neuronal *UNet-Paralela* supera a *UNet3D-Densa* en el conjunto de datos base en la columna “Precisión *normal (%)*”. Esto indica que la *UNet-Paralela* ha logrado una mejor identificación de las diferencias entre los casos clasificados como *normal* y aquellos que pertenecen a las demás categorías.

Sin embargo, aunque la *UNet-Paralela* demuestra un mejor desempeño en la clasificación de la clase *normal*, la columna “Precisión *benign* y *cancer (%)*” revela la razón por la cual su exactitud global (*accuracy*) es inferior. La red muestra un rendimiento menos eficiente al diferenciar entre los casos *benign* y *cancer*, lo que resulta en una precisión menor en estas categorías.

En el siguiente capítulo, procederemos a discutir estos resultados de manera exhaustiva. Esta discusión incluirá una comparación detallada con los resultados obtenidos a partir de otros modelos analizados a lo largo de este trabajo. Asimismo, contrastaremos los hallazgos actuales con los antecedentes presentados en capítulos anteriores, lo que nos permitirá situar nuestros resultados en el contexto de la literatura existente y evaluar cómo se alinean o difieren de estudios previos.

<sup>2</sup>*Transfer learning*: Técnica que permite utilizar un modelo previamente entrenado en un problema similar, aprovechando el conocimiento adquirido para mejorar el rendimiento en un nuevo problema.

# **Capítulo 9**

## **Conclusiones**

En este capítulo se exponen las conclusiones generales del proyecto una vez analizados los resultados obtenidos en el capítulo anterior. Este análisis permitirá identificar las fortalezas y debilidades de cada enfoque, proporcionando una visión global del progreso logrado.

Adicionalmente, se presentarán reflexiones personales sobre el trabajo realizado, destacando las lecciones aprendidas y las áreas en las que se podría haber mejorado el enfoque. Estas reflexiones ofrecerán una perspectiva crítica y autocrítica que contribuirá a la comprensión profunda del proceso de desarrollo.

Finalmente, se propondrán una serie de desarrollos futuros, orientados a complementar o perfeccionar el proyecto actual. Estas propuestas estarán enfocadas en superar las limitaciones identificadas, optimizar los resultados obtenidos, y explorar nuevas direcciones de investigación.

### **9.1. Conclusiones del proyecto**

El objetivo principal de este trabajo de fin de grado fue evaluar la viabilidad de utilizar técnicas de *deep learning* para la predicción del cáncer de mama, así como investigar si es posible mejorar el rendimiento de estos sistemas mediante distintas arquitecturas y estrategias de entrenamiento.

## 9.1. CONCLUSIONES DEL PROYECTO

---

Los resultados obtenidos permiten extraer varias conclusiones clave:

- **Impacto del *transfer learning*:** Las redes *InceptionV3* y *ResNetV2*, ambas beneficiadas por el uso de *transfer learning*, sobresalieron significativamente en términos de precisión. Esto pone de manifiesto la utilidad de estas técnicas en la detección temprana del cáncer, ya que permiten aprovechar el conocimiento preexistente de redes entrenadas en conjuntos de datos masivos, no necesariamente médicos, lo que incrementa la capacidad del modelo para generalizar en el ámbito clínico.
- **Desempeño de arquitecturas sin *transfer learning*:** A pesar de no haber utilizado *transfer learning*, las redes basadas en *U-Net* demostraron un rendimiento notable. Este hecho resalta la adaptabilidad y robustez de estas arquitecturas en tareas de segmentación y clasificación médica, especialmente en escenarios con restricciones de hardware. Además, su cercanía en rendimiento a las arquitecturas más avanzadas sugiere que, con optimizaciones adicionales o un mayor volumen de datos, podrían competir directamente en términos de precisión y eficiencia.
- **Efectividad del aumento de datos:** La aplicación de técnicas de aumento de datos ha mostrado efectos variados. Mientras que *InceptionV3-Paralela* y *ResNetV2-Paralela* mantuvieron un alto rendimiento, otras redes experimentaron una ligera disminución en la precisión. Esto sugiere que, aunque el aumento de datos es generalmente beneficioso, en este caso no ha dado mejores resultados.

### 9.1.1. Comparación con el modelo de referencia

En el capítulo 7, *Desarrollo del proyecto*, se utilizó un modelo básico *InceptionV3* como punto de referencia, el cual presentó resultados modestos, con una pérdida de 0,7103 y una precisión (*accuracy*) del 60,83 %. Este modelo inicial sirvió como base comparativa. Durante el desarrollo de este trabajo, se implementaron mejoras

## CAPÍTULO 9. CONCLUSIONES

significativas en el diseño de las redes neuronales, superando ampliamente los resultados del modelo de referencia. La creación de arquitecturas personalizadas, como las redes paralelas y las redes tridimensionales en *UNet3D-Densa*, fue clave para este avance, adaptándose específicamente al desafío de la clasificación de imágenes médicas.

Además, el diseño y ajuste cuidadoso de las capas de cada modelo, junto con la selección óptima de hiperparámetros de cada capa, jugaron un papel crucial en la mejora del rendimiento. Un desafío constante fue la lucha contra el sobreajuste, que se mitigó mediante técnicas de regularización y validación cruzada.

Otro aspecto clave fue la identificación y superación de los máximos locales<sup>1</sup>, que en varios puntos dificultaron el progreso del aprendizaje. Para mitigar este problema, se ajustó la complejidad de la red mediante la modificación de capas y la adición de hiperparámetros adicionales.

Gracias a estas mejoras y esfuerzos de diseño, los resultados finales superaron significativamente los del modelo de referencia inicial, demostrando la efectividad de las estrategias implementadas y estableciendo una base sólida para futuras investigaciones y aplicaciones en la detección temprana de cáncer de mama.

### 9.1.2. Comparativa con los sistemas CADe

Al comparar los resultados obtenidos con los sistemas CADe explicados en el capítulo 3, se observa que el rendimiento de las redes neuronales desarrolladas en este proyecto es competitivo y, en varios casos, superior. El sistema CADe con mayor rendimiento documentado, según Chu *Chu y cols. (2015)*, alcanzó una precisión del 98,5 %. Todas las redes implementadas en este proyecto, utilizando el conjunto de datos base, superan esta métrica, lo que indica una mejora significativa en la capacidad de detección.

Además, si consideramos los datos sobre el rendimiento de varios sistemas CADe,

<sup>1</sup>Máximos locales: Valores de una función en los que esta alcanza un máximo dentro de una región específica, pero que no representan el máximo global. En redes neuronales, estos pueden obstaculizar el proceso de optimización durante el entrenamiento.

## 9.2. CONCLUSIONES PERSONALES

---

encontramos que la precisión de dichos sistemas varía notablemente. Por ejemplo, [Chu y cols. \(2015\)](#) reportan una precisión del 98,5% utilizando datos públicos (DDSM). Por su parte, [Mudigonda y cols. \(2001\)](#) lograron un 81% utilizando también datos públicos, mientras que [Yang y cols. \(2007\)](#) alcanzaron entre un 95% y un 98% con datos privados.

Comparando estos resultados con los obtenidos en este proyecto:

- **InceptionV3-Paralela** obtuvo una precisión del 98,87% en el conjunto de datos base, superando claramente a todos los sistemas CADe mencionados en la tabla.
- **ResNetV2-Paralela** logró una precisión del 98,33%, también superior a los sistemas CADe.
- **UNet3D-Densa** alcanzó una precisión del 97,02%, y la red **UNet-Paralela** obtuvo un 96,76%, ambas superando ampliamente los resultados de la mayoría de los sistemas CADe.

Es importante destacar, como se menciona en este documento, la inexistencia de una base de datos abierta y estandarizada para la evaluación de estos sistemas. Las comparaciones realizadas deben tomarse con cautela, ya que cada sistema CADe se mide con conjuntos de datos diferentes, muchos de los cuales son privados, y no se tiene conocimiento sobre el desbalance de las clases o las categorías específicas manejadas en dichos conjuntos de datos. Sin embargo, los resultados obtenidos en este proyecto sugieren que las redes neuronales desarrolladas ofrecen un rendimiento superior en la tarea de detección del cáncer de mama, al menos dentro del marco de los datos utilizados en esta investigación.

## 9.2. Conclusiones Personales

La realización de este Trabajo de Fin de Grado ha supuesto un desafío significativo que ha implicado un esfuerzo constante y considerable a lo largo de todo el proceso. A diferencia de un enfoque centrado en una única arquitectura, en este trabajo

## CAPÍTULO 9. CONCLUSIONES

---

se han diseñado y desarrollado cuatro redes neuronales distintas, cada una con características y objetivos específicos. Inicialmente, el trabajo estaba programado para ser presentado en el mes de julio; sin embargo, debido a dificultades relacionadas con la obtención y manejo de datos, así como a la falta de tiempo disponible, se tomó la decisión de aplazar su presentación hasta septiembre. Este aplazamiento, aunque inesperado, permitió un análisis más profundo de los modelos desarrollados, así como la incorporación de una nueva arquitectura: la red neuronal *UNet-Paralela*.

La red *UNet-Paralela* fue concebida con la intención de adaptar el enfoque desarrollado para las redes Inception a una red del tipo U-Net. Sin embargo, los resultados obtenidos fueron peores en comparación con las otras redes evaluadas. Este resultado subrayó la eficacia del enfoque tridimensional empleado en la red *UNet3D-Densa*, demostrando que la arquitectura y el tratamiento de los datos en tres dimensiones eran más adecuados para el problema en cuestión. Este hallazgo no solo proporcionó una validación del enfoque elegido para la red *UNet3D-Densa*, sino que también ofreció una perspectiva crítica sobre la adaptación de ciertas técnicas a diferentes tipos de redes y problemas.

El tiempo adicional obtenido por el aplazamiento de la entrega también permitió dedicar más atención a la redacción de este documento, lo que resultó en una presentación más detallada y comprensible del trabajo realizado. De este modo, el proceso de escritura se benefició de un enfoque más reflexivo, posibilitando una mejor organización y explicación de los conceptos desarrollados.

El inicio del desarrollo de este trabajo se remonta al mes de septiembre, con una fase preliminar dedicada a la lectura exhaustiva de dos libros, *Python Deep Learning* de Torres (2020) y *Deep Learning with Python* de Chollet, Francois (2021). Estos textos proporcionaron una base teórica sólida para el diseño de las redes y sirvieron de guía en la comprensión de las técnicas y metodologías más avanzadas en el ámbito de la inteligencia artificial. Además, la búsqueda, procesamiento y adaptación de los datos adecuados para entrenar los modelos representaron etapas que requirieron una inversión de tiempo y esfuerzo equiparable al proceso de diseño de las diferentes redes.

### 9.3. DESARROLLOS FUTUROS

---

En términos generales, la realización de este trabajo me ha permitido explorar de manera exhaustiva y profunda el campo de las redes neuronales, una subárea de la inteligencia artificial que ha despertado mi interés desde siempre dentro de la informática. Este proyecto no solo ha mejorado mis conocimientos técnicos y prácticos en el diseño, implementación y evaluación de modelos de aprendizaje profundo, sino que también ha consolidado mi deseo de especializarme en este campo en el futuro. Mi aspiración profesional, ya sea en el ámbito de la investigación o en el sector empresarial, es seguir con el desarrollo de soluciones con el uso de técnicas de inteligencia artificial.

Este trabajo, en resumen, ha sido tanto un reto como una oportunidad de crecimiento personal y profesional. Las dificultades encontradas a lo largo del camino han fortalecido mi capacidad para abordar problemas complejos y han reafirmado mi pasión por el aprendizaje profundo y las redes neuronales. A través de la experimentación, la adaptación de enfoques y el análisis crítico de los resultados, he adquirido no solo habilidades técnicas avanzadas, sino también una comprensión más profunda de los desafíos y oportunidades que presenta el campo de la inteligencia artificial.

## 9.3. Desarrollos futuros

Los resultados obtenidos en este proyecto abren la puerta a múltiples líneas de investigación y desarrollo que podrían ampliar y mejorar significativamente los logros actuales. A continuación, se detallan algunas de las direcciones más prometedoras para futuras investigaciones.

### 9.3.1. Aumento de la cantidad de datos

Uno de los mayores limitantes en el desarrollo de modelos de *deep learning* para la detección de cáncer de mama ha sido, sin duda, la cantidad limitada de datos disponibles. Se han utilizado técnicas de aumento de datos (*data augmentation*)

## CAPÍTULO 9. CONCLUSIONES

para mitigar este problema sin mucho éxito. El acceso a nuevos conjuntos de datos, especialmente aquellos que están específicamente enfocados en esta tarea, podría mejorar considerablemente los resultados sin necesidad de realizar grandes modificaciones en los modelos actuales. En particular, la recopilación de datos que incluyan una mayor diversidad de casos, tanto en términos de demografía como de variabilidad en las imágenes, permitiría a los modelos aprender de manera más generalizada y reducir el riesgo de sobreajuste a un conjunto de datos específico.

### 9.3.2. Posibilidades de optimización futura

Existen diversas oportunidades para optimizar y mejorar aún más las redes desarrolladas. La optimización del proceso de aumento de datos y la exploración exhaustiva de nuevos hiperparámetros son dos áreas clave que podrían ser objeto de futuras investigaciones. Un enfoque más sistemático para ajustar estos hiperparámetros podría conducir a mejoras en la precisión y robustez de los modelos. Además, la implementación de técnicas de *ensemble*<sup>2</sup>, en las cuales se combinan los resultados de diferentes arquitecturas, podría aprovechar las fortalezas de cada modelo individual y compensar sus debilidades, llevando a un rendimiento superior en la tarea de clasificación.

### 9.3.3. Segmentación de imágenes

Hasta ahora, este trabajo de fin de grado se ha centrado exclusivamente en la tarea de clasificación de imágenes de mamografías. Sin embargo, los conjuntos de datos empleados contienen información suficiente no solo para clasificar las imágenes en distintas categorías, sino también para realizar una segmentación precisa, identificando y localizando las masas malignas dentro de las imágenes. La incorporación de técnicas de segmentación de imágenes en futuras investigaciones podría complementar los esfuerzos de clasificación, proporcionando un diagnóstico más detallado y visualmente

<sup>2</sup>*Ensemble*: Técnica que combina múltiples modelos para mejorar el rendimiento general mediante la reducción de la varianza, sesgo o errores de los modelos individuales.



### 9.3. DESARROLLOS FUTUROS

---

interpretativo. Este enfoque no solo permitiría identificar la presencia de masas malignas, sino también observar su ubicación y extensión dentro del tejido mamario, mejorando así la precisión del diagnóstico y la planificación del tratamiento.

## **Anexos**



# Apéndice A

## Contenido teórico

### A.1. Matrices de confusión

Las matrices de confusión presentadas en este documento son herramientas esenciales para evaluar el rendimiento de los modelos de clasificación, mostrando tanto las predicciones correctas como las incorrectas para cada clase del conjunto de datos. Estas matrices se estructuran en dos ejes que reflejan la comparación entre las etiquetas verdaderas y las predichas:

- **Eje Horizontal (*Predicción*):** Representa las etiquetas asignadas por el modelo para cada instancia. Este eje muestra cómo el modelo clasifica cada ejemplo en las distintas clases posibles.
- **Eje Vertical (*Valor*):** Representa las etiquetas verdaderas de las instancias en el conjunto de datos de prueba. Este eje indica la clase real a la que pertenece cada ejemplo, según el conjunto de datos anotado.

En la figura A.1, se presenta una matriz de confusión que ilustra la relación entre las predicciones del modelo y los valores reales para cuatro clases diferentes: *Actionable*, *Benign*, *Cancer* y *Normal*. Cada celda de la matriz contiene dos valores:

- El número absoluto de instancias que corresponden a cada combinación de etiqueta verdadera y predicción del modelo.

## A.1. MATRICES DE CONFUSIÓN

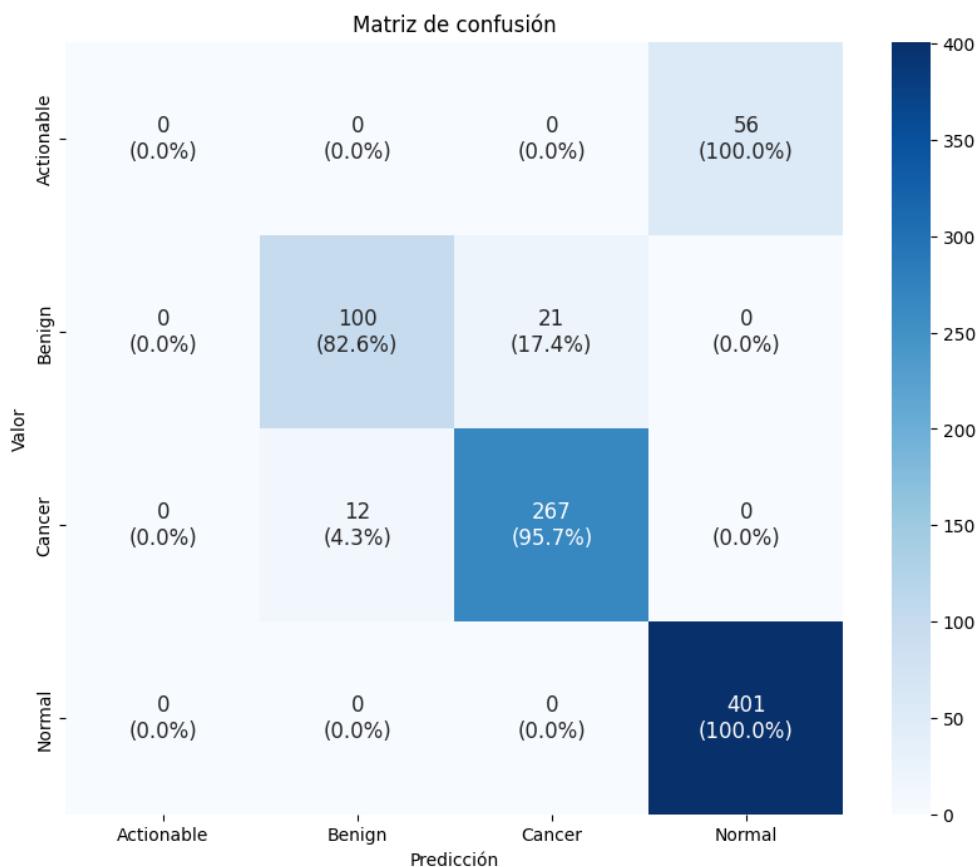


Figura A.1: Matriz de confusión de ejemplo.

- El porcentaje correspondiente al total de instancias reales de esa clase, que indica la proporción de ejemplos clasificados correctamente o incorrectamente.

Por ejemplo, la celda ubicada en la intersección de la fila correspondiente a la clase *Cancer* y la columna correspondiente a la predicción *Cancer* (celda 3,3), muestra que de 279 instancias reales de cáncer, 267 fueron correctamente clasificadas como *Cancer* por el modelo, lo que representa un 95.7% de precisión en esta clase. En contraste, la celda (3,2) muestra que 12 instancias de cáncer fueron incorrectamente clasificadas como *Benign*, lo que equivale a un 4.3% de error en la predicción de esta clase.

Las diagonales principales de la matriz (de la esquina superior izquierda a la esquina inferior derecha) representan las predicciones correctas para cada clase, mientras que las celdas fuera de la diagonal indican errores de clasificación. En el ejemplo presentado, se observa que las clases *Actionable* y *Normal* tienen una

## APÉNDICE A. CONTENIDO TEÓRICO

---

precisión del 100 %, mientras que las clases *Benign* y *Cancer* presentan ciertos grados de confusión, evidenciados por los valores fuera de la diagonal.

Este análisis es crucial para identificar patrones de errores y comprender las fortalezas y debilidades del modelo en la clasificación de distintas categorías, lo que a su vez permite orientar los esfuerzos de mejora del sistema.



### A.1. MATRICES DE CONFUSIÓN

---

# Apéndice B

## Código

### B.1. Weighted categorical crossentropy

Código *Python* utilizado en la función de pérdida:

```
def weighted_categorical_crossentropy(weights):
    weights = tf.constant(weights, dtype=tf.float32)
    def loss(y_true, y_pred):
        y_true = tf.cast(y_true, tf.float32)
        y_pred = tf.cast(y_pred, tf.float32)
        cross_entropy = categorical_crossentropy(y_true, y_pred)
        weights_tensor = tf.reduce_sum(weights * y_true, axis=-1)
        weighted_loss = cross_entropy * weights_tensor
        return tf.reduce_mean(weighted_loss)

    return loss
```

Vamos a ver la explicación de este código a continuación:

La función toma como entrada un *array* de pesos con cuatro valores, uno para cada categoría. Estos pesos se convierten a un tensor utilizando la función `tf.constant` y asegurando que los valores sean de tipo `float32`. Dentro se define la función de pérdida `loss(y_true, y_pred)`, que será utilizada durante el entrenamiento del

## B.2. VALIDACIÓN CRUZADA

---

modelo. Esta función toma como argumentos las etiquetas verdaderas (`y_true`) y las predicciones del modelo (`y_pred`).

Para asegurar la compatibilidad con las operaciones posteriores, tanto `y_true` como `y_pred` se convierten a tensores de tipo `float32` utilizando la función `tf.cast`. Luego, se calcula la entropía cruzada categórica entre las etiquetas verdaderas y las predicciones mediante la función `categorical_crossentropy(y_true, y_pred)`. Esta métrica mide la discrepancia entre la distribución real y la predicha para cada clase.

El siguiente paso es calcular un tensor de pesos sumando los productos de los pesos y las etiquetas verdaderas a lo largo del eje de las clases, lo que da como resultado un valor de peso para cada instancia en el lote de datos. Luego, la pérdida de entropía cruzada se multiplica por este tensor de pesos, ajustando la contribución de cada instancia a la pérdida total según su clase y el peso asignado a esa clase. Finalmente, se devuelve el promedio de la pérdida ponderada utilizando `tf.reduce_mean(weighted_loss)`, proporcionando una única medida de error para el lote de datos.

## B.2. Validación cruzada

El código *Python* expuesto a continuación ayudará a comprender de primera mano la creación de los directorios de validación cruzada y el del conjunto de datos de prueba. Este código se ejecuta sobre cada una de las categorías de datos, en este caso es para la categoría *actionable*. Esto se hizo de la misma manera para las otras 3 categorías.

```
input_directory = "E:/Numpys/Array/TodasBenignMalignant/Con DA/Actionable"  
output_directory = "E:/Numpys/Array/TodasBenignMalignant/Con DA/Division"  
  
for division in ["Training", "Validation", "Test"]:  
    division_dir = os.path.join(output_directory, division)
```

## APÉNDICE B. CÓDIGO

---

```
if not os.path.exists(division_dir):
    os.makedirs(division_dir)

npy_files = [f for f in os.listdir(input_directory) if f.endswith('.npy')]
image_prefixes = []
for npy_file in npy_files:
    image_prefixes.append(npy_file)

num_prefixes = len(image_prefixes)
num_test_prefixes = int(0.2 * num_prefixes) # 20% para test
random.shuffle(image_prefixes) # Seleccionar aleatoriamente los prefijos para
test_prefixes = image_prefixes[:num_test_prefixes]
remaining_prefixes = image_prefixes[num_test_prefixes:]

kf = KFold(n_splits=5, shuffle=True, random_state=42)
fold = 1
for train_index, val_index in kf.split(remaining_prefixes):
    training_prefixes = [remaining_prefixes[i] for i in train_index]
    validation_prefixes = [remaining_prefixes[i] for i in val_index]

    training_dir = os.path.join(output_directory,
        f"Training/Fold_{fold}/Actionable")
    validation_dir = os.path.join(output_directory,
        f"Validation/Fold_{fold}/Actionable")
    if not os.path.exists(training_dir):
        os.makedirs(training_dir)
    if not os.path.exists(validation_dir):
        os.makedirs(validation_dir)
```

## B.2. VALIDACIÓN CRUZADA

---

```
for prefix in training_prefixes:
    source_file = os.path.join(input_directory, prefix)
    destination = os.path.join(training_dir, prefix)
    shutil.copy(source_file, destination)

for prefix in validation_prefixes:
    source_file = os.path.join(input_directory, prefix)
    destination = os.path.join(validation_dir, prefix)
    shutil.copy(source_file, destination)

fold += 1

test_dir = os.path.join(output_directory, "Test/Actionable")
if not os.path.exists(test_dir):
    os.makedirs(test_dir)
for prefix in test_prefixes:
    source_file = os.path.join(input_directory, prefix)
    destination = os.path.join(test_dir, prefix)
    shutil.copy(source_file, destination)
```

En primer lugar, el código crea los directorios donde se guardarán los diferentes datos, después selecciona el número de casos que tendrá cada división de forma aleatoria, siendo el 20% para *test* y el resto se utilizará en la validación cruzada.

Haciendo uso de la librería *KFold* se especifican cinco *folds* distintos. Para cada *fold*, se generan listas de prefijos para los datos de entrenamiento y validación utilizando los índices proporcionados por *KFold*. A continuación, se definen los directorios donde se almacenarán estos datos para la partición actual. Si estos directorios no existen, se crean.

Luego, se copian los archivos correspondientes a los prefijos de entrenamiento desde el directorio de entrada al directorio de entrenamiento. Posteriormente, se realiza

## APÉNDICE B. CÓDIGO

---

el mismo proceso para los datos de validación, copiando los archivos correspondientes al directorio de validación.

Tras completar las iteraciones para todas las particiones, se define y crea, si es necesario, un directorio para los datos de *test*. Finalmente, se copian los archivos correspondientes a los prefijos de prueba desde el directorio de entrada al directorio de *test*.



## B.2. VALIDACIÓN CRUZADA

---

# Referencias

- Balleyguier, C., Kinkel, K., Fermanian, J., Malan, S., Djen, G., Taourel, P., y Helenon, O. (2005). Computer-aided detection (CAD) in mammography: does it help the junior or the senior radiologist? *European journal of radiology*, 54(1), 90–96.
- Baum, F., Fischer, U., Obenauer, S., y Grabbe, E. (2002). Computer-aided detection in direct digital full-field mammography: initial results. *European Radiology*, 12, 3015–3017.
- Best Buy. (2024). *Msi nvidia geforce rtx 3070 ti ventus 3x oc 8gb gddr6 pci express 4.0 graphics card*. Descargado de <https://www.bestbuy.com/site/msi-nvidia-geforce-rtx-3070-ti-ventus-3x-oc-8gb-gddr6-pci-express-4-0-graphics-card/6467500.p?intl=nosplash&skuId=6467500> (Accessed: 2024-08-31)
- Brzakovic, D., Luo, X. M., y Brzakovic, P. (1990). An approach to automated detection of tumors in mammograms. *IEEE Trans. Med. Imaging*, 9, 233–241.
- Buda, M., Saha, A., Walsh, R., Ghate, S., Li, N., Swiecicki, A., ... Mazurowski, M. (2020). *Breast Cancer Screening – Digital Breast Tomosynthesis (BCS-DBT) (Version 5)*. Descargado de <https://doi.org/10.7937/E4WT-CD02> (Data set <https://www.cancerimagingarchive.net/collection/breast-cancer-screening-dbt> [Accedido el 6 de junio de 2024]) doi: 10.7937/E4WT-CD02

## REFERENCIAS

---

Buda, M., Saha, A., Walsh, R., Ghate, S., Li, N., Świecicki, A., . . . Mazurowski, M. A. (2021, 08). A Data Set and Deep Learning Algorithm for the Detection of Masses and Architectural Distortions in Digital Breast Tomosynthesis Images. *JAMA Network Open*, 4(8), e2119100-e2119100. Descargado de <https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2783046> (<https://jamanetwork.com/journals/jamanetworkopen/fullarticle/2783046> [Accedido el 6 de junio de 2024]) doi: 10.1001/jamanetworkopen.2021.19100

Choi, J. Y., Kim, D. H., Plataniotis, K. N., y Ro, Y. M. (2016). Classifier ensemble generation and selection with multiple feature representations for classification applications in computer-aided detection and diagnosis on mammography. *Expert Syst. Appl.*, 46, 106–121.

Chollet, Francois. (2021). *Deep learning with Python* (2.<sup>a</sup> ed.). Manning.

Chu, J., Min, H., Liu, L., y Lu, W. (2015). A novel computer aided breast mass detection scheme based on morphological enhancement and SLIC superpixel segmentation. *Medical Physics*, 42, 3859–3869.

Des Plantes, B. Z. (1932). Un nuevo método para la diferenciación en la radiografía (planigrafía). *Acta radiologica*(2), 182–192.

Ganesan, K., Acharya, U. R., Chua, C. K., Min, L. C., y Abraham, T. K. (2014). Automated diagnosis of mammogram images of breast cancer using discrete wavelet transform and spherical wavelet transform features: A comparative study. *Technol. Cancer Res. Treat.*, 13, 605–615.

Görgel, P., Sertbas, A., y Uçan, O. N. (2015). Computer-aided classification of breast masses in mammogram images based on spherical wavelet transform and support vector machines. *Expert Syst*, 32, 155–164.

He, K., Zhang, X., Ren, S., y Sun, J. (2016). Identity mappings in deep residual networks. En *Computer Vision–ECCV 2016: 14th European Conference*,

## REFERENCIAS

---

- Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part IV 14* (pp. 630–645).
- Huo, Z., Giger, M. L., Vyborny, C. J., Wolverton, D. E., Schmidt, R. A., y Doi, K. (1998). Automated computerized classification of malignant and benign masses on digitized mammograms. *Academic Radiology*, 5, 155–168.
- Instituto Nacional de Estadística, INE. (2024). *Tasa de mortalidad atribuida a las enfermedades cardiovasculares, el cáncer, la diabetes o las enfermedades respiratorias crónicas por comunidad autónoma, edad, sexo y periodo*. Descargado de <https://www.ine.es/up/H5IW2m1Bi4> (<https://www.ine.es/up/H5IW2m1Bi4> [Accedido el 9 de julio de 2024])
- Instituto Nacional del Cáncer (NCI). (2024). *PDQ® sobre los exámenes de detección y la prevención. PDQ Exámenes de detección del cáncer de seno (mama)*. (Resúmen actualizado el 9 de mayo de 2024. Disponible en: <https://www.cancer.gov/espanol/tipos/seno/pro/deteccion-seno-pdq>. Fecha de acceso: 07/06/2024)
- Karssemeijer, N., y te Brake, G. M. (1996). Detection of stellate distortions in mammograms. *IEEE Transactions on Medical Imaging*, 15, 611–619.
- Kim, S. J., Moon, W. K., Cho, N., Cha, J. H., Kim, S. M., y Im, J. G. (2006). Computer-aided detection in digital mammography: Comparison of craniocaudal, mediolateral oblique, and mediolateral views. *Radiology*, 241, 695–701. Descargado de <https://pubmed.ncbi.nlm.nih.gov/17114620/> doi: 10.1148/radiol.2413051145
- Lee, R. S., Gimenez, F., Hoogi, A., Miyake, K. K., Gorovoy, M., y Rubin, D. L. (2017). A curated mammography data set for use in computer-aided detection and diagnosis research. *Scientific Data*, 4, 170177. Descargado de <https://doi.org/10.1038/sdata.2017.177> (<https://www.nature.com/articles/sdata2017177> [Accessed June 6, 2024]) doi: 10.1038/sdata.2017.177

## REFERENCIAS

---

- Li, L., Clark, R. A., y Thomas, J. A. (2002). Computer-aided diagnosis of masses with full-field digital mammography. *Academic Radiology*, 9, 4–12.
- Liu, S., Babbs, C. F., y Delp, E. J. (2001). Multiresolution detection of spiculated lesions in digital mammograms. *IEEE Transactions on Image Processing*, 10, 874–884.
- Mariño, M. A. G. (2022). Tomosíntesis digital para el cribado de cáncer de mama: revisión sistemática y evaluación de la calidad de los metaanálisis. *Revista de Senología y Patología Mamaria*, 35(4), 236–242. Descargado de <https://www.elsevier.es/es-revista-revista-senologia-patologia-mamaria--131-articulo-tomosintesis-digital-el-cribado-cancer-S0214158221001250> (<https://www.elsevier.es/es-revista-revista-senologia-patologia-mamaria--131-articulo-tomosintesis-digital-el-cribado-cancer-S0214158221001250>) [Accedido el 6 de junio de 2024])
- Martínez, J. (2017, 06). Análisis de minería de datos distribuida con Weka Parallel en computadoras con múltiples procesadores físicos y lógicos. *Economía y Administración (E&A)*, 6, 155. doi: 10.5377/eya.v6i2.4307
- Mudigonda, N. R., Rangayyan, R. M., y Desautels, J. E. (2000). Gradient and texture analysis for the classification of mammographic masses. *IEEE Trans. Med. Imaging*, 19, 1032–1043.
- Mudigonda, N. R., Rangayyan, R. M., y Desautels, J. E. (2001). Detection of breast masses in mammograms by density slicing and texture flow-field analysis. *IEEE Transactions on Medical Imaging*, 20, 1215–1227.
- Niklason, L. T., Christian, B. T., Niklason, L. E., Kopans, D. B., Castleberry, D. E., Opsahl-Ong, B., ... others (1997). Digital tomosynthesis in breast imaging. *Radiology*, 205(2), 399–406.
- Niklason, L. T., Niklason, L. E., y Kopans, D. B. (1999, febrero 16). *Tomosynthesis system for breast imaging*. Google Patents. (US Patent 5,872,828)

## REFERENCIAS

---

- Organización Mundial de la Salud, OMS. (2024). Cáncer de mama. *World Health Organization*. (<https://www.who.int/es/news-room/fact-sheets/detail/breast-cancer> [Accedido el 6 de junio de 2024])
- Qiu, Y., Wang, Y., y Zheng, B. (2016). Computer-aided classification of mammographic masses using the deep learning technology: a preliminary study. En *SPIE, Medical Imaging 2016: Computer-Aided Diagnosis* (Vol. 9785).
- Rangayyan, R. M., Mudigonda, N. R., y Desautels, J. E. (2000). Boundary modelling and shape analysis methods for classification of mammographic masses. *Med. Biol. Eng. Comput.*, 38, 487–496.
- Ronneberger, O., Fischer, P., y Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. En *Medical image computing and computer-assisted intervention—MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III* 18 (pp. 234–241).
- Roy, S., Saha, R., Sarkar, S., Mehera, R., Pal, R. K., y Bandyopadhyay, S. K. (2023). Brain tumour segmentation using S-Net and SA-Net. *IEEE Access*, 11, 28658–28679.
- Sadaf, A., Crystal, P., Scaranello, A., y Helbich, T. (2011). Performance of computer-aided detection applied to full-field digital mammography in detection of breast cancers. *European Journal of Radiology*, 77, 457–461.
- Sahiner, B., Chan, H. P., Petrick, N., Helvie, M. A., y Hadjiiski, L. M. (2001). Improvement of mammographic mass characterization using spiculation measures and morphological features. *Med. Phys.*, 28, 1455–1465.
- Sawyer-Lee, R., Gimenez, F., Hoogi, A., y Rubin, D. (2016). *Curated Breast Imaging Subset of Digital Database for Screening Mammography (CBIS-DDSM)*. Descargado de <https://doi.org/10.7937/K9/TCIA.2016.7002S9CY> (Dataset — <https://doi.org/10.7937/K9/TCIA.2016.7002S9CY> [Accedido el 6 de junio de 2024]) doi: 10.7937/K9/TCIA.2016.7O02S9CY

## REFERENCIAS

---

- Szegedy, C., Ioffe, S., Vanhoucke, V., y Alemi, A. (2017). Inception-v4, inception-resnet and the impact of residual connections on learning. En *Proceedings of the AAAI conference on artificial intelligence* (Vol. 31).
- Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., y Wojna, Z. (2016). Rethinking the inception architecture for computer vision. En *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).
- The, J. S., Schilling, K. J., Hoffmeister, J. W., y Mcginnis, R. (2009). Detection of breast cancer with full-field digital mammography and computer-aided detection. *American Journal of Roentgenology*, 192, 337–340.
- Timp, S., Varela, C., y Karssemeijer, N. (2007). Temporal change analysis for characterization of mass lesions in mammography. *IEEE Trans. Med. Imaging*, 26, 945–953.
- Torres, J. (2020). *Python deep learning: Introducción práctica con Keras y TensorFlow* 2. Alpha Editorial.
- Vedantham, S., Karellas, A., Vijayaraghavan, G. R., y Kopans, D. B. (2015). Digital breast tomosynthesis: state of the art. *Radiology*, 277(3), 663–684.
- Walsh, J., Othmani, A., Jain, M., y Dev, S. (2022). Using U-Net network for efficient brain tumor segmentation in MRI images. *Healthcare Analytics*, 2, 100098.
- Wikipedia - Alan Turing. (2024). *Alan Turing*. Descargado de [https://es.wikipedia.org/wiki/Alan\\_Turing](https://es.wikipedia.org/wiki/Alan_Turing) ([https://es.wikipedia.org/wiki/Alan\\_Turing](https://es.wikipedia.org/wiki/Alan_Turing) [Accedido el 9 de julio de 2024])
- Yang, S. K., Moon, W. K., Cho, N., Park, J. S., Cha, J. H., Kim, S. M., ... Im, J.-G. (2007). Screening mammography—detected cancers: Sensitivity of a computer-aided detection system applied to full-field digital mammograms. *Radiology*, 244, 104–111. Descargado de <https://pubs.rsna.org/doi/abs/>



## REFERENCIAS

---

[10.1148/radiol.2441060756?journalCode=radiology](https://doi.org/10.1148/radiol.2441060756?journalCode=radiology) doi: 10.1148/radiol  
.2441060756