Escuela Politécnica Superior de Jaén

Grado en Ingeniería Informática

UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior de Jaén

Trabajo Fin de Grado

Identificación de tumores de mama a partir de análisis de imágenes con técnicas de aprendizaje profundo

Alumno: Javier Martinez Portellano

Tutor: Francisco Charte Ojeda

Dpto: Informática

UNIVERSIDAD DE JAÉN

D./Dª Francisco Charte Ojeda, tutor(es) del Trabajo Fin de Grado titulado: **Identificación de tumores de mama a partir de análisis de imágenes con técnicas de aprendizaje profundo**, que presenta Javier Martinez Portellano, autoriza(n) su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, junio de 2021

El estudiante

El tutor

CHARTE OJEDA FRANCISCO - 25996290B

Firmado digitalmente por CHARTE OJEDA FRANCISCO - 25996290B
Fecha: 2021.06.29 17:34:21 +02'00'

Javier Martinez Portellano

Francisco Charte Ojeda

Agradecimientos

En primer lugar agradecer a mis padres y hermana por el apoyo incondicional que me han brindado durante todo el grado. Ha sido una tarea ardua, y aunque hubo momentos en los que pensé que no lo conseguiría, ellos siempre estuvieron ahí.

Agradecer también a todos mis amigos y cactus en concreto sin los cuales todo el camino habría sido bastante más difícil.

En concreto agradecer a Irene Moreno Rubio por ser un pilar fundamental de apoyo y cariño sin el cual no tengo dudas de que no habría podido superar esta etapa universitaria en la sin duda gracias a ella hoy en día estoy aquí.

A Antonio Manuel Milla Lara por ser mi compañero y un buen amigo durante todo el proceso de realización de este proyecto ayudándome y aclarándome dudas siempre que lo necesitaba.

Por último, pero no menos importante, agradecer a Francisco Charte Ojeda todo el trabajo que ha realizado con el fin de instruirme y guiarme en el campo de la inteligencia artificial y la realización de este proyecto. También agradecerle la paciencia que ha tenido pese a los malos ratos que le he hecho pasar en más de algún momento. A partir de este punto será un referente que tendré en vista como una figura a alcanzar por su desempeño y capacidad de trabajo.

Tabla de contenidos

Lista de figuras

Lista de tablas

Capítulo 1

Introducción

En este primer capítulo se definirá el marco contextual en el cual se ha realizado este proyecto, así como las razones de interés temático y la motivación que han impulsado la investigación en el campo del *Machine Learning* (ML). Finalmente se podrá encontrar un breve resumen con la estructura de esta memoria donde se indicará el contenido de cada capítulo seguido por una tabla donde se presentarán los acrónimos necesarios para la compresión del lector.

1.1. Interés temático

Los tumores son una de las principales causas de muerte por enfermedad en todo el mundo tal y como se explica en Jemal et al. [2011]. Esta enfermedad cuenta con un elemento clave en la supervivencia de los pacientes, la fase de detección. Es necesario ser capaz de detectar la existencia de células cancerígenas con la máxima anticipación posible.

En el campo de la oncología el número de técnicas de detección que involucran el trabajo con imágenes es sumamente alta, tal y como se muestra en artículos como Barentsz et al. [2001] o R.A et al. [2006]. La cantidad de imágenes generadas con este fin exige la existencia de un alto número de especialistas que puedan ser capaces de clasificar correctamente esas imágenes generadas indiferentemente de las técnicas usadas para ello.

Como se puede apreciar en la tabla 1.1 los tumores que más incidencia presentan son aquellos que se desarrollan en la próstata y en las mamas. Una de las estadísticas que permite conocer la mortalidad y agresividad de cada tumor es la supervivencia

Tumor	Incidencia por 100.000 personas	Casos totales	Porcentaje de nuevos tumores
Colorrectal	45.29	3060245.30	8.20
Hígado	8.06	544614.20	2.40
Leucemia	14.24	962196.80	3.40
Linfoma	20.93	1414240.10	4.30
Páncreas	12.76	862193.20	3.20
Piel	23.28	1573029.60	5.60
Próstata	**158.65**	10719980.50	10.60
Pulmón	60.39	4080552.30	12.70
Riñón	16.00	1081120.00	4.10
Mama	**128.63**	8691529.10	15.30
Tiroides	13.25	895302.50	2.90
Útero	26.15	1766955.50	3.60
Vejiga	21.43	1448025.10	4.50

Tabla. 1.1: Tumores más comunes y estadísticas principales

relativa a cinco años, la cual indica qué porcentaje de personas diagnosticadas con tumor maligno siguen vivas tras cinco años desde la fecha de diagnóstico.

Tumor	Supervivencia relativa 5 años
Colorrectal	64.6 %
Hígado	19.6 %
Leucemia	63.7 %
Linfoma	72.7 %
Páncreas	10.0 %
Piel	92.7 %
Próstata	**97.8 %**
Pulmón	20.5 %
Riñón	75.2 %
Mama	**90.0 %**
Tiroides	98.3 %
Útero	81.2 %
Vejiga	76.9 %

Tabla. 1.2: Probabilidad de supervivencia tras 5 años tras la detección

Tras los tumores más comunes presentados en la tabla 1.2, la motivación en la selección de la mama como objetivo sobre el que se realizará el estudio viene dada por los datos presentes en la tabla mencionada al inicio de este párrafo, donde se puede apreciar una menor supervivencia en los tumores desarrollados en mamas respecto a los desarrollados en la próstata, además de la presencia de una mayor población nacional femenina según Dis [2021].

Cabe aclarar que el porcentaje de supervivencia anteriormente indicado representa

la supervivencia media entre los posibles 4 estadios, los cuales describen el tamaño y ubicación del tumor, así como el crecimiento de este en torno a los tejidos adyacentes. Si se realizase un estudio de supervivencia separado por estadios, se podrían apreciar los resultados presentados en la tabla 1.3

Estado de detección	Porcentaje de supervivencia
1	97.90
2	89.60
3	72.00
4	26.20
Desconocido	69.10

Tabla. 1.3: Supervivencia por estadios a 5 años

Se puede apreciar la repercusión del estadio en el que se detecta la presencia del tumor en relación a la supervivencia del paciente. Esto nos indica que el trabajo a realizar tiene que centrarse en poder detectar los tumores en el estadio más reciente posible.

Una vez seleccionado el tipo de tumor a tratar, es necesario, debido a la existencia de múltiples técnicas de *Imaging*, seleccionar de entre las posibles, aquellas que otorguen mejores resultados. Posteriormente en esta memoria se presentará la reflexión seguida para realizar la selección final del tipo de imagen sobre la que trabajar.

1.1.1. Motivación

Tras el análisis del problema existente se plantea una oportunidad que permite el beneficio de ambas partes involucradas en el proceso de detección de un tumor maligno mediante técnicas de *Imaging*, es decir, el paciente, y el especialista.

En los últimos años se ha producido un auge desmesurado en el campo del ML debido a la imparable cantidad de nuevas tecnologías y recursos que siguen surgiendo aun hoy día. La filosofía y razón de este proyecto reposa en la idea de convertir una tarea mecánica y que requiere un alto grado de atención, como es la detección localizada y etiquetado de un tumor, en una tarea rápida e infalible.

Esta tarea siempre ha sido gestionada por los profesionales, pero tal y como se indicó al inicio del capítulo, se está convirtiendo en una tarea demasiado densa como para que la clasificación y etiquetado presenten una alta calidad dentro de un tiempo aceptable sin necesitar un elevado número de profesionales que, en general, no pueden ser mantenidos por el sistema debido a limitaciones económicas.

La utilización del sistema propuesto necesitaría una revisión para cada imagen mucho menos exhaustiva, lo que agilizaría enormemente el proceso de clasificación y permitiría aumentar la eficacia del trabajo realizado debido a que en el caso de ser imágenes positivas, es decir, que sí exista una presencia anómala, no sería necesaria la supervisión de un profesional ya que estos casos podrían filtrarse con posteriores técnicas más exhaustivas, pero que son realizadas a un menor número de pacientes. Para que el sistema a desarrollar pueda considerarse un éxito, es necesario que la probabilidad de acierto en la clasificación generada sea muy elevada, además de una presencia de falsos negativos extremadamente baja.

1.1.2. Estructura de la memoria

Finalmente, como se indicó en la introducción, se procede a presentar un resumen por capítulos de la memoria realizada.

Capitulo 1. Introducción

Capítulo en el que se encuentra el lector y en el cuál se busca presentar el contexto de este proyecto, así como la información necesaria para la compresión de los objetivos propuestos.

Capitulo 2. Antecedentes

En este capítulo se proporcionarán todos los conocimientos teóricos respecto a las técnicas utilizadas necesarios para poder abordar y comprender el problema tratado, además de la información de índole médica indispensable para la contextualización de las elecciones realizadas en torno al tipo de imagen usada.

Capitulo 3. Objetivos

Definición tanto del objetivo principal de este proyecto como de los objetivos más específicos que se buscarán alcanzar.

Capitulo 4. Materiales y métodos

Se expondrán todas las herramientas necesarias para la realización del proyecto, así como los materiales utilizados y las técnicas implementadas con las que se buscarán alcanzar los objetivos.

Capitulo 5. Resultados

Fase de experimentación del proyecto donde se aplicarán las técnicas mencionadas, generando así la información pertinente para realizar un estudio en torno a los datos obtenidos.

Capitulo 6. Conclusiones

Capítulo final donde, con la información generada en el anterior capítulo, se observará si se han alcanzado los objetivos propuestos al inicio del proyecto junto con propuestas para continuar la línea de estudio seguida.

1.2. Glosario de términos

Finalmente se expondrá en la tabla 1.4 los acrónimos usados a lo largo de toda la memoria con el fin de facilitar la lectura. Durante el texto, aunque aparezcan las definiciones en castellano, se utilizarán las siglas en inglés.

Español	Inglés	Siglas en Inglés
Aprendizaje profundo	Deep learning	DL
Aprendizaje automático	Machine learning	ML
Inteligencia artificial	Artificial intelligence	AI
Perceptrón multicapa	Multilayer perceptron	MLP
Arboles de clasificación y regresión	Clasification and regression trees	CART
Clasificador bayesiano ingenuo	Naive bayesian clasifier	NBC
Maquina de vector soporte	Support vector machine	SVM
Descenso del gradiente estocástico	Stochastic gradient descent	SGD
Función de base radial	Radial base function	RBF
Lógica difusa	Fuzzy logic	FL
Bosque aleatorio	Random forest	RF
Red bayesiana dinámica	Dinamic bayesian networks	DBN
Método de conjuntos	Ensemble machine learning	EML
K-vecinos más cercanos	k-nearest neightbours	KNN
Red neuronal convolucional	convolutional neuronal network	CNN
Mapa de características	Feature map	FM
Red neuronal recurrente	recurrent neuronal network	RNN
Busqueda de arquitectura de la red	Network architecture search	NAS
Tomografia computarizada	Computed tomography	CT
Unidad de procesamiento gráfica	Graphics processing unit	GPU

Tabla. 1.4: Acrónimos usados a lo largo de la memoria

Capítulo 2

ANTECEDENTES

2.1. Introducción

Este capítulo presentará en primer lugar los fundamentos teóricos necesarios para la comprensión de esta memoria. En segundo lugar, se expondrá un breve resumen de los clasificadores simples utilizados hasta la fecha en el campo de la clasificación de imágenes para la detección de tumores acompañados cada uno con una referencia por si se desea ampliar conocimiento de alguno de ellos. Posteriormente se nombrarán aquellas técnicas más avanzadas y se profundizará en aquella utilizada para la realización de los experimentos. Para terminar con el capítulo se presentarán los interpoladores utilizados para el reescalado de las imágenes y un último apartado sobre el estado del arte.

2.2. Fundamentos teóricos

Antes de explicar qué es exactamente el DL, es necesario enmarcar el contexto en el que existe. DL es una parte del campo del ML, que a su vez es una parte de la AI, por lo que en primer lugar se realizará una explicación de los tres campos.

2.2.1. Inteligencia artificial

Se trata de una rama de conocimiento en el campo de la informática que busca imitar los comportamientos inteligentes humanos como aprender, imitar o razonar. Es

un campo de gran amplitud que abarca técnicas de diversa dificultad, desde niveles básicos como sentencias *if* hasta la capacidad de crear nueva información a partir de características aprendidas a lo largo de un proceso.

A la hora de definir exactamente qué es la AI y delimitar su alcance resulta extremadamente difícil debido a que antes se debe de definir lo que es la inteligencia como tal. Es una tarea ardua ya que depende también del entendimiento que tengamos de la inteligencia humana y su funcionamiento. En artículos como Dobrev [2012] se realiza un estudio sobre lo que significa y conlleva la AI.

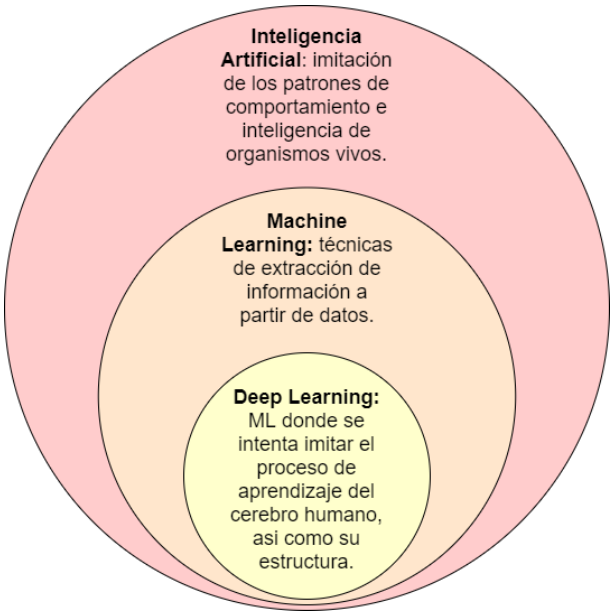

Figura 2.1: Jerarquía del DL,ML e AI inspirada en Opperman

2.2.2. *Machine Learning*

Se refiere al ámbito de la AI donde se generan los sistemas inteligentes en torno a la capacidad de realizar una extracción de características basadas en patrones, a partir de los cuales es capaz de predecir comportamientos futuros sin la necesidad de supervisión humana intentando imitar el comportamiento humano. Uno de los procesos indispensables por los que tiene que pasar la mayoría de técnicas en este campo es el aprendizaje, el cual puede ser de tres tipos en función del etiquetado de los datos usados para el proceso de extracción de características.

- **Entrenamiento supervisado**: los elementos que debe aprender a diferenciar cuentan con una etiqueta o valor continuo dependiendo del tipo de tarea que se esté realizando con la que tras realizar una predicción la red puede observar si

la salida que ha generado para ese elemento es correcta, pudiendo actuar en consecuencia del resultado.

- **Entrenamiento no supervisado**:Cuando se busca explorar todos los patrones posibles en los datos introducidos se omiten las etiquetas con las que comparar las salidas obtenidas, ofreciendo así resultados no esperados, es decir, se trata de un modo en el que el sistema no cuenta con ese tipo de información.

- **Entrenamiento por refuerzo**: este último método de aprendizaje está basado en la teoría conductual haciendo que la red busque maximizar una "recompensa". Cuando se realiza una clasificación correcta, se favorece a la red, en caso de que falle, se penaliza. El objetivo es encontrar un punto medio entre los dos métodos anteriores intentando alcanzar un equilibrio exploración-explotación.

Una vez aprendida la información aportada, existe una clasificación que separa en dos grupos en función del objetivo que se busca con el sistema. La diferencia consiste en si se desea describir la estructura de datos u obtener un modelo predictivo. Se puede encontrar más información sobre los temas tratados a continuación en Charte [2015].

- **Sistemas predictivos**: el objetivo es generar un modelo que ante unos datos de entrada sea capaz de reconocer patrones, etiquetando así la información o realizar una predicción en el valor de salida. Esta clase de sistemas suelen recibir un entrenamiento supervisado.

- **Sistemas descriptivos**: como su nombre indica, buscan describir una estructura de datos con el fin de reconocer información no trivial, así como la relación existente entre los datos. En este caso los sistemas suelen recibir un entrenamiento no supervisado.

Una vez estudiado los tipos de sistemas, y antes de nombrar aquellos más básicos, se van exponer las diferentes tareas que pueden realizarse, todo ello en función de los dos tipos de sistemas expuestos en el párrafo anterior. Así pues, se pueden distinguir cinco tareas diferentes.

- **Clasificación**: con unos datos distribuidos en categorías mediante etiquetas, el objetivo será que el sistema sea capaz de clasificar datos en la categoría correcta sin conocer la etiqueta de cada elemento.

- **Regresión**: el objetivo es aproximar una función matemática, usualmente continua, con el objetivo de tener un modelo que sea capaz de predecir un valor de salida en función de una entrada concreta.
- **Agrupamiento**: tomando como partida un conjunto de datos no etiquetados y un criterio de agrupamiento, se buscará encajar cada elemento perteneciente según métricas de similitud definidas.
- **Asociación**: tomando como inicio un entrenamiento no supervisado se buscará encontrar las asociaciones existentes en un conjunto de datos con el objetivo de generar unas reglas de asociación que facilite la interpretación del conjunto.
- **Tratamiento de series temporales**: a diferencia del resto de posibles tareas a realizar en esta lista, en este tipo de caso se cuenta con una dependencia temporal en los datos. Esta tarea se basa en la regresión con importancia temporal por parte de los datos, dicho de otra manera, se trata de una tarea de pronóstico.

2.3. Técnicas básicas

En este apartado se expondrán brevemente aquellas técnicas clásicas que presentan poca complejidad utilizadas para realizar las tareas indicadas en el apartado anterior.

2.3.1. Clasificadores bayesianos

Existen diversos tipos de clasificadores bayesianos por lo que en este apartado se expondrán los más conocidos.

Clasificador bayesiano ingenuo

Habitualmente, para detectar un elemento en una escena se toman las características de un objeto y si se localizan esa serie de patrones, se indica que el objeto está en la escena. En los clasificadores bayesianos ingenuos, como se indica en Bustamante et al. [2006] toman las características de un objeto de forma independiente a la hora de indicar si el elemento se encuentra en la escena, no se indica si está o no, o qué elemento es, lo que se expresa es la probabilidad de que el elemento estudiado esté en la escena en función de la cantidad de características que se encuentren.

El funcionamiento se basa en un modelo gráfico para representar la dependencia entre variables aleatorias, así pues, tal y como se muestra en el teorema de Bayes, la probabilidad de una variable cuando se encuentra condicionada respecto a otra es:

$$P(A_i|B) = \frac{P(B|A_i)P(A)}{\sum_{k=1}^{n} P(B|A_k)P(A_k)}$$

Redes bayesianas dinámicas

Las redes bayesianas clásicas tratan de representar el estado de las variables en un momento concreto del tiempo. Por otro lado, las RDB, como se explica en Ghahramani [1998], sirven para representar procesos dinámicos. Este tipo de redes Se basan en la representación de los estados del proceso en un tiempo y las relaciones temporales entre los diferentes procesos. Para la definición de una RBD tomaremos como premisas que el estado actual solo depende del estado anterior y que las probabilidades en el modelo no cambian con el tiempo.

La inferencia en las RBD es la misma que en las RB clásicas, debido al uso de los mismos métodos, pero con una mayor complejidad. Esta característica genera que sea común utilizar métodos basados en simulación estocástica.

Las RBD constan de dos componentes, los cuales son la estructura base y la red de transición, por lo que el proceso de aprendizaje consta de dos etapas diferenciadas:

- Aprendizaje de la estructura base: se toman en cuenta los datos de todas las variables en cada instante de tiempo, permitiendo así obtener las interdependencias sin considerar relaciones temporales.
- Aprendizaje de la estructura de transición mediante medidas de ajuste o búsqueda.

2.3.2. Máquina de vectores de soporte

En el caso de las máquina de vectores de soporte se observa, tal y como se describe en Noble [2006], un algoritmo de aprendizaje el cual se alimenta con una gran cantidad de datos para entrenar el sistema y poder diferenciar o reconocer patrones realizando una clasificación basada en etiquetas. Un posible ejemplo seria presentar una hortaliza en una escena y poder diferenciar entre una serie de hortalizas las cuales el modelo ha sido preentrenado para diferenciarlas. El funcionamiento interno trata

de encontrar un plano que permita separar dos nubes de puntos dejando la máxima distancia entre el punto de cada clase más cercano entre sí tal y como se aprecia en la figura 2.2.

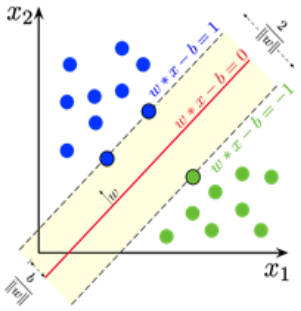

Figura 2.2: Plano objetivo en una SVM Larhman [2018]

2.3.3. Lógica difusa

En la lógica tradicional, como se observa en Carlos Martínez [2018], se trabaja con la premisa de los posibles valores de que un elemento pertenezca a un conjunto, a nombrar, (0,1). Sin embargo, en la lógica difusa, existe la posibilidad de que el valor no sea estricto, si no que pueda presentar un valor en el intervalo [0,1] siendo así posible que se encuentre parcialmente en el conjunto.

2.3.4. Métodos de conjuntos en aprendizaje automático (*ensembles*)

Son algoritmos, como se indica en Polikar [2012], que se basan en la ejecución de un algoritmo de aprendizaje base repetidas veces, formando posteriormente una votación de las hipótesis resultantes.

Existen dos tipos de aproximaciones al diseño de este tipo de algoritmos:

- La primera construye cada hipótesis de manera independiente de tal forma que el conjunto final de hipótesis sea precisa y diversa, es decir, cada hipótesis individual tiene un error razonablemente bajo a la hora de realizar nuevas predicciones y aun así no coincidir con el resto de hipótesis en muchas de las predicciones. En caso de que funcione correctamente se obtendrá un clasificador de mayor calidad, debido a que en caso de opiniones contrarias entre las hipótesis que lo componen generaría aquél caso con superioridad de votos.

- Por otra parte, la segunda aproximación trata de construir las hipótesis en parejas por moda estadística de tal manera que el peso del voto de las hipótesis ofrezca un buen ajuste con los datos.

Bosque aleatorio

Se trata de una extensión de los árboles de clasificación donde se agrupan varios de estos árboles. El funcionamiento reside, al igual que se describe en Cutler et al. [2012], en la interconexión de estos árboles de clasificación, en el cual cada árbol depende de los valores de un vector aleatorio de la muestra de manera independiente y con la misma distribución que el resto de árboles del bosque. Tienen como características principales:

- Debido a la selección aleatoria de características, usada para la división individual de los nodos que produce tasas de error, se realiza una comparativa favorable a AdaBoost.
- Clasificador fuerte debido a su robustez ante el ruido.
- Internamente se supervisan el error, la fuerza y la correlación. Estos datos supervisados son utilizados para obtener la medida de la variable de importancia.

2.3.5. K-vecinos más cercano

Este algoritmo surgió por la necesidad de crear un método de realizar análisis discriminatorio donde las estimaciones de probabilidad de densidad eran difíciles de determinar. Se basa en la distancia, la cual puede ser euclídea entre otras, entre una muestra de ejemplo y las muestras de entrenamiento. Se describe una explicación más profunda en Peterson [2009].

2.3.6. Regresiones

Una de las tareas más comunes en el campo del ML además de la ya explicada clasificación es la regresión, por lo que se darán unas nociones básicas sobre el funcionamiento y los tipos que existen.

La regresión se basa en el uso de un modelo matemático para ajustar una función a un conjunto de datos. Ese ajuste permite establecer una relación entre los datos, pudiendo definir nuevos datos fuera del conjunto inicial de datos mediante la función previamente definida. Existen tres categorías de regresión en función de la cantidad de variables a tener en cuenta y la correlación que exista entre estas. Entre estos tipos de regresión encontramos.

- **Lineal simple**: la más utilizada y sencilla debido a que solo existe una variable dependiente.
- **Lineal múltiple**: a diferencia del anterior, hay presentes más de una variable dependiente.
- **No lineal**: permite una aproximación de los valores de la variable dependiente en un entorno no lineal, generando como resultado un entorno más complejo. Existen tres tipos de regresiones no lineales.
    - Exponencial
    - Logarítmica
    - Polinómica

Dentro de la familia de las regresiones lineales nos encontramos con las regresiones logísticas. Es la vertiente aplicada a clases de las regresiones. Es un análisis de regresión donde se intenta predecir la categoría que tomará una variable, la cual solo puede tomar una categoría dentro de un conjunto limitado a partir de las variables predictoras (independientes).

Arboles de clasificación y regresión

Los árboles de clasificación y regresión son métodos que nos permiten la construcción de modelos de predicción a partir de datos tal y como se explica en Loh [2011]. El funcionamiento se parece a una estructura de datos *quadtree* ya que se basa en la partición recursiva del espacio de datos ajustando un modelo de predicción para cada partición. Visualmente se muestra el resultado mediante un árbol de decisión. Dependiendo del objetivo, se usará un árbol de clasificación o de regresión.

- **Árbol de clasificación**: diseñado para variables dependientes que toman un número discreto de valores desordenados con una predicción de error medido en términos de coste de error de clasificación.

- **Árbol de regresión**: este tipo de árboles se utilizan variables dependientes que toman valores discretos ordenados o continuos, con la predicción de error que suele medirse con la diferencia al cuadrado de los valores observados y predichos.

2.4. Fundamentos: el perceptrón y redes neuronales

Para la realización de este proyecto tal y como se indicó en la introducción, el objetivo y la tarea a completar es una clasificación. Como se vio anteriormente, la clasificación es una tarea predictiva desarrollada sobre un proceso de aprendizaje etiquetado.

La unidad mínima de información en el campo del DL y que suele usarse en la mayoría de estructuras para la clasificación es el perceptrón Rosenblatt [1958], el cual está compuesto por un peso por cada conexión entre las posibles entradas y un umbral ajustable. Su función principal es resolver una salidas binarias generada por las entradas x_n, donde cada x_i cuenta con un peso w_i. Si el sumatorio del producto de cada entrada por su peso es mayor que el umbral de activación del perceptrón la clasificación generada por la salida y será uno, en caso contrario será 0. Se puede apreciar más claramente la estructura en la figura 2.3.

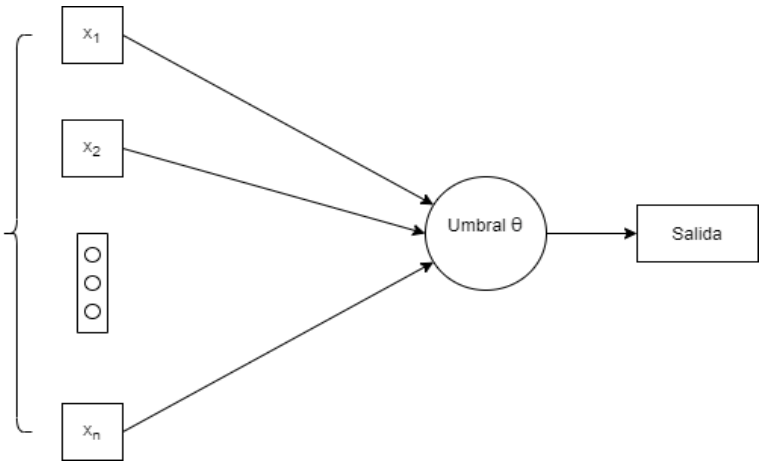

Figura 2.3: Estructura de un perceptrón

Cada perceptrón cuenta con un umbral de activación previamente mencionado. El umbral cuenta con una función de activación, que es una función matemática que determina bajo que circunstancias se activará el perceptrón. Existen diferentes tipos de funciones de activación que convendrá usar en diferentes ocasiones.

2.4.1. Funciones de activación

- **Lineal**: la función más simple de las existentes, la cual genera un resultado proporcional a los elementos de entrada. Presente en la figura 2.4.

Figura 2.4: Función de activación lineal Laughsinthestocks [2015b]

- **Escalonada**: en este caso la única salida posible existente es cero o uno y, precisamente su limitación reside en que se trata de una función no derivable, algo que es esencial para realizar *backpropagation*, no se recomienda su uso en el proceso de aprendizaje. Presente en la figura 2.5.

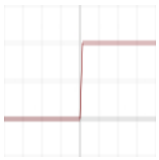

Figura 2.5: Función de activación escalonada Laughsinthestocks [2015a]

- **Sigmoide o logística**: a diferencia de la escalonada, en este caso la salida está en el rango [0, 1] ya que normaliza los valores de salida en ese umbral, y es gracias a esta normalización [0, 1] que resulta ideal para los problemas de clasificación binaria, por lo que suele usarse en las capas de salida para problemas con dos categorías. Presente en la figura 2.6.

Figura 2.6: Función de activación sigmoide Laughsinthestocks [2015d]

- **Tahn**: se trata de una función muy similar a la sigmoide con la diferencia de que el rango está normalizado en [-1, 1] con el objetivo de dejar el cero en el centro del intervalo. Presente en la figura 2.7.

- **ReLU**: se trata de la función más extendida.Es una función a trozos que solo se diferencia de la binaria en que presenta valor alto si la salida es mayor estricto que cero. Presente en la figura 2.8.

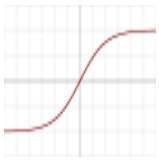

Figura 2.7: Función de activación sigmoide Laughsinthestocks [2015e]

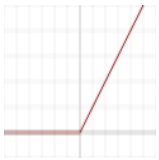

Figura 2.8: Función de activación ReLU Laughsinthestocks [2015c]

- **LeakyReLU**: es la versión en la cual se soluciona *dying ReLU problem* el cual se da en la función ReLU. Como consecuencia de solucionar el inconveniente mencionado se pierde consistencia en los valores negativos. Presente en la figura 2.9.

Figura 2.9: Función de activación LeakyReLU Laughsinthestocks [2016]

- **Softmax**: mientras que hasta ahora todas las funciones recibían un elemento de entrada, en este caso se recibe un vector. Los resultados de salida al igual que la función sigmoide se normaliza en el rango [0, 1], pero en este caso, los ajusta mediante división entre el sumatorio de los resultados. El resultado final es una función que indica la probabilidad de pertenencia a una de las posibles clases existentes.
- **Funcionen de base radial**: son una combinación lineal de traslaciones de una función radialmente simétrica tal y como se describe en Arias et al. [2009]. Suelen usarse para interpolar datos dispersos debido a que el sistema asociado a ecuaciones lineales es invertible, incluso cuando la distribución de puntos no es regular.

Perceptrón multicapa

Es posible generar resultados más potentes si agrupamos en una capa paralela un conjunto de perceptrones simples. Al unir varias de estas capas obtenemos la estruc-

tura más simple de una red neuronal, conocida como perceptrón multicapa (MLP).

El objetivo final es que el MLP sea capaz de realizar una tarea, la cual puede ser de clasificación, regresión o agrupación entre otras, sin embargo, antes debe de saber cómo tratar la información que se le aporta.

Antes de continuar es necesario aclarar que los clasificadores simples que se van a exponer a continuación no solo sirven para realizar como su nombre indica clasificaciones, también sirven para tareas de regresión.

2.5. *Deep Learning y redes neuronales modernas*

Conocido en castellano como aprendizaje profundo, se trata del subcampo del ML en el cual se centran los diseños basados en arquitecturas las cuales admiten múltiples transformaciones de datos en formato matricial o tensorial.

Tras contextualizar la definición del DL, se puede apreciar en la figura 2.1 la jerarquía existente y una definición breve de la materia que ocupa cada campo.

Una red moderna cuenta con múltiples neuronas, pero con una diferencia respecto al MLP, puesto que en este caso existe una especialización de neuronas, es decir, mientras que en un MLP solo existen neuronas simples, en una red moderna encontramos múltiples tipos de neuronas que cumplen con distintos objetivos, alcanzando así una mayor similitud al diseño neuronal humano.

Aunque las arquitecturas modernas sean muy variadas suele existir un factor común en todas ellas, y es que en el proceso de aprendizaje es necesario un reajuste de los pesos en función de cada clasificación realizada con el objetivo de mejorar el rendimiento de la red. Los dos algoritmos más comunes utilizados actualmente son el descenso del gradiente, explicado en Hauser [2012] o en Ojeda [2016], y *backpropagation* el cual se relata extensamente en Rojas [1996].

Los algoritmos indicados anteriormente giran en torno a la función de activación que se explicó al inicio de esta sección. Por otro lado, contamos con herramientas de optimización que pueden ser usadas en distintos casos, como es el caso del descenso del gradiente estocástico.

Descenso del gradiente estocástico

Se trata de una función cuyo objetivo, indicado en Bottou [2012], es minimizar el coste de predicción de la función $l(y, \hat{y})$, la cual mide el coste de predecir $\hat{y}$ cuando la respuesta es y al trabajar con un par $z = (x, y)$ donde x es entrada arbitraria e y una salida escalar. Se trata de un método de gradiente descendente optimizado mediante la ratio de convergencia donde se toman, a diferencia del original, los elementos como independientes. Normalmente se mide la calidad de los resultados observando los valores obtenidos con el riesgo empírico, que mide el comportamiento de los datos usados para entrenar.

2.5.1. Redes convolucionales

Una red neuronal convolucional (CNN) Albawi et al. [2017] está destinada a la clasificación de imágenes, y para ello, sigue una estructura en dos partes. En la primera parte, encontramos las llamadas capas convolucionales, las cuales están compuestas por dos tipos de neuronas, las convolucionales, y las reductoras de muestreo que aunque estrictamente hablando no pertenezcan al proceso de convolución aparece después de este. En la segunda parte encontramos un perceptrón multicapa cuyo propósito será realizar el aprendizaje y la posterior clasificación. Se puede apreciar el diseño en la figura 2.10. Las neuronas pertenecientes a la parte convolucional se utilizan para extraer la información de las imágenes y transformar la información a un formato entendible por el MLP.

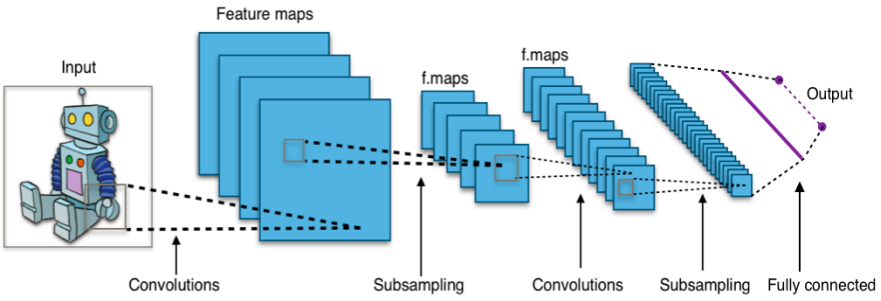

Figura 2.10: Estructura de una CNN por Aphex34 [2015]

Neuronas convolucionales

El primer paso que realiza una red neuronal convolucional se desarrolla en estas neuronas. En esta parte de la red se realiza la extracción de características. Este

proceso se basa en la aplicación de filtros específicos de tamaño variable sobre una imagen con se puede apreciar en la figura 2.11, permitiendo así extraer los patrones que la componen, los cuales generarán la información que permita diferenciar los elementos existentes.

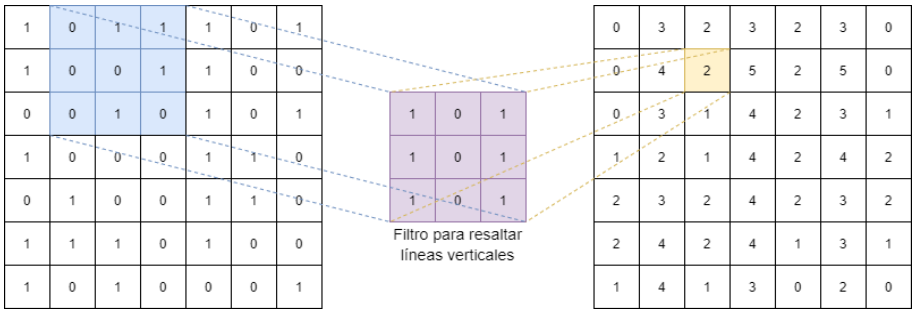

Figura 2.11: Aplicación de un filtro en una capa convolucional

Los filtros utilizados en la detección de patrones tienen diseños que buscan la extracción de normalmente un patrón concreto. Todos los filtros aplicados siguen la misma ecuación para el calculo del píxel destino, es decir, $\sum a_{i_j} \cdot b_{i_j}$.

Sobre una matriz al comienzo de una capa convolucional pueden aplicarse numerosos filtros a la vez, y por cada filtro se generará un llamado mapa de características (FM), que es en esencia una copia modificada de la imagen original que resalta alguna propiedad en específico, como en la figura 2.11 donde se resaltan las lineas verticales.

Si se observara el resultado generado en una primera capa convolucional habiendo aplicado 32 filtros, se observaría que se han generado 32 mapas de características. Si a continuación se aplicase una segunda capa convolucional de 64 filtros, se generarían $32 \cdot 64$ FM, es decir, 2048.

Neuronas de reducción de muestreo

Como se ha indicado antes, la extracción de características permite obtener los patrones de una imagen. Sin embargo, este proceso sin la presencia de esta capa resultaría en la posibilidad de que dos imágenes muy parecidas, con una pequeña diferencia en el desplazamiento de píxeles, pudieran acabar clasificadas en diferentes clases. Para evitar esta posibilidad esta capa utiliza técnicas como *max-pooling*, donde se reduce el tamaño de una imagen manteniendo el valor más alto de cada región (normalmente 2x2) permitiendo así que, con una resolución menor, las mismas características se correspondan con una mayor zona de activación en la imagen.

El la figura 2.12 se puede apreciar cómo el tamaño del FM inicial se ve reducido.

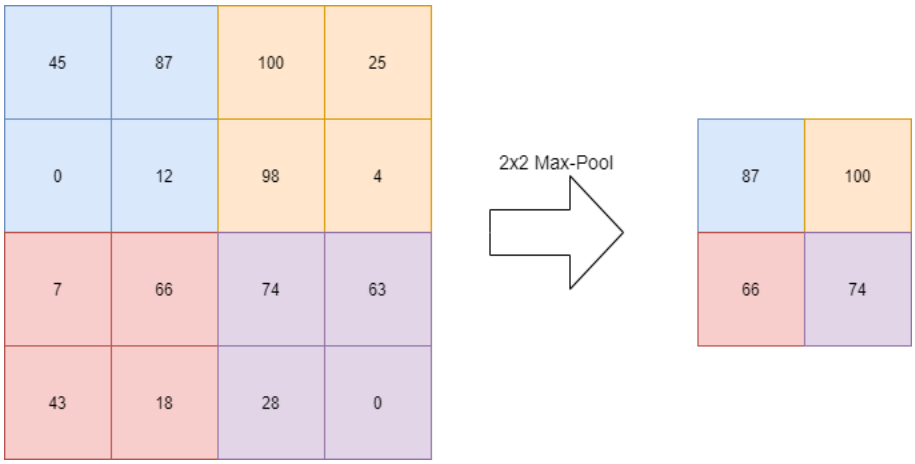

Figura 2.12: Aplicación de Max-pooling con tamaño 2x2

La aplicación de este tipo de técnicas es necesario debido a que la entrada recibida por el MLP debe de ser de una sola dimensión y en el proceso de reducir la dimensionalidad de las imágenes tratadas no puede perderse información si se busca generar un aprendizaje de calidad.

Neuronas de clasificación

Una vez extraídas las características de una imagen, se introducirán los datos en la fase de entrenamiento, donde en un perceptrón multicapa *fully connected* (todas las neuronas de una capa conectadas con todas las neuronas de las capas anterior y siguiente) calculará el peso de cada neurona para posteriormente obtener la clasificación por clases que se busca.

2.5.2. *Automated Machine Learning (autoML)*

El proceso de generación de un modelo de CNN que genere buenos resultados puede ser tedioso y de un alto coste, por esa razón se desarrollaron técnicas de autoML las cuales permiten auto configurar una red para obtener un aumento en la calidad del modelo. La definición de redes neuronales mediante este tipo de técnicas no asegura el máximo global ya que en la búsqueda de arquitectura que se busca no solo se tiene en cuenta la *accuracy* obtenida, si no que parámetros como el coste computacional o el tiempo de búsqueda se tienen en cuenta como factores limitantes.

El procedimiento para generar una arquitectura sigue el proceso de NAS que busca obtener un modelo robusto aplicando las operaciones básicas que se han indicado en

el espacio de búsqueda y que otorgue buenos resultados. Este procedimiento consta de 3 componentes.

1. El espacio de búsqueda para encontrar la arquitectura objetivo. Puede ser:
   - Completamente estructurado
   - Basado en celdas
   - Basado en morfismos
   - Jerárquico.
2. La estrategia que se seguirá durante el proceso de búsqueda.
3. Los métodos que se usarán para el la estimación de los modelos. Pueden ser:
   - Aprendizaje reforzado.
   - Algoritmos evolutivos.
   - Gradiente descendente.
   - Optimización basada en modelos sustitutos.

Si se desea más información sobre el funcionamiento interno de cada uno de los elementos nombrados, así como un poco de historia sobre autoML se recomienda acudir a He et al. [2021].

Una de las aplicaciones en este campo viene de la mano de Tensorflow y recibe el nombre de AutoKeras. Se trata de una librería que permite generar redes neuronales tanto para clasificación como para tareas de regresión, y no solo en imágenes ya que puede usarse sobre texto también entre otros elementos. En Jin et al. [2019] se explica con más detenimiento el funcionamiento y objetivos de esta herramienta, pero la idea es que cualquier persona pueda tener acceso a redes neuronales sin la necesidad de contar con una gran base en la materia.

2.5.3. *Transfer Learning*

El tamaño de los algoritmos de DL está continuamente aumentando. Estos algoritmos descritos en artículos como Kurková et al. [2018] se entrenan con cantidades masivas de datos, lo que desemboca en una inevitable dependencia de datos si se comparan con los algoritmos clásicos de ML, debido a que en los nuevos algoritmos los patrones a detectar son sumamente más concretos. Para evitar que se produzca

el problema de que la inexistencia de suficientes datos para el entrenamiento de un algoritmo es posible utilizar aprendizaje de transferencia, o *Transfer Learning*, el cual intenta traspasar el conocimiento del dominio de origen hacia el dominio destino tomando como premisa que los datos de entrenamiento no tienen que ser exactamente independientes y estar idénticamente distribuidos como los datos utilizados para los test. La premisa reside en el uso del aprendizaje almacenado que se ha obtenido en la resolución de un problema anterior buscando solucionar otro, con características similares, en el que no se cuenta con la cantidad de datos necesaria para obtener un buen entrenamiento. Existen cuatro categorías de este tipo de aprendizaje:

- **Basado en instancias**: utiliza instancias en el dominio de origen con un peso apropiado.
- **Basado en mapeado**: mapea instancias de los dos dominios en un nuevo espacio de datos con una mejor similitud.
- **Basado en redes**: reutiliza la parte preentrenada de la red en el dominio de origen.
- **Basado en redes generativas adversarias**: utiliza tecnología de las redes adversarias para encontrar funcionalidades transferibles que se ajuste para los dos dominios.

2.5.4. Redes neuronales recurrentes

Redes utilizadas para hacer que persista la información durante un número mayor de épocas debido a que las salidas de las capas presentan realimentación a su misma capa generando así un tipo de red neuronal muy eficiente para datos secuenciales.

2.5.5. Redes neuronales probabilísticas

Encontramos la diferencia respecto al resto, como se indica en Juan Andres Cadena [2018], en que no se realiza reajuste de pesos con el objetivo de minimizar el error, es decir, en este tipo de redes contamos con:

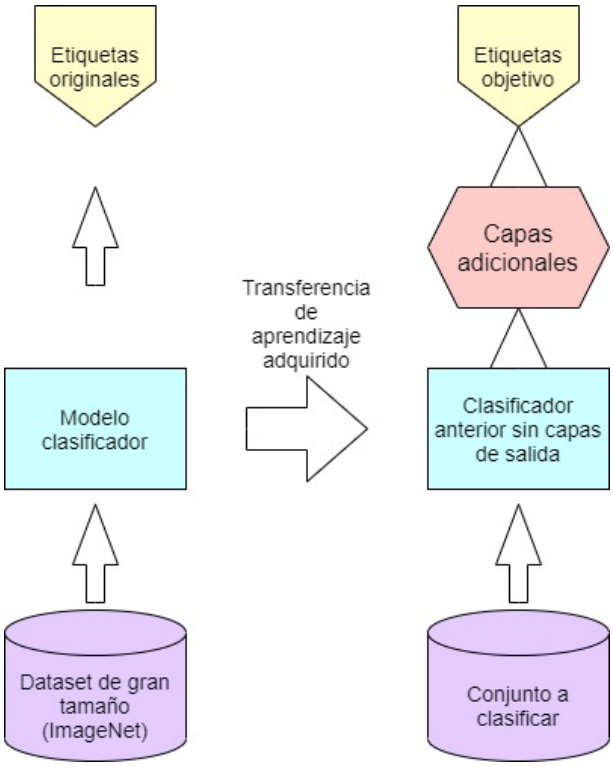

Figura 2.13: Descripción Transfer Learning

- **Capa de entrada**: calcula las distancias entre el vector de entrada entrenado y produce un vector donde los elementos señalan la proximidad a la entrada respecto a la entrada preentrenada.

- **Capa competitiva**: suma las contribuciones para cada clase de entrada y generar un vector de salida con probabilidades. Posteriormente y mediante una función de transferencia en la salida de la segunda capa selecciona la opción más probable generando uno en esa clase y cero en el resto.

- **Capa de salida**: presenta el resultado obtenido de la capa anterior.

2.6. Interpolación de imágenes

A la hora de modificar la resolución de la imagen uno de los métodos posibles es realizar una interpolación de los píxeles. Esta técnica permite concentrar o dispersar, en función de si se quiere aumentar o disminuir la resolución, la información de varios píxeles en uno o viceversa. El objetivo es perder la mínima información posible contenida en la imagen y generar un resultado que mantenga las características de esta.

Existen diversos métodos de interpolación. Sin embargo, en este apartado se van a describir aquellos con los que posteriormente se va a trabajar.

- **Más cercano**: encuentra el punto más cercano en la imagen original para posteriormente establecer el valor del píxel de ese punto en la imagen de destino.
- **Lineal**: calcula el valor del píxel en la imagen de destino mediante el píxel de la imagen origen y la intensidad en la imagen origen en ese mismo píxel.
- **Área**: realiza un remuestreo de la imagen mediante el uso de una relación de área para cada píxel.
- **Cubico**: se trata de una interpolación bi-cubica donde la zona de muestreo tratada es de 4x4 píxeles.
- **Lanczos4**: compuesto por un interpolador tipo Lanczos, es decir, que señala la proporción de uso de cada píxel en la imagen original. En el caso de este interpolador, la proporción de uso se señala en regiones de 8x8 píxeles.

Los interpoladores expuestos se dividen en dos categorías, aquellos que se utilizan para aumentar la resolución de una imagen, como es el caso del más cercano o el lineal, y aquellos usados para reducir la resolución de la imagen, como Lanczos4.

2.7. Conocimiento médico

Con el objetivo de contextualizar el problema tratado y permitir al lector la comprensión del objetivo de las pruebas realizas y la aproximación tomada, se expondrán a continuación los conceptos médicos básicos.

2.7.1. Definición de tumor

Un tumor es, según la definición del instituto nacional del cáncer en Estados Unidos NIH [2021], una masa de tejido anormal generado por un error de replicación celular. En el proceso de replicación que ocurre constantemente pueden producirse errores en los cuales se crean células generadas erróneamente debido a un fallo genético en el núcleo de esas células. Si estas células son eliminadas por el sistema nervioso cuando deben y siguen el proceso de subdivisión, el tejido compuesto por esas células recibe

el nombre de tumor. También es posible que el tumor se produzca a partir de una multiplicación demasiado acelerada. Existen dos tipos de tumores posibles en función de si existe la posibilidad de diseminación, es decir, si el tumor puede expandirse e invadir a otras zonas del organismo hablamos de tumor maligno, en caso de que no exista es posibilidad, recibe el nombre de tumor benigno.

Sistema de clasificación BI-RADS

Para la clasificación de tumores analizados en técnicas de *imaging* existe este sistema, el cual permite clasificar la presencia de un tumor en imágenes de tipo ecografía o resonancia magnética. Aunque este sistema sirve para la clasificación de diferentes tumores, en este proyecto se aplicará este sistema de clasificación sobre los tumores en mamas. Cada mama se clasificará atendiendo a cuatro criterios.

- **Grasa**: si la mama presenta una composición dominante de grasa.
- **Densidad media**: si el tejido fibroglandular es heterogéneo respecto a su densidad.
- **Heterogeneidad**: mama con tejido glandular hetereogéneamente denso.
- **Densa**: presencia de parénquima glandular altamente denso, lo cual es posible que oculte lesiones.

Una vez realizada la especificación de las características de la mama, lo cual permite contextualizar los posibles elementos anómalos encontrados, se procede a una clasificación basada en siete niveles, la cual estudia las lesiones encontradas. Estos niveles indican la probabilidad de que un elemento anómalo encontrado sea maligno, además del procedimiento médico a seguir.

- **BI-RADS 0**: se considera no concluyente debido a una lectura incompleta, es decir, existe la necesidad de realizar más pruebas para determinar la presencia de elementos anómalos.
- **BI-RADS 1**: mama sana. La categorización de una mama sana no indica que no existan elementos anómalos en esta, pero sí que no tienen rasgos malignos y por lo tanto no significan un peligro actual para la paciente.
- **BI-RADS 2**: elementos benignos detectados. Aunque sean benignos, es necesario revisarlos cada dos años para comprobar que no se han desarrollado.

- **BI-RADS 3**: si se han hallado elementos anómalos benignos pero la certeza de que pertenezcan a esa categoría no es absoluta, es decir, que puedan ser malignos. Esta clasificación señala una probabilidad de malignidad del 2 %.
- **BI-RADS 4**: existe una posibilidad entre el 29 % y el 70 % de que el elemento anómalo localizado sea maligno. Esta categoría cuenta con una subcategorización en función de la sospecha sobre la malignidad del elemento.
  - Categoría 4-A: si existe una baja sospecha de malignidad.
  - Categoría 4-B: la sospecha sobre la malignidad es intermedia.
  - Categoría 4-C: la preocupación pasa a ser moderada, pero no clásica de malignidad
- **BI-RADS 5**: seguramente maligno con una seguridad superior al 70 %.
- **BI-RADS 6**: malignidad en la anomalía confirmada.

El proceso de clasificación es mucho mas complejo que la explicación superior, si se quiere profundizar en la materia se recomienda L. et al. [2011].

2.8. Estado del arte

En este último apartado van a explorarse las líneas de investigación seguidas en torno a la detección de tumores mediante técnicas usadas en el campo del ML y se hará una revisión de algunos resultados obtenidos hasta la fecha.

A la hora de estudiar las técnicas aplicadas en el campo de ML para la detección de tumores, es posible encontrar artículos con varios enfoques, aunque se puede apreciar un uso predominante de técnicas como CNN diseñadas manualmente o SVM como en Ragab et al. [2019]. Podemos encontrar artículos previos que reúnan diferentes técnicas y algoritmos utilizados en Ganesan et al. [2013b].

El uso de técnicas como *transfer learning*, la cual será usada en este proyecto, ha llegado a presentar resultados excelentes, como es en el caso de Garcia-Dominguez et al. [2020] donde se puede ver un estudio con todas las redes disponibles para aplicar *transfer learning* y los resultados obtenidos en cada caso.

Por otro lado se encuentran pocos artículos en los cuales se haya aplicado Auto-Keras, la otra técnica a usar en este proyecto, a imágenes con el objetivo de detectar

tumores. Sí es posible encontrar algunos casos como Marconi Ramos et al. [2019] donde, además de tratar con imágenes no relacionadas con las CT, los resultados obtenidos son pésimos.

No se ha encontrado ningún artículo en el cual se realice un estudio sobre el funcionamiento de AutoKeras al tratar imágenes mamarias de tipo CT.

A continuación, en la tabla 2.1 se muestra un resumen con todas las técnicas de *imaging* usadas en la detección de tumores, las cuales indican la posibilidad del desarrollo de sistemas basados en extracción de características en imágenes pertenecientes al campo del DL.

Tumor	Técnicas									
	sigmoidoscopia	Colonoscopia	TC	Ecografía	Biopsia	PET	NMR	EUS	Termografía	Radiografía
Colorrectal	X	X	X							
Hígado			X	X						
Leucemia					X					
Linfoma			X			X				
Páncreas			X				X	X		
Piel					X					
Próstata					X		X	X		
Pulmón			X		X					X
Riñón			X				X			
Mama					X		X		X	X
Tiroides				X						
Útero				X	X					
Vejiga					X					

Tabla. 2.1: Técnicas en las cuales se realiza un diagnostico basado en imágenes.

Relación técnica-articulo

Para finalizar el capítulo se va exponer una lista en la cual se enlazará cada técnica utilizada para la detección de tumores con los artículos de referencia en los cuales se utilizan los algoritmos expuestos en el apartado 2.3

- Ganesan et al. [2013a] se tratan Arboles de decisión, redes neuronales, algoritmos gausianos, red neuronal difusa, SVM y regresiones lineales.
- En Sha et al. [2020] se utiliza *transfer learining* y CNN
- En Pradhan and Chawla [2020] se mencionan prácticamente todos los clasificadores expuestos anteriormente.
- Bataineh [2019] se exponen MLP, KNN, CART, NB, SVM
- Belciug and Gorunescu [2012] Se utilizan modelos híbridos usando clasificadores simples.

- Hussain et al. [2018] expone también modelos híbridos donde suelen combinarse con SVM.
- Juan Andres Cadena [2018] Se utilizan redes probabilísticas

Capítulo 3

OBJETIVOS

En esta sección se expondrán aquellos hitos que se buscan alcanzar a lo largo este proyecto. Internamente, este capítulo se dividirá en categorías para que la compresión y visualización de los mismos sea más clara.

Antes de comenzar es necesario aclarar que algunos de los objetivos no serán apreciables en la memoria ya que persiguen una motivación personal que permanece ajena a la búsqueda de los objetivos indicados en los párrafos siguientes, como es la calidad del sistema. En el apartado 6.5 del capítulo final de esta memoria se incluirá una reflexión estudiando si se han alcanzados los objetivos personales marcados a lo largo de este proyecto.

3.1. Objetivo general

El objetivo primordial, y aquel que se busca alcanzar con el desarrollo de este proyecto, es la creación de un modelo de CNN que permita la detección de anomalías en mamas con la intención de minimizar la carga de trabajo de los profesionales encargados de realizar esa clasificación de forma manual, permitiendo de esta manera que se agilice el proceso de diagnóstico donde se vean involucradas imágenes de tipo CT.

La motivación del objetivo general reside en el interés de agilizar todo el trámite necesario para el diagnóstico de un tumor lo más rápido posible, ya que como se indicó mediante la supervivencia a cinco años en la introducción, un aspecto clave es la fase de detección de un tumor y si en un caso concreto la citación de una persona para realizarse una TC se tiene que postergar demasiado a causa de lista de espera, el desenlace podría ser fatal.

3.2. Objetivos específicos

Una vez definido el objetivo general del proyecto, se procede a a definir los objetivos específicos, los cuales serán metas a alcanzar en el proceso de desarrollo del proyecto.

Ciencia de datos

Obtención de una visión general del campo de Ciencia de datos y los métodos de minería de datos esenciales, en particular de redes neuronales artificiales y técnicas de aprendizaje profundo.

Al comienzo de este proyecto no se tiene prácticamente ningún conocimiento previo en este campo, salvo nociones básicas en el campo del procesamiento del lenguaje natural.

En todo el desarrollo la intención respecto a este campo es formar las bases que permitan no solo completar este trabajo si no también adquirir los conocimientos necesarios para en el futuro poder realizar otros proyectos relacionados con la materia.

Adecuación de un conjunto de datos

- Preparación de un conjunto de datos representativo que permita implementar un detector capaz de identificar la presencia de tumores.

- Preprocesamiento mínimo adecuado para conseguir una vista objetiva de la comparativa a realizar.

  El objetivo principal de esta parte será partir de un conjunto de datos muy reducido para poder aplicar técnicas de aumento del conjunto de datos como *data augmentation* y ver la repercusión que tiene en los resultados y técnicas de procesamiento de imágenes como la interpolación para comprender la influencia del estado de las imágenes y sus propiedades sobre al final de la experimentación. Todas estas pruebas que se han señalado buscan comprender todo el proceso desde el punto de partida del conjunto con el que se empieza a trabajar hasta que se tiene un conjunto usable para el entrenamiento de una red neuronal.

Experimentación

De igual manera que en el apartado anterior se buscaba adquirir los conocimientos sobre el proceso de creación de un conjunto de imágenes, en esta parte se buscará adquirir conocimientos sobre el proceso de entrenamiento de una red neuronal. Se busca conocer exactamente qué es necesario en el proceso de entrenamiento y qué repercusión tiene cada paso en los resultados finales.

El objetivo secundario más importante después del principal se comprende con la experimentación ya que se busca alcanzar una mejora no solo en la velocidad de diagnóstico si no también en la calidad, y es que es necesario que el sistema presente rendimiento prácticamente perfecto debido a que para delegar la clasificación y diagnóstico de imágenes de tipo TC en el sistema es indispensable que este no cometa errores.

Se ha optado por seguir los tres procedimientos indicados anteriormente con el objetivo de realizar una comparación entre ellos de no solo los parámetros indicados si no también del tiempo que conlleva entrenar cada una de las CNN generadas. Esto se debe a que se busca no solo comprender el funcionamiento de las CNN, sino también de técnicas más avanzadas.

Así pues, después de todas las valoraciones expresadas el resumen de este apartado sería el siguiente.

- Experimentación y definición de los parámetros pertenecientes a los elementos de preprocesamiento indispensables indicados (resolución, interpolador...).
- Definición de la CNN posteriormente utilizada para la comparación de los procedimientos usados.
- Entrenamiento de los modelos generados por diferentes procedimientos utilizados y obtención de la calidad de clasificación para cada uno de ellos.

Comparativa y evaluación

- Se realizará una comparativa final de los tres procedimientos usados para generar el modelo de CNN buscado. Se estudiarán métricas no comunes en el estudio de calidad de una CNN como el tiempo invertido en el desarrollo, y recursos necesarios además de las ampliamente conocidas como *accuracy*, *precision*, *recall*.

- Selección del procedimiento que otorgue los mejores resultados, teniendo en cuenta los elementos descritos en el comparativa mencionada.

En la comparativa final se busca a nivel personal no solo los objetivos indicados en la lista superior. Debido a que nunca se ha realizado ninguna memoria con estas características es importante comprender cómo definir los parámetros que determinen la elección del mejor modelo y qué pautas son necesarias seguir a la hora de realizar una comparativa. También se busca invitar y motivar a la continuación de este proyecto según las lineas establecidas con el fin de seguir profundizado en el campo de la detección de tumores mediante CNN, haciendo posible en algún momento un modelo usable en la vida real y en casos prácticos reales.

Capítulo 4

MATERIALES Y MÉTODOS

En este capítulo se describirán todos los procesos y técnicas que han contextualizado la realización de los experimentos, así como todos los recursos que han sido necesarios para el desarrollo del proyecto. Dicho pues, se podrán encontrar tanto los criterios establecidos para la selección de la base de datos, las técnicas aplicadas al conjunto de imágenes para realizar el preprocesamiento necesario o herramientas a nivel de software y bibliotecas usadas.

4.1. Procedimientos usados

Para la realización de este trabajo se han tenido en cuenta tres procedimientos para alcanzar un modelo con el objetivo de realizar la clasificación de las imágenes mediante una red neuronal convolucional. El objetivo es obtener una comparativa entre los resultados obtenidos por parte de estos modelos, así como las horas dedicadas a cada uno de ellos para observar en cuál merece la pena invertir más tiempo. Los tres procedimientos que se van a desarrollar son:

- **CNN diseñada a mano**: el objetivo será encontrar la configuración que genere el mejor resultado, invirtiendo un número de horas similares a las necesarias para generar el modelo en AutoKeras. La idea es aclarar si es rentable invertir tiempo en la construcción de un modelo diseñado a mano, con las imágenes que se trabajan en este proyecto, habiendo soluciones como las dos siguientes descritas.
- **CNN generada mediante AutoKeras**: esta herramienta permite obtener la configuración óptima para una red de forma automática, existe la posibilidad de que

el modelo se estanque en un óptimo local debido a las búsquedas limitadas que se tendrán que establecer como consecuencia de la capacidad de cómputo limitada. Autokeras es una librería de AutoML (*Automated Machine Learning*) que permite a usuarios sin experiencia en la materia alcanzar complejos sistemas gracias a su búsqueda automática de la arquitectura de la red y sus hiperparámetros.

- **Red neuronal convolucional obtenida mediante *Transfer Learning***: En el caso de este proyecto, se realizarán pruebas con los tres modelos más comunes de Keras los cuales son VGG16, ResNet50 y Xception. Estas redes se usarán como punto de partida, y sobre cada una de ellas, se añadirá una capa de clasificación binaria debido a la naturaleza binaria de la distribución de clases en el conjunto de imágenes de trabajo.

  - VGG16: se trata de una red indicada para la clasificación grandes volúmenes de imágenes mediante CNN muy profundas. Esta red se compone por seis divisiones donde cada una de las cinco primeras de 2 a 4 capas convolucionales y una última capa de *pooling*. La última división está compuesta por tres capas densas.

  - ResNet50: Como su nombre indica, está compuesto por 50 capas de diferentes tipos las cuales están separadas en cinco divisiones, donde cada una de ellas está compuesta por un bloque de capas convolucionales y otras de identificación. La diferencia de esta red respecto a la anterior es que se aplica *skip connection* que permite transmitir información Entre neuronas separadas por más de una capa.

  - Xception: es el modelo superior a Inception. Al igual que las anteriores es usada para la clasificación de imágenes mediante CNN. Este diseño también aplica *skip connection* al igual que ResNet50, pero presenta la mayor arquitectura de todas tal y como se observa en la figura 4.1.

Por último, se recurrirá a la técnica de *fine-tunning* para terminar de entrenar cada una de las redes con nuestro conjunto de imágenes. finalmente se intentará mejorar los resultados anteriores recurriendo a esta técnica que permite usar un modelo preentrenado con un alto número de imágenes y que está desarrollado para la resolución de un problema concreto, aplicándolo a otro problema,

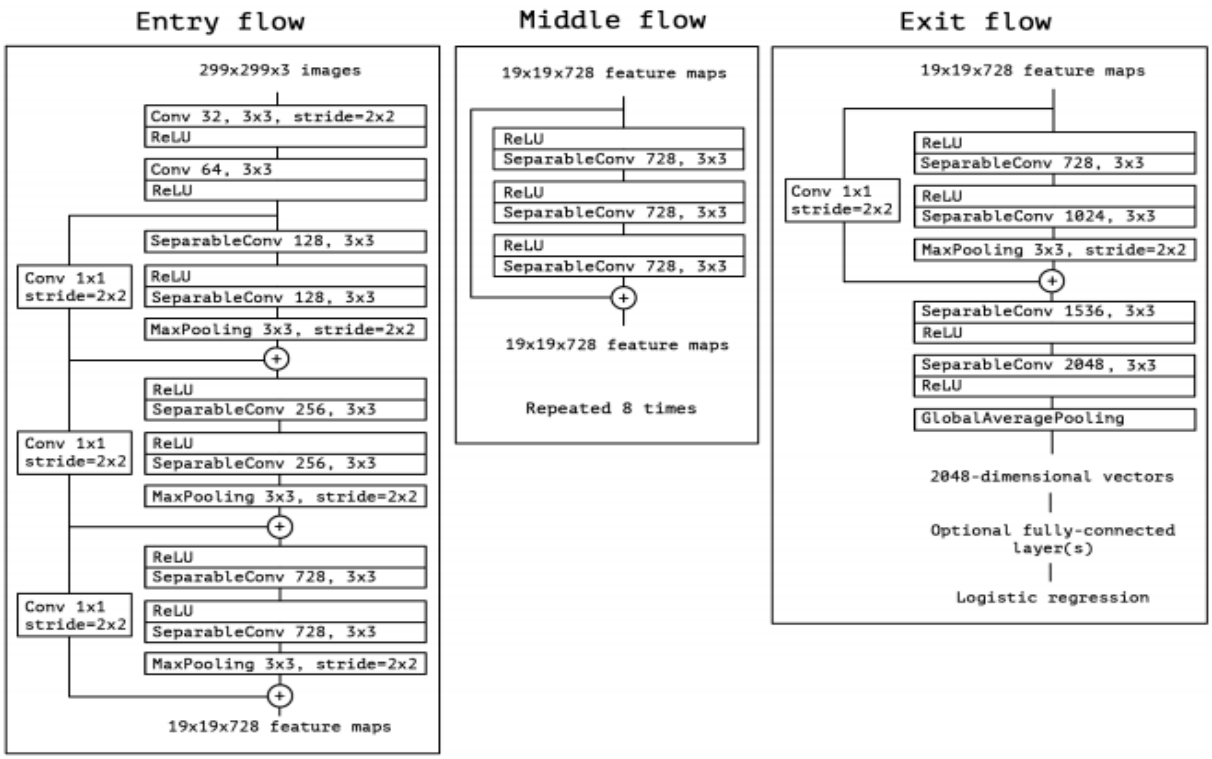

Figura 4.1: Arquitectura de la red Xception Chollet [2016]

4.2. Conjunto de datos

El objetivo dentro de la búsqueda de un conjunto de datos era encontrar uno de tamaño reducido para poder realizar diversos experimentos y comparar los resultados en función del número de imágenes disponibles aplicando técnicas como *data augmentation*. Para ello, se irán realizando técnicas de ampliación de conjuntos de datos hasta conocer el número mínimo necesario para generar un clasificador que aporte buenos resultados.

Se han tenido en cuenta tres bases de datos a la hora de seleccionar cuál se usará para entrenar los diferentes procedimientos para crear el modelo objetivo.

- **DDSM, *Digital Database for screening mammography***: la alta cantidad de datos que contiene este conjunto no concuerda con los objetivos planteados de estudiar la influencia de diferentes tamaños de un mismo conjunto de datos. En total, este conjunto de datos consta de 2620 casos, donde cada caso contiene en torno a diez imágenes en formato JPG, pero como se ha indicado previamente, el objetivo es tomar como punto de partida un conjunto reducido.
- ***Mammographic Mass Dataset*** del instituto de radiología de la universidad de Erlangen-Nuremberg: contiene una cantidad de imágenes no muy extensa y po-

dría ser un conjunto de datos apropiado para el objetivo que se busca. La base de datos está compuesta por 961 instancias, seis atributos por imagen, aunque debido a que no se ha podido acceder al contenido, no se conoce el número de clases iniciales. No se ha seleccionado finalmente debido a que fue necesario solicitar a los creadores de dicho conjunto de datos, y por desgracia, no se recibió respuesta.

- **Mini-MIAS *database of mamograms***: finalmente se tuvo en cuenta esta base de datos donde encontramos un conjunto de imágenes muy reducido, el cual se compone de 322 imágenes separadas en tres clases con imágenes de resolución 1024x1024 y formato PGM, el cual representa una escala de grises.

Selección del conjunto de datos

Finalmente, se seleccionó la última base de datos (Mini-MIAS) por la flexibilidad que aporta a la hora de la experimentación gracias a su reducido número de imágenes puesto que permite analizar el proceso de aprendizaje de la red desde el punto de partida de un conjunto de datos muy escaso. La cantidad de clases que presenta también favorece esta elección debido a la estandarización en dos clases que se persigue puesto que el proceso de reducción resulta mucho más sencillo al haber tan poca diferencia entre las clases iniciales y las perseguidas. Por último, señalizar que esta es la única base de datos a la que se pudo acceder sin necesidad de ningún software adicional. La distribución de imágenes iniciales que contenía la base de datos era la siguiente.

Tipo de imagen	N.º de imágenes	Porcentaje de imágenes sanas
Sanas	208	
Benignas	63	64.596 %
Malignas	51	
Total	322	208 sanas / 114 anormales

Tabla. 4.1: Distribución de imágenes inicial

4.3. Adecuación del conjunto de datos

Inicialmente, el conjunto de datos presentaba una estructura dividida en tres clases, sin embargo, nuestro objetivo se basa en una clasificación binaria debido a que se busca encontrar anomalías de cualquier tipo, lo cual implica una división dos partes del sistema BI-RADS, teniendo como clases BI-RADS 1 por un lado y BI-RADS del 2 al 6

por otro, por lo que el primer paso antes de preprocesar las imágenes fue reordenarlas reduciendo las clases a la presencia de anomalías, es decir, tras la estandarización del conjunto de datos el contenido de este estaba separado en dos clases, aquellas mamas que estaban completamente sanas, y aquellas que sí presentaban alguna anomalía, pudiendo esta ser de carácter benigno o maligno. Tras la adecuación del conjuntos de datos a los requisitos de las pruebas, se procedió al tratamiento de las imágenes.

Tipo de imagen	N.º de imágenes	Porcentajes de imágenes sanas
Sanas	208	64.596 %
Anómalas	114	
Total	322	

Tabla. 4.2: Distribución de imágenes tras normalizar

4.3.1. *Data augmentation*

Inicialmente el conjunto de imágenes dispone de un número bastante escaso como se indicó inicialmente, por lo que tras realizar pruebas con una CNN básica se descubrió que la red aprendía demasiado rápido.

La solución a este problema es realizar *Data Augmentation* sobre el conjunto actual de imágenes, la cual es una técnica mediante la que es posible obtener múltiples imágenes a partir de una sola imagen utilizando transformaciones, como por ejemplo rotaciones o desplazamientos. Además, se puede aprovechar la utilización de esta técnica para obtener una distribución homogénea por clases del conjunto de imágenes, es decir, que exista el mismo número de imágenes en las dos clases existentes.

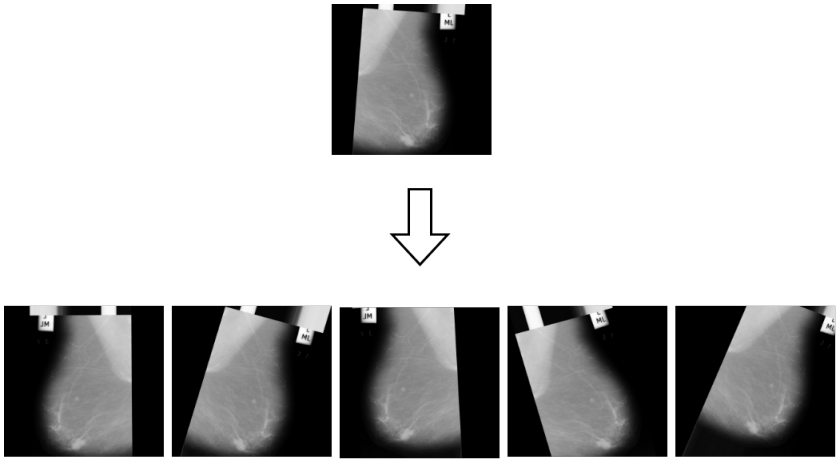

Figura 4.2: Ejemplo de *Data Augmentation*

El objetivo es generar un número de imágenes que presenten un grupo no muy grande debido los criterios de selección de la base de datos que se indicaron al inicio, por lo tanto, se fijará una segunda fase en la cual la base de datos estará compuesta por en torno a 2000 imágenes. La distribución después de aplicar esta técnica quedaría de la siguiente forma:

Tipo de imagen	N.º de imágenes	Porcentajes de imágenes sanas
Sanas	1200	50.18 %
Anómalas	1191	
Total	2391	

Tabla. 4.3: Distribución de imágenes tras aplicar *Data augmentation*

Finalmente los resultados obtenidos trabajan sobre esta base de datos debido a que los resultados presentados en el siguiente capitulo aclaran que no es necesario volver a aplicar *Data Augmentation*.

4.4. Organización del conjunto de imágenes

En este apartado se relatará de manera breve cuál ha sido la forma de trabajar con el conjunto de datos en los experimentos realizados.

Es necesario recalcar que para desarrollar una CNN se subdivide el conjunto de datos en tres partes. La subdivisión ha de realizarse de manera que exista una aleatoriedad en los datos, pero manteniendo la proporción entre las imágenes de cada clase que compone cada subdivisión. Estas subdivisiones suelen ser tres, aunque es posible reducirlas a dos gracias a Keras.

En primer lugar, se debe aclarar que todos los resultados mostrados en el apartado siguiente se han realizado manteniendo la siguiente proporción:

- **Conjunto de entrenamiento**: este será el conjunto usado para el aprendizaje de la red, en el cual se extraerán las características necesarias para clasificar nuestro conjunto de imágenes en caso de que el aprendizaje sea exitoso. El 60 % de las imágenes irán destinadas a este grupo.
- **Conjunto de validación**: a la hora de entrenar, en cada época es necesario validar que las características extraídas son las correctas. Este grupo sirve para que dinámicamente se corrija el entrenamiento de nuestra red. También sirve

para aportar una idea de la *accuracy* una vez el modelo haya sido entrenado. Este conjunto estará compuesto por un 20 % del conjunto de datos.

- **Conjunto de test**: por último, contamos con el conjunto de datos mediante el cual obtendremos la calidad de nuestro modelo, siempre y cuando el número de imágenes utilizadas para la clasificación sea suficiente para evitar obtener una precisión no representativa. El 20 % de las imágenes se destinarán a este grupo.

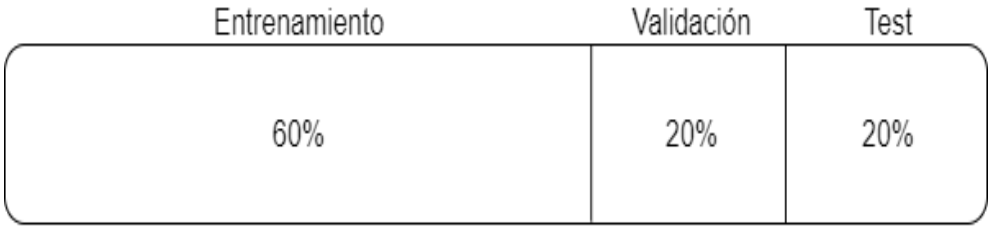

Figura 4.3: Distribución usada para el desarrollo de los experimentos

Como se ha indicado, es posible trabajar sin conjunto de validación en este caso concreto debido a que la propia librería divide internamente el conjunto de entrenamiento en entrenamiento y validación, aun así, en este proyecto, se ha trabajado con los tres subconjuntos.

4.5. Procesamiento del conjunto de imágenes

En el caso de las tomografías computarizadas, contamos con imágenes con una resolución de 1024x1024 y relación de aspecto 1:1. Son imágenes con formato PGM, el cual se trata de un formato compuesto en escala de grises, aunque, internamente, cada imagen cuenta con los tres canales de color RGB.

Las imágenes de CT son, en esencia, una imagen de rayos X generada a través de una maquina rotatoria especializada. Estas imágenes presentan una apariencia de escala de grises, aunque como se ha indicado previamente no lo son. El objetivo de su apariencia es facilitar la detección sencilla de elementos anómalos.

Posteriormente, se han reescalado las imágenes a un tamaño de 256 píxeles, manteniendo la proporción inicial 1:1, debido a que la resolución de partida (1024px) es demasiado elevada para que la red trabaje con ellas. Como solución CV2 OpenCV [2021] nos permite trabajar con varios interpoladores. Para prevenir el caso de que existan imágenes con una resolución menor a 256x256 píxeles, se van a utilizar todos los interpoladores disponibles, ya que como se indicó en 2.6 algunos de esos interpoladores están centrados en el aumento de resolución y otros en la disminución.

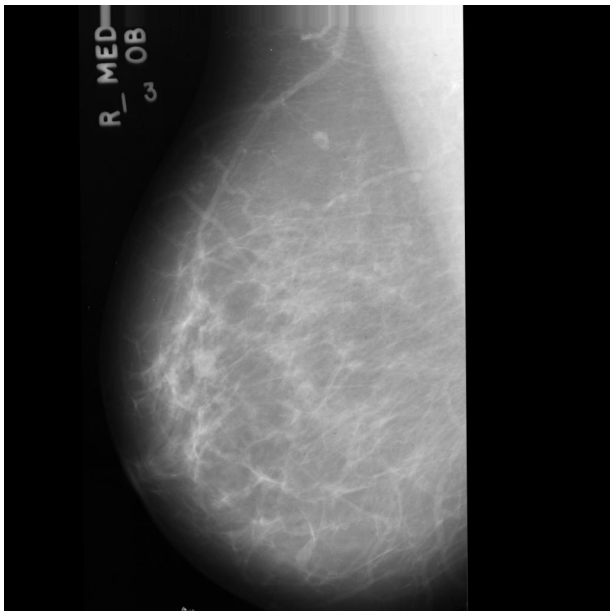

Figura 4.4: Imagen de una mama sin reescalado

De entre todos los interpoladores estudiados para realizar el reescalado, se ha recurrido a un interpolador lineal. Tanto la selección del interpolador como de la resolución de imagen con la que se realizarán los experimentos se han seleccionado atendiendo a los resultados que se mencionan en el próximo capitulo.

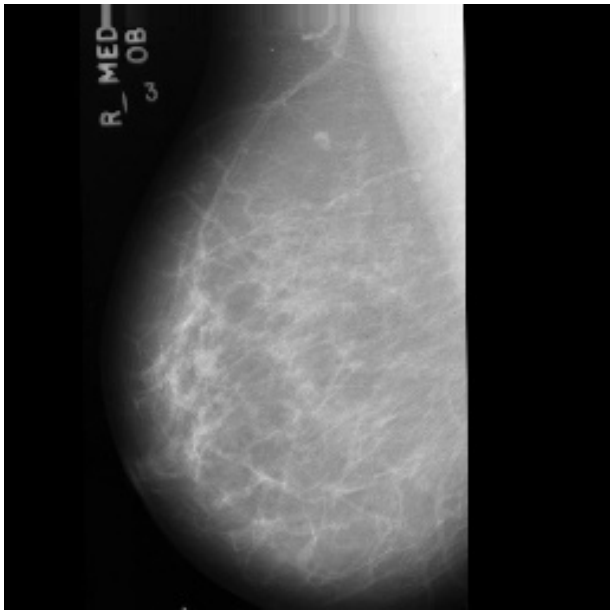

Figura 4.5: Imagen de una mama reescalada

4.6. Herramientas adicionales

Para la realización de este proyecto se ha seleccionado una serie de herramientas que faciliten el trabajo a realizar y generen unos resultados con una presentación correcta. Se van a mencionar las principales herramientas utilizadas, señalando qué función han cumplido dentro del proyecto.

Familia Office

Durante todo el proyecto se han llevado diversos cuadernos de bitácora que plasman el trabajo que se ha ido realizando cronológicamente. Estos cuadernos han sido escritos en archivos DOCXS pertenecientes al software Word perteneciente a la familia.

En segundo lugar, se ha utilizado Excel para realizar todos los cálculos necesarios a la hora de tratar los resultados generados por los diversos experimentos debido a su versatilidad y al fácil acceso debido a que pertenece al paquete, además de la experiencia previa sobre este software.

Overleaf

Para la redacción de la memoria final se ha recurrido a la herramienta Overleaf Overleaf [2021]. La motivación reside en la facilidad para establecer el estilo a seguir, fijado por la Escuela Politécnica de Jaén. Esto se debe a la posibilidad de tomar como punto de partida una plantilla que ya cuente con un estilo preestablecido. Por otro lado, la facilidad que ofrece este software para trabajar con aspectos matemáticos convierte a esta herramienta en la opción más preferible.

PyCharm

Por último, la primera tarea a la hora de seleccionar la herramienta de desarrollo fue establecer el lenguaje de programación que se iba a usar. Finalmente se seleccionó Python debido a la gran flexibilidad que ofrece gracias a su amplia gama de bibliotecas que ofrecen la posibilidad de trabajar con muchos ámbitos independientes. Por otro lado, Python es el lenguaje predilecto para el desarrollo de redes neuronales al trabajar con programación de alto nivel, como es el caso.

Una vez seleccionado el lenguaje a usar, se procedió a la elección del framework, resultando esta búsqueda en la elección de PyCharm PyCharm [2021] debido a la experiencia previa con este software, lo cual facilitó los primeros pasos en el desarrollo del proyecto. Conjuntamente, se han utilizado dentro del proyecto creado archivos PY para el código que conlleva un largo periodo de ejecución, y Jupyter, el cual es una herramienta que permite la ejecución por bloques, para aquellos códigos que necesitaban cambios frecuentes o que interesaba la ejecución de solo un fragmento del código.

4.6.1. Bibliotecas

Para la realización de este proyecto, han sido necesarias bibliotecas para Python tanto para el preprocesamiento de imágenes, como para la construcción de las redes neuronales. A continuación, se van a nombrar las bibliotecas estudiadas para las tareas indicadas, y finalmente, aquellas que han sido seleccionadas.

Procesamiento de imágenes

Dada la poca cantidad de preprocesamiento que requieren las imágenes, y la simplicidad de los mismos, la selección de la biblioteca para trabajar se ha basado en la experiencia previa con la herramienta. La librería que cumple con esta característica es OpenCV.

Se trata de una librería ampliamente conocida debido a su capacidad para la modificación y extracción de información respecto a las imágenes, por otro lado, se trata de una biblioteca Open Source, lo cual ha decantado más la balanza a favor de la selección de esta librería.

Redes neuronales

En el caso de la librería a utilizar para el desarrollo de las redes necesarias encontramos varias alternativas. A diferencia del procesamiento de imágenes, apenas se cuenta con experiencia previa a este proyecto. Las librerías que se han tenido en cuenta son:

- scikit-learn scikit learn [2021]: Se trata de una librería para Machine Learning y

análisis de datos. La diferencia respecto al resto de las que se van a estudiar es su gran versatilidad debido a que ofrece la posibilidad de trabajar con numerosas técnicas de aprendizaje automático, como arboles binarios, maquinas vectores soporte o bosques de clasificación. Sin embargo, puesto que el objetivo de este proyecto es la comparativa de CNNs, se descarta como opción útil.

- TensorFlow(Keras) Abadi et al. [2015]: Es una de las librerías más potentes en el campo de aprendizaje profundo, se trata de una herramienta bastante potente, desarrollada por Google que ofrece los recursos necesarios para la construcción de las redes neuronales que se persiguen. Tensorflow cuenta con una biblioteca de más alto nivel llamadas Keras con la que resulta extremadamente sencillo construir redes de múltiples tipos, además de ofrecer recursos de redes preentrenadas u optimizadores de la estructura de la red, los cuales son necesarios para realizar este proyecto.

- PyTorch Paszke et al. [2019]: Otra opción posible, y que ofrece al igual que Tensorflow los elementos necesarios para desarrollar redes neuronales de numerosos tipos. Se trata de una versión desarrollada por Facebook. A diferencia de Tensorflow, en esta biblioteca es posible la definición de modelos dinámicos, sin embargo, para este proyecto no serán necesarios.

Finalmente se ha seleccionado Keras como librería de trabajo para el desarrollo de los modelos de redes neuronales que se buscan debido a la difusión con la que cuenta dentro del campo del aprendizaje profundo. Por otro lado, la mayoría de la documentación encontrada pertenece a esta librería. El último factor que se ha tenido en cuenta es la escasa pero existente experiencia en comparación con PyTorch.

4.7. Equipo usado para la experimentación

Puesto que el objetivo de este proyecto es encontrar el modelo óptimo teniendo en cuenta trabajo y tiempo invertidos, es necesario exponer los componentes del equipo con el que se han realizado los experimentos para generar una contextualización justa de todas las pruebas realizadas. En la tabla inferior se indicarán todos los componentes relevantes para la realización de los experimentos.

Componente	Modelo
CPU	Intel Core i5 4570 22nm
Memoria Ram	12GB DDR3 3600Hz
Disco duro	Kingston A400 SSD SA400S37/480G
GPU	Nvidia GFORCE GTX 1060 6GB by MSI
Placa base	Asus H87M-E

Tabla. 4.4: Hardware perteneciente al equipo de experimentación

Capítulo 5

RESULTADOS

5.1. Introducción

En este capítulo se describirán todos los experimentos realizados necesarios para alcanzar el objetivo del proyecto, así como se presentarán los resultados obtenidos. Inicialmente se tratarán todos los experimentos para fijar los parámetros a usar en los tres modelos de CNN, es decir, se estudiará si el conjunto de imágenes inicial es suficiente, qué resolución de imagen otorga mejores resultados y qué interpolador realza con mayor calidad las características de cada imagen. Posteriormente se presentarán los experimentos relativos a los modelos finales generados, así como el proceso hasta alcanzarlos.

5.2. Estudio sobre el tamaño del conjunto de imágenes

En este apartado vamos a comprobar si el tamaño inicial del conjunto, el cual contiene 322 imágenes, es suficiente para realizar el resto de experimentos. Para ello, se va a trabajar con la estructura de una CNN simple. La primera prueba consistirá en un estudio de la *accuracy* obtenida en el entrenamiento. Para ello, la red con la que se realizarán las pruebas será la siguiente:

La red neuronal expuesta en la figura 5.1 tendrá como función de activación ReLU para todas las capas debido a que como se explicó en el capítulo de antecedentes, es la más usada debido a los buenos resultados que suele aportar. Respecto la cantidad

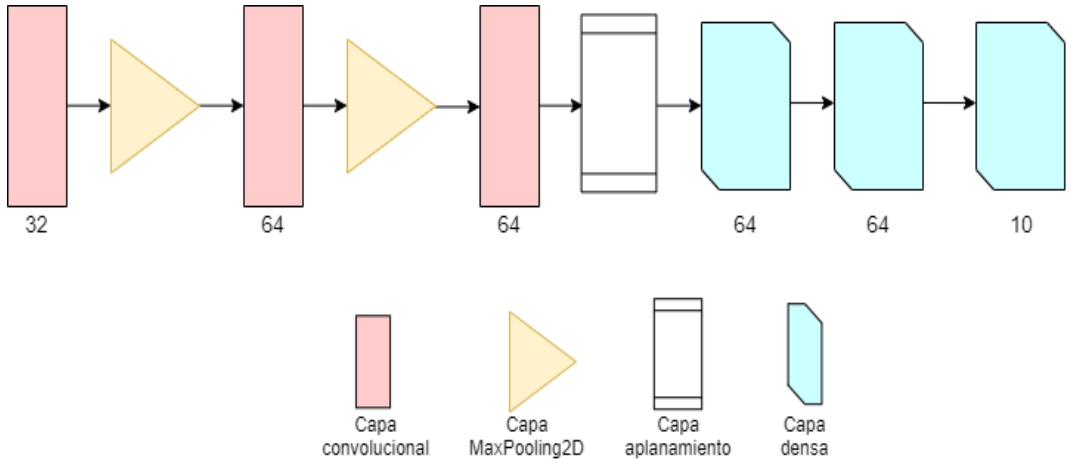

Figura 5.1: CNN básica. Los números presentes bajo las capas indican la cantidad de filtros aplicados por capa en el caso de las convolucionales y numero de neuronas en el caso de las densas

de filtros aplicados en cada uno de los filtros y la cantidad de neuronas por cada capa densa no siguen un objetivo concreto más allá de comparar el tamaño con los generados por el resto de procedimientos.

Los experimentos realizados para estudiar el tamaño del conjunto de datos consistirán en ejecutar diez veces todo el proceso de definición de la estructura de la red (sin modificarla), entrenamiento y clasificación con el objetivo de estudiar la variabilidad de los datos entre ejecuciones sin modificar ningún elemento. Tras haber realizado las diez ejecuciones, los parámetros de rango de datos y desviación típica, ambos respecto a *accuracy*, otorgarán la visión de si el sistema es estable, es decir, que no exista una variación muy significativa de resultados entre ejecuciones.

Épocas	Accuracy media (entrenamiento)	Rango (test)	Accuracy media (test)	σ^2 (test)
30	71.96 %	[47.61, 95.23]	72.15 %	14.85
40	91.76 %	[61.90, 78.57]	69.25 %	6.48

Tabla. 5.1: Resultados previos a realizar *Data Augmentation*

Los resultados obtenidos con el conjunto inicial en la tabla 5.1 han sido generados sobre solo 64 imágenes (20 % del conjunto test), y como podemos observar, los resultados son muy variables. Las desviaciones típicas presentadas son demasiado elevadas, lo cual nos indica, junto con el rango, que no estamos ante un clasificador estable. Para evitar resultados erróneos debido al bajo número de imágenes con el que se han obtenido, se va a recurrir a la técnica de *Data Augmentation*, la cual se explicó en el capítulo anterior, con el objetivo de aumentar el conjunto de imágenes hasta alcanzar una cifra de en torno a 2000 imágenes.

Épocas	Accuracy media (entrenamiento)	Rango (test)	Accuracy media (test)	σ^2 (test)
30	98.86 %	[38.07, 52.71]	43.78 %	5.24
40	99.88 %	[50, 61.9]	50.19 %	3.86

Tabla. 5.2: Resultados posteriores a realizar *Data Augmentation*

Tras realizar los mismos experimentos, pero con el conjunto de datos ampliados, se puede apreciar en la tabla 5.2 unos resultados de menor calidad en lo que a *accuracy* se refiere, sin embargo, los datos se estabilizan mucho antes y presentan menor dispersión. Además los cálculos de *accuracy* están realizado sobre un conjunto de 478 imágenes (20 % del conjunto test tras *Data augmentation*), por lo que existe la certeza de que el valor es mucho más representativo.

Los resultados obtenidos indican que influye positivamente recurrir a *Data augmentation* para resolver este tipo de situaciones ya que interesa tener un sistema solido que permita resultados estables con el objetivo de que exista la certeza de tener una *accuracy* fija, y aunque esta haya haya bajado, se debe tener en cuenta que las pruebas han sido realizadas con una CNN muy simple, por lo que los resultados no se esperaba que fueran prometedores.

5.3. Selección del interpolador

A la hora de preprocesar las imágenes, la única técnica aplicada ha sido la interpolación para ajustar la dimensión de original de cada imagen a la de entrada de la red, por lo que el siguiente paso antes de probar modelos más complejos, y una vez conocemos el conjunto de datos, con lo que se trabajará será la comprobación de cuál de los interpoladores disponibles en CV2 ofrece un mejor rendimiento.

Este experimento ha consistido en un total de diez ejecuciones para cada interpolador con cuatro cantidades de épocas diferentes, esto se debe a que también se intentará descubrir cuánto afecta el proceso de aprendizaje de la red en la posición que ocupa cada interpolador respecto al resto. En la siguiente tabla se mostrará un ranking con las medias de *accuracy* en test obtenidas y una última fila que mostrará la media de posiciones obtenida en los rankings de cada época para cada interpolador.

Como se puede observar en la tabla 5.3, el interpolador lineal es aquel, aunque por poco, que otorga los mejores resultados, de ahora en adelante, para todas las pruebas se fijará el interpolador lineal como algoritmo de reescalado para el conjunto

Épocas	Más cercano	Lineal	Área	Cúbico	Lanczos4
5	2.8	2.7	3.1	3.2	3.1
10	2.7	2.8	2.6	4.2	2.6
15	2.8	2.2	4.1	3.2	2.7
20	2.3	2.8	2.4	3.5	3.8
Media de ranking	10.6	**10.5**	12.2	14.1	12.2

Tabla. 5.3: Ranking sobre el mejor interpolador

de imágenes.

5.4. Selección de la resolución de imagen

Como se indicó inicialmente, la resolución de las imágenes es demasiado elevada tal y como se indica en Sabottke and Spieler [2020] por lo que, ante los recursos limitados del equipo de experimentación, se tomará como punto de partida la resolución de 256x256 píxeles. El siguiente paso consistirá en encontrar la resolución óptima para el conjunto de imágenes con el objetivo de encontrar los mejores resultados tomando como resolución la indicada como punto de partida. El primer experimento consistirá en una prueba con las mismas medidas realizadas que las seleccionadas en la búsqueda del mejor interpolador, es decir, una media de ranking. Para este caso solo se estudiarán las resoluciones de 128x128 y 256x256 debido a que en caso de que los mejores resultados sean generados por las imágenes de 256x256, no será necesario seguir realizando pruebas en torno a la resolución debido a que la resolución máxima factible por limitaciones de recursos del sistema es 256x256 píxeles.

Épocas	128px	256px
5	1.7	1.3
10	1.9	1.1
15	1.7	1.3
20	1.7	1.3
Media de ranking	7	**5**

Tabla. 5.4: Ranking sobre la mejor resolución

Los resultados obtenidos en la tabla 5.4 nos indican que la mejor resolución para trabajar es 256x256px, por lo tanto, de ahora en adelante, se fijará esta resolución como constante a la hora de continuar con los experimentos.

5.5. Procedimiento de experimentación

Una vez fijados todos los parámetros necesarios con los experimentos previos, solo resta demostrar cuál de los tres procedimientos a comparar otorga mejores resultados. El objetivo es estudiar el tiempo que ha llevado generar cada sistema, así como la *accuracy* sobre un conjunto de imágenes diferente del utilizado para el proceso de entrenamiento, obteniendo al final de las pruebas que se realicen el mejor modelo tiempo-*accuracy*.

En el campo de la medicina es preferible una falso positivo a un falso negativo debido a lo que conlleva. Un falso positivo se podrá aclarar con una prueba más exhaustiva y no tendrá más repercusión que la molestia del paciente por tener que acudir a ser inspeccionado nuevamente. Por otro lado, un falso negativo podría tener como repercusión el fallecimiento del paciente debido a la tardía detección del tumor. Así pues, se introducirán, además de *accuracy*, las metricas de *recall* y *precision* para estudiar la cantidad de falsos positivos y negativos que clasifica cada modelo.

5.5.1. Procedimiento generado por AutoKeras

Con el fin de conocer qué resultados se pueden alcanzar como máximo, manteniendo el tiempo de búsqueda invertido relativamente bajo, es preferible obtener el modelo generado por AutoKeras en primer lugar debido a los buenos resultados que suele producir tal y como se expone en Waring et al. [2020], así pues, con los parámetros fijados que se establecieron en los anteriores experimentos, se procedió a obtener el modelo.

El procedimiento con el cuál se realiza el entrenamiento de AutoKeras es extremadamente simple. Una vez se ha preprocesado el conjunto de imágenes, se separan en dos subconjuntos, entrenamiento, y test, en este caso no encontramos el conjunto de validación debido a que AutoKeras utiliza una parte de entrenamiento que considere oportuna para realizar la validación, por lo que en este caso, el subconjunto de imágenes de entrenamiento estaba compuesta por el 80 % de las imágenes totales. Lo único que resta es entrenar el modelo. AutoKeras encontrará la configuración óptima dentro del número de oportunidades concretas. Una vez entrenado, existirán tantas arquitecturas como oportunidades se hayan fijado, salvo que se encuentre un óptimo y AutoKeras decida detener el entrenamiento. Una vez disponemos de nuestro modelo entrenado se intentará clasificar el subgrupo de test.

Parámetro	Valor
Accuracy	97.48 %
Loss	0.06
Precision	95.58 %
Recall	99.58 %

Tabla. 5.5: Resultados generados por AutoKeras

Como se puede observar en 5.5, los resultados obtenidos son excelentes debido a la alta *accuracy* y la baja cantidad de falsos negativos y positivos, tomando el proceso de entrenamiento en torno a cinco horas de ejecución en un equipo personal con las características nombradas en el anterior capítulo. Estos valores obtenidos nos permitirán trabajar con la idea de cuál podría ser el óptimo global, por lo tanto, el resto de sistemas que a continuación se nombrarán se juzgarán bajo la métrica obtenida en este apartado.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_1 (InputLayer)         [(None, 256, 256, 3)]     0
_________________________________________________________________
cast_to_float32 (CastToFloat (None, 256, 256, 3)       0
_________________________________________________________________
normalization (Normalization (None, 256, 256, 3)       7
_________________________________________________________________
random_translation (RandomTr (None, 256, 256, 3)       0
_________________________________________________________________
random_flip (RandomFlip)     (None, 256, 256, 3)       0
_________________________________________________________________
efficientnetb7 (Functional)  (None, None, None, 2560)  64097687
_________________________________________________________________
global_average_pooling2d (Gl (None, 2560)              0
_________________________________________________________________
dense (Dense)                (None, 1)                 2561
_________________________________________________________________
classification_head_1 (Activ (None, 1)                 0
=================================================================
Total params: 64,100,255
Trainable params: 63,789,521
Non-trainable params: 310,734
_________________________________________________________________
```

Figura 5.2: Mejor modelo generado por AutoKeras

La estructura de la red mostrada en la figura 5.2 quedaría con las siguientes capas:

- `input_1`: se trata de la capa de entrada donde se indican los parámetros de la resolución de las imágenes y la cantidad de canales por las que están compuestas, además de otros elementos como el *batch size*
- `cast_to_float32`: transforma un tensor en un nuevo tipo, en este caso, en elementos de tipo coma flotante de 32 bits.
- `normalization`: modifica todos los elementos recibidos en una distribución centrada en cero con una distribución estándar uno.

- `random_translation`: realiza una traslación aleatoria de cada imagen, es decir, desplaza los píxeles y rellena el espacio que queda vacío resultante de la traslación.

- `random_flip`: se trata de otra transformación al igual que la capa anterior, solo que este caso, la operación aplicada sobre cada imagen es una rotación.

- `efficientnetb7`: en el estudio realizado en Tan and Le [2019] se demostró la arquitectura óptima hasta la fecha para un modelo de clasificación de imágenes. B7 es la estructura que mejores resultados aportó. Este tipo de modelos pueden cargar los pesos obtenidos con un preentrenamiento realizado en torno a ImageNet. Esta red se trata de una estructura diseñada para recursos con pocas capacidades por lo que presenta un tamaño y complejidad no muy elevadas.

- `global_average_pooling2d`: tras obtener la información de la capa anterior se está reduciendo la dimensionalidad de los datos para obtener datos de solo una dimensión que puedan ser comprendidos por la capa siguiente.

- `dense`: esta penúltima capa es la que generará el entrenamiento de la red neuronal. Aquí será donde se ajustarán los pesos durante el entrenamiento para luego poder realizar una buena clasificación.

- `classification_head_1`: se trata de la última capa en la cual durante el proceso de clasificación se generará el resultado por clases esperado.

Tras estudiar las diversas capas del modelo generado por AutoKeras se puede observar que en realidad tiene un diseño basado en *transfer learning* ya que las capas presentes son de transformaciones sobre las imágenes y una capa preentrenada con Imagenet.

5.5.2. Procedimiento generado con Transfer Learning

Para este caso, el preprocesamiento, como se indicó, sigue siendo el mismo que para AutoKeras. Como ya se explicó en el capítulo anterior, una vez cargada la información de cada una de las redes, y habiendo añadido la capa de clasificación binaria, solo resta aplicar *fine-tunning* para poder realizar el entrenamiento de la red generada con nuestro conjunto de entrenamiento. Para el caso de este procedimiento el número de épocas ha ido variando en función de la *accuracy* obtenida con el objetivo de evitar *overfitting*.

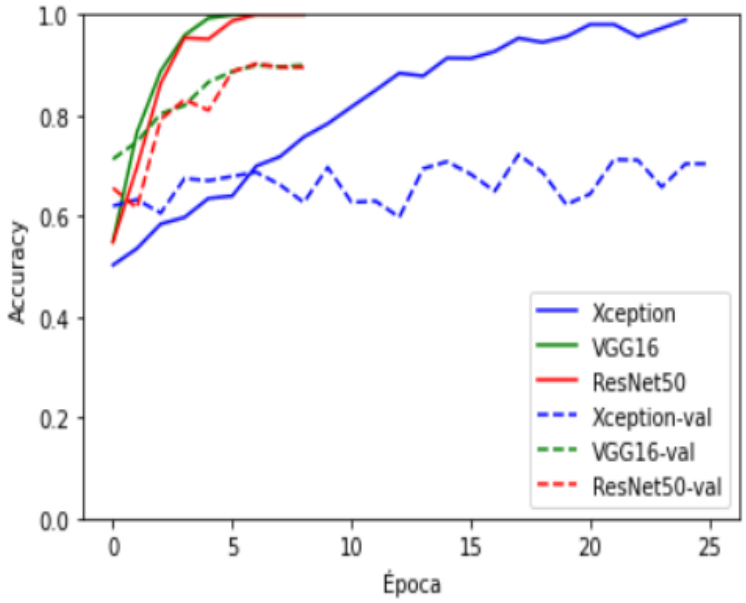

Figura 5.3: *accuracy* obtenida durante el entrenamiento de las redes

Se puede apreciar claramente la superioridad del uso de las redes VGG16 y ResNet50 sobre Xception, sin embargo, resta presentar la *accuracy* obtenida durante la fase de test. Para eliminar toda duda, aunque claramente se aprecie en las gráficas 5.3 y 5.4 la superioridad de las dos redes indicadas, se presentarán los resultados de las tres redes con las que se han realizado los experimentos.

Épocas	Red	Accuracy	Loss	Precision	Recall
9	ResNet50	90.32 %	0.35	87.54 %	94.14 %
	VGG16	92.63 %	0.25	92.18 %	93.72 %
26	Xception	67.16 %	1.38	64.35 %	77’82 %

Tabla. 5.6: Resultados generados con *Transfer Learning*

En vista de los resultados obtenidos en la tabla 5.6 se observa que el modelo generado con la red VGG16 como red base para realizar *Transfer Learning* otorga los mejores resultados en comparación con el resto de rivales. Es cierto que ResNet50 muestra una ligera superioridad sobre VGG16 respecto a los falsos negativos, sin embargo, es una diferencia demasiado baja como para seleccionar ResNet50 como la mejor opción, además, VGG16 es superior en el resto de parámetros, por lo tanto en la comparativa final, cuyo objetivo como se indicó previamente es aclarar qué procedimiento genera un modelo con mayor calidad, se seleccionará VGG16 como el mejor representante.

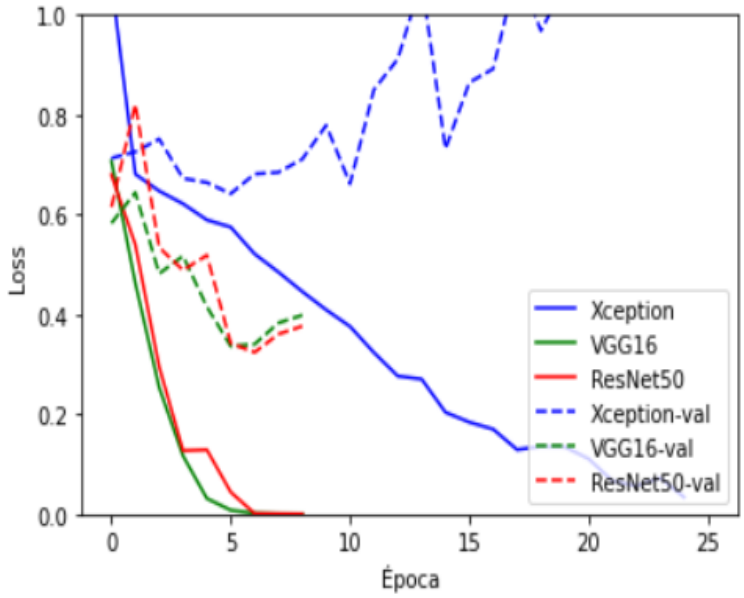

Figura 5.4: Loss durante el entrenamiento de las redes

5.5.3. Modelo generado manualmente

Como último método contamos con la red generada que se describió al comienzo de la experimentación. Una vez que se fijaron todos los parámetros ajenos a la propia estructura de la red, se procedió a la experimentación realizada bajo diez ejecuciones debido a que este es el único procedimiento en el que, manteniendo tanto el preprocesamiento de las imágenes como la estructura de la red, se obtenían resultados de *accuracy* con el conjunto test cuyo resultado variaba.

Rango de precisiones	Accuracy media	Desviación típica
[50, 61.9]	50.19 %	3.86

Tabla. 5.7: Resultados obtenidos con el procedimiento manual

Debido a los resultados de baja calidad que se mostraron anteriormente, como bien se aprecia en la tabla 5.7, respecto a este procedimiento se va a descartar para la comparativa final. Esto se debe a que por un lado la *accuracy* que presenta respecto al resto de procedimientos es bastante peor. Por otro lado, existe una alta variabilidad en esa métrica entre diferentes entrenamientos, lo cual no se da en los otros procedimientos. El tiempo de entrenamiento es semejante al necesario para *transfer learning* con VGG16. No tiene sentido seguir con este modelo hacia la comparativa final puesto que no resalta en ningún aspecto.

5.5.4. Comparativa de procedimientos

Por último, tras los procedimientos propuestos, se expondrán los resultados finales de manera que toda la experimentación realizada cobre forma. Así pues, en la tabla 5.8 se puede apreciar la diferencia de *accuracy* en torno al subgrupo de imágenes test.

Procedimiento	Accuracy	Loss	Precision	Recall	Tiempo invertido	GPU
AutoKeras	97.48 %	0.06	95.58 %	99.58 %	5 horas	40 %
VGG16	92.63 %	0.25	92.18 %	93.72 %	8 minutos	

Tabla. 5.8: Comparativa final de los mejores procedimientos

Antes de analizar la información presente en la tabla 5.8 se realizará una explicación breve de las variables que componen la tabla superior que no se han visto hasta ahora.

- **Tiempo invertido**: se trata del tiempo invertido en el entrenamiento y clasificación de la red. No puede exponerse de otra manera debido a que el tiempo necesario para la implementación es compartido por varios de los procedimientos debido a la formación previa que requieren, además, muchos de los conocimientos adquiridos a lo largo de la implementación de uno de los procedimientos han reducido el tiempo necesario para el resto métodos.
- **GPU**: se trata del uso de la GPU por parte del sistema en el proceso de entrenamiento.

En primer lugar, se observa que en términos de calidad basados en la *accuracy* obtenida, el mejor resultado es aportado por AutoKeras, superando por una calidad relativamente pequeña al obtenido mediante *Transfer Learning*, por lo que si se buscan los mejores resultados la elección es obvia, el procedimiento a escoger sería AutoKeras. No obstante, se puede apreciar que el tiempo de entrenamiento que exige cada procedimiento es bastante dispar. En el caso de que se prefiera priorizar el tiempo necesario para contar con un modelo operativo, *Transfer Learning* podría ser un procedimiento apto. Si se elige este último procedimiento se ha de tener en cuenta que si decidiera aplicarse en casos reales no podrá operar sin supervisión humana debido a la cantidad de falsos positivos y negativos.

Para completar este apartado, se van a facilitar las matrices de confusión para aportar una imagen clara de los tan importantes falsos positivos y negativos.

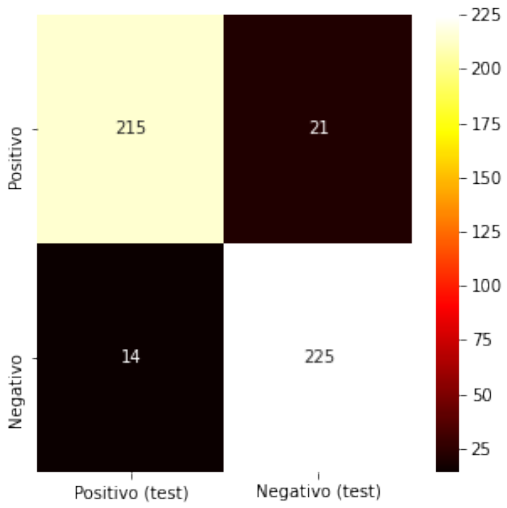

Figura 5.5: Matriz de confusión con VGG16

Se distingue por una pequeña diferencia en las figuras 5.5 y 5.6 viendo una superioridad de VGG16 respecto a ResNet50 en los falsos negativos, mientras que ambos generan el mismo número de falsos positivos. Indudablemente los resultados obtenidos presentan una alta calidad. Sin embargo, no son perfectos, y como se indicó anteriormente existe la necesidad de que sean así para contar con un modelo usable.

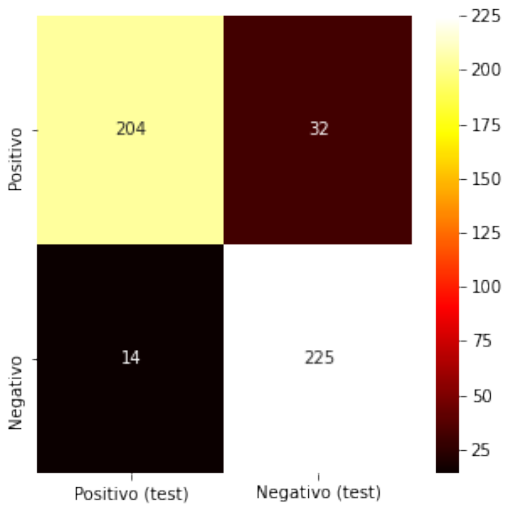

Figura 5.6: Matriz de confusión con ResNet50

Capítulo 6

CONCLUSIONES

6.1. Introducción

En este último capítulo se analizará si se ha alcanzado con éxito el objetivo propuesto al inicio del proyecto. En primer lugar, se realizará un repaso por el proceso de experimentación, posteriormente se analizarán los resultados obtenidos, y se continuará con una serie de proposiciones para continuar la investigación en torno a esta temática. Finalmente se expondrá una valoración personal sobre el trabajo realizado.

6.2. Resumen del proceso de experimentación

Para comenzar con una vista genérica del proceso de experimentación. Se puede apreciar la alta calidad de los resultados obtenidos al final. Inicialmente se comenzó con resultados bajos generados por el procedimiento manual, los cuales mejoraron drásticamente a lo largo de todo el proceso de experimentación.

Cierto es que el conjunto de datos no presenta una gran extensión, sin embargo, se ha podido demostrar la utilidad de técnicas como *Data Augmentation*, arrojando unos resultados claramente superiores respecto al conjunto de datos inicial, afirmando por lo tanto, que en el caso de este conjunto de datos es necesario contar con un mayor número de imágenes que las disponibles al inicio de la experimentación. La claridad en la información que presentaban las imágenes ha contribuido a que los modelos generados posean una calidad considerable.

Por otro lado, se puede apreciar que los resultados generados por el procedimiento

del diseño manual de una CNN no son lo suficientemente buenos como para considerar el uso del modelo. Sería necesario revisar la aproximación al problema, así como los preprocesamientos necesarios para el conjunto de imágenes, con el objetivo de intentar mejorar los resultados obtenidos mediante este procedimiento.

La utilización de técnicas avanzadas como AutoKeras o *Transfer Learning* han ayudado en gran medida a alcanzar los datos presentados. La priorización del uso de estas técnicas avanzadas ante una aproximación manual puede resultar una decisión a tener en cuenta con el objetivo de generar un ahorro del tiempo necesario para alcanzar un buen modelo.

Aunque se hayan alcanzado valores de tan alta calidad, no son perfectos. Esto define la utilidad de los sistemas desarrollados, ya que en caso de usarse en casos prácticos dentro de hospitales o centros médicos es necesario que no exista ningún caso de falso negativos debido a que puede conllevar incluso al fallecimiento de un paciente, por lo que es necesario que cualquier sistema que no cuente con una calidad absoluta, esté en todo momento bajo supervisión humana.

6.3. Conclusiones finales

La decisión del procedimiento final para generar el modelo seleccionado reside en esencia en el tiempo disponible para contar con un modelo operativo. Es necesario tener presente que los resultados realizados trabajan sobre un conjunto de 2319 imágenes, y que por lo tanto, los tiempos necesarios giran en torno a las imágenes disponibles. En caso de querer trabajar con un conjunto mayor, sería objeto de estudio la escalabilidad del tiempo necesario para realizar el entrenamiento de las CNN respecto al tamaño del conjunto de datos.

6.4. Propuestas de futuro

Brevemente se aportarán algunas ideas sobre la posible continuación respecto a este proyecto.

6.4.1. Desarrollo de software

Sería interesante el desarrollo de un software que permita la utilización de los modelos generados en el ámbito médico. El objetivo sería implementar una versión usable. El software podría tener varios tipos de fines, desde una implantación en los equipos de generación de imágenes de tipo CT de un sistema dinámico de clasificación de imágenes, hasta el desarrollo de un software externo que permita el procesamiento y clasificación de las imágenes generadas para cada paciente.

6.4.2. Implantación dinámica en equipos

Una de las principales motivaciones existentes al comienzo de este proyecto fue la implantación dinámica de este tipo de software, es decir, la posibilidad de que un radiólogo pudiera realizar pruebas en las cuales se indique gráfica y dinámicamente la probabilidad de existencia de anomalías con información probabilística sobre la imagen generada para aportar una primera aproximación e indicar al profesional si es necesaria una búsqueda más exhaustiva.

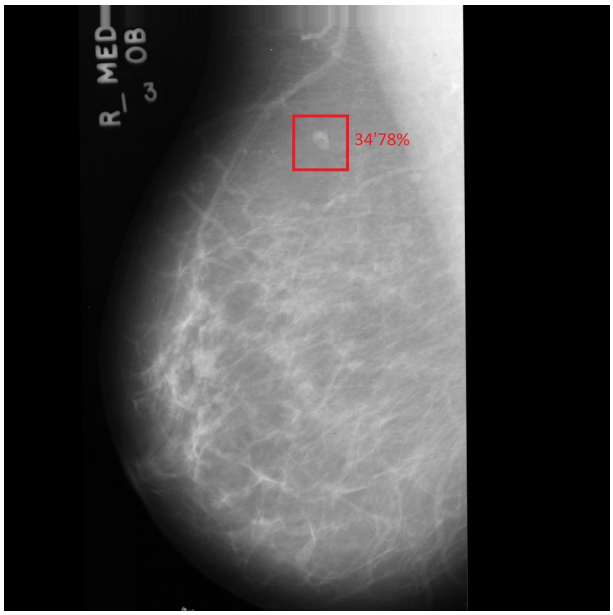

Figura 6.1: Ejemplo práctico de software dinámico a implementar

6.5. Valoración personal

Al principio del proyecto apenas tenía ningún conocimiento respecto al campo del ML. Comenzar con este proyecto supusieron muchas horas de investigación que parecían no aportar ningún resultado. En el proceso de realización fui adquiriendo diversos conocimientos mediante ensayo y error, sin embargo, conforme iba avanzando el proyecto empezaban a dar sus frutos las horas de investigación.

No me cabe duda de que, si ahora comenzara con un proyecto de la misma temática desde cero, tardaría muchísimo menos en realizar uno de mayor calidad. Este TFG ha supuesto mi comienzo en el campo de la AI y ha contribuido enormemente a entender el funcionamiento de las redes neuronales y el preprocesamiento de imágenes, algo que, sin duda me servirá en el futuro.

Por otro lado, no solo he adquiridos conocimientos del campo de la informática ya que también he aprendido sobre exactamente qué es un tumor, los sistemas de clasificación que existen, cuáles son los más comunes y cómo de peligrosos son.

Las horas de trabajo que han exigido este TFG son difíciles de valorar debido a que de desarrollo y redacción no han sido muy elevadas, sin embargo, todas las horas de investigación y corrección por la falta de experiencia han sido bastantes.

Bibliografía

españa: población por género 2020, 2021. URL https://es.statista.com/estadisticas/472331/poblacion-de-espana-por-genero. comprobado en 2021-6-18.

M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL https://www.tensorflow.org/. Software available from tensorflow.org.

S. Albawi, T. A. Mohammed, and S. Al-Zawi. Understanding of a convolutional neural network. *2017 International Conference on Engineering and Technology (ICET)*, 2017. doi: 10.1109/icengtechnol.2017.8308186.

Aphex34. Typical cnn.png, 2015. URL https://commons.wikimedia.org/wiki/File:Typical_cnn.png.

E. Arias, J. A. M. Losada, B. Moreno, F. J. Bravo, and Alejandro. Mejoramiento de imÁgenes usando funciones de base radial. *Revista Ingenierías Universidad de Medellín*, 2009. ISSN 1692-3324. URL https://www.redalyc.org/articulo.oa?id=75017199004.

J. Barentsz, S. Takahashi, W. Oyen, R. Mus, P. De Mulder, R. Rodney, O. Matthijs, and W. Mali. American cancer society guidelines for the early detection of cancer: Update of early detection guidelines for prostate, colorectal, and endometrial cancers: Also: Update 2001–testing for early lung cancer detection. *CA: A Cancer Journal for Clinicians*, 51:38–75, 2001. doi: 10.3322/canjclin.51.1.38.

A. A. Bataineh. A comparative analysis of nonlinear machine learning algorithms for

breast cancer detection. *International Journal of Machine Learning and Computing*, 9(3):248–254, 2019. doi: 10.18178/ijmlc.2019.9.3.794.

S. Belciug and F. Gorunescu. A hybrid neural network/genetic algorithm applied to breast cancer detection and recurrence. *Expert Systems*, 30(3):243–254, 2012. doi: 10.1111/j.1468-0394.2012.00635.x.

L. Bottou. Stochastic gradient descent tricks. *Lecture Notes in Computer Science*, pages 421–436, 2012. doi: 10.1007/978-3-642-35289-8_25.

C. Bustamante, L. Garrido, and R. Soto. Comparing fuzzy naive bayes and gaussian naive bayes for decision making in robocup 3d. *Lecture Notes in Computer Science*, pages 237–247, 2006. doi: 10.1007/11925231_23.

F. P. Carlos Martínez, Gerardo Colmenares. Uso de las técnicas de preprocesamiento de datos e inteligencia artificial (lógica difusa) en la clasificación/predicción del riesgo bancario", 2018. URL http://becker.faces.ula.ve/CDCHT/CDCHT%20E2170309B/PREGRADO/TESIS_CARLOS_MARTINEZ.pdf.

F. Charte. Nuevos métodos híbridos de computación flexible para clasificación multietiqueta. 6 2015. doi: 10.6084/m9.figshare.1451245.v1. URL https://figshare.com/articles/thesis/Nuevos_m_todos_h_bridos_de_computaci_n_flexible_para_clasificaci_n_multietiqueta/1451245.

F. Chollet. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016. URL http://arxiv.org/abs/1610.02357.

A. Cutler, D. R. Cutler, and J. R. Stevens. Random forests. *Ensemble Machine Learning*, pages 157–175, 2012. doi: 10.1007/978-1-4419-9326-7_5.

D. Dobrev. A definition of artificial intelligence. *CoRR*, abs/1210.1568, 2012. URL http://arxiv.org/abs/1210.1568.

K. Ganesan, U. R. Acharya, C. K. Chua, L. C. Min, K. T. Abraham, and K.-H. Ng. Computer-aided breast cancer detection using mammograms: A review. *IEEE Reviews in Biomedical Engineering*, 6:77–98, 2013a. doi: 10.1109/rbme.2012.2232289.

K. Ganesan, U. R. Acharya, C. K. Chua, L. C. Min, K. T. Abraham, and K.-H. Ng. Computer-aided breast cancer detection using mammograms: A review. *IEEE Reviews in Biomedical Engineering*, 6:77–98, 2013b. doi: 10.1109/rbme.2012.2232289.

M. Garcia-Dominguez, C. Dominguez, J. Heras, E. Mata, and V. Pascual. Frimcla: A framework for image classification using traditional and transfer learning techniques. *IEEE Access*, 8:53443–53455, 2020. doi: 10.1109/access.2020.2980798.

Z. Ghahramani. *Learning dynamic Bayesian networks*, pages 168–197. Springer Berlin Heidelberg, Berlin, Heidelberg, 1998. ISBN 978-3-540-69752-7. doi: 10.1007/BFb0053999. URL https://doi.org/10.1007/BFb0053999.

K. Hauser. B553 lecture 4: Gradient descent. 2012. URL http://people.duke.edu/~kh269/teaching/b553/gradient_descent.pdf.

X. He, K. Zhao, and X. Chu. Automl: A survey of the state-of-the-art. *Knowledge-Based Systems*, 212:106622, 2021. ISSN 0950-7051. doi: https://doi.org/10.1016/j.knosys.2020.106622. URL https://www.sciencedirect.com/science/article/pii/S0950705120307516.

L. Hussain, A. Ahmed, S. Saeed, S. Rathore, I. A. Awan, S. A. Shah, A. Majid, A. Idris, and A. A. Awan. Prostate cancer detection using machine learning techniques by employing combination of features extracting strategies. *Cancer Biomarkers*, 21(2): 393–413, 2018. doi: 10.3233/cbm-170643.

A. Jemal, F. Bray, M. M. Center, J. Ferlay, E. Ward, and D. Forman. Global cancer statistics. *CA: A Cancer Journal for Clinicians*, 61(2):69–90, 2011. doi: 10.3322/caac.20107.

H. Jin, Q. Song, and X. Hu. Auto-keras: An efficient neural architecture search system. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1946–1956. ACM, 2019.

S. M. P. L. Juan Andres Cadena, Juan Mauricio Cadena. AplicaciÓn de redes neuronales probabilÍsticas en la deteccion de fallas incipientes en transformadores. *Scientia et Technica*, 14(39):48–53, 2018. doi: 10.3233/cbm-170643.

V. Kurková, Y. Manolopoulos, B. Hammer, L. Iliadis, and I. Maglogiannis. artificial neural networks and machine learning – icann 2018$_2$018. *Lecture Notes in Computer Science*, $2018.doi$: .

A. L., A. Santalla, M. L. Criado, I. González-Pérez, M. Calderón, J. Gallo, and J. F. Parra. Clasificación radiológica y manejo de las lesiones mamarias. *Clínica e Investigación en Ginecología y Obstetricia*, 38(4):141–149, 2011. 10.1016/j.gine.2010.10.016.

Larhman. Svm margin.png, 2018. URL https://commons.wikimedia.org/wiki/user:Larhmam.

Laughsinthestocks. Activation binary step.svg, 2015a. URL https://en.wikipedia.org/wiki/File:Activation_binary_step.svg.

Laughsinthestocks. Activation identity.svg, 2015b. URL https://en.wikipedia.org/wiki/File:Activation_identity.svg.

Laughsinthestocks. Activation rectified linear.svg, 2015c. URL https://en.wikipedia.org/wiki/File:Activation_rectified_linear.svg.

Laughsinthestocks. Activation logistic.svg, 2015d. URL https://en.wikipedia.org/wiki/File:Activation_logistic.svg.

Laughsinthestocks. Activation tahn.svg, 2015e. URL https://commons.wikimedia.org/wiki/File:Activation_tanh.svg.

Laughsinthestocks. Activation prelu.svg, 2016. URL https://en.wikipedia.org/wiki/File:Activation_prelu.svg.

W. Loh. Classification and regression trees. *WIREs Data Mining and Knowledge Discovery*, 1(1):14–23, 2011. 10.1002/widm.8.

R. Marconi Ramos, C. Ghedini Ralha, T. M. Kurc, J. H. Saltz, and G. Teodoro. Increasing accuracy of medical cnn applying optimization algorithms: An image classification case. *2019 8th Brazilian Conference on Intelligent Systems (BRACIS)*, 2019. 10.1109/bracis.2019.00049.

T. NIH, 2021. URL https://www.cancer.gov/espanol.

W. S. Noble. What is a support vector machine? *Nature Biotechnology*, 24(12):1565–1567, 2006. 10.1038/nbt1206-1565.

L. R. Ojeda. Método del gradiente de máximo descenso. *MATEMÁTICA ESPOL-FCNM JOURNAL*, 14, 2016. URL http://www.revistas.espol.edu.ec/index.php/matematica/article/view/469/350.

T. OpenCV. Opencv, 2021. URL https://opencv.org.

A. Opperman. URL https://www.deeplearning-academy.com/p/ai-wiki-machine-learning-vs-d

T. Overleaf. Overleaf, 2021. URL https://es.overleaf.com.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. Pytorch: An imperative

style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019. URL http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learni pdf.

L. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009. 10.4249/scholarpedia.1883.

R. Polikar. Ensemble learning. *Ensemble Machine Learning*, pages 1–34, 2012. 10.1007/978-1-4419-9326-7_1.

K. Pradhan and P. Chawla. Medical internet of things using machine learning algorithms for lung cancer detection. *Journal of Management Analytics*, 7(4):591–623, 2020. 10.1080/23270012.2020.1811789.

T. PyCharm. Pycharm, 2021. URL https://www.jetbrains.com/es-es/pycharm.

S. R.A, v. E. A.C, W. R., L. B., b. T., R. D., C. W., C. C., R. C., S. D., C. V., and E. H. Commonly used imaging techniques for diagnosis and staging. *Journal of Clinical Oncology*, 24:3234–3234, 2006. 10.1200/jco.2006.06.5946.

D. A. Ragab, M. Sharkas, S. Marshall, and J. Ren. Breast cancer detection using deep convolutional neural networks and support vector machines. *PeerJ*, 7:e6201, 2019. 10.7717/peerj.6201.

R. Rojas. The backpropagation algorithm. *Neural Networks*, pages 149–182, 1996. 10.1007/978-3-642-61068-4_7.

F. Rosenblatt. The perceptron: A probabilistic model for information storage and organization in the brain. *APA PsycNet*, 1958. https://doi.org/10.1016/j.knosys.2020.106622. URL https://psycnet.apa.org/record/1959-09865-001.

C. F. Sabottke and B. M. Spieler. The effect of image resolution on deep learning in radiography. *Radiology: Artificial Intelligence*, 2(1):e190015, 2020. 10.1148/ryai.2019190015.

T. scikit learn. scikit-learn, 2021. URL https://scikit-learn.org/stable.

Z. Sha, L. Hu, and B. D. Rouyendegh. Deep learning and optimization algorithms for automatic breast cancer detection. *International Journal of Imaging Systems and Technology*, 30(2):495–506, 2020. 10.1002/ima.22400.

M. Tan and Q. V. Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *CoRR*, abs/1905.11946, 2019. URL http://arxiv.org/abs/1905.11946.

J. Waring, C. Lindvall, and R. Umeton. Automated machine learning: Review of the state-of-the-art and opportunities for healthcare. *Artificial Intelligence in Medicine*, 104: 101822, 2020. 10.1016/j.artmed.2020.101822.