



Universitat
Oberta
de Catalunya

APLICACIÓN WEB PARA LA IDENTIFICACIÓN DE TIPOS DE CÁNCER DE MAMA EN IMÁGENES MAMOGRAFICAS MEDIANTE LA IMPLEMENTACIÓN DE UNA RED NEURONAL CONVOLUCIONAL

Laura Alejos Collado

Máster Universitario en Bioinformática y Bioestadística
Área 4

Consultora: Romina Astrid Rebrij

Profesor responsable de la asignatura: Antoni Pérez Navarro

Diciembre 2021



Esta obra está sujeta a una licencia de Reconocimiento-NoComercial-SinObraDerivada [3.0 España de Creative Commons](https://creativecommons.org/licenses/by-nc-nd/3.0/es/)

FICHA DEL TRABAJO FINAL

Título del trabajo:	<i>Aplicación web para la identificación de tipos de cáncer de mama en imágenes mamográficas mediante la implementación en una red neuronal convolucional</i>
Nombre del autor:	<i>Laura Alejos Collado</i>
Nombre del consultor/a:	<i>Romina Astrid Rebrij</i>
Nombre del PRA:	<i>Antoni Pérez Navarro</i>
Fecha de entrega:	<i>12/2021</i>
Titulación:	<i>Máster Universitario en Bioinformática y Bioestadística</i>
Área del Trabajo Final:	<i>Área 4</i>
Idioma del Trabajo:	<i>Castellano</i>
Número de créditos:	<i>15</i>
Palabras clave:	<i>Cáncer de mama, Aprendizaje profundo, Mamografías</i>
Resumen del Trabajo:	
<p>El cáncer de mama es el tipo de cáncer más detectado en todo el mundo. Las mamografías son el método de detección más común para el cáncer de mama. En España, el sistema público de salud las realiza periódicamente a partir de los 50 años, adelantándose hasta los 45 en algunas comunidades autónomas.</p> <p>Las técnicas convencionales de detección de cáncer requieren bastante cantidad de tiempo de parte de los radiólogos y los errores son altamente probables. Con el fin de incrementar la eficacia médica se están desarrollando diferentes algoritmos de inteligencia artificial que, no solo reducen el tiempo empleado en el diagnóstico, sino que, además, mejoran su precisión.</p> <p>En este Trabajo de Fin de Máster se ha propuesto un modelo de red neuronal convolucional y se ha intentado mejorar mediante la utilización de técnica de transferencia de aprendizaje. El proyecto se ha desarrollado en el lenguaje de programación Python mediante el uso de librerías como Tensorflow y Keras.</p> <p>Para el entrenamiento del modelo se ha utilizado la base de datos de imágenes <i>The Cancer Imaging Archive</i>. Dichas imágenes se encuentran previamente clasificados por médicos en tumores benignos y malignos.</p> <p>Finalmente, se ha realizado la comparación de los modelos creados. El modelo mejor considerado ha sido utilizado para crear la aplicación web mediante el paquete Flask.</p>	
Abstract:	
<p>Breast cancer is the most detected cancer around the world. Mammographies are the common breast cancer detection method. In Spain, women start having them done in their 50's via the public health system. In some regions they start 5 years earlier, beginning at 45.</p> <p>The conventional cancer detection techniques demand a great deal of time from radiologists and the likelihood of mistakes is too high. In order to improve the medical efficacy, different artificial intelligence algorithms are being</p>	

developed. Not only to reduce the diagnostic time but also to improve the precision.

In this Master's Thesis, a convolutional neural network model has been proposed and an attempt has been made to improve it through the use of transfer learning technique. The whole project has been developed in Python programming language, using libraries such as Tensorflow and Keras, among others.

For the training model, a database of breast cancer's images has been used, downloaded from The Cancer Imaging Archive. Such images were previously classified by doctors in two types, benign and malignant tumours.

Finally, a comparison between the created models has been made and the considered best model has been used to create the web application developed with the Flask package.

Índice

1. Resumen.....	8
1.1. Antecedentes	8
1.2. Método	8
1.3. Resultados	8
1.4. Conclusiones.....	8
2. Introducción	9
2.1. Contexto y justificación del trabajo.....	9
2.1.1. Descripción general	9
2.1.2. Contexto	9
2.1.3. Justificación	10
2.2. Objetivos del Trabajo	10
2.2.1. Objetivo general:.....	10
2.2.2. Objetivos específicos:	11
2.3. Enfoque y método seguido	11
2.4. Planificación del trabajo	12
2.4.1. Tareas.....	12
2.4.2. Calendario.	14
2.4.3. Hitos.....	15
2.4.4. Análisis de riesgos	16
2.5. Breve resumen de contribuciones y productos obtenidos	17
3. Estado del arte	17
3.1. Información relevante.....	17
3.2. Artículos relacionados	19
3.3. Inteligencia Artificial.....	21
3.3. Aprendizaje automático.....	22
3.4. Aprendizaje profundo	23
3.5. Redes neuronales convolucionales.....	25
4. Metodología.....	28
4.1. El conjunto de datos.....	28
4.2. Entorno de trabajo.....	28
4.3. Lenguaje de programación.....	28
4.4. Entorno de programación	29
4.5. Librerías utilizadas	29

4.6.	Importación de datos.....	30
4.7.	Análisis exploratorio de los datos.....	30
4.8.	Preprocesamiento de los datos.....	32
4.9.	Construcción del modelo CNN.....	34
4.10.	Entrenamiento y validación del modelo.....	35
4.11.	Mejora del modelo	35
4.12.	Diseño de la aplicación web.....	37
4.13.	Publicación de la aplicación web	37
5.	Resultados	38
5.1.	Exposición de los resultados de cada modelo	38
5.1.1	Modelo general	38
5.1.2.	Modelo con transferencia de aprendizaje	40
5.2.	Comparación de modelos	41
5.3.	Elección final de un modelo	41
5.4.	Uso de la aplicación web	41
6.	Discusión.....	43
7.	Conclusiones	44
7.1.	Conclusiones generales	44
7.2.	Líneas de futuro	44
7.3.	Seguimiento de la planificación.....	45
8.	Glosario	46
9.	Bibliografía	47

Lista de ilustraciones

Ilustración 1: Número de incidencias estimadas por tipo de cáncer en el mundo, teniendo en cuenta ambos sexos y todas las franjas de edad [3].	9
Ilustración 2: Diagrama de Gantt: Calendario del proyecto. Elaboración propia.	14
Ilustración 3: Hitos del proyecto. Elaboración propia.	15
Ilustración 4: Subcampos de la inteligencia artificial [23].	23
Ilustración 5: Estructura de una neurona [27].	23
Ilustración 6: Estructura de una ANN [27].	24
Ilustración 7: Max-Pooling [17].	27
Ilustración 8: Ejemplo de imagen de mamografía con tumor benigno. Elaboración propia.	31
Ilustración 9: Ejemplo de imagen de mamografía con tumor maligno. Elaboración propia.	31
Ilustración 10: Modelo Secuencial de CNN. Elaboración propia.	35
Ilustración 11: Características principales de la red VGG16. Elaboración propia.	36
Ilustración 12: Lanzamiento de la aplicación a través de Heroku. Elaboración propia.	38
Ilustración 13: Gráfica de la precisión durante el entrenamiento y validación del modelo general. Elaboración propia.	39
Ilustración 14: Gráfica de la función de pérdida durante el entrenamiento y validación del modelo general. Elaboración propia.	39
Ilustración 15: Gráfica de la precisión durante el entrenamiento y validación del modelo con transferencia de aprendizaje. Elaboración propia.	40
Ilustración 16: Gráfica de la pérdida durante el entrenamiento y validación del modelo con transferencia de aprendizaje. Elaboración propia.	40
Ilustración 17: <i>Página principal de la página web. Elaboración propia.</i>	42
Ilustración 18: Selección de la imagen para importarla en la página web. Elaboración propia.	42
Ilustración 19: Resultado de la predicción de la aplicación. Elaboración propia.	43

Lista de tablas

Tabla 1: Análisis de riesgos del Trabajo. Elaboración propia.	16
Tabla 2: Anomalías y características principales del CM. Elaboración propia.	18
Tabla 3: Resumen artículos relacionados para este trabajo. Elaboración propia.	20
Tabla 4: Métricas del modelo general. Elaboración propia.	39
Tabla 5: Métricas del modelo con transferencia de aprendizaje. Elaboración propia.	41
Tabla 6: Comparación de las métricas de los modelos. Elaboración propia.	41

1. Resumen

1.1. Antecedentes

El cáncer de mama es el tipo de cáncer más diagnosticado en todo el mundo. Se caracteriza por producir un exceso de células malignas con comportamiento y reproducción descontrolado en el tejido mamario.

El método más utilizado para su detección son las mamografías, que se hacen de manera periódica entre las mujeres a partir de los 50 años en España (45 dependiendo de la comunidad autónoma). Esto permite la detección precoz del cáncer que a su vez conlleva a una probabilidad alta de éxito al administrar los distintos tratamientos disponibles.

1.2. Método

Se ha diseñado un modelo de red neuronal convolucional que es capaz de clasificar las mamografías en dos clases: tumores benignos y malignos. Para el entrenamiento del modelo se han utilizado imágenes descargadas del repositorio público *The Cancer Imaging Archive*. Con el objetivo de mejorar el rendimiento del modelo se ha utilizado la técnica de aprendizaje por transferencia. El modelo con mejor rendimiento se ha utilizado para la creación de la página web interactiva, creada con el paquete Flask.

1.3. Resultados

El modelo con mejor rendimiento es el modelo realizado con transferencia de aprendizaje conectada a la red VGG16. No obstante, la red general es la que se ha utilizado para crear la página web. Dicha aplicación procesa la imagen subida por un usuario y realiza la predicción de clase.

1.4. Conclusiones

Se ha conseguido crear una red neuronal convolucional con un nivel de precisión del 62.8%. Esta precisión podría aumentarse implementando los modelos en entornos de trabajo GPU con mayor memoria, en los que se podría añadir un número más elevado de épocas.

El aprendizaje profundo ha demostrado ser de gran utilidad en la detección del cáncer de mama ya que ayuda al pronto diagnóstico y, por lo tanto, mejora la probabilidad de éxito del tratamiento.

Con este trabajo se pretende mostrar una ayuda para los profesionales sanitarios a la hora de detectar el cáncer de mama. La aplicación web tiene un funcionamiento muy sencillo que podría ser mejorado e implementado en el entorno hospitalario.

2. Introducción

2.1. Contexto y justificación del trabajo

2.1.1. Descripción general

Se desarrollará una aplicación web interactiva donde un usuario pueda ingresar una imagen de detección mamográfica y se obtenga la clasificación del tipo de cáncer de mama tipo benigno o maligno.

2.1.2. Contexto

La palabra “cáncer” agrupa una serie de enfermedades en las que el organismo produce un exceso de células malignas cuyo comportamiento y reproducción es descontrolado. Cuando la pérdida de control ocurre en las células del tejido mamario se habla de cáncer de mama (CM) [1]. Según las estadísticas del *International Agency for Research on Cancer*, este se ha convertido en el tipo de cáncer más diagnosticado en todo el mundo, sobre todo entre las mujeres (Organización Mundial de la Salud, 2021).

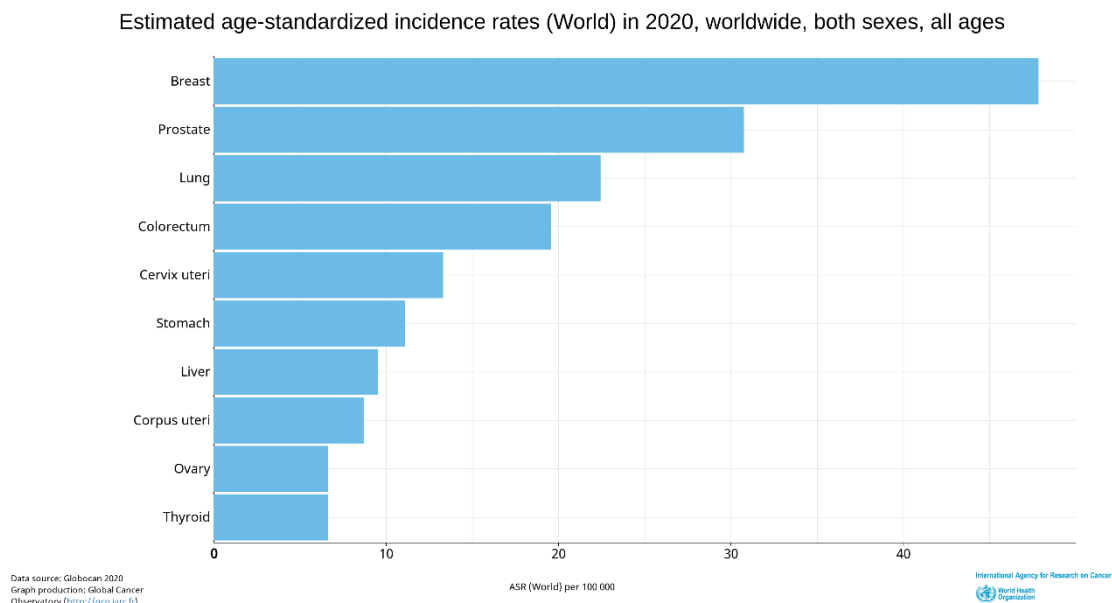


Ilustración 1: Número de incidencias estimadas por tipo de cáncer en el mundo, teniendo en cuenta ambos sexos y todas las franjas de edad [3].

Las mamografías son el método de diagnóstico más común en el CM. Los radiólogos utilizan esta técnica para visualizar si existe la presencia de alguna anomalía en el tejido mamario [4].

Existen dos tipos de clasificación del CM, que tienen que ver con su método de detección: La clasificación histológica y la molecular. La primera se realiza a través del análisis de una muestra de tejido extraída, mientras que para la

clasificación molecular se utiliza el análisis de microarray, que detecta la expresión génica de los siguientes biomarcadores principales [5].

Su detección precoz es primordial para aumentar la esperanza de vida de los pacientes y poder administrar a tiempo las terapias adecuadas [8].

2.1.3. Justificación

Las mamografías son el método de diagnóstico más común en el CM. En España, la sanidad pública las realiza a partir de los 50 años, aunque en algunas comunidades autónomas se adelantan hasta los 45 [10]. Aunque existen dos tipos de mamografía, la convencional (radiografías fotográficas) o la digital (almacenadas en una computadora), la más utilizada hoy en día es esta última [11].

Las técnicas convencionales de detección del cáncer requieren una buena cantidad de tiempo y los errores son altamente probables. Es por ello, que, con el fin de poder mejorar la eficiencia médica de la interpretación de imágenes de los pacientes con cáncer, surge la necesidad de desarrollar modelos que predican el tipo de tumor y detectan la metástasis [12].

El aprendizaje profundo (DL, por sus siglas en inglés), mediante sus fases de procesamiento y posterior clasificación de imágenes, podría ser de gran utilidad para reducir el tiempo invertido en el diagnóstico y mejorar la precisión de diagnósticos médicos. No solo porque son capaces de reconocer patrones, sino que también pueden extraer información relevante de un conjunto de datos complejos de imágenes médicas [12].

Concretamente, para el estudio del cáncer de mama, se han probado diferentes algoritmos de DL que realizan la detección a partir de distintos tipos de imágenes de detección. Por ejemplo, se ha diseñado una red neuronal convolucional (CNN, por sus siglas en inglés) que permite extracción de características aplicadas en imágenes histopatológicas de la mama que, en particular, detectaban la mitosis [12]. En otros estudios, han utilizado otro tipo de imágenes como las mamografías para realizar la CNN, cuya precisión ha resultado tan alta que es capaz de distinguir y minimizar los falsos positivos y los falsos negativos [13].

2.2. Objetivos del Trabajo

2.2.1. Objetivo general:

Desarrollar una aplicación web interactiva que permita al usuario ingresar una mamografía y obtener la clasificación del tipo benigno o maligno.

2.2.2. Objetivos específicos:

A continuación, se detallan los objetivos específicos:

- ✓ Obtener las principales características del método de detección de cáncer de mama por imágenes.
- ✓ Determinar el mejor modelo de predicción en base a una precisión mínima del 80%.
- ✓ Desarrollar una aplicación web interactiva que implemente el algoritmo de predicción de aprendizaje profundo que clasifique las imágenes en las clases benigno o maligno.

2.3. Enfoque y método seguido

Para el desarrollo del proyecto se ha escogido una base de datos del repositorio Cancer Imaging Archive (TCIA), la base de datos llamada CMMD (The Chinese Mammography Database), dispone de un total de 1775 pacientes, cuyo diagnóstico fue confirmado para un cáncer de mama de tipo benigno o maligno [14]. Además, para 749 pacientes, se incluyen los subtipos moleculares de cada cáncer de mama maligno. Además, se adjuntan dos archivos, uno en .xlsx, donde se incluye toda esta información, y el segundo un archivo .csv, donde se encuentran los metadatos (información asociada para cada imagen).

Las imágenes están en formato DICOM que es un formato estándar aceptado en medicina en todo el mundo. Este formato, se utiliza para “*ver, almacenar, recuperar y compartir imágenes médicas*” (POSTDICOM, 2021)¹⁵. La característica principal de este tipo de imagen es que mantiene la precisión de la información de las imágenes médicas [15].

Los métodos de DL estructuran los algoritmos en capas para crear redes neuronales artificiales (ANN, por sus siglas en inglés) que son capaces de aprender y tomar decisiones por sí mismas. Generalmente, se utilizan conjuntos de datos muy grandes, analógicos etiquetados como audio, video, texto o imágenes [12].

Para la realización del trabajo se han escogido métodos de DL ya que se trabaja con imágenes que tienen muchas características. En concreto, estas imágenes son de 2294 x 1914 píxeles y son en blanco y negro.

Los principales algoritmos de aprendizaje profundo son [16]:

- ✓ Redes neuronales recurrentes (RNN)
- ✓ Redes neuronales convolucionales (CNN)
- ✓ Redes de memoria a corto plazo (LSTM)
- ✓ Máquinas de Boltzmann profundas (DBMS)
- ✓ Codificadores automáticos apilados
- ✓ Redes de creencias profundas (DBN)

Para realizar el algoritmo del trabajo, que es un algoritmo de clasificación basado en imágenes, se ha considerado el más apropiado el de las redes neuronales convolucionales (CNN). En dicho algoritmo, las capas se organizan en tres dimensiones: alto, ancho y profundidad. Se suele decir, que da la capacidad de “ver” al ordenador, intentando imitar el córtex visual del ojo humano. Para ello, la CNN tiene varias capas ocultas especializadas que siguen una jerarquía: las primeras capas pueden detectar líneas, curvas y se van especializando, hasta que las capas más profundas pueden reconocer ya formas más complejas [17].

Para el desarrollo del algoritmo de este trabajo, se ha decidido utilizar el lenguaje de programación Python debido a que dispone de una capacidad de memoria más amplia en comparación con R, además de que el manejo de grandes conjuntos de datos es más rápido.

La metodología utilizada ha sido la siguiente:

- ✓ Exploración de los datos: Extracción y visualización de las características de los datos.
- ✓ Pre-procesamiento de los datos: Acondicionamiento y estandarización de los datos de imagen.
- ✓ Construcción del modelo: Creación de la red neuronal convolucional desde 0.
- ✓ Entrenar el modelo: Se entrena el modelo con los datos de entrenamiento.
- ✓ Validar el modelo: Se valida el modelo con el conjunto de datos de prueba.
- ✓ Mejorar el modelo: Se mejora el modelo aplicando técnicas como por ejemplo la transferencia de aprendizaje.
- ✓ Validar el modelo: Se comparan todos los modelos obtenidos entre sí.

Para el desarrollo de la aplicación web, en Python existen varias posibilidades como Django y Flask, siendo este último el más fácil de utilizar y por tanto el elegido para el trabajo ya que no se disponía de experiencia previa en la creación de aplicaciones web.

Finalmente, para que la aplicación web pueda ser utilizada desde cualquier tipo de dispositivo electrónico, se ha intentado publicar en las plataformas PythonAnywhere y Heroku.

2.4. Planificación del trabajo

2.4.1. Tareas

Se detallan las tareas generales que surgen de cada uno de los objetivos específicos:

“Obtener las principales características del método de detección de cáncer de mama por imágenes.”

“Determinar el mejor modelo de predicción en base a una precisión mínima del 80%.”

- ✓ Entrenar el modelo con los datos.
- ✓ Analizar el rendimiento del modelo, evaluando su precisión.
- ✓ Mejoras en el rendimiento del modelo.

“Desarrollar una aplicación web interactiva que implemente el algoritmo de predicción de aprendizaje profundo que clasifique las imágenes en benigno vs maligno”.

- ✓ Comprender el manejo de la creación de la aplicación web utilizando Flask.
- ✓ Crear la estructura de la aplicación web.
- ✓ Desarrollar la aplicación web.
- ✓ Comprobar el buen funcionamiento de la misma.

2.4.2. Calendario.

A continuación, se muestra la planificación temporal de las tareas y las subtareas. Para ello, se ha utilizado el programa GanttProject.

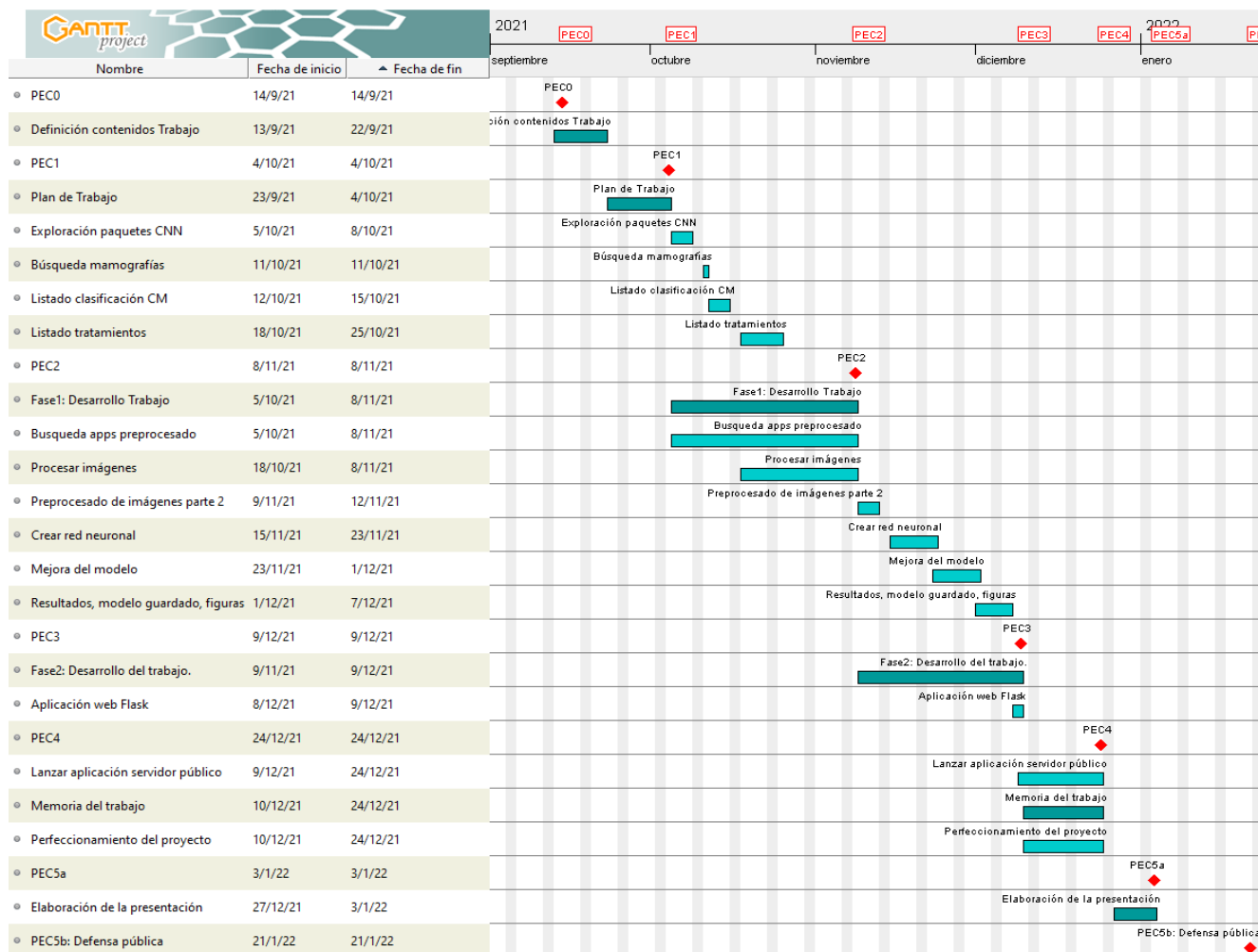


Ilustración 2: Diagrama de Gantt: Calendario del proyecto. Elaboración propia.

2.4.3. Hitos

Aquí, se muestran las fechas clave para la obtención de los objetivos del trabajo.

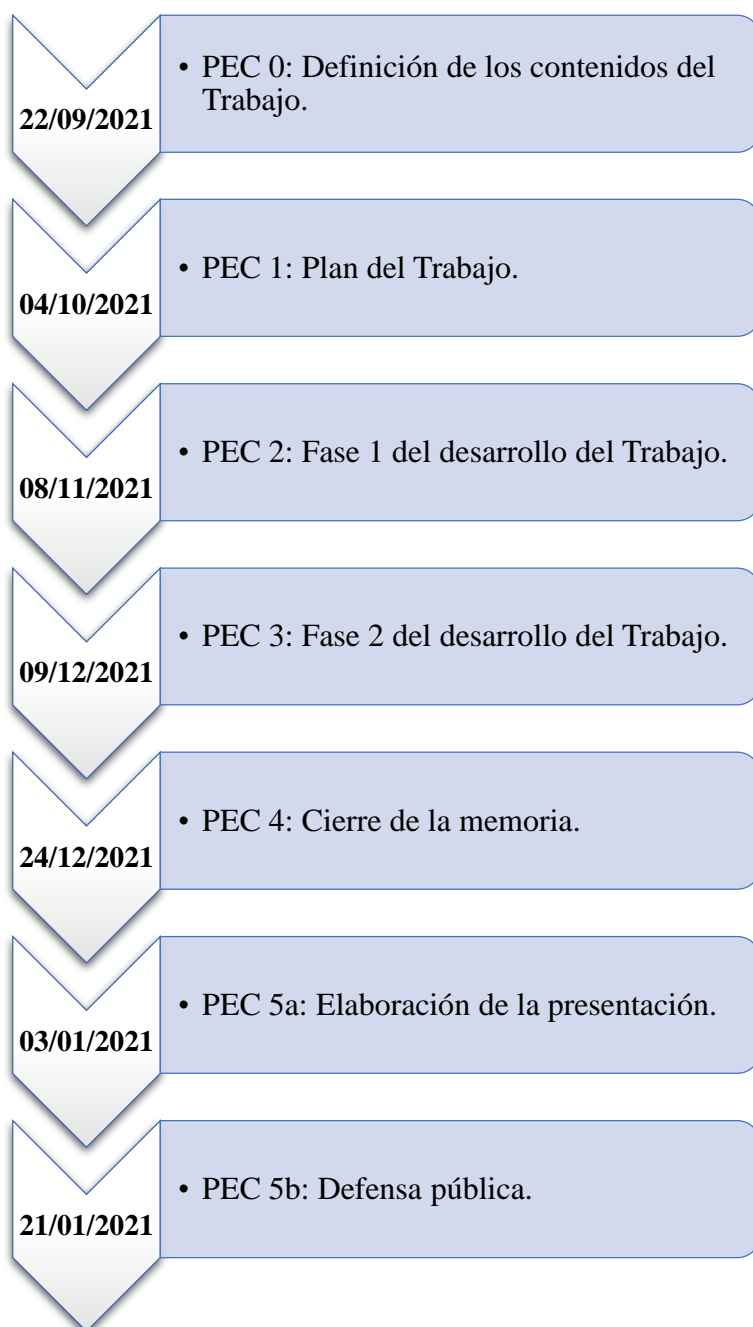


Ilustración 3: Hitos del proyecto. Elaboración propia.

2.4.4. Análisis de riesgos

A continuación, se detallan los riesgos según la planificación de las actividades:

Descripción del riesgo	Severidad	Probabilidad	Mitigación
Descartar demasiadas muestras en el preprocesado de imágenes	Alta	Media	Buscar otros repositorios para obtener más imágenes
No funcionen las librerías de R	Alta	Baja	Investigar alternativas, obtener ayuda en distintos foros, obtener código alternativo
No obtener la precisión establecida en los objetivos	Alta	Alta	Añadir más capas a la CNN, obtener más imágenes
Listado de tratamientos muy voluminoso que dificulte la visualización en la herramienta	Alta	Moderada	Resumir la información asociada a cada tratamiento, facilitar la presentación de la información
No poder concretar la elaboración de una aplicación web con Shiny	Moderada	Baja	Emplear como alternativa un archivo Markdown
Preprocesado de imágenes muy largo que conlleve un alto tiempo de cómputo	Moderada	Alta	Acortar el tiempo de cómputo de otras tareas como la búsqueda de información

Tabla 1: Análisis de riesgos del Trabajo. Elaboración propia.

2.5. Breve resumen de contribuciones y productos obtenidos

En este apartado se enumeran los productos obtenidos tras finalizar el proyecto.

✓ **Memoria.** En la memoria se recolecta toda la información relacionada con el desarrollo del proyecto desde la planificación con los objetivos, el contexto y la justificación del mismo, hasta la explicación de la metodología utilizada, analizando tanto los resultados y las conclusiones obtenidas, como las referencias bibliográficas.

✓ **Repositorio de GitHub.** Aquí es donde se encuentra todo el código creado y utilizado para el desarrollo del trabajo. Se puede acceder a él a través de este [enlace](#).

✓ **Aplicación web.** La aplicación es capaz de realizar las predicciones para las imágenes subidas por un usuario. Se puede acceder a la demostración a través de este [enlace](#).

2.6. Breve descripción de los demás capítulos de la memoria

Los próximos capítulos de la memoria tratarán sobre:

✓ **Estado del arte.** Se realiza el resumen y la comparación de otros artículos que tratan sobre proyectos con objetivos parecidos a los de este trabajo. Además, se darán explicaciones de conceptos clave utilizados para el correcto desarrollo del trabajo como la inteligencia artificial, el aprendizaje profundo y las redes neuronales convolucionales.

✓ **Metodología.** En este apartado se explica detalladamente los pasos seguidos para el desarrollo del proyecto, exponiendo tanto los métodos como los materiales que se han utilizado.

✓ **Resultados.** Aquí se exponen mediante uso de distintos gráficos de visualización las comparaciones de precisión entre los distintos modelos creados. Además, se incluye la muestra visual del funcionamiento de la página web.

✓ **Discusión.** Se interpretan los resultados obtenidos en el trabajo teniendo en cuenta su contexto y la relevancia del mismo.

✓ **Conclusión.** Se exponen tanto las conclusiones del trabajo como las posibles líneas futuras del mismo.

3. Estado del arte

3.1. Información relevante

Existen varios tipos de cáncer de mama, de los cuales solo el 5-10% es hereditario en ambos sexos. Aunque el 90% restante tenga un componente genético, son esporádicos o no heredables. En definitiva, la enfermedad es provocada por múltiples factores genéticos y no genéticos que cuando se

combinan provocan la alteración celular que da lugar al crecimiento descontrolado, que en algunos casos puede provocar la invasión a otros tejidos o metástasis [2].

Las mamografías son la prueba de imagen más utilizada para detectar el CM. En ellas, la presencia de un bulto o tumor se muestra como un área blanca concentrada que puede tener distintos tipos de relieve o forma. Las anomalías incluyen [4]:

Anomalías	Características principales
Quistes	Pequeños sacos de líquido con pared delgada.
Calcificaciones	Depósitos de calcio. Los más grandes se denominan macrocalcificaciones, que son el resultado de la acumulación de pequeños depósitos (microcalcificaciones).
Fibroadenomas	Tumores redondos, generalmente benignos.
Tejido cicatricial	Aparece como un tejido blanco en la mamografía.

Tabla 2: Anomalías y características principales del CM. Elaboración propia.

Una masa detectada en una mamografía, ya sea cancerosa o no, puede ser del tipo quiste, calcificación o fibroadenoma. Una vez determinado si el tumor es benigno o maligno, se realizan pruebas adicionales para comprobar el subtipo de CM [4].

La clasificación del CM, tiene que ver con su método de detección y puede ser una clasificación histológica o molecular. La primera se realiza a través del análisis de una muestra de tejido extraída, mientras que para la clasificación molecular se utiliza el análisis de microarray, que detecta la expresión génica de los siguientes biomarcadores principales [5]:

- ✓ Receptores de Estrógenos (ER, por sus siglas en inglés). Los ER pertenecen a la familia de los receptores esteroideos. Son factores de transcripción inducibles por ligandos (receptores nucleares) que activan y regulan la transcripción [6, 7].

- ✓ Receptores de Progesterona (PR, por sus siglas en inglés). Al igual que los ER, pertenecen a la familia de los receptores esteroideos por lo que comparten las funciones principales [6, 7].

- ✓ HER2. Cuya sobreexpresión en la superficie de las células tumorales da lugar a un crecimiento descontrolado ya que su función es enviar señales a las proteínas encargadas de la división en el ciclo celular [6, 7].

En base a todo ello, la clasificación molecular del CM se divide en cuatro subgrupos [5]:

1. Luminal A: Este tipo de CM tiene el patrón ER+/PR+/HER2- y corresponde al 40% de los casos detectados. El crecimiento de sus células tumorales es más lento debido a una baja concentración de la proteína Ki67, por lo que permite un buen pronóstico. Al ser positivo a hormonas, responde bien a tratamientos

hormonales como los tamoxífenos (antagonistas de ER) o inhibidores de la enzima aromatasa (anastrozol) [5].

2. Luminal B: Este tipo de CM tiene el patrón ER+/PR-/HER2+/- y corresponde al 20% de los casos. Tiene una alta concentración de proteína Ki67, por lo que la tasa de crecimiento es más disparada en comparación con el luminal A, y hace que el diagnóstico sea algo más complicado. Además de los tratamientos hormonales mencionados en el tipo luminal A, hacen falta terapias citotóxicas adicionales como la quimioterapia [5].
3. HER 2+: Aproximadamente corresponde al 10-15% de los casos detectados. Se caracteriza por ser ER-/PR-/HER2+. Debido a que la sobreexpresión del receptor provoca un crecimiento agresivo del tumor, la quimioterapia convencional tiene una respuesta pobre sobre el mismo [5]. Existen terapias adicionales llamadas inmunoterapias, que son altamente innovadoras y muy prometedoras que se están utilizando para este tipo de CM: los Anticuerpos Conjugados (ADC, por las iniciales en inglés) y los Receptores de Antígenos Quiméricos (CARs, por sus siglas en inglés) [8].
4. Triple negativo: Este tipo de CM es negativo para todos los biomarcadores anteriores. Suelen ser el 15-20% de los casos [5]. Además de presentar mutaciones en los genes supresores de tumores, se caracteriza por tener una sobreexpresión de la proteína Trop-2. Este es el tipo de CM más agresivo de todos, con el peor pronóstico de los tres y el más difícil de tratar. A parte de la quimioterapia, hay tipo de inmunoterapia tipo ADC en fase 3, que está presentando unos resultados muy prometedores [9].

Esta clasificación de los diferentes subtipos moleculares del cáncer de mama es primordial, no solo para aumentar la esperanza de vida de los pacientes por la detección temprana, sino también para poder administrar las inmunoterapias adecuadas [8].

3.2. Artículos relacionados

Se ha realizado una recopilación de artículos con objetivos parecidos a los de este trabajo, se resumen las técnicas de aprendizaje profundo utilizadas para la clasificación de cáncer de mama. Dicho resumen se muestra en la tabla que viene a continuación:

Trabajo	Imágenes	Modelo	Parámetros utilizados	Clasificación	Precisión	Anotaciones
1 [18]	Imágenes histopatológicas tomadas a microscopio a distintos tamaños: 40x, 100x, 200x y 400x.	Utilización de la red preentrenada AlexNet con algunas modificaciones	3 capas de convolución con la función de activación ReLu en todas las capas	<u>Binaria:</u> - Benigno - Maligno	Distintas precisiones dependiendo del tamaño de imagen entre un 80-85%	Extracción de parches de imagen para el <i>train</i> y la combinación de dichos parches para la clasificación final.
2 [19]	Imágenes de resonancia magnética	2 modelos utilizados: CNN y CLSTM ¹ con <i>transfer learning</i>	7 capas de convolución con la función de activación ReLu en todas las capas	<u>3 subtipos:</u> - HR+ ² - HER2+ - TN	Precisión final de un 78% en la CNN y 74% en la CLSTM.	Se combinan ambos modelos con <i>transfer learning</i> utilizando los pesos de la red preentrenada
3 [20]	Mamografías	Se utilizaron 2 modelos preentrenados: - VGG16 - ResNet50	Se preentrenaron ambas redes utilizando distintos números de epoch para cada una	<u>4 tipos:</u> - Masa benigna - Masa maligna - Calcificación benigna - Calcificación maligna	Precisión final de un 84% para VGG16 y un rango de 76-99% para ResNet50	Para la red ResNet50 se hicieron varias pruebas tanto con preentrenamiento como sin él
4 [21]	Mamografías. Se utilizaron un set de 4 imágenes: L-CC ³ R-CC L-MLO R-MLO	Se creó una MV-DCN ⁴	Se aplicaron varias capas de convolución.	<u>3 tipos:</u> BI-RADS0: Incompleta BI-RADS1: Normal BI-RADS2: Benigno	Para cada tipo de clasificación precisión distinta. Rango en torno a 54-76%.	Toda la red se entrenó conjuntamente mediante gradiente estocástico descendiente con retropropagación

Tabla 3: Resumen artículos relacionados para este trabajo. *Elaboración propia.*

¹ CLSTM significa, por sus siglas en inglés, *Convolutional Long Short-Term Memory*, derivado de las redes de memoria a corto plazo (LSTM).

² HR+ significa, por sus siglas en inglés, *Hormonal Receptor +* que es positivo a receptores de estrógenos y/o progesterona y HER2-.

³ L y R corresponden a izquierda y derecha, respectivamente. A su misma vez, CC significa Craneal Caudal y MLO Medio lateral Oblicua.

⁴ MV-DCN significa *Multi-View Deep Convolutional Neural Network*

Se puede observar que se han utilizado distintas técnicas con el mismo objetivo: la clasificación del cáncer de mama. Algunos, han utilizado directamente las imágenes histopatológicas y han creado una red que tenía distinta precisión dependiendo del tamaño de imagen obtenida a microscopio [19].

Otros, han utilizado otro tipo de imágenes como las de resonancia magnética para crear varios modelos distintos y realizar un tipo de clasificación por subtipos moleculares. No obstante, la cantidad de imágenes utilizada para entrenar el modelo era bastante pequeña [20].

Finalmente, otros trabajos incluyen las mamografías como imagen para construir las redes neuronales. Sin embargo, una de ellas no entrena la red con sus datos [21] y la otra no consigue una clasificación completa ya que no es capaz de clasificar los tumores malignos [22].

En nuestro trabajo, se utilizarán mamografías y se entrenará una red desde cero, para clasificar los tumores de mama en Benigno vs Maligno.

3.3. Inteligencia Artificial

La inteligencia artificial (IA) es la habilidad de una máquina para presentar las mismas capacidades que los humanos, como el razonamiento, el aprendizaje, la creatividad y la capacidad de planear [23].

El punto de partida, considerado en los años 50, fue cuando Alan Turing, considerado el padre de la IA publicaba en la revista Mind un artículo con el título “Computing machinery and intelligence” donde se hacía la pregunta: ¿pueden las máquinas pensar? y explicaba un método para averiguarlo [23].

Sin embargo, no fue hasta 1956, cuando los considerados padres de la IA moderna, John McCarty, Marvin Misky y Claude Shannon, presentaron formalmente el término durante una conferencia [23].

Un breve resumen de la historia de la IA con sus respectivos productos incluye [23]:

- ✓ El supercomputador Deep Blue de IBM en 1997, que ganó al campeón mundial de ajedrez.
- ✓ El supercomputador Watson de IBM en 2011, que ganó el concurso televisivo estadounidense Jeopardy, en el que se realizan preguntas de todo tipo de cultura.
- ✓ Los asistentes virtuales creados por Apple (Siri, 2011), Google (Google Now, 2012) o Microsoft (Cortana, 2014).
- ✓ La IA se asentó en sectores importantes como la Automoción, Turismo, Movilidad o Seguros a partir del 2018, construyendo avances relevantes en la conducción autónoma, o modelos predictivos que evaluaban la probabilidad de

sufrir un accidente.

✓ A partir de 2020, inducido en parte por la situación de la pandemia de la covid-19, se favorece el uso de la IA en el ámbito de la salud, utilizando sensores térmicos automatizados o herramientas de Big Data para la detección temprana de pacientes cero y control de focos de contagio.

En resumen, la IA tiene un gran impacto en la vida de las personas, especialmente en áreas como el trabajo, educación, bienestar, relaciones sociales o la salud [23].

3.3. Aprendizaje automático

El término “Aprendizaje automático” lo acuñó Arthur Samuel en 1959 mientras trabajaba en IBM. Este término surgió de la búsqueda de IA, ya que, en sus primeros días, los científicos buscaban que las máquinas aprendieran los datos [24].

A pesar de que la IA y el aprendizaje automático son dos términos que la mayoría de veces se confunden, la IA es según el Instituto de Ingeniería del Conocimiento el *“concepto general de máquina pensante o la toma de decisiones automatizada”*²⁴ mientras que el término descrito por Arthur Samuel se refiere a la capacidad de aprendizaje de las computadoras sin estar explícitamente programadas para ello [24].

El aprendizaje automático en sí mismo, se centra en métodos como la predicción, la extracción de datos o el pre-procesamiento para mejorar la precisión. Existen tres tipos de aprendizaje automático: aprendizaje supervisado, no supervisado y reforzado [25].

Aprendizaje supervisado

La computadora se presenta con entradas de ejemplo y salidas deseadas, dadas por un “maestro”. El objetivo es aprender la regla general que asigne entradas a salidas [25].

Aprendizaje no supervisado

No se asignan etiquetas al algoritmo de aprendizaje, dejándolo solo para encontrar la estructura en su entrada. Este tipo de aprendizaje puede ser un objetivo en sí mismo (bien para encontrar patrones ocultos en los datos) o un medio hacia un fin (aprendizaje de características) [25].

Aprendizaje de refuerzo

Consiste en aprendizaje por “experiencia”, es decir, al algoritmo se le da un refuerzo positivo cada vez que acierta [25].

3.4. Aprendizaje profundo

El aprendizaje profundo (DL, por sus siglas en inglés) forma parte del subconjunto del aprendizaje automático. Se caracteriza, principalmente, por el uso de las redes neuronales profundas, con múltiples capas. El DL se inspira en el funcionamiento del cerebro humano, así como las redes neuronales se basan en la actividad de las redes neuronales humanas [25].

La tarea de aprendizaje hace que las redes neuronales sean aptas para la identificación de patrones y la clasificación de los datos, de manera similar a la del cerebro humano. Por lo que imitan funciones básicas de las redes neuronales biológicas [25].

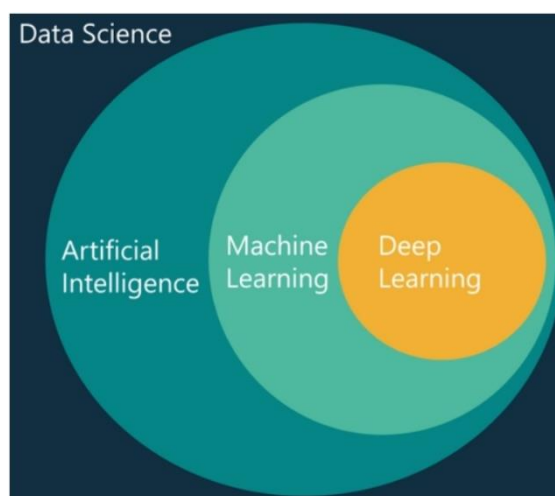


Ilustración 4: Subcampos de la inteligencia artificial [23].

Red neuronal biológica vs Red neuronal artificial

Para explicar el funcionamiento de las redes neuronales artificiales es conveniente explicar primero la estructura típica de una neurona. Se caracterizan por tener tres partes: el cuerpo celular o soma, los axones y las dendritas [26].

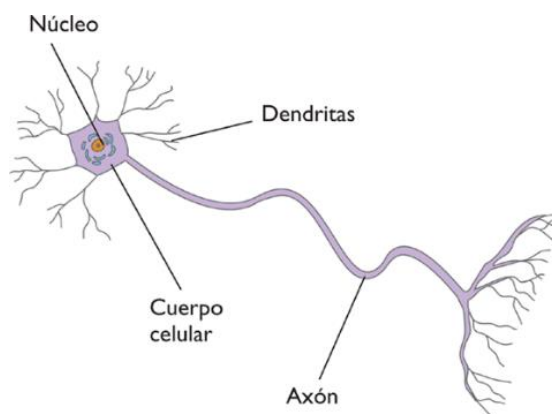


Ilustración 5: Estructura de una neurona [27].

De forma muy resumida, las señales recibidas por las neuronas entran a través de las dendritas y emiten señales a través del axón. Se le llama sinapsis al proceso por el cual la señal es transferida desde el axón de una neurona y recibida a través de las dendritas de otra [26].

Las señales se transmiten en forma de impulso nervioso y estos pueden ser eléctricos o químicos. Cuando el impulso llega al axón se liberan unas sustancias llamadas neurotransmisores al espacio sináptico (espacio que existe al final de una neurona y el inicio de otra). Esta señal química se recibe a través de la dendrita de la neurona adyacente, y así, se produce el impulso eléctrico [26].

En condiciones de reposo, las neuronas presentan una determinada carga eléctrica que es diferente a la carga que hay en el exterior de la neurona. A esta carga se le llama potencial de reposo [26].

Cuando una neurona recibe una estimulación superior a un umbral se desencadena el potencial de acción, de manera que la neurona se despolariza produciendo un impulso nervioso que viaja hasta el axón y permite la liberación de los neurotransmisores, que a su misma vez estimulan los receptores de la neurona postsináptica (colindante) y producen cambios en su potencial. Estos cambios pueden ser excitatorios o inhibitorios. Los primeros facilitan la despolarización necesaria para llegar al umbral y que se genere el potencial de acción, haciendo que la membrana neuronal se vuelva más positiva, mientras que los segundos hacen que la distancia para llegar al potencial de acción sea mayor, volviendo la membrana más negativa [26].

La Red Neuronal Artificial (ANN), está formada por un conjunto de nodos interconectados al que se les llama neuronas. Las conexiones entre las neuronas se establecen a través de valores numéricos llamados pesos [27].

Se puede observar el modelo de funcionamiento de una red neuronal biológica con una artificial en la siguiente imagen [27]:

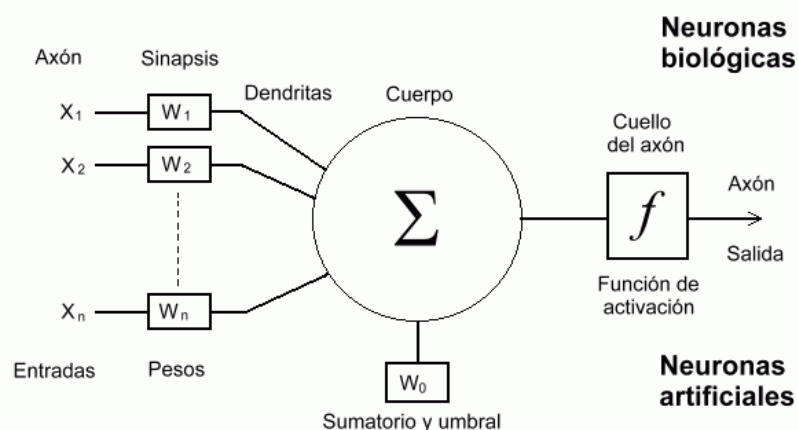


Ilustración 6: Estructura de una ANN [27].

En la imagen se visualiza un ejemplo de red neuronal que tiene n entradas. “Las entradas (X_n) son el estímulo que recibe la neurona artificial del entorno que la rodea, y la salida representa al estímulo”²⁷. La neurona es capaz de acondicionarse a su medio periférico y aprender de él cambiando el valor de sus

pesos sinápticos (W_n). Por ello, los pesos se conocen por ser los parámetros libres del modelo, ya que se pueden cambiar y ajustar para realizar una tarea determinada [27].

Las redes neuronales se caracterizan por tener las siguientes características [29]:

- ✓ Función de activación. Es la función que hace la transformación de todas las señales de entrada (datos) en una sola señal de salida (resultado). Existen múltiples tipos de función de activación como la función sigmoide, la tangente hiperbólica, Softmax, ReLU (Rectified Lineal Unit, por su acotación en inglés), y Leaky ReLU. Las últimas dos se comportan bien en imágenes y tienen un buen desempeño en redes convolucionales [29].

- ✓ Arquitectura o topología de red. “Consiste en la organización y disposición de las neuronas en la red formando las capas o las agrupaciones de neuronas. Es la base del aprendizaje”³⁰. Como se ha explicado anteriormente, al conjunto de neuronas se le llama nodo. En una ANN la sinapsis conecta los nodos, y estos se van agrupando formando las capas. El conjunto de una o más capas se denomina red neuronal [29]. Entre sus características principales se destacan: el número de capas, el número de nodos dentro de una misma capa, la capacidad que tiene la información de viajar hacia atrás. Así pues, cuantas más capas tenga una red, más compleja se vuelve y a su vez, mayor capacidad de aprendizaje y complejidad en las decisiones [30].

- ✓ Algoritmo de entrenamiento. Se establecen los pesos para excitar o inhibir las neuronas [30].

3.5. Redes neuronales convolucionales

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) se pueden definir como un tipo de red neuronal artificial con aprendizaje supervisado en el que las neuronas ejercen como campos receptivos de forma semejante a como lo hacen las neuronas de la corteza visual [17].

Las CNN están formadas por tres tipos de capas [17]:

- ✓ Capas convolucionales.
- ✓ Capas de reducción o pooling: reduce el número de parámetros quedándose los más comunes.
- ✓ Capa totalmente conectada que proporciona la clasificación: proporciona el resultado final.

Funcionamiento de una CNN

Las CNN aprenden por sí solas a reconocer objetos y patrones dentro de imágenes, a partir de una gran cantidad de estas. Es importante que el tamaño muestral sea grande para que la red sea capaz de percibir las características exclusivas de cada objeto y al mismo tiempo, poder generalizarlo [17].

➤ Píxeles de una imagen

Las CNN toman como entrada los píxeles de una imagen. La matriz de píxeles viene determinada por el alto x ancho x los canales de la imagen. Los canales son igual a 1 si la imagen está en escala de grises, o bien igual a 3 con una imagen a color [17]. Por ejemplo, si se tuviera una imagen a color de 26x26x3 habría en total 2028 neuronas de entrada, que equivaldrían a nuestra capa de entrada.

Antes de alimentar la red, los valores de la imagen hay que normalizarlos. Los colores de los píxeles tienen valores en un rango de 0-255, por lo que la conversión de cada pixel será el valor/255, obteniendo un resultado entre 0 y 1 [17].

➤ Convoluciones

Esta etapa es un paso distintivo en las CNN. Las convoluciones consisten en coger “*grupos de píxeles cercanos de la imagen de entrada*”¹⁷ y hacer operaciones matemáticas (productos escalares) contra una pequeña matriz llamada kernel. El kernel detecta todas las neuronas de la capa de entrada (en dirección izquierda-derecha, arriba-abajo) para generar una nueva capa de neuronas ocultas (matriz de salida). Cuando las imágenes son a color (tres canales), se obtendrán tres kernels que se suman para obtener dicha matriz de salida [17].

Los kernels se ajustan mediante *backpropagation*, ya que al principio van tomando valores aleatorios. Al conjunto de kernels se le llama filtro. A medida que se desplaza el kernel, se obtiene una nueva imagen filtrada [17]. Por lo que una red con 16 filtros, daría lugar a 16 matrices de salida y por tanto sería como si se obtuvieran 16 imágenes filtradas nuevas. Dichas imágenes estarán trazando características específicas de la imagen original que ayudarán a realizar la futura clasificación.

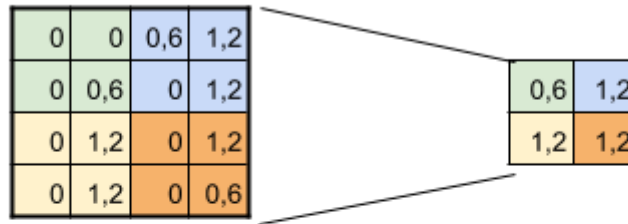
➤ Subsampling

El *subsampling* se realiza como paso previo a una nueva convolución. Aquí se reduce la cantidad de neuronas [17]. Como se viene explicando, a partir de nuestra imagen a color de 26x26x3 se tendrían en total 2028 neuronas de entrada en la primera capa de entrada, además, en la capa de la primera convolución se tendrían 16 filtros o mapas de características de 26x26x3 que equivaldrían a 32448 neuronas en la capa oculta.

Si se hiciera una nueva convolución a partir de esta capa, el número de neuronas sería muy elevado (lo que implica también un mayor procesamiento). Para reducir el tamaño de la próxima capa se realiza el subsampling, que consiste en reducir el tamaño de las imágenes filtradas, pero manteniendo las características principales captadas por cada filtro [17].

Aunque existen muchos tipos de *subsampling*, el más utilizado, y el que se utilizará en el trabajo, es el *Max-Pooling* [17].

Siguiendo con el ejemplo anterior, suponiendo que se aplica un *Max-Pooling* de 2x2 significaría que se van a recorrer cada una de nuestras 16 imágenes de características obtenidas de 26x26x3, pero en vez de tomar 1 pixel, se tomará el valor más alto de esos 4 píxeles (2x2). Cuando esto ocurre, la imagen se reduce a la mitad ya que sería de 13x13 píxeles. Luego, de 32448 neuronas se reducirían a 8112 que siguen almacenando la información más relevante para la detección de las características deseadas [17].



SUBSAMPLING:
 Aplico Max-Pooling de 2x2
 y reduzco mi salida a la mitad

Ilustración 7: Max-Pooling [17].

➤ Clasificación

Después de aplicar todas las capas de convolución necesarias, se construye la última capa que está íntegramente conectada donde da como resultado la clasificación [17].

En la última capa, el número de neuronas corresponderá con el número de clases que se quieren predecir. Además, se debe de “aplanar” la última convolución con una (o más) capas de neuronas ocultas y convertirla en una capa de neuronas “tradicionales”. La salida se hace mediante una función, como por ejemplo la antes mencionada *SoftMax* [17].

4. Metodología

4.1. El conjunto de datos

El conjunto de datos utilizado en este trabajo se ha obtenido a través del repositorio de *The Cancer Imaging Archive*, un repositorio gratuito que dispone de una amplia gama de imágenes de detección de cáncer al alcance de los usuarios.

La base de datos recibe el nombre de *The Chinese Mammography Database* (CMMD) cuyo propietario es Tracy Nolan. La base de datos se construyó en colaboración con la Escuela de Ciencias de la Computación e Ingeniería de la Universidad de Tecnología del Sur de China y fue apoyado parcialmente por la Investigación y Desarrollo de Área Clave de la provincia de Guangdong, la Fundación Nacional de Ciencias Naturales de China, la Fundación de Ciencias Naturales de Guangdong y la Fundación de Salud y Medicina Proyecto de innovación colaborativa de la ciudad de Guangzhou [31].

Este proyecto fue publicado en 2021 y se creó realizando un examen de mamografía a 1775 pacientes de China con tumores en la mama. Posteriormente, dichos tumores fueron confirmados a través de una biopsia, y en 749 pacientes se incluyó el subtipo molecular del cáncer. Los datos de imágenes se adquirieron en un sistema de mamografía GE Senographe DS [31].

El conjunto de datos ocupa en total 22,9GB y son 3728 mamografías en formato DICOM [31].

4.2. Entorno de trabajo

El trabajo se realizó en un ordenador personal de la marca Lenovo. Tiene un procesador Intel(R) Core(TM) i7-7700HQ CPU con 8 GB de RAM. Cuenta con un sistema operativo de 64 bits, procesador basado en x64. Su sistema operativo es Windows 10.

4.3. Lenguaje de programación

El lenguaje de programación elegido inicialmente para el proyecto fue R pero debido a problemas de RAM para almacenar grandes cantidades de imágenes DICOM finalmente, se procedió al uso de Python.

Python es un lenguaje muy apropiado para el procesamiento de grandes cantidades de imágenes y cuenta con varios paquetes que facilitan la implementación de técnicas de aprendizaje automático. Además, permite la creación de códigos entendibles siendo un lenguaje con sintaxis simple y directa [32].

La versión de Python utilizada para el desarrollo del proyecto es 3.9.

4.4. Entorno de programación

Para la utilización de Python en un sistema operativo Windows se ha utilizado Anaconda.

Anaconda es una distribución de Python que funciona como un gestor de entorno, paquetes y tiene una colección amplia de código abierto [33].

Anaconda Distribution se agrupa en cuatro sectores [33]:

- ✓ Anaconda Navigator: Interfaz gráfica de Anaconda Python.
- ✓ Anaconda Project.
- ✓ Librerías de Ciencia de Datos
- ✓ Conda: El gestor de código de Anaconda Python.

La instalación de Anaconda se hizo desde su página web oficial, seleccionando la opción de *Individual Edition* [34]. Una vez descargada, se utilizó la interfaz gráfica de *Anaconda Navigator* para el desarrollo del proyecto.

Se creó un entorno virtual llamado “tfm” para evitar las incompatibilidades entre paquetes y garantizar la estabilidad del sistema.

Se activó el entorno virtual creado y se utilizó la aplicación *Jupyter Notebook* para crear y ejecutar el código Python, pudiendo incluir texto, imágenes o gráficos además del código. En dicho cuaderno se ha ejecutado todo el código necesario para el desarrollo de la red neuronal convolucional.

Además, se utilizó el paquete Flask para el desarrollo de la aplicación web, un marco web de Python que permite crear aplicaciones de manera sencilla.

Se instaló Flask en el entorno virtual creado y se creó un archivo llamado app.py donde se ejecutó el código de la aplicación. También se instalaron otras librerías necesarias para el correcto funcionamiento de la aplicación.

Para la creación del archivo con extensión .py se utilizó la aplicación *Visual Code*, disponible en *Anaconda Navigator*.

4.5. Librerías utilizadas

Las librerías instaladas y utilizadas en el entorno virtual creado han sido las siguientes:

- ✓ Tensorflow

- ✓ Keras
- ✓ Os
- ✓ Numpy
- ✓ Pandas
- ✓ Matplotlib
- ✓ Sys
- ✓ Sklearn
- ✓ Pydicom
- ✓ Flask
- ✓ Cv2
- ✓ Glob

4.6. Importación de datos

Una vez el conjunto de imágenes descargado de la página web de *Cancer Imaging Archive* fueron almacenadas en una carpeta dentro de nuestro ordenador llamada “Imágenes TFM/CMMD”.

Una vez aquí, se crearon dos subcarpetas con la clase de las imágenes Benignos o Malignos, y se introdujo cada tipo de imagen en su correspondiente carpeta. Además, teniendo en cuenta los datos del Excel de la base de datos se eliminaron las fotos repetidas, esto se hizo basándose en la columna del Excel de “número de imagen” que ponía en cuál de las imágenes se identificaba la patología. También se eliminaron las imágenes con mala calidad. Para ello, se utilizó el programa *Dicom viewer* descargado a través de página web oficial [35].

Se establecieron tres directorios distintos uno principal, donde están los datos de los Excel descargados, y los otros con las dos subcarpetas de benignos o malignos.

Para cargar las imágenes y poder trabajar con ellas, se utilizó la librería *os*. Las imágenes se importaron por clase y definiendo una función para ello. Para la lectura de las imágenes se utilizó el paquete *pydicom* con la función “*dcmread*”. Las imágenes fueron guardadas en dos listas respectivas a Benignos o a Malignos.

4.7. Análisis exploratorio de los datos

El análisis exploratorio de los datos es lo primero que hay que hacer antes de implementar las técnicas de aprendizaje automático, esto se realiza con el objetivo de tener más información acerca de los datos.

Se visualizó una imagen aleatoria de entre la lista de benignos:

La imagen mostrada es la nº 150 y corresponde a la etiqueta Benignos

```
Out[9]: <matplotlib.image.AxesImage at 0x26ae9cd3070>
```

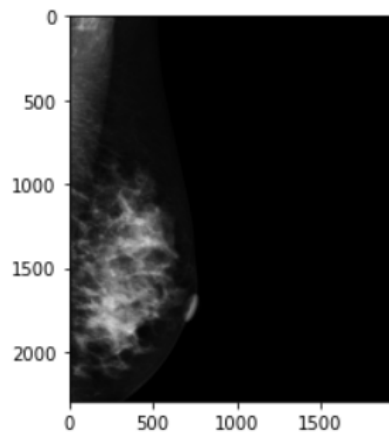


Ilustración 8: Ejemplo de imagen de mamografía con tumor benigno. Elaboración propia.

Después, se hizo lo mismo con la lista de imágenes malignos:

La imagen mostrada es la nº 202 y corresponde a la etiqueta Malignos

```
Out[20]: <matplotlib.image.AxesImage at 0x2296df1a190>
```

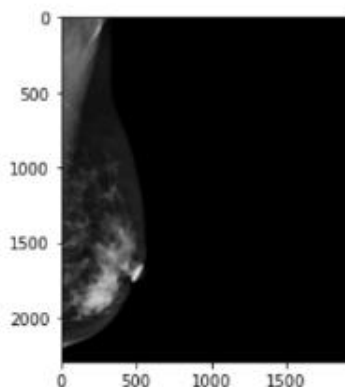


Ilustración 9: Ejemplo de imagen de mamografía con tumor maligno. Elaboración propia.

Como las imágenes DICOM tienen asociadas un objeto tipo dataset por cada imagen, también se visualizó la información asociada.

Para analizar mejor los datos se crea un *data frame* extrayendo distintas características de las imágenes:

- Clase: Clase de imagen.
- Tamaño: Nos indica el tamaño de la imagen.
- Modo: Nos indica el modo de imagen, 1-correspondiente a el modo en una escala de grises, 3- correspondiente a RGB.

El nuevo conjunto de datos se creó con el paquete *pandas* y la función *pd.DataFrame()*. Con la función *Print()* se visualizan las primeras filas y columnas del conjunto de datos.

4.8. Preprocesamiento de los datos

➤ Reducción y balanceado del conjunto de datos

El total del conjunto de datos estaba formado por 1749 imágenes. 1283 perteneciendo a la clase malignos y 466 a la clase benignos. Para que el conjunto de datos esté balanceado, se escogieron 466 imágenes al azar de la lista cargada con las imágenes de clase maligna. Para ello se utilizó la función *sample* del paquete *random*.

➤ Carga de las etiquetas de imagen en una lista

Se cargan las clases de imagen benignas o malignas en una nueva lista.

➤ Preprocesamiento de las imágenes

Antes de introducir los datos de las imágenes en el modelo es necesario realizar un procesamiento. Para ello, se creó una función llamada *Preprocesamiento()* que se encarga de lo siguiente:

- Transformación de las imágenes en un array. La red neuronal funciona con los datos de la imagen, por ello, lo primero es transformar las imágenes en un vector formado por los píxeles de cada imagen. Para ello, se utiliza la función *pixel_array* de la librería *pydicom*.
- Transformación del modo de imagen a RGB. El modo de imagen se transforma para que tenga 3 canales. Dicha transformación es importante para que todas las imágenes tengan la misma cantidad de canales. Para ello, se utiliza el paquete *cv2* y se utiliza la función *cvtColor()* con un argumento llamado *cv2.COLOR_GRAY2RGB*.
- Redimensionamiento de las imágenes. El tamaño de las imágenes de nuestro conjunto de datos es 2294x1914 píxeles por 3 canales (RGB). Debido al tamaño elevado de las imágenes y el coste computacional tan grande, se reducen las imágenes a 256x256. Para esto, se utiliza la función *resize()* de la librería *cv2*.
- Normalización de los píxeles. Los valores de los píxeles se encuentran entre 0 y 255. Para hacer la normalización se deben dividir los valores de los píxeles de imagen entre 255 para obtener valores dentro del intervalo 0 y 1. Esto ayuda a que el algoritmo funcione más rápido. Además, es importante para poner todas las características (píxeles) en la misma escala y que ninguna esté dominada por otra.

Esta función se aplica para todas las imágenes que se utilizan para la creación del modelo. Al aplicar la función las imágenes preprocesadas se guardan independientemente de su clase en una nueva lista.

➤ Codificación de la variable clase

Se codifican las clases asignando valores numéricos de entre 0 y 1 a cada valor de la variable clase ("Benigno", "Maligno"). Esto es necesario para que el algoritmo funcione.

Se crea un diccionario de manera que la variable clase benigno coja el valor de 0 y la clase maligno corresponda al 1. Para asignar a cada clase su valor numérico se utiliza el método `get()` que itera por la lista guardada con las clases.

➤ Transformación de las listas en vectores

Se realiza la transformación de las listas en vectores con la función `array()` del paquete Numpy. Esto es necesario ya que el formato lista no es aceptado para los modelos de CNN.

➤ Creación de los conjuntos de *train*, *test* y *validation*

Se divide el conjunto de datos en tres subconjuntos:

- Conjunto de datos de *train* o entrenamiento: Formado por las imágenes que se utilizan para entrenar y ajustar el modelo.
- Conjunto de datos del *test* o prueba: Imágenes utilizadas para evaluar el modelo final. No forman parte del entrenamiento y nos proporcionan la información del error real que ha cometido el modelo.
- Conjunto de *validation* o validación: Corresponde al conjunto de imágenes utilizadas para evaluar el modelo sin tener una tendencia determinada cuando se ajustan los hiperparámetros del modelo.

Se divide el 75% de las imágenes al entrenamiento y el 25% para la prueba. De las imágenes destinadas al entrenamiento, se destina a su vez, el 25% para la validación.

Para dividir los datos se utiliza la función `train_test_split()` de la librería Sklearn.

Finalmente, se obtienen tres subconjuntos formados por 524 imágenes para el entrenamiento, 233 para la prueba y 175 para la validación.

➤ One-hot encoding

Se debe realizar un one-hot encoding de la variable clase. Esto significa crear una columna para cada clase y colocar un 1 en la columna que corresponde con

la clase y 0 en lo demás. Para ello, se ha utilizado la función *to_categorical()* de la librería keras.

4.9. Construcción del modelo CNN

A continuación, se detallan las distintas características utilizadas para crear la red neuronal convolucional:

- En primer lugar, se utiliza la función *Sequential()* de la librería Keras que agrupa un conjunto de capas. A este modelo se le siguen añadiendo las capas con una función llamada *add()*.
- En total, el modelo está formado por tres capas de convolución añadidas con la clase *Conv2D()* donde se especifica el número de filtros y su tamaño. Además, se añade la función de activación utilizada en las capas convolucionales: ReLu. Como se ha explicado anteriormente, esta función es una de las más adecuadas cuando los datos de entrada de la red son imágenes y es la más utilizada en las CNNs.
- La primera capa de convolución viene determinada por el tamaño de nuestros datos de entrada, que en el caso del trabajo es de 256x256x3 habiendo reducido previamente las imágenes durante el Preprocesamiento.
- El tamaño del filtro o *kernel* en todas las capas convolucionales ha sido de 4x4.
- El número de filtros va aumentando en todas las capas, siendo respectivamente de 32,64 y 128.
- Después de cada capa de convolución se añade la capa de reducción. El tipo de reducción utilizado es el *Max Pooling* y se añade con una función llamada *MaxPooling2D()*. Esta capa sirve para reducir el coste computacional ya que reduce la muestra de entrada tomando los valores más característicos de los datos. Su tamaño ha sido de 2x2.
- Después de las capas convolucionales se añade la capa de aplanamiento. Esto se hace con la función *Flatten()* así se aplanan las dimensiones de la información de entrada.
- Añadimos una capa completamente conectada (la capa de neuronas tradicionales) con 128 filtros y la función de activación ReLu.
- Por último, se añade la capa de salida, formada por 2 neuronas ya que hay dos clases a predecir. Esto devuelve un resultado en forma de probabilidad, que corresponde a la probabilidad de que la clase pertenezca realmente a su clase. En este caso, la función de activación ha sido la sigmoidea.

Una vez definido el modelo se debe compilar, esto se hace mediante la función *compile()*.

La función de coste calcula el error entre el estimado y el real para optimizar los parámetros de la red. En este caso se ha utilizado la función *binary_crossentropy*.

Model: "sequential"

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 253, 253, 32)	1568
max_pooling2d (MaxPooling2D)	(None, 126, 126, 32)	0
conv2d_1 (Conv2D)	(None, 123, 123, 64)	32832
max_pooling2d_1 (MaxPooling2D)	(None, 61, 61, 64)	0
conv2d_2 (Conv2D)	(None, 58, 58, 128)	131200
max_pooling2d_2 (MaxPooling2D)	(None, 29, 29, 128)	0
flatten (Flatten)	(None, 107648)	0
dense (Dense)	(None, 128)	13779072
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 2)	258

=====
 Total params: 13,944,930
 Trainable params: 13,944,930
 Non-trainable params: 0
 =====

Ilustración 10: Modelo Secuencial de CNN. Elaboración propia.

4.10. Entrenamiento y validación del modelo

El modelo ha sido entrenado con la función *fit()* con las imágenes y sus respectivas etiquetas del conjunto de datos de train o entrenamiento.

El *batch_size* o tamaño de lote indicado ha sido de 32. Esto significa que el modelo utiliza 32 imágenes en cada actualización de gradiente.

Para entrenar el modelo se han fijado 50 épocas.

Una vez que el modelo ha sido entrenado, se evalúa su precisión con la función *evaluate()* y el conjunto de imágenes y etiquetas del test o prueba.

4.11. Mejora del modelo

Para aumentar la precisión del modelo se ha realizado la siguiente técnica:

Transferencia de aprendizaje

Se utiliza un modelo previamente entrenado sobre los datos.

Un modelo previamente entrenado es un modelo que ha sido entrenado con gran cantidad de imágenes de manera que las características aprendidas por dicho modelo son útiles para nuestra clasificación, aunque las clases de ambos conjuntos de datos sean totalmente diferentes.

El modelo previamente entrenado escogido es la arquitectura VGG16 que fue creada en el año 2015 por Karen Simonyan y Andrew Zisserman. Este modelo fue entrenado con unas 1.4 millones de imágenes y 10.000 clases diferentes [36].

Para utilizar dicho modelo se ha realizado la extracción de características que consiste en utilizar las representaciones que han sido aprendidas por un modelo previamente entrenado para que en las nuevas muestras se extraigan características interesantes. Dichas características se extraen tomando la parte convolucional de la red VGG16 y ejecutando los datos a través de ella para finalmente entrenar un clasificador densamente conectado en la parte superior.

Primero, se importa el modelo VGG16 mediante la función VGG16() de la librería Keras. Se especifica el argumento “include_top = False” para que el clasificador de la red preentrenada no se incluya en el modelo ya que nuestras clases son distintas.

Model: "vgg16"		
Layer (type)	Output Shape	Param #

input_1 (InputLayer)	[(None, 256, 256, 3)]	0
block1_conv1 (Conv2D)	(None, 256, 256, 64)	1792
block1_conv2 (Conv2D)	(None, 256, 256, 64)	36928
block1_pool (MaxPooling2D)	(None, 128, 128, 64)	0
block2_conv1 (Conv2D)	(None, 128, 128, 128)	73856
block2_conv2 (Conv2D)	(None, 128, 128, 128)	147584
block2_pool (MaxPooling2D)	(None, 64, 64, 128)	0
block3_conv1 (Conv2D)	(None, 64, 64, 256)	295168
block3_conv2 (Conv2D)	(None, 64, 64, 256)	590880
block3_conv3 (Conv2D)	(None, 64, 64, 256)	590880
block3_pool (MaxPooling2D)	(None, 32, 32, 256)	0
block4_conv1 (Conv2D)	(None, 32, 32, 512)	1180160
block4_conv2 (Conv2D)	(None, 32, 32, 512)	2359808
block4_conv3 (Conv2D)	(None, 32, 32, 512)	2359808
block4_pool (MaxPooling2D)	(None, 16, 16, 512)	0
block5_conv1 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv2 (Conv2D)	(None, 16, 16, 512)	2359808
block5_conv3 (Conv2D)	(None, 16, 16, 512)	2359808
block5_pool (MaxPooling2D)	(None, 8, 8, 512)	0

Total params: 14,714,688		
Trainable params: 14,714,688		
Non-trainable params: 0		

Ilustración 11: Características principales de la red VGG16. Elaboración propia.

Después, se procede a la creación y al establecimiento de nuestro clasificador densamente conectado y se entrena el modelo con los datos.

4.12. Diseño de la aplicación web

La aplicación web ha sido creada con Flask, un *framework* de Python que se utiliza de forma muy sencilla para crear aplicaciones.

En el entorno virtual de nuestro proyecto, “tfm”, se realiza la instalación de Flask. Una vez instalado se utiliza Visual Code Studio, disponible en *Anaconda Navigator* y se crea un archivo .py.

Para la visualización del contenido de la aplicación web, Flask tiene una función llamada *render_template()* que permite el uso de plantillas Jinja, que son archivos en código HTML.

Para utilizar el modelo de red convolucional creado y realizar las predicciones se utiliza el modelo previamente guardado con la extensión .json. Se cargan tanto el modelo como sus pesos.

En la plantilla con el formato HTML se diseña la pantalla principal de nuestra página web, en dicha pantalla aparece el título de nuestra aplicación. En dicha pantalla se indica al usuario que debe de seleccionar una imagen desde su ordenador para poder realizar la predicción.

Una vez se obtiene la imagen seleccionada se realiza su procesamiento de manera similar al realizado en el apartado de preprocesado:

- ✓ Se convierte la imagen en una matriz de píxeles,
- ✓ Se convierte a modo RGB.
- ✓ Se redimensionan las imágenes.

Cuando se hace la selección de la imagen desde la pantalla principal se pulsa el botón “*predict*” para realizar la predicción.

4.13. Publicación de la aplicación web

La publicación de la página web ha sido realizada a través de PythonAnywhere y Heroku, aunque sin éxito en ninguna de las dos.

En el caso de PythonAnywhere, el problema reside en que las cuentas de prueba tienen una memoria de 500Mb gratuita establecida [37] y para lanzar la aplicación creada con Flask hay dos opciones: la primera, utilizar la versión de Python 3.8 donde ya vienen todos los programas instalados. Opción que no ha sido posible utilizar, dado que no se puede sobrescribir la carpeta de los

paquetes instalados en esa versión de Python, cosa necesaria para instalar nuestro paquete de lectura de las imágenes DICOM (paquete pydicom). Por otro lado, otra solución podría ser utilizar la versión de Python 3.9, en la cual sí que se pueden instalar nuevos paquetes. El inconveniente aquí, ha sido el ya mencionado problema de la memoria gratuita, ya que al instalar las librerías Tensorflow, Flask y Pydicom (entre otras) se ocupa toda la memoria disponible.

En el caso de Heroku [38], ocurre exactamente lo mismo. Cuando se realiza la compilación de los paquetes la página muestra un error diciendo que el tamaño máximo que deja compilar es de 500Mb. Y el tamaño total a compilar con los paquetes utilizados para la página web es de 586.6Mb. En la siguiente ilustración se adjunta la captura de pantalla de Heroku:

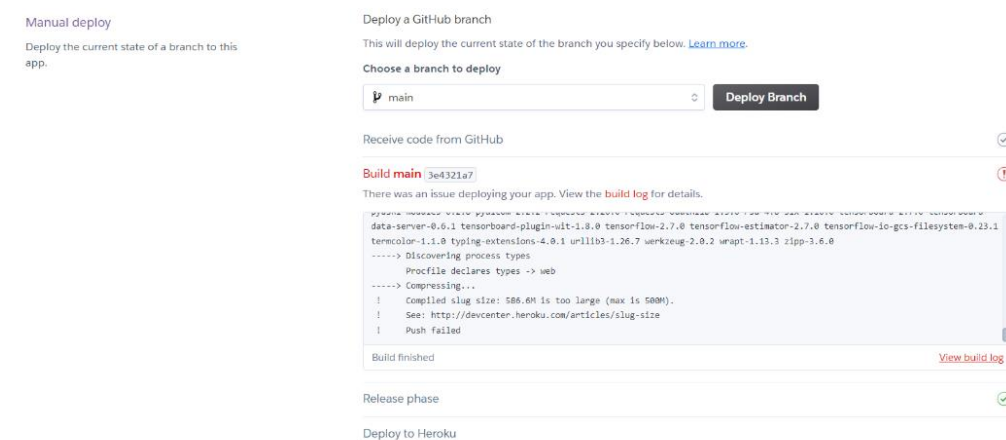


Ilustración 12: Lanzamiento de la aplicación a través de Heroku. Elaboración propia.

No obstante, se adjunta un tutorial en forma de vídeo para mostrar el funcionamiento de la página web, accediendo desde [aquí](#).

5. Resultados

5.1. Exposición de los resultados de cada modelo

A continuación, se detallan las características de precisión y pérdida de cada modelo creado junto con la visualización de la etapa de entrenamiento.

5.1.1 Modelo general

Modelo formado por tres capas de convolución. Ha sido entrenado con 466 imágenes de cada clase.

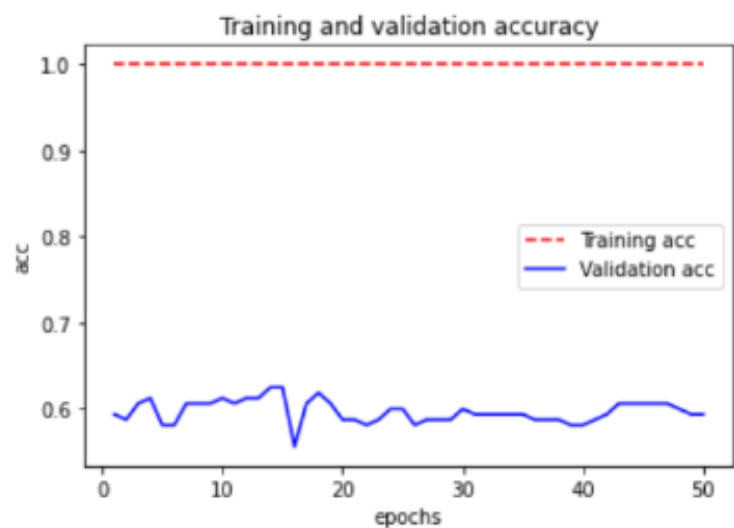


Ilustración 13: Gráfica de la precisión durante el entrenamiento y validación del modelo general. Elaboración propia.

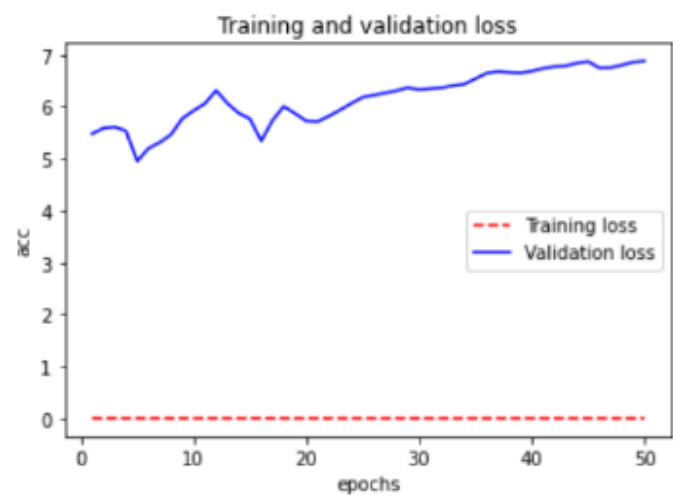


Ilustración 14: Gráfica de la función de pérdida durante el entrenamiento y validación del modelo general. Elaboración propia.

Precisión	0.5622
Pérdida	7.9857

Tabla 4: Métricas del modelo general. Elaboración propia.

En los gráficos adjuntos se puede observar el típico patrón de sobreajuste. La precisión del entrenamiento se conserva en el 100% desde el principio, pero, la de validación se queda estancada a partir de las 20 épocas. En la pérdida, se puede observar como con los datos del entrenamiento se mantiene en cero, sin

embargo, para los datos de validación aumenta a medida que aumentan las épocas.

El denominado *Overfitting* o sobreajuste es muy habitual cuando hay tan pocos datos ya que el modelo aprende características muy específicas en lugar de generalizar los resultados (solo se consideran válidos los datos idénticos a los de entrenamiento).

5.1.2. Modelo con transferencia de aprendizaje

Se ha utilizado la red pre-entrenada VGG16, pero con un clasificador personalizado a nuestra clasificación densamente conectado.

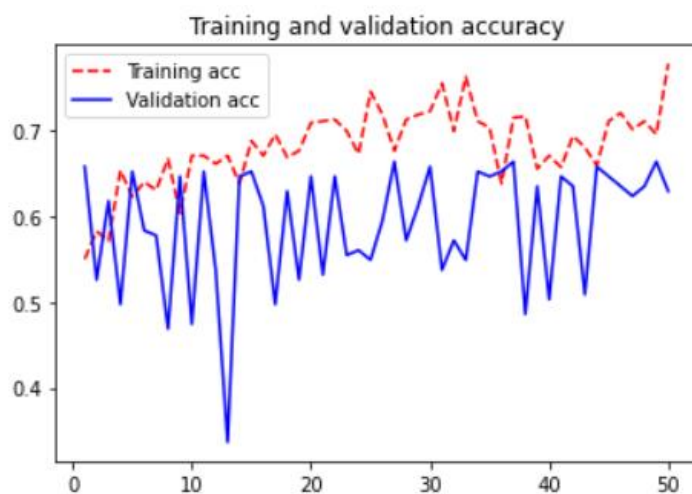


Ilustración 15: Gráfica de la precisión durante el entrenamiento y validación del modelo con transferencia de aprendizaje. Elaboración propia.

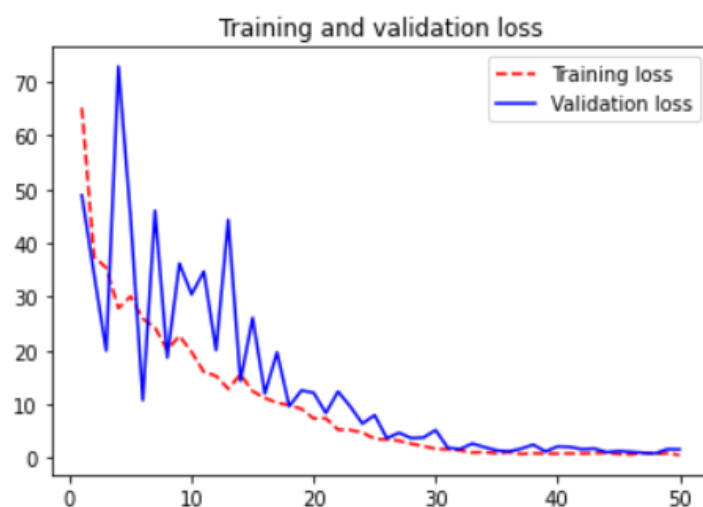


Ilustración 16: Gráfica de la pérdida durante el entrenamiento y validación del modelo con transferencia de aprendizaje. Elaboración propia.

Precisión	0.6285
Pérdida	1.6401

Tabla 5: Métricas del modelo con transferencia de aprendizaje. Elaboración propia.

En estos gráficos se puede observar como las métricas de entrenamiento y validación tienen más o menos la misma tendencia. Además, el sobreajuste se produce desde el principio.

La pérdida es de 1.64, lo que indica que hay sobreajuste.

5.2. Comparación de modelos

Modelo	Precisión	Pérdida	Épocas	Seg/step	Seg/época	Tiempo
General	0.5622	7.9857	50	2seg	25seg	21 min
VGG16	0.6285	1.6401	50	23seg	444seg	6h 16min

Tabla 6: Comparación de las métricas de los modelos. Elaboración propia.

Se puede destacar el elevado coste computacional para el modelo con transferencia de aprendizaje. También se puede observar como el mejor modelo es al que se le ha realizado dicha transferencia.

5.3. Elección final de un modelo

El modelo que se ha considerado mejor es el de la transferencia de aprendizaje con el modelo VGG16. Aunque el modelo general tiene un ajuste de los datos de entrenamiento del 100% luego tiene mucho sobreajuste. Sin embargo, el tiempo computacional empleado es mucho mayor para el modelo VGG16, por ello, se ha utilizado el modelo general para realizar la página web.

5.4. Uso de la aplicación web

A continuación, se muestra un breve tutorial del uso de la página web:

La página principal de la página web se muestra en la siguiente ilustración.

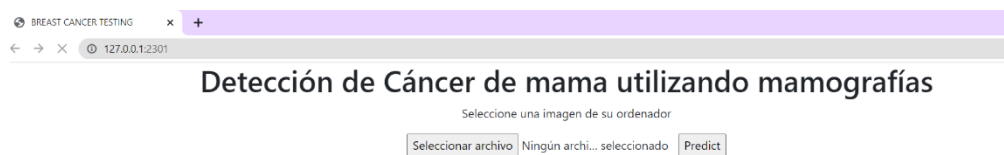


Ilustración 17: Página principal de la página web. Elaboración propia.

En la pantalla principal aparece un botón para indicarle al usuario que deberá seleccionar una imagen de su ordenador. Como se puede ver en la siguiente imagen, al pulsar este botón se tiene acceso a todas las carpetas existentes en el computador.

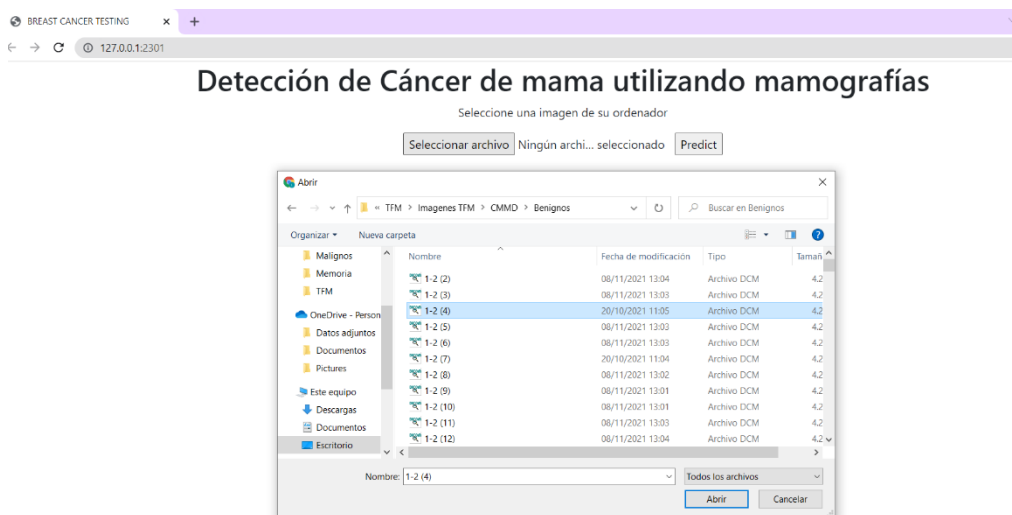


Ilustración 18: Selección de la imagen para importarla en la página web. Elaboración propia.

Cuando el usuario selecciona la imagen deseada se pulsa en el botón “abrir” para cargarla.

Una vez que la imagen está cargada se deberá pulsar en el botón “Predict” para realizar la predicción.

Finalmente, como se puede observar en la siguiente imagen, la aplicación devuelve el tipo de tumor detectado en la imagen junto a su precisión:



Ilustración 19: Resultado de la predicción de la aplicación. Elaboración propia.

6. Discusión

El cáncer es uno de los mayores retos de la medicina actual. En concreto, el cáncer de mama se ha convertido en el tipo de cáncer más diagnosticado en todo el mundo. Esto hace que cada vez haya más proyectos de investigación tanto para mejorar la detección precoz como para tratarlo.

Con este trabajo no se pretende diagnosticar a una persona cáncer de mama, sin embargo, puede ser una herramienta muy útil para ayudar a los profesionales y conseguir así, una detección precoz y en consecuencia mayores expectativas para el paciente al administrar los tratamientos.

Se ha demostrado que con pocas líneas de código se puede lograr una herramienta revolucionaria en el ámbito de la medicina. Sin embargo, un ordenador personal no es suficiente para crear una precisión muy alta ya que no tiene un servidor tan potente como en los artículos de referencia mencionados.

Este proyecto está enfocado de manera similar a los existentes en la bibliografía como por ejemplo en el de Geras et al., 2017²¹ que utiliza como datos de entrada mamografías o en el de Shen et al., 2019²⁰ que además de utilizar mamografías, utiliza la misma red preentrenada para hacer la transferencia de aprendizaje VGG16.

Por otro lado, al igual que en el estudio de Spanhol, Oliveira, Petitjean, & Heutte, 2016¹⁸ la red entrenada desde cero en este trabajo, consta de 3 capas convolucionales con activación ReLu en todas ellas. Además, también se hace una clasificación del tipo benigno vs maligno.

Sin embargo, en todos los modelos se alcanzó un nivel de precisión mayor al alcanzado con nuestro algoritmo, debido a que en la mayoría el número de

épocas utilizadas fue mucho mayor, y esto es difícil hacerlo mediante un ordenador personal debido al alto coste computacional.

7. Conclusiones

7.1. Conclusiones generales

Se han conseguido desarrollar dos modelos de redes neuronales convolucionales con una precisión mínima del 56% para la clasificación de tumores de mama. El modelo que ofrece un mayor rendimiento, aunque el gasto computacional fuera muy elevado, es el realizado por transferencia de aprendizaje conectada a la red VGG16, llegando a una precisión del 62%.

No obstante, el modelo que se ha utilizado para el desarrollo de la página web ha sido el modelo general, debido al tiempo computacional y al peso del archivo de la red.

Al realizar el proyecto, ha quedado constancia de la importancia de disponer de una base de datos amplia para entrenar las redes neuronales y, además, utilizar entornos con mayor capacidad como el sistema operativo Linux o el acceso a plataformas como Google Colab que tiene acceso gratuito a servidores GPUs.

7.2. Líneas de futuro

Existen varias modificaciones que se le pueden aplicar a este proyecto:

- ✓ El modelo es mejorable en varios aspectos:
 - En primer lugar, se podría implementar el modelo en un entorno GPU modificando varios parámetros de entrenamiento o aumentando el número de imágenes.
 - Un adecuado preprocesamiento de imágenes podría incluir un aislamiento del tumor creando un código que sea capaz de eliminar el músculo mamario e identificar las masas y/o calcificaciones. Aunque dicha práctica se intentó como primer método de preprocesado el tiempo y espacio requerido computacionalmente era enorme para el uso de un ordenador personal, como el utilizado en este proyecto.

✓ Aumento de la base de datos para el entrenamiento. Lo verdaderamente útil de la aplicación sería que al introducir un usuario una nueva imagen pasara a formar parte de la base de datos, consiguiendo así una base de datos más amplia. Además, también se le podría pedir al profesional que indicara un nivel de satisfacción con la clase predicha por el modelo.

✓ Construcción de un modelo que prediga los subtipos de cáncer de mama. La primera idea de proyecto fue la de crear una aplicación que predijera los subtipos de cáncer de mama a partir de las mamografías. Realmente esta sería una opción interesante ya que para obtener un diagnóstico de este tipo se necesita realizar la técnica de microarray (que no es tan asequible como una simple mamografía). El mayor problema al intentar hacer esto fue que no existe una base de datos amplia que tenga las mamografías previamente clasificadas en distintos subtipos moleculares de cáncer de mama. Por lo que primero haría falta compilar este tipo de información.

✓ Desplegar la página web en un dominio accesible para todo el mundo, ya sea por la compra de un dominio web o por el pago para aumentar la memoria de serie que traen las plataformas PythonAnywhere y Heroku.

El aprendizaje profundo es una técnica de apoyo muy útil para el diagnóstico de imágenes médicas por los radiólogos. En un futuro estas técnicas deberían estar implementadas en los programas que utilizan los sanitarios para visualizar las mamografías.

7.3. Seguimiento de la planificación

La planificación se intentó llevar al día según el calendario inicial, pero en el transcurso del proyecto se ha tenido que ir modificando debido a imprevistos que han ido surgiendo.

En primer lugar, el proyecto estaba inicialmente pensado para realizarse en Rstudio, pero debido a la cantidad de RAM requerido por R se tuvo que abortar el proceso y cambiar el lenguaje de programación a Python con sus consecuentes instalaciones de programas.

Por otro lado, el preprocesado de las imágenes ocupó más tiempo del debido, ya que la importación de archivos tipo DICOM dificultaba el trabajo y requería más espacio de almacenamiento. Además, como se ha comentado en el apartado anterior, el preprocesado inicial de las imágenes estaba pensado para realizarse de otra manera, pero los requerimientos computacionales eran demasiado para mi ordenador personal.

Finalmente, algunas actividades se planearon con una duración inadecuada, especialmente la “creación de la aplicación web” y la “publicación” de la misma, ya que al requerir más tiempo en otras actividades no se pudieron hacer a tiempo.

8. Glosario

- ANN: del inglés, Artificial Neural Network.
- CM: Cáncer de Mama.
- CNN: del inglés, Convolutional Neural Network.
- DL: del inglés, Deep Learning.
- ER: del inglés, Estrogen Receptor.
- IA: Inteligencia Artificial.
- PR: del inglés, Progesterone Receptor.
- RGB: del inglés, Red Green Blue.

9. Bibliografía

A continuación, se enumeran las fuentes bibliográficas utilizadas en el proyecto:

1. **Pérez García JM, Muñoz Couselo E, Cortés Castán J.** ¿Qué es el cáncer de mama? Comprender el cáncer mama, Barcelona: Amat Editorial; 2013.
2. **Torrades S.** El origen genético del cáncer de mama. *Offarm* 2003, 22, 108-12.
3. **Organización Mundial de la Salud (2021).** Lyon (France): Cancer Today - International Agency for Research on Cancer. [Accedido el 21 de septiembre de 2021]. < <https://gco.iarc.fr/today/home> >
4. **Medical News Today (2021).** “¿¿Cómo se ve el cáncer de mama en una mamografía?”. [Artículo en línea]. [Accedido el 16 de diciembre del 2021]. <<https://www.medicalnewstoday.com/articles/es/imagenes-de-mamografia>>
5. **Monnot GC, Romero P.** Rationale for immunological approaches to breast cancer therapy. *Breast* 2018;37:187-95.
6. **Pareja F, Marchiò C, Reis-Filho JS.** Molecular diagnosis in breast cancer. vol. 24. Elsevier Ltd; 2018.
7. **Alberts B.** Principios generales de la señalización celular. *Introd. a la Biol. Cel., Médica Panamericana*; 2011, p. 531-71.
8. **Antón A.** Inmunología tumoral e inmunoterapia del cáncer. Zaragoza: AMAZING BOOKS; 2018.
9. **Bardia A, Hurvitz SA, Tolaney SM, Loirat D, Punie K, Oliveira M, et al.** Sacituzumab Govitecan in Metastatic Triple-Negative Breast Cancer. *N Engl J Med* 2021;384:1529-41. doi:10.1056/nejmoa2028485.
10. **La Información (2013).** “¿A qué edad hay que hacerse la primera mamografía?”. [Artículo en línea]. [Accedido el 2 de octubre de 2021]. <https://www.lainformacion.com/asuntos-sociales/a-que-edad-hay-que-hacerse-la-primera-mamografia_8pqhgw2ilmbcfdxjzrfhm6/>
11. **Breast Cancer (2012).** “Mammography Technique and Types”. [Artículo en línea]. [Accedido el 14 de noviembre de 2021]. <<https://www.breastcancer.org/symptoms/testing/types/mammograms/types>>
12. **Munir K, Elahi H, Ayub A, Frezza F, Rizzi A.** Cancer diagnosis using deep learning: A bibliographic review. *Cancers (Basel)* 2019;11:1-36. doi:10.3390/cancers11091235
13. **Shen L, Margolies LR, Rothstein JH, Fluder E, McBride R, Sieh W.** Deep Learning to Improve Breast Cancer Detection on Screening Mammography. *Sci Rep* 2019;9:1-12. doi:10.1038/s41598-019-48995-4.
14. **National Cancer Institute's (2021).** Cancer Imaging Archive. Saint Louis (USA): The Cancer Imaging Archive. [Accedido el 4 de octubre de 2021]. <Recuperado de: www.cancerimagingarchive.net/ >
15. **Post DICOM (2021).** “Un curso acelerado sobre el manejo de datos de imágenes médicas de DICOM”. [Artículo en línea]. [Accedido el 16 de diciembre de 2021]. < <https://www.postdicom.com/es/blog/handling-dicom-medical-imaging-data> >

16. **ICHI.PRO (2020).** “Matemáticas y estadísticas detrás del aprendizaje automático – PARTE 3”. [Artículo en línea] [Accedido el 10 de octubre de 2021]. <<https://ichi.pro/es/matematicas-y-estadisticas-detras-del-aprendizaje-automatico-parte-3-252750161062386> >
17. **Aprende Machine Learning (2018).** “¿Cómo funcionan las Convolutional Neural Networks? Visión por Ordenador”. [Artículo en línea]. [Accedido el 10 de octubre de 2021]. < <https://www.aprendemachinelearning.com/como-funcionan-las-convolutional-neural-networks-vision-por-ordenador/> >
18. **Spanhol FA, Oliveira LS, Petitjean C, Heutte L.** Breast cancer histopathological image classification using Convolutional Neural Networks. *Proc Int Jt Conf Neural Networks* 2016;2016-October:2560-7. doi:10.1109/IJCNN.2016.7727519.
19. **Zhang Y, Chen JH, Lin Y, Chan S, Zhou J, Chow D, et al.** Prediction of breast cancer molecular subtypes on DCE-MRI using convolutional neural network with transfer learning between two centers. *Eur Radiol* 2021;31:2559-67. doi:10.1007/s00330-020-07274-x.
20. **Shen L, Margolies LR, Rothstein JH, Fluder E, McBride R, Sieh W.** Deep Learning to Improve Breast Cancer Detection on Screening Mammography. *Sci Rep* 2019;9:1-12. doi:10.1038/s41598-019-48995-4.
21. **Geras KJ, Wolfson S, Shen Y, Wu N, Kim SG, Kim E, et al.** High-Resolution Breast Cancer Screening with Multi-View Deep Convolutional Neural Networks 2017:1-9.
22. **Berzal Fernando (2017).** “Breve historia de la inteligencia artificial: el camino hacia la empresa”. [Artículo en línea]. [Accedido el 22 de noviembre de 2021]. <<https://www.cesce.es/es/w/asesores-de-pymes/breve-historia-la-inteligencia-artificial-camino-hacia-la-empresa>>
23. **Ramos López Beatriz (2019).** “¿Cuáles son las diferencias entre ¿Cuáles son las diferencias entre *Data Science*, *Machine Learning*, *Deep Learning* y *Artificial Intelligence*?”. [Artículo en línea]. [Accedido el 22 de noviembre de 2021]. < <https://www.cursosgis.com/cuales-son-las-diferencias-entre-data-science-machine-learning-deep-learning-y-artificial-intelligence/> >
24. **Handelman, GS., Kok, HK., Chandra, RV., Razavi, AH., Lee, MJ., Asadi, H (2018).** “eDoctor: machine learning and the future of medicine (review)”. *J Intern Med* [Artículo en línea]. 603-619 (vol. 284). [Accedido el 22 de noviembre de 2021]. < <https://onlinelibrary.wiley.com/doi/full/10.1111/joim.12822> >
25. **HISOUR (2021).** “Aprendizaje automático”. [Artículo en línea]. [Accedido el 22 de noviembre de 2021]. < <https://www.hisour.com/es/machine-learning-42773/> >
26. **Lafarga Coscojuela MA.** *Biología Celular de la neurona y de la sinapsis*, Santander. Editorial: Servicio de Publicaciones de la Universidad de Cantabria, D.L; 1994.
27. **Escuelapedia (2021).** “Sobre las neuronas”. [Artículo en línea]. [Accedido el 22 de noviembre de 2021]. < <https://www.escuelapedia.com/sobre-las-neuronas/> >
28. **Sancho Caparrini Fernando (2019).** “Redes Neuronales: una visión superficial”. [Artículo en línea]. [Accedido el 18 de noviembre de 2021].

- < <http://www.cs.us.es/~fsancho/?e=72>>
29. **Calvo Diego (2018)**. "Función de activación – Redes neuronales". [Artículo en línea]. [Accedido el 15 de noviembre de 2021]. <<https://www.diegocalvo.es/funcion-de-activacion-redes-neuronales/>>
 30. **Sociedad Andaluza de Educación Matemática Thales (2017)**. "Características de las Redes Neuronales". [Artículo en línea]. [Accedido el 15 de noviembre de 2021]. <<https://thales.cica.es/rd/Recursos/rd98/TecInfo/07/capitulo3.html>>
 31. **Tracy Nolan (2021)**. "La base de datos china de mamografías (CMMD)". [Repositorio Cancer Imaging Archive]. [Descargadas el día 19 de octubre de 2021]<<https://wiki.cancerimagingarchive.net/pages/viewpage.action?pageId=70230508#70230508bcab02c187174a288dbcbf95d26179e8>>
 32. **Graph Everywhere**. "Machine Learning en Python". [Artículo en línea]. [Accedido el 18 de diciembre de 2021]. <<https://www.grapheverywhere.com/machine-learning-en-python/>>
 33. **Tokio School (2021)**. "Ciencia de datos: Anaconda Python". [Artículo en línea]. [Accedido el 19 de diciembre de 2021]. <<https://www.tokioschool.com/noticias/ciencia-datos-anaconda-python/>>
 34. **Anaconda Documentation**. "Getting started with Anaconda". [Guía de usuario en línea]. [Accedido el 11 de octubre de 2021]. <<https://docs.anaconda.com/anaconda/user-guide/getting-started/>>
 35. **ezDICOM DICOM viewer**. "ezDICOM for users". [Guía de usuario en línea]. [Accedido el 5 de octubre de 2021]. <<https://people.cas.sc.edu/rorden/ezdicom/index.html>>
 36. **Rivera Mariano (2019)**. "Reeuso de Redes Preentrenadas". [Tutorial en línea]. [Accedido el 22 de noviembre de 2021]. <http://personal.cimat.mx:8181/~mrivera/cursos/aprendizaje_profundo/preentrenadas/preentrenadas.html>
 37. **PythonAnywhere (2021)**. "Setting up Flask applications on PythonAnywhere". [Documentación en línea]. [Accedido el 11 de diciembre de 2021]. <<https://help.pythonanywhere.com/pages/Flask/>>
 38. **Heroku Dev Center (2021)**. "Documentation". [Documentación en línea]. [Accedido el 15 de diciembre de 2021]. <<https://devcenter.heroku.com/categories/reference>>