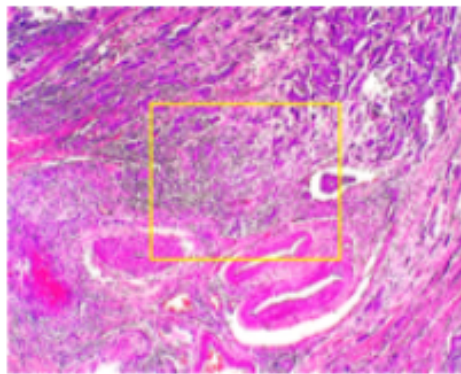
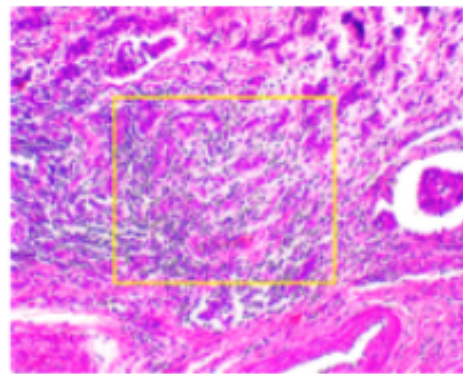


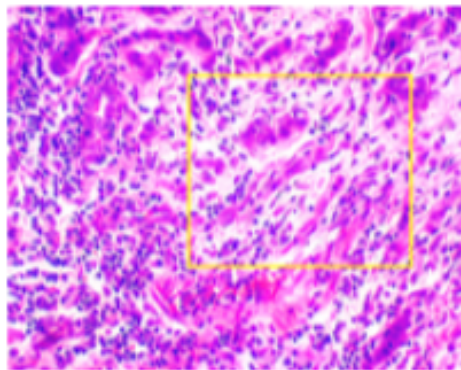
# Detección de cáncer de mama a través de redes convolucionales



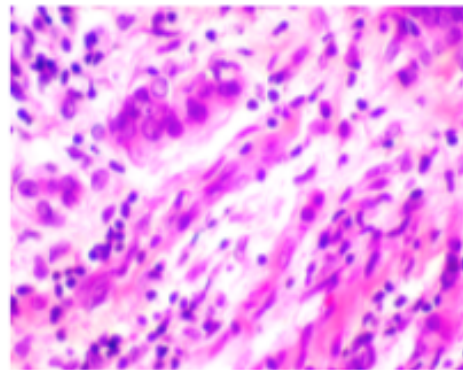
(a)



(b)



(c)



(d)

REALIZADO POR:

Marina Calero López  
Lucas Manuel Herencia Solís  
Juan Antonio Moreno Moguel

May 6, 2025

# Resumen

Este proyecto se enmarca en la asignatura de Procesamiento de Imágenes Digitales (PID) y tiene como objetivo desarrollar un sistema que detecte el cáncer de mama mediante la identificación y comparación de células cancerígenas con células sanas. La metodología se basará en el uso de redes neuronales convolucionales (CNN) para analizar imágenes digitales de tejido mamario y extraer patrones característicos. Se realizan procesos de preprocesamiento y clasificación según el modelo descrito del documento científico elegido, el cuál será entrenado dependiendo la amplitud respecto al ojo humano de la que se habrá realizado la imagen (x40, x100, x200 y x400), con el cuál se permitirá la clasificación de células malignas y benignas. Con este enfoque, se busca crear una herramienta de apoyo al diagnóstico clínico, que contribuya a una detección temprana y más precisa de la enfermedad.

**Palabras clave:** cáncer de mama, redes neuronales convolucionales (CNN), células, benigno, maligno.

# Índice

<b>1. Introducción</b>	<b>3</b>
<b>2. Planteamiento del Teórico</b>	<b>5</b>
2.1. Objetivos del Proyecto . . . . .	5
2.2. Tecnologías Utilizadas . . . . .	5
2.2.1. Enfoque 1: Redes Neuronales Convolucionales (CNN) . . . . .	6
2.2.2. Enfoque 2: Autoencoder y KNN sobre la Capa Latente . . . . .	6
<b>3. Implementación</b>	<b>8</b>
3.1. Descripción del dataset . . . . .	8
3.2. Arquitectura de la aplicación . . . . .	8
3.2.1. Arquitectura CNN . . . . .	9
3.2.2. Arquitectura KNN . . . . .	10
3.3. Evaluación del Sistema . . . . .	12
<b>4. Experimentación</b>	<b>13</b>
<b>5. Manual de usuario</b>	<b>13</b>
<b>6. Conclusiones</b>	<b>14</b>
<b>7. Autoevaluación de cada miembro del equipo</b>	<b>14</b>
<b>8. Tabla de tiempos</b>	<b>14</b>
<b>Bibliografía</b>	<b>14</b>

# 1. Introducción

El cáncer de mama es una de las principales causas de mortalidad en el mundo, habiendo sido la responsable de 670.000 muertes en 2022 siendo a su vez el tipo de cáncer más común en las mujeres según [1]. Esto provoca que durante los últimos años se haya estado realizando una labor social enorme referente a la concienciación sobre el cáncer de mama dando gran importancia a su pronta detección.

Gracias a los avances en Deep Learning [2], se han desarrollado métodos innovadores para analizar imágenes histológicas y detectar el cáncer de mama. En este contexto, existen dos aproximaciones que se han estudiado en la literatura, como en el artículo ***“Classification of breast cancer based on histology images using convolutional neural networks”*** [3], soportada por los 445 artículos en los que se realiza un estudio entre dos enfoques.

Este proyecto tiene como objetivo el desarrollo de una aplicación capaz de detectar cáncer de mama a partir de imágenes histológicas, utilizando dos enfoques complementarios. El *primero* se basa en una red neuronal convolucional (CNN) que realiza clasificación directamente a la imagen, y un *segundo enfoque*, basado en extracción de características seguida de la clasificación mediante el algoritmo K-Nearest Neighbors (KNN).

El *primer enfoque* se fundamenta en redes neuronales convolucionales, que han demostrado gran efectividad en la clasificación automática de imágenes biomédicas. Específicamente, el trabajo de [4] implementa un autoencoder basado en CNN para la recuperación de imágenes de cáncer de mama, permitiendo identificar imágenes similares dentro de grandes bases de datos. Esta capacidad de aprendizaje profundo permite al modelo extraer representaciones jerárquicas y semánticas directamente desde los datos crudos, sin necesidad de ingeniería manual de características.

Este enfoque aprovecha la potencia de las Redes Neuronales Convolucionales (CNN), un tipo de red neuronal feedforward inspirada en la percepción visual [5]. Las CNN aplican filtros o “kernels” a lo largo de las imágenes para extraer automáticamente patrones y características como bordes, texturas y formas. Esta capacidad de aprender de forma automática elimina la necesidad de una extracción manual de atributos y, al reducir significativamente el número de conexiones en comparación con una red completamente conectada (FC), permite una convergencia más rápida y una actualización de pesos más eficiente durante el entrenamiento.

Por otro lado, el *segundo enfoque* está inspirado en el trabajo de [3], del cuál se ha utilizado parte de su propuesta dedicada a la recuperación de imágenes de células histográficas de cáncer de mama. En concreto, se ha extraído la parte del encoder para poder realizar una adecuada extracción de características de las imágenes, con las cuáles se lleva a cabo posteriormente un análisis mediante KNN utilizando dichas características extraídas.

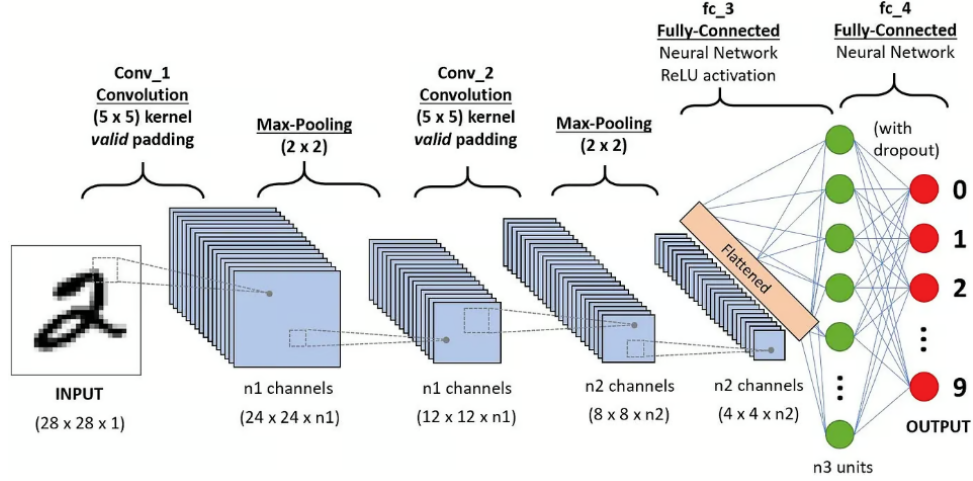


Figura 1: Capas convolucionales [6]

Ambos enfoques son implementados y evaluados en este proyecto, comparando su rendimiento y su viabilidad para una futura aplicación en el diagnóstico asistido por computadora.

El objetivo de este proyecto es desarrollar un sistema que permita detectar el cáncer de mama a partir de imágenes histológicas, utilizando técnicas de aprendizaje automático y redes neuronales convolucionales. Se espera que este sistema contribuya a la mejora en la detección temprana del cáncer de mama, facilitando el trabajo de los profesionales médicos y aumentando la tasa de supervivencia de las pacientes.

El objetivo de nuestro trabajo es desarrollar una red convolucional capaz de detectar el cáncer de mama a partir de imágenes histológicas obtenidas de un dataset público [7] y comparar distintas aproximaciones para evaluar la eficacia en la detección temprana de esta enfermedad.

## 2. Planteamiento del Teórico

### 2.1. Objetivos del Proyecto

Este trabajo tiene como objetivo principal analizar y comparar diferentes metodologías de clasificación de imágenes médicas, con especial atención en la detección temprana del cáncer de mama. Para ello, se estudian y contrastan dos enfoques representativos: uno híbrido basado en un autoencoder y el algoritmo k-Nearest Neighbors (KNN), y otro basado en redes neuronales convolucionales (CNN).

El **primer enfoque** emplea un autoencoder como técnica de reducción de dimensionalidad y extracción automática de características. En este esquema, la parte *encoder* del autoencoder extrae una representación comprimida de las imágenes, capturando patrones relevantes de manera no supervisada. Estas representaciones latentes son luego utilizadas como entrada para el clasificador KNN, que asigna una clase a cada imagen en función de la similitud con las representaciones más cercanas en el espacio latente. Este enfoque combina las fortalezas de la codificación no supervisada con la simplicidad del clasificador KNN.

El **segundo enfoque** se basa en redes neuronales convolucionales (CNN), las cuales aprenden directamente tanto las características como la función de clasificación. Las CNN están especialmente diseñadas para trabajar con datos visuales, y mediante el uso de capas convolucionales, capturan patrones espaciales jerárquicos presentes en las imágenes. Este CNN en concreto será creado a través de las capas descritas en el documento escogido y entrenado posteriormente en 4 diferentes casos dependiendo de la amplitud respecto al ojo humano de la imagen. Gracias a esta capacidad, han demostrado un alto rendimiento en tareas de diagnóstico médico, incluyendo la detección temprana del cáncer de mama.

Los objetivos específicos del proyecto son los siguientes:

1. **Crear** una aplicación que permita a los médicos detectar el cáncer de mama solo dando la imagen.
2. **Desarrollar** distintos modelos que permitan la clasificación binaria de las imágenes.
3. **Evaluar** el rendimiento de distintos modelos, tanto CNN en tareas de clasificación binaria como KNN.
4. **Realizar experimentación para análisis** del impacto de las técnicas de aumento de datos sobre el desempeño de los modelos.

### 2.2. Tecnologías Utilizadas

Para el desarrollo del proyecto se ha utilizado el lenguaje de programación **Python**, ampliamente adoptado en la comunidad científica por su simplicidad, legibilidad y amplio ecosistema de bibliotecas para el análisis de datos, visión por computador y aprendizaje automático.

El desarrollo de la interfaz de nuestra aplicación ha sido desarrollado utilizando **Tkinter**, permitiendo de esta forma una selección y visualización de resultados cómoda de las imágenes junto a su clasificación en benigna o maligna.

Entre estas bibliotecas, se ha seleccionado **TensorFlow** como herramienta principal para la implementación y comparación de modelos. TensorFlow es una librería de código abierto desarrollada por Google que permite construir, entrenar y desplegar modelos de *machine learning*, especialmente redes neuronales profundas. Su flexibilidad y eficiencia lo convierten en una plataforma ideal tanto para métodos tradicionales como para arquitecturas más complejas basadas en aprendizaje profundo.

### 2.2.1. Enfoque 1: Redes Neuronales Convolucionales (CNN)

Una **Red Neuronal Convolutiva** (CNN, por sus siglas en inglés) es un tipo de arquitectura de red neuronal especializada en procesar datos con una estructura de tipo cuadrícula, como las imágenes. Su principal fortaleza radica en su capacidad para aprender representaciones espaciales jerárquicas mediante capas convolucionales.

#### Estructura típica de una CNN:

1. **Capa Convolutiva (Convolutional Layer):** Esta capa aplica filtros (o *kernels*) sobre la imagen de entrada para extraer características locales, como bordes, texturas o patrones específicos. El proceso de convolución genera mapas de activación que representan la presencia de ciertas características.
2. **Función de Activación (ReLU):** Tras cada convolución, se aplica normalmente una función de activación como ReLU (Rectified Linear Unit) para introducir no linealidades, lo que permite a la red aprender funciones más complejas.
3. **Capa de Agrupamiento (Pooling Layer):** Reduce la dimensionalidad de los mapas de activación, conservando las características más relevantes. El tipo más común es *max pooling*, que selecciona el valor máximo en regiones locales.
4. **Capas Completamente Conectadas (Fully Connected Layers):** Al final de la red, estas capas interpretan las características extraídas para realizar la predicción final, como clasificar si una imagen muestra tejido sano o indicios de cáncer.
5. **Capa de Salida:** Generalmente una capa *softmax* (para clasificación multiclase) o una sigmoide (para clasificación binaria), que devuelve la probabilidad de pertenencia a cada clase.

Las CNN se entrenan mediante retropropagación (backpropagation) y optimización mediante algoritmos como Adam o SGD, ajustando los pesos para minimizar una función de pérdida, habitualmente la entropía cruzada (*binary cross-entropy* en problemas binarios).

### 2.2.2. Enfoque 2: Autoencoder y KNN sobre la Capa Latente

Un **autoencoder** es una red neuronal no supervisada diseñada para aprender una representación comprimida (latente) de los datos, útil para tareas como reducción de dimensionalidad, eliminación de ruido o extracción de características.

#### Estructura de un autoencoder:

1. **Codificador (Encoder):** Esta parte reduce progresivamente la dimensión de la entrada, transformándola en una representación más compacta. Se compone de varias capas densas o convolucionales, según el tipo de dato.
2. **Capa Latente (Bottleneck):** Es la representación más comprimida de la entrada. Idealmente, contiene la información esencial del patrón original y descarta el ruido o redundancias. Esta es la capa que se utilizará para aplicar métodos de clasificación tradicionales, como KNN.
3. **Decodificador (Decoder):** Reconstruye la entrada original a partir de la capa latente, tratando de minimizar la diferencia entre entrada y salida (p. ej., usando el error cuadrático medio como función de pérdida).

Una vez entrenado el autoencoder, se prescinde del decodificador y se conserva únicamente el **codificador**, que transforma las imágenes en vectores de baja dimensión. Sobre estos vectores comprimidos se aplica el algoritmo **K-Nearest Neighbors (KNN)**, que clasifica nuevas imágenes comparándolas con las más cercanas en el espacio latente. Esta estrategia permite combinar lo mejor de ambos mundos: una representación no lineal potente (extraída por el autoencoder) y un clasificador sencillo, interpretable y eficaz como KNN.

En concreto, en este proyecto, el modelo CNN ya se encontraba predefinido en el artículo [3], por lo que solo ha sido necesario construir la red neuronal la cual ha sido entrenada con el dataset **BreakHis**, en concreto se han generado 5 modelos, uno entrenado con el dataset completo y uno generado por cada magnificación disponible del dataset (40X, 100X, 200X, 400X), por otra parte, el modelo KNN, se divide en dos partes, la extracción de características y la clasificación, la extracción de características se ha obtenido a partir del autoencoder descrito en el artículo [4] y el clasificador KNN realizado manualmente.



### 3. Implementación

El desarrollo de este proyecto se fundamenta en el uso de técnicas de aprendizaje automático y profundo para la clasificación de imágenes histológicas de cáncer de mama. Con el objetivo de comparar el rendimiento entre distintos enfoques, se han implementado y entrenado dos tipos de modelos: uno basado en un clasificador K-Nearest Neighbors (KNN) acompañado de un proceso de extracción de características, y otro basado en una red neuronal convolucional (CNN). Ambos modelos han sido entrenados utilizando un conjunto de imágenes histológicas públicamente disponible, asociado al trabajo [1]. Este dataset incluye imágenes con diferentes niveles de magnificación (40x, 100x, 200x y 400x), aunque el artículo original no proporciona demasiados detalles sobre la obtención de las imágenes ni sobre las características clínicas asociadas.

#### 3.1. Descripción del dataset

Para el desarrollo de este proyecto se utilizó el **dataset de imágenes histológicas de cáncer de mama BreaKHis** [8], asociado al trabajo de [3]. Este conjunto de datos es ampliamente utilizado en la literatura científica para la investigación en diagnóstico asistido por computadora en cáncer de mama. BreaKHis contiene **7909 imágenes histológicas** de biopsias mamarias, distribuidas en dos clases principales: **benignas** y **malignas**, en concreto, **2480 imágenes benignas**, subdivididas en cuatro subtipos: adenosis, fibroadenoma, tumor filoides y tejido conectivo y **5429 imágenes malignas**, subdivididas en carcinoma ductal, carcinoma lobulillar, carcinoma mucinoso y carcinoma papilar. Las imágenes fueron adquiridas a partir de muestras teñidas con hematoxilina y eosina (H&E) y capturadas con un microscopio óptico en diferentes niveles de aumento: **40x, 100x, 200x y 400x**.

Las imágenes tienen una resolución de  $700 \times 460$  píxeles en formato PNG y están organizadas por paciente, clase y magnificación. Una característica importante del dataset es que mantiene la trazabilidad del origen de las imágenes (es decir, a qué paciente pertenecen), lo que permite realizar particiones de entrenamiento y prueba sin mezclar muestras del mismo paciente entre ambos conjuntos, reduciendo así el riesgo de sobreajuste.

#### 3.2. Arquitectura de la aplicación

La aplicación desarrollada para el análisis de cáncer de mama mediante modelos KNN y CNN ofrece una interfaz gráfica intuitiva basada en Tkinter, que permite al usuario interactuar fácilmente con el sistema. Una de sus funcionalidades principales es la carga de imágenes, donde el usuario puede seleccionar desde su dispositivo una imagen médica, la cual se muestra en una vista previa dentro de la interfaz para su verificación.

Además, la aplicación permite la selección de distintos modelos preentrenados, tanto de tipo KNN (como “KNN(all)” o “KNN(aux)”) como de redes neuronales convolucionales (CNN), con distintas configuraciones que ofrecen flexibilidad en el análisis. Esta selección de modelo es esencial para personalizar el tipo de procesamiento que se aplicará sobre la imagen cargada.

Una vez seleccionados la imagen y el modelo, el usuario puede iniciar el análisis pulsando un botón. El sistema entonces ejecuta el modelo correspondiente, determinando si el caso es benigno o maligno. El resultado se muestra de forma clara mediante un mensaje textual, un icono visual asociado al diagnóstico, y un conjunto de estadísticas adicionales que aportan más contexto al resultado obtenido.

La aplicación también gestiona de forma inteligente el estado de la interfaz: desactiva el botón de análisis si no se han cumplido los requisitos previos (imagen y modelo seleccionados), y limpia automáticamente los resultados anteriores cuando se cargan nuevos datos. Gracias a su diseño estructurado y al uso de componentes *ttk*, la experiencia del usuario es fluida y profesional, facilitando la interpretación del diagnóstico por parte de usuarios tanto técnicos como clínicos.

### 3.2.1. Arquitectura CNN

En el ámbito del aprendizaje automático, las redes neuronales convolucionales (CNN) se han convertido en una tendencia dominante, alcanzando un notable éxito en diversas áreas como la visión por computador y el reconocimiento de voz. En este contexto, el artículo [3] propone una arquitectura CNN específicamente diseñada para la clasificación de imágenes histológicas de cáncer de mama teñidas con hematoxilina y eosina (H&E), diferenciando entre tejidos benignos y malignos. La arquitectura fue implementada utilizando el popular framework BVLC Caffe [9]. Este entorno permite diseñar e implementar redes CNN de forma sencilla, con interfaces disponibles tanto para *MATLAB* como para Python, y soporte para entrenamiento en CPU y GPU. El modelo CNN propuesto se especifica mediante archivos de configuración que definen la estructura de la red y los parámetros de optimización.

La arquitectura propuesta consta de *5 capas convolucionales* seguidas por dos capas completamente conectadas y una capa softmax de salida. En detalle, la *primera* capa convolucional utiliza filtros de tamaño  $3 \times 3$  y genera 64 mapas de características; la *segunda* también emplea filtros de  $3 \times 3$  pero produce 96 mapas; la *tercera* capa, con filtros del mismo tamaño, genera 128 mapas de características; la *cuarta* capa incrementa esta cifra a 256 mapas, al igual que la *quinta* capa. Posteriormente, se añaden dos capas completamente conectadas, la primera con 2000 unidades ocultas y la segunda con tantas unidades como clases tenga el conjunto de datos. Finalmente, una capa softmax se encarga de la clasificación. Todas las capas convolucionales y completamente conectadas aplican la función de activación ReLU (**Rectified Linear Unit**), definida como:

$$f(x) = \max(0, x) \quad (1)$$

lo que introduce no linealidades y acelera la convergencia del entrenamiento.

Además, se utiliza max pooling tras las capas convolucionales primera, segunda y quinta. Esta técnica de reducción de dimensión espacial se aplica después de ReLU con un filtro de tamaño  $3 \times 3$  y un desplazamiento (**stride**) de 2. Las capas tercera y cuarta no incorporan pooling, probablemente para preservar la resolución espacial de las características extraídas.

El peso de las capas se inicializa mediante una distribución gaussiana con desviación estándar baja (0,01), mientras que la regularización se aborda con varias estrategias. Por un lado, se incluye una capa de *dropout* después de la primera capa completamente conectada, con una probabilidad de retención (*keep probability*) de  $\rho = 0,5$ , lo cual ayuda a mitigar el sobreajuste. Por otro lado, se aplica una penalización de tipo  $L_2$  (*weight decay*) con un valor  $\lambda = 10^{-3}$  para restringir los valores de los pesos durante el entrenamiento.

El modelo se entrena mediante el algoritmo de descenso de gradiente estocástico (SGD), con un tamaño de lote (*batch size*) igual a 32. La tasa de aprendizaje inicial se establece en 0,001 y se ajusta dinámicamente durante el entrenamiento utilizando una política de decaimiento inverso cada cinco épocas. Este parámetro controla la magnitud de los pasos que da la red en cada iteración para minimizar la función de pérdida. Además, se incorpora un factor de momento (*momentum*) de 0,9, que favorece una convergencia más estable y ayuda a evitar quedarse atrapado en mínimos locales. Finalmente, los datos de entrada se barajan aleatoriamente antes de cada época para prevenir sesgos asociados a un orden fijo en el conjunto de entrenamiento.

En conjunto, esta arquitectura CNN logra una extracción jerárquica de características efectivas, desde patrones locales hasta representaciones globales, optimizando así el rendimiento en la clasificación de imágenes histológicas. Los resultados reportados demuestran que esta red supera a los métodos clásicos basados en características manuales, lo que refuerza la utilidad de las CNN en aplicaciones de diagnóstico asistido por computadora en el ámbito de la patología digital.

### 3.2.2. Arquitectura KNN

El artículo [4], propone una metodología innovadora para la extracción automática de características en imágenes histológicas de cáncer de mama, basada en autoencoders convolucionales (CNN-AE). Este enfoque permite representar imágenes complejas en forma de vectores latentes que conservan la información más relevante para tareas posteriores como clasificación o recuperación por similitud.

**Arquitectura del Autoencoder:** El autoencoder se entrena mediante un proceso iterativo en el que se minimiza la diferencia entre la imagen original y su reconstrucción (error de reconstrucción). Este error se calcula usando una función de pérdida (como el error cuadrático medio), y se optimiza con métodos como el descenso de gradiente estocástico (SGD) o el optimizador Adam. Durante el entrenamiento, se emplea un conjunto de datos dividido en subconjuntos de **entrenamiento** y **validación**, lo cual permite evaluar el rendimiento general del modelo y evitar el sobreajuste.

De forma específica, el proceso incluye tres pasos fundamentales:

1. **Codificación (Encoder):** compresión de la imagen en una representación latente compacta.
2. **Decodificación (Decoder):** reconstrucción de la imagen a partir del vector latente.

3. **Cálculo de la función de pérdida y optimización:** ajuste iterativo de los pesos para minimizar el error de reconstrucción.

Además, se implementa una política de **almacenamiento del mejor modelo**, en la cual se guarda la versión del autoencoder que presenta el **menor valor de pérdida** en el conjunto de validación. Este modelo óptimo se utilizará posteriormente **para la extracción de características y recuperación de imágenes**.

#### **Extracción de Características para Clasificación:**

Una vez entrenado, el **codificador** se desacopla del decodificador y se utiliza de manera independiente para transformar imágenes en vectores latentes. Esta operación consiste en alimentar una imagen al codificador y capturar la activación de su capa más profunda, justo antes de la entrada al decodificador. El vector de salida obtenido en esta etapa representa una **caracterización compacta** de la imagen de entrada y se conforma de 48 valores, preservando información estructural crucial para la identificación de patrones asociados al cáncer de mama. Estos vectores de características se almacenan en una base de datos y se emplean como entrada para clasificadores como k-NN.

#### **Clasificación mediante k-NN utilizando Vectores Latentes:**

Tras el proceso de extracción de características mediante el autoencoder convolucional, cada imagen del conjunto de datos fue transformada en un vector latente de dimensión reducida, representando las características más relevantes de la imagen histológica. Estos vectores latentes, junto con su etiqueta de clase correspondiente (benigno o maligno), fueron almacenados en archivos **JSON**, actuando como base de datos para el sistema de clasificación.

Para realizar la clasificación de una nueva imagen, se sigue el siguiente procedimiento:

1. **Extracción del vector latente:** La imagen de entrada se procesa a través del codificador previamente entrenado, obteniendo su correspondiente vector de características.
2. **Cálculo de distancias:** Se calcula la **distancia euclídea** entre el vector extraído de la imagen a predecir y todos los vectores almacenados en el archivo JSON. La distancia euclídea entre dos vectores  $x$  e  $y$  se define como:

$$d(x, y) = \sqrt{\sum_{i=1} (x_i - y_i)^2} \quad (2)$$

3. **Selección de los vecinos más cercanos:** Una vez calculadas todas las distancias, se seleccionan los **k vectores más cercanos**, siendo  $k=5$  en esta implementación. Estos representan las imágenes más similares en el espacio latente.
4. **Clasificación por mayoría:** Se toma la clase más común entre los cinco vectores más próximos, y esta se asigna como la **predicción final** para la imagen de entrada.

Este enfoque no requiere un entrenamiento explícito del clasificador, ya que el método k-NN funciona de forma **perezosa** (lazy learning), basándose directamente en las instancias almacenadas y las comparaciones entre vectores. Gracias a la alta calidad de las características extraídas por el autoencoder, el sistema logra clasificaciones precisas sin necesidad de arquitecturas más complejas.

### 3.3. Evaluación del Sistema

Los modelos fueron evaluados mediante un conjunto de métricas de clasificación binaria, entendiendo que los benignos serán los casos negativos y los malignos los casos positivos:

1. **Precisión (accuracy):** mide la proporción de predicciones correctas respecto al total de muestras evaluadas. Se calcula como:

$$\text{accuracy} = \frac{VP + VN}{VP + VN + FP + FN} \quad (3)$$

2. **Sensibilidad (recall):** evalúa la capacidad del modelo para identificar correctamente las muestras positivas. Se calcula como:

$$\text{recall} = \frac{VP}{VP + FN} \quad (4)$$

3. **Precisión (precision):** cuantifica la proporción de muestras clasificadas como positivas que realmente lo son. Se calcula como:

$$\text{precision} = \frac{VP}{VP + FP} \quad (5)$$

4. **F1 Score:** proporciona una medida balanceada entre la precisión y la sensibilidad, siendo especialmente útil en contextos donde existe un desequilibrio entre clases. Se calcula como:

$$F1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (6)$$

Estas métricas fueron calculadas de forma individual para cada uno de los modelos entrenados, lo que permitió realizar un análisis cuantitativo del impacto que tiene el tipo de arquitectura como *K-Nearest Neighbors* (KNN) y *Convolutional Neural Networks* (CNN) sobre el rendimiento en la tarea de clasificación.

Adicionalmente, se evaluó la influencia de distintos niveles de *data augmentation* aplicados a las imágenes de entrada, tanto para KNN como con CNN con el fin de observar cómo afectan la capacidad de generalización de los modelos.

Como parte del trabajo, se llevó a cabo una experimentación adicional orientada a evaluar el impacto de una integración más profunda entre los enfoques basados en KNN y CNN. En esta etapa, se creó un sistema de almacenamiento que registró todas las representaciones extraídas de las imágenes mediante el modelo KNN, sin restricciones en cuanto a la dimensión o amplitud de dichas características. Paralelamente, se entrenó un modelo CNN utilizando la totalidad del conjunto de imágenes disponibles, y se extrajeron los pesos resultantes tras el entrenamiento completo del modelo. Esta experimentación se realizó con el fin de observar qué resultados podían obtenerse al disponer de ambas fuentes de representación, y explorar su posible utilidad en tareas de clasificación médica.

## 4. Experimentación

Para llevar a cabo la experimentación, se realizaron pruebas sobre un conjunto de datos al que se le aplicará *data augmentation* con el objetivo de evaluar la robustez y capacidad de generalización de los modelos. Las técnicas de aumento de datos incluirán transformaciones como rotaciones de un rango entre  $-30^\circ$  a  $30^\circ$  y modificaciones en la intensidad de las imágenes, con un factor de entre 0,7 (oscurece) a 1,3 (ilumina), simulando variaciones realistas que podrían encontrarse en escenarios del mundo real. Esto permitirá analizar cómo afecta la variabilidad de entrada al rendimiento de los modelos evaluados.

En esta etapa, se llevará a cabo una comparación de rendimiento entre dos enfoques de clasificación: una red neuronal convolucional (*CNN*) entrenada sobre cada variedad de las imágenes aumentadas y un entrenamiento con todas las imágenes en global, y un modelo *K-Nearest Neighbors* (*KNN*) diseñado de forma personalizada para cada uno de las posibles variaciones de imágenes respecto a su aumento. Esta comparación permitirá observar las diferencias entre un enfoque basado en aprendizaje profundo y otro basado en métodos clásicos de aprendizaje supervisado, en cuanto a métricas de *precisión*, *recall* y *F1-score*.

## 5. Manual de usuario

Esta herramienta permite cargar una imagen desde el dispositivo y realizar un análisis para determinar si corresponde a un caso **benigno** o **maligno**. Los pasos a seguir son los siguientes:

### Paso 1: Seleccionar una Imagen

#### Opción 1: Cargar por Ruta

- Haz clic en el botón **Seleccionar Imágenes**.
- Navega por tus carpetas y selecciona las imágenes que desees analizar.
- Las imágenes seleccionadas aparecerán en el panel de previsualización.

#### Opción 2: Arrastrar y Soltar

- Puedes arrastrar la imagen directamente desde tu explorador de archivos y soltarla en el área indicada de la aplicación.
- Las imágenes se cargarán automáticamente en la aplicación.

### Paso 2: Iniciar el Análisis

- Una vez que hayas cargado la imagen correctamente, haz clic en el botón **Analizar**.
- La aplicación procesará la imagen y mostrará si corresponde a un caso maligno o benigno.

### Paso 3: Ver el Resultado

- Después de un breve momento, se mostrará el resultado en pantalla.
- Si la imagen es **benigna**, se mostrará un mensaje y un círculo indicador en color **verde**.
- Si la imagen es **maligna**, se mostrará un mensaje y un círculo indicador en color **rojo**.

### Paso 4: Realizar Nuevos Análisis

- Puedes cargar otras imágenes haciendo clic nuevamente en el botón **Seleccionar** o arrastrándolas directamente.
- El proceso de análisis es el mismo para cada imagen que cargues.

## 6. Conclusiones

Aquí van las conclusiones.

## 7. Autoevaluación de cada miembro del equipo

Criterio	Lucas Manuel Herencia Solís	Marina Calero López	Juan Antonio Moreno Moguel
Comprensión y dominio	Excelente		
Exposición didáctica	Excelente		
Integración del equipo	Excelente		
Objetivos	Excelente		
Aspectos didácticos	Excelente		
Experimentación y conclusiones	Excelente		
Contenidos	Excelente		
Divulgación de contenidos	Excelente		
Bibliografía / Recursos científicos	Excelente		

## 8. Tabla de tiempos

Aquí van las conclusiones.

## Referencias

- [1] World Health Organization, «Breast Cancer,» 2025, Accessed: 2025-04-30.
- [2] P. P. Shinde y S. Shah, «A review of machine learning and deep learning applications,» págs. 1-6, 2018.

- [3] D. Bardou, K. Zhang y S. M. Ahmad, «Classification of breast cancer based on histology images using convolutional neural networks,» *Ieee Access*, vol. 6, págs. 24 680-24 693, 2018.
- [4] A. E. Minarno, K. M. Ghufro, T. S. Sabrila, L. Husniah y F. D. S. Sumadi, «Cnn based autoencoder application in breast cancer image retrieval,» págs. 29-34, 2021.
- [5] D. H. Hubel y T. N. Wiesel, «Receptive fields, binocular interaction and functional architecture in the cat's visual cortex,» *The Journal of physiology*, vol. 160, n.º 1, pág. 106, 1962.
- [6] DataCamp, «Capas convoluciones: CNN con TensorFlow en Python,» 2025, Accessed: 2025-04-30.
- [7] M. Nasser y U. K. Yusof, «Deep learning based methods for breast cancer diagnosis: a systematic review and future direction,» *Diagnostics*, vol. 13, n.º 1, pág. 161, 2023.
- [8] Google Drive, «Carpeta compartida de recursos,» 2025, Accessed: 2025-04-30.
- [9] Y. Jia et al., «Caffe: Convolutional architecture for fast feature embedding,» págs. 675-678, 2014.