

PROYECTO FIN DE GRADO

TÍTULO: Inteligencia Artificial aplicada al Análisis de Imágenes

AUTOR/A: Victoria Muñoz Murillo

TITULACIÓN: Grado en Ingeniería de Sonido e Imagen

DIRECTOR/A: María Salomé Merino Menéndez

TUTOR/A: José Manuel Díaz López

DEPARTAMENTO: Ingeniería Audiovisual y Comunicaciones

VºBº TUTOR/A

Miembros del Tribunal Calificador:

PRESIDENTE/A: Luis Narvarte Fernández

TUTOR/A: José Manuel Díaz López

SECRETARIO/A: José Luis Rodríguez Vázquez

Fecha de lectura: Septiembre 2023

Calificación:

El Secretario/La Secretaria,

AGRADECIMIENTOS

Quiero agradecer a mi familia el gran apoyo que me han brindado. Muchísimas gracias a mi madre y a mi madrina por estar siempre ahí, por su infinita paciencia y por confiar siempre en mí, a mi novio por todo su apoyo y ayuda en los momentos más duros.

También agradecer a mi tutor, José Manuel Díaz López, por guiarme y ayudarme con el PFG, también agradecer a César Díaz Martín por ayudarme a obtener los recursos y conocimientos básicos necesarios para llevar a cabo este proyecto. A su vez, agradecer a la doctora María Salomé Merino y el doctor Juan Arrazola del Hospital Clínico San Carlos por darme la oportunidad y los medios necesarios para poder realizar este proyecto.

Muchas gracias a mis amigos por todos los buenos momentos y por amenizar estos intensos 4 años.

Muchas gracias a todos por vuestro apoyo

ÍNDICE DE CONTENIDO

1. INTRODUCCIÓN -----	PÁG. 13
2. MARCO TECNOLÓGICO -----	PÁG. 15
2.1. IA de generación de imágenes -----	pág. 15
2.1. Redes Neuronales Convolucionales -----	pág. 18
2.3. Marco teórico médico -----	pág. 24
3. ESTADO DEL ARTE -----	PÁG. 29
3.1. IA de generación de imágenes -----	pág. 29
3.2. IA (CNN) en clasificación de imágenes médicas -----	pág. 32
4. METODOLOGÍA -----	PÁG. 35
4.1. Recopilación de datos -----	pág. 35
4.2. Preprocesamiento de datos -----	pág. 38
4.3. Arquitecturas de CNN utilizadas -----	pág. 40
5. IMPLEMENTACIÓN -----	PÁG. 45
5.1. Software y hardware empleado -----	pág. 45
5.2. Hiperparámetros -----	pág. 46
6. PRUEBAS Y RESULTADOS TÉCNICAS DE PROMPTING -----	PÁG. 49
7. ENTRENAMIENTO Y RESULTADOS MODELOS DE CNN -----	PÁG. 61
8. PRESUPUESTO -----	PÁG. 69
9. IMPACTO DEL PROYECTO -----	PÁG. 69
10. CONCLUSIONES -----	PÁG. 71
11. TRABAJO FUTURO -----	PÁG. 73
12. BIBLIOGRAFÍA -----	PÁG. 75
13. ANEXO A -----	PÁG. 79

ÍNDICE DE FIGURAS

Figura 1: Diagrama de bloques general de una GAN -----	pág.15
Figura 2: Representación esquemática de una red neuronal de 4 capas -----	pág.18
Figura 3: Representación de los parámetros ajustables en relación a las neuronas de la red -----	pág.19
Figura 4: Representación de convolución con un kernel de 3x3 -----	pág.21
Figura 5: Filtro Sobel para obtener bordes horizontales (izq) y verticales (dcha) -----	pág.21
Figura 6: Ejemplo filtro de media 3x3 -----	pág.22
Figura 7: Ejemplo aplicación kernel 3x3 a imagen RGB -----	pág.22
Figura 8: Ejemplo aplicación de máx pooling con kernel 2x2 -----	pág.23
Figura 9: Ramos del nervio facial y glándula parótida -----	pág.24
Figura 10: Sección de la glándula parótida -----	pág.25
Figura 11: TC axial con tumor en la glándula parótida -----	pág.26
Figura 12: Toma de ecografía en la glándula parótida -----	pág.26
Figura 13: Comparación en resultados de imágenes generales con el nivel de detalle de los prompts -----	pág.30
Figura 14: Representación de proceso de colonoscopia sin IA (izq) vs con IA (dcha) -----	pág.34
Figura 15: Ecógrafo Toshiba Aplio 500 -----	pág.36
Figura 16: Imágenes de ecografías de la glándula parótida con marcas superpuestas -----	pág.36
Figura 17: Imágenes de ecografías Doppler de la glándula parótida -----	pág.37
Figura 18: Representación de overfitting en curva de pérdida -----	pág.39
Figura 19: Bloque de aprendizaje residual [3] -----	pág.40
Figura 20: Ejemplos de arquitecturas para ImageNet [3] -----	pág.41
Figura 21: Bloque Inception-A perteneciente a la red Inception-v4 [37] --	pág.42
Figura 22: Esquema de módulo Inception-ResNet-C de la red Inception-ResNet-v2 [37] -----	pág.43
Figura 23: Reemplazado de aplicación de convolución tradicional (a) por convolución depthwise (b) seguido de convolución pointwise (c) [54] -----	pág.44
Figura 24: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 1, Tema 1 -----	pág.49
Figura 25: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 2, Tema 1 -----	pág.50
Figura 26: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 3, Tema 1 -----	pág.50
Figura 27: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 4, Tema 1 -----	pág.51
Figura 28: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 5, Tema 1 -----	pág.51
Figura 29: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 6, Tema 1 -----	pág.52
Figura 30: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 7, Tema 1 -----	pág.53
Figura 31: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 8, Tema 2 -----	pág.54
Figura 32: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 8 + negative prompts, Tema 2 -----	pág.54

Figura 33: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 9, Tema 2 -----	pág.55
Figura 34: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 10, Tema 3 -----	pág.56
Figura 35: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 11, Tema 3 -----	pág.57
Figura 36: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 12, Tema 3 -----	pág.58
Figura 37: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 13, Tema 4 -----	pág.58
Figura 38: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 14, Tema 4 -----	pág.59
Figura 39: Prueba 1 ResNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas) -----	pág.62
Figura 40: Prueba 2 ResNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)-----	pág.63
Figura 41: Prueba 1 Inception-ResNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)	pág.64
Figura 42: Prueba 2 Inception-ResNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)	pág.65
Figura 43: Prueba 1 MobileNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)	pág.67
Figura 44: Prueba 2 MobileNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)	pág.68

RESUMEN

Este proyecto tiene como finalidad abordar la temática de Inteligencia Artificial (IA) en los ámbitos de análisis, clasificación y generación de imágenes. Para ello, en primer lugar se ha llevado a cabo un estudio y comparación de las principales IAs generadoras de imágenes para posteriormente seleccionar 3 de las mismas en función de la coherencia de sus resultados respecto a las instrucciones proporcionadas. Una vez seleccionadas, se ha realizado una serie de pruebas de *prompting* y se ha llevado a cabo una comparación de la adecuación de los resultados obtenidos con cada una de ellas, en distintas temáticas y diferentes estructuras de *prompts*. Finalmente, tras dichas pruebas y análisis de resultados, se ha obtenido una estructura de *prompt* eficiente que permite al usuario indicarle a la IA generadora las características de la imagen deseada de manera concisa.

Por otro lado, para abordar el ámbito de análisis y clasificación de imágenes por IA se ha decidido crear una red neuronal convolucional basada en arquitecturas de ResNet, Inception-ResNet y MobileNet utilizando *data augmentation* y distintos hiperparámetros para poder detectar la malignidad en tumores de glándulas salivales tan solo con la ecografía de las mismas. Para la comparación de resultados, se han utilizado las gráficas de *accuracy* y *loss* de los datos de entrenamiento y validación. Para el entrenamiento y validación de la red se han utilizado imágenes de ecografías tomadas entre los años 2012 y 2021 del Hospital Clínico San Carlos de Madrid.

ABSTRACT

The goal of this Thesis is the approach of Artificial Intelligence topics to achieve a development in the creation, classification and analysis of images. To achieve this, it was carried out a broad study and exemplification paralleling the main AIs Image generator, narrowing the image selection to the best three images for our purpose. Once finished with the selection process, a series of Prompting Tests were performed on each image individually using different Prompt structures and topics allowing us to compare similarities, connection, suitability making possible the creation of a new efficient Prompt structure that will let the user ask the Generating Artificial Intelligence for specific or required characteristics to the desired Image in a more concise/brief way.

On the other hand, to address the field and classification of AI images, it has been decided to establish a convolutional neural network based on architectures such as ResNet, Inception ResNet and MobileNet, using for this data augmentation and different hyperparameters in order to identify and detect the malignancy of tumours in salivary glands merely with a simple ultrasonography. For the comparison of results, there have been used graphics of accuracy and loss from the training and validation data. For the training and validation of the network, there have been used ultrasound images taken at Hospital Clínico de Madrid between the years 2012 and 2021.

1. INTRODUCCIÓN

Este proyecto tiene como finalidad abordar la temática de Inteligencia Artificial (IA) en los ámbitos de análisis, clasificación y generación de imágenes. En primer lugar, se estudiarán las posibilidades que ofrecen actualmente las principales IAs generadoras de imágenes, además de las técnicas utilizadas para obtener dichas imágenes. Adicionalmente, se expondrán algunas técnicas de búsqueda para utilizar estos generadores de manera más eficiente y acorde a la voluntad del usuario.

Por otro lado, en este proyecto se abordará el análisis de imágenes a través del estudio de ecografías de tumores en la glándula parótida mediante el uso de Inteligencia Artificial, más concretamente, a través de la creación de un modelo de red neuronal convolucional basada en las arquitecturas ResNet (Residual Network), Inception-ResNet y MobileNet.

2. MARCO TECNOLÓGICO DEL PROYECTO

2.1. Inteligencia Artificial de generación de imágenes

En la actualidad, las IAs generadoras de imágenes continúan en evolución, la cual en los últimos años ha aumentado exponencialmente. Estas inteligencias artificiales se basan en diferentes técnicas de deep learning, las cuales utilizan redes neuronales con múltiples capas con la finalidad de aprender los distintos patrones que presentan los datos de entrenamiento, además de obtener características jerárquicas y abstracciones en los datos visuales. A su vez, cabe destacar en este ámbito la arquitectura de las redes generativas adversarias (GAN), consistente en la competencia de dos redes neuronales, una encargada de la creación de las imágenes (generador) y la otra encargada de determinar la verosimilitud de las mismas (discriminador), comparando así su parecido con imágenes procedentes de fuentes reales.

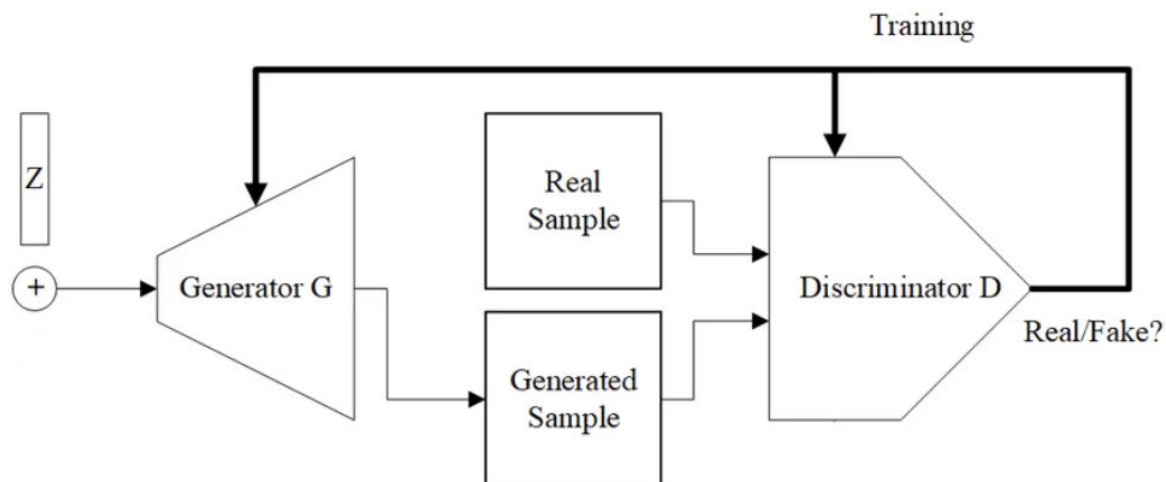


Figura 1: Diagrama de bloques general de una GAN

Teniendo esto en cuenta, se presentan a continuación algunas de las IAs principales en la generación de imágenes:

- **Deep Convolutional Generative Adversarial Networks (DCGAN):** usan redes convolucionales profundas tanto en el generador como en el discriminador, tomando como entrada un ruido aleatorio.
- **Super Resolution Generative Adversarial Networks (SRGAN):** se basa en el uso de GAN para aumentar la resolución de las imágenes de entrada, utilizando a su vez una función de pérdida perceptual que usa características obtenidas por una red previamente entrenada.

- **CycleGAN:** utiliza aprendizaje sin supervisión para la traducción de imágenes entre dos dominios.
- **StyleGAN y StyleGAN2:** versiones de GAN que permiten mayor control sobre las características de las imágenes generadas.
- **BigGAN:** se trata de una versión más grande de la arquitectura GAN que permite obtener imágenes con gran resolución,
- **Vector Quantized Variational Autoencoder 2 (VQ-VAE2):** utiliza un proceso de cuantificación vectorial para generar imágenes que representen diferentes muestras del espacio latente.

A continuación, se presentan brevemente 10 de los principales generadores de imágenes que utilizan técnicas de Inteligencia Artificial para su cometido:

1. Lexica: se trata del creador de imágenes de Bing, es gratuito y utiliza tecnología procedente de DALL-E.

2. Leonardo.AI: para comenzar a utilizar este generador de imágenes es necesario registrarse o iniciar sesión con una cuenta, y proporciona 150 búsquedas de calidad estándar gratuitas cada 8h-9h. Permite seleccionar el modelo generador de imágenes. En caso de tratarse de contenido a generar de mayor calidad, el número de búsquedas gratuito permitido en este margen de tiempo disminuye. Permite diversas personalizaciones, entre las que se encuentran la selección de la resolución de la imagen, el número de imágenes a generar o la relación de aspecto de las mismas.

3. PlaygroundAI: esta herramienta de generación de imágenes también permite editarlas utilizando inteligencia artificial. En su versión gratuita ofrece crear 1000 imágenes por día. Por otro lado, permite seleccionar el tamaño de la imagen a obtener, el número de columnas en el que mostrar las imágenes generadas y el número de imágenes a generar en cada búsqueda. También permite generar variaciones respecto a una imagen de las generadas o una que adjunte el usuario.

4. MidJourney: se trata de una herramienta de pago generadora de imágenes en un servidor de Discord, por lo que es necesario tener una cuenta de Discord para utilizarlo.

5. BlueWillow: se trata de una herramienta generadora de imágenes en un servidor de Discord, permite generar variaciones respecto a una imagen de las generadas y está disponible su versión gratuita. Dicha versión permite generar 25 imágenes gratuitas, posteriormente es necesaria una suscripción mensual.

6. Catbird.AI: se trata de un generador de imágenes gratuito que permite seleccionar el número de imágenes por búsqueda a generar, además de ofrecer una lista de distintos modelos generadores a seleccionar para generar la imagen. Proporciona 12 generaciones (*tokens*) gratuitas, después es necesaria una suscripción mensual.

7. Dream Studio Creator: es gratuita, es necesario iniciar sesión ya sea mediante google, discord o una cuenta propia de Dream Studio. La IA gestiona las generaciones a las que se tiene acceso mediante créditos, proporcionando alrededor de 60 imágenes gratuitas inicialmente. **stability ai**

8. PicFinder: se trata de una herramienta gratuita generadora de imágenes a partir de un prompt, destaca por su rapidez y permite seleccionar la forma de la imagen (cuadrada/alargada/ancho). Genera de manera gratuita hasta 1000 imágenes por día y proporciona numerosas imágenes por búsqueda, y no requiere de iniciar sesión para comenzar a generar imágenes.

9. Clipdrop: esta herramienta, además de ofrecer la generación de imágenes a partir de una descripción textual (prompt), permite reescalar imágenes, modificar luces, eliminar fondos u objetos de imágenes. Se trata de una IA generadora gratuita, utiliza el modelo de Stable Diffusion XL. **stability ai**. Proporciona 400 generaciones de imágenes por día dejando

10. DALL-E: herramienta generadora de imágenes creada por OpenAI, continúa en evolución.

2.2. Redes Neuronales Convolucionales

El término “**red neuronal**” hace referencia a una serie de algoritmos que describen el comportamiento y las conexiones entre las neuronas que forman las capas de las que consta dicha red. Esta estructura de capas formadas por neuronas para poder aprender patrones y procesarlos de manera “automática” está inspirada en la estructura de las neuronas en el interior del cerebro y la evolución de las mismas y sus conexiones durante el proceso de aprendizaje.

Cada neurona realiza una operación utilizando los valores a su entrada. Además, cada neurona posee un sesgo o *bias*, valor que se suma a su entrada antes de procesarla. De este modo, cuando el dato de entrada llega a una neurona, esta le suma su *sesgo* (*bias*, garantiza que la salida de la neurona sea mayor a un determinado valor) y opera sobre ello, obteniendo así un resultado a su salida. Este resultado, tras ser procesado por una **función de activación** que le otorga el comportamiento no lineal a la red neuronal, será la entrada de las neuronas de la siguiente capa. La función de activación de la última capa dará como salida la probabilidad de que la imagen corresponda a cada una de las categorías de salida. De esta forma, las neuronas de una capa reciben la misma información de salida de la capa anterior.

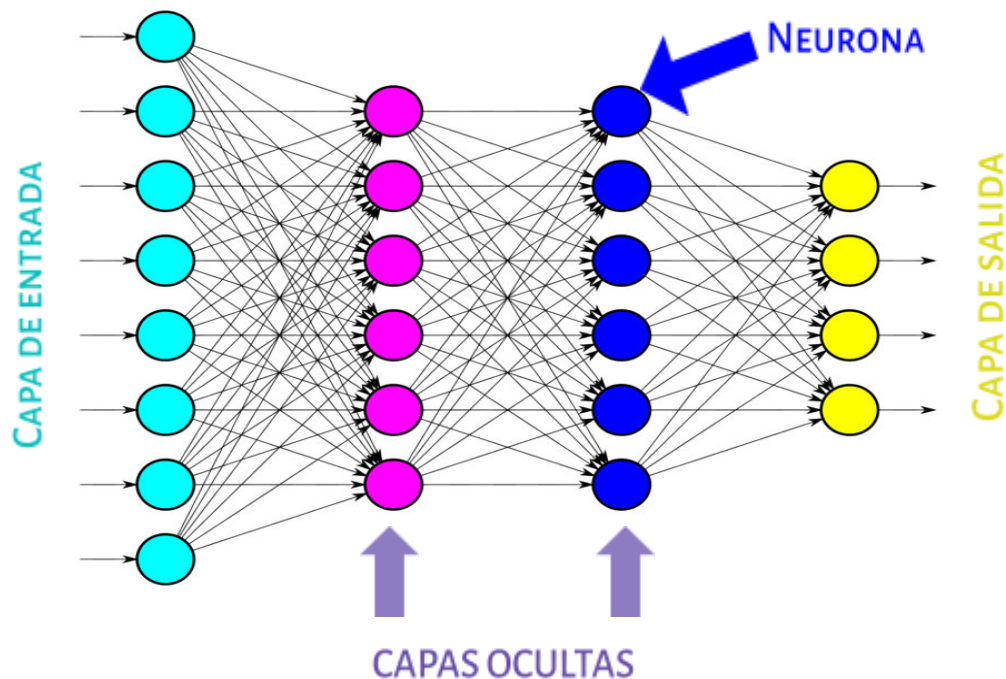


Figura 2: Representación esquemática de una red neuronal de 4 capas

Así, **cuanto mayor es el orden de la capa, más compleja será la información que procese**. Esta profundidad en la cantidad de capas es la que genera lo que se denomina el *deep learning*, ya que la red cuenta con más de una capa oculta.

Las variables de entrada de la red neuronal llegan a la primera capa, denominada capa de entrada, por lo tanto, la capa de entrada tendrá tantas neuronas como variables de entrada. En el caso de que la entrada sea una imagen, la entrada tendrá tantas neuronas como píxeles la imagen. De la misma forma, la última capa, la que proporciona las variables de salida, se denomina capa de salida. A las demás capas se les denomina capas ocultas. Se puede ver una esquematización de un ejemplo de distribución de las neuronas y las capas de una red neuronal en la figura 2.

La fortaleza de las conexiones entre las neuronas están determinadas por su *peso*, el cual se va ajustando de acuerdo a una **función de pérdida** en cada iteración (época) durante el entrenamiento de la red. Al comienzo del entrenamiento, los pesos de todas las conexiones de la red neuronal se encuentran inicializados aleatoriamente. De este modo, los ajustes disponibles en relación a una neurona artificial se podrían observar de un modo similar al representado en la figura 3.

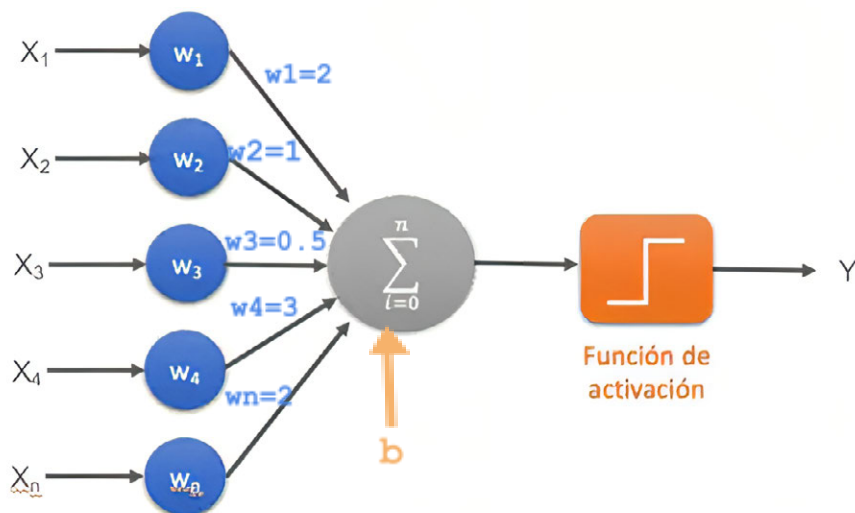


Figura 3: Representación de los parámetros ajustables en relación a las neuronas de la red

A modo de síntesis, al finalizar cada época, la función de pérdida se encarga de comparar el resultado correcto con el obtenido en esa época. Con el resultado de dicha comparación, el **optimizador se encargará de ajustar los pesos de las conexiones internas, y los sesgos (biases) de cada neurona**.

Uno de los métodos de optimización más extendidos es el de **descenso de gradiente** (*gradient descent*), el cual se basa principalmente en el cálculo del gradiente de la función de pérdida (dependiente de los sesgos y los pesos de las conexiones entre neuronas). Tras la realización de este cálculo, se ajustan los valores de los sesgos y pesos en la dirección en la que el gradiente tenga mayor pendiente en sentido descendente, ya que indica el sentido en el que la función de pérdida disminuye más rápidamente. La medida en la que se modifican los sesgos y los pesos en cada época viene determinada por el ratio de aprendizaje (*learning rate*). De este modo, se va llevando a cabo este proceso desde la salida de la última capa hasta la de la primera, lo que se conoce como **retropropagación del gradiente** (*backpropagation*). El alcance de ajuste de parámetros del método de retropropagación del gradiente, sin embargo, se ve limitado cuanto más profunda es la red, llegando incluso a aumentar el error de entrenamiento y disminuir la exactitud (*accuracy*) de la red con el aumento de las capas. Esto ocurre debido a que al aumentar la profundidad de la red, al realizar la retropropagación del gradiente, el valor del gradiente puede alcanzar valores muy pequeños o incluso tender a 0, reduciendo la capacidad de aprendizaje de la red. Por lo tanto, se crea un problema denominado *degradación* (*degradation problem*) en la red que, en ocasiones, es tratado mediante inicializaciones normalizadas y capas intermedias de normalización.

Sin embargo, estos ajustes y proceso de aprendizaje resultan mejorables respecto a visión artificial, análisis y clasificación de imágenes. Por ello, surgieron las **redes neuronales convolucionales**. Este tipo de red fue especialmente creada para tareas relacionadas con el procesamiento de imágenes, incluyen **capas convolucionales** que utilizan *filtros* que se convolucionan con la imagen para obtener diferentes características de interés: bordes, texturas, patrones, ... Además, este tipo de redes contienen también **capas de agrupación**, las cuales se encuentran generalmente después de una capa convolucional, y su función principal se basa en reducir las dimensiones de las imágenes a procesar y resaltar sus características más esenciales.

2.2.1. Filtros en redes neuronales convolucionales

Un filtro consiste en una matriz generalmente cuadrada cuyos valores se pueden modificar en función de las propiedades de la imagen que se deseen resaltar. Dicha matriz, denominada núcleo o kernel, posee un píxel principal en el centro sobre el que situarán todos los píxeles de la imagen para realizar la convolución. Así, cuando haya terminado el proceso de filtrado, el kernel se habrá convolucionado con todos los grupos de píxeles del mismo tamaño de la imagen, habiéndose utilizado todos los píxeles como píxel principal del núcleo. En cada época, los valores del filtro serán ajustados por el optimizador, además de los demás parámetros nombrados en la sección de redes neuronales.

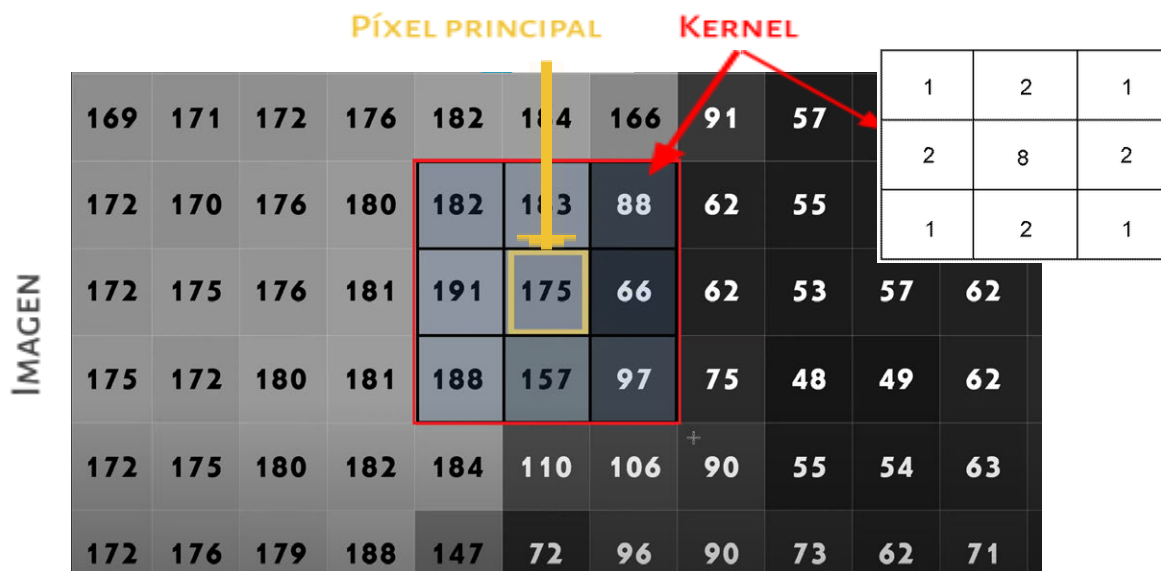


Figura 4: Representación de convolución con un kernel de 3x3

Sabiendo esto, se presentan algunos de los filtros más comunes junto a su utilidad:

- Filtro para obtención de bordes (altas frecuencias): destacan, entre otros, los operadores de Sobel (figura 5) y algoritmo de Canny.

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

Figura 5: Filtro Sobel para obtener bordes horizontales (izq) y verticales (dcha)

- Filtro para reducción de bordes (muestra de bajas frecuencias): destacan los filtros de media (Figura 6) y de mediana.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figura 6: Ejemplo filtro de media 3x3

Sabiendo esto, se puede concluir que las **capas que forman una red neuronal** pueden ser principalmente de los siguientes tipos:

- **Convolutivo:** convoluciona su entrada con un filtro con coeficientes que se determinan durante el entrenamiento. Este tipo de capa se define mediante el número de filtros (*kernels*) que aplicar a la imagen y el tamaño de los mismos. También es posible definir un número de *strides* en la capa convolutiva, es decir, el número de píxeles que se desplaza el *kernel* hacia la derecha o hacia abajo en la imagen. En el caso de las imágenes de entrada RGB, se realiza la convolución con los filtros de manera independiente por cada canal para luego sumar las 3 salidas y el sesgo, lo que es denominado como *mapa de activaciones 2D*. En ocasiones, previo a la aplicación de filtros, se aplica *padding*, que consiste en agregar píxeles con valor 0 a los bordes de la imagen para que la resultante de aplicar los filtros tenga las mismas dimensiones que la imagen original.

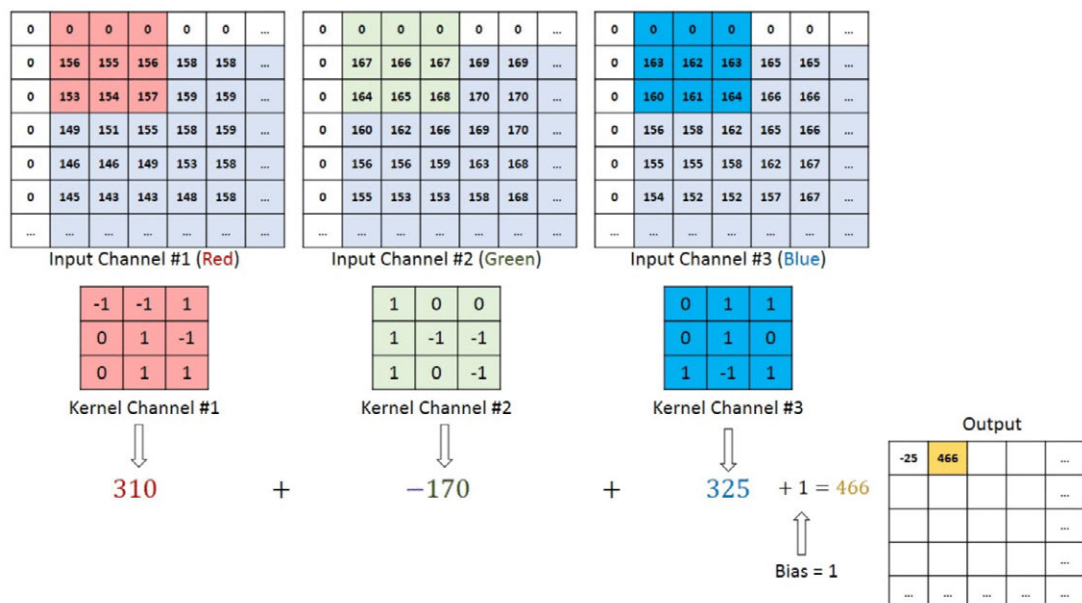


Figura 7: Ejemplo aplicación kernel 3x3 a imagen RGB

- **Densa:** son las capas más sencillas y utilizadas, la salida de cada neurona de esa capa va a todas las neuronas de la capa siguiente.
- **Capas de pooling:** se trata de una capa que aplica filtros para reducir el tamaño de los mapas de activación, permitiendo así el análisis por regiones de la imagen. Uno de los tipos más comunes de pooling es el *máx-pooling*, en el que la imagen se divide en regiones del tamaño del filtro y se extrae el píxel de mayor valor de la región. Se muestra en la figura 8 un ejemplo de la aplicación de máx pooling con un kernel 2x2.



Figura 8: Ejemplo aplicación de máx pooling con kernel 2x2

- **Capas de normalización:** se utilizan para regular y estabilizar el proceso de entrenamiento de una red, actuando sobre los mapas de activación. Entre sus propósitos principales se encuentra reducir el riesgo del problema del desvanecimiento del gradiente, además de aumentar la robustez de la red. Generalmente, el proceso de normalización consiste en calcular la media y la desviación estándar de los datos, luego se lleva a cabo una transformación lineal y finalmente la normalización para ajustar la distribución. En el caso de la capa de *normalización por lotes* (Batch Normalization), los datos se procesan en subconjuntos (mini-batches), mientras que en el caso de la capa de *normalización por capas* (Layer Normalization) se opera sobre los datos de una capa específica.

2.3. Marco teórico médico

Las glándulas salivares son encargadas de la producción de saliva y pueden ser menores o mayores, las primeras de muy pequeño tamaño y distribuidas a todo lo largo de la vía aerodigestiva con especial concentración en la zona superior, y las mayores anatómicamente identificables y en número de seis: glándulas parótidas, glándulas submandibulares y glándulas sublinguales. Los tumores de glándulas salivales constituyen un 3%-6% de todos los tumores de cabeza y cuello. Los tumores benignos tienden a ocurrir en personas con edades cercanas a 45 años, mientras que los malignos suelen afectar en mayor medida a personas cuya edad está comprendida en el rango de 55 a 65 años.

Además, los tumores de las glándulas parótidas resultan entre un 70% y un 80% de los que tienen lugar en cabeza y cuello, precedidos por los de la glándula submandibular (8%-15%) y los de las glándulas salivales menores (5%-8%), siendo los tumores de la glándula sublingual los menos frecuentes. Por otro lado, cuanto más pequeña sea la glándula, mayor será la probabilidad de lesión maligna, resultando en un 80% de los casos de tumores en glándulas salivales menores, entre un 37% y un 43% de los de la glándula submandibular y entre un 15% y un 25% de los de la glándula parótida.

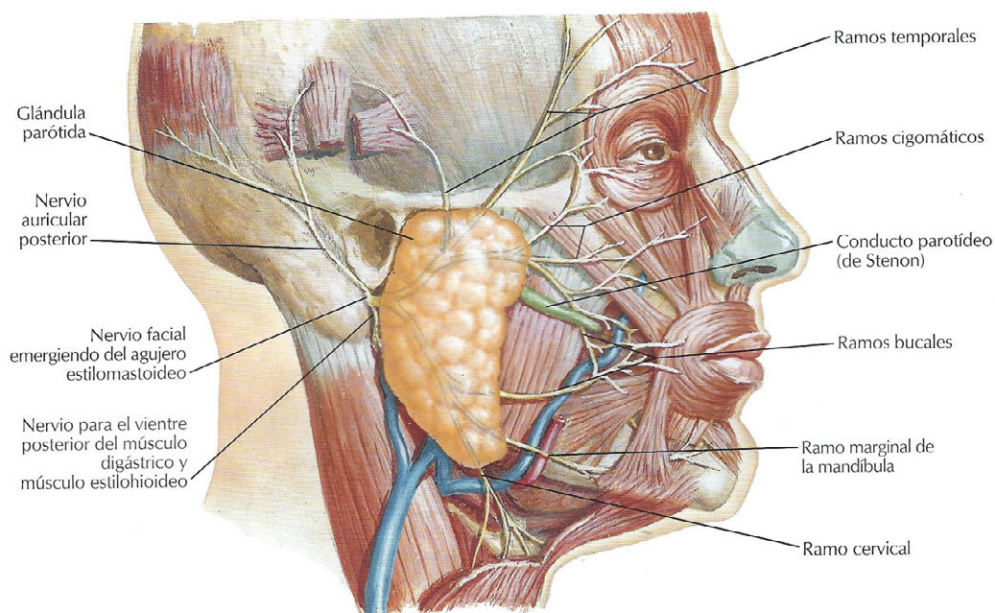


Figura 9: Ramos del nervio facial y glándula parótida

La glándula parótida es la mayor de las tres glándulas salivales principales de la cabeza y a través de ella discurren numerosas estructuras. Se sitúa anterior y caudal a la mitad inferior de la oreja; superficial, posterior y profunda a la rama de la mandíbula. Se extiende caudalmente hasta el borde inferior de la mandíbula y cranealmente hasta el arco cigomático. Posteriormente cubre la parte anterior del músculo esternocleidomastoideo y se extiende anteriormente hasta la mitad del músculo masetero. [1]

La glándula parótida produce saliva acuosa y amilasa salival, necesarias para la formación del bolo alimenticio, la digestión oral y para facilitar un descenso suave del bolo alimenticio hacia los tramos superiores del aparato gastrointestinal.

Los tumores más frecuentes de la glándula parótida son benignos y por regla general suelen afectar a la porción superficial de la glándula. Entre estos tumores se incluyen el adenoma pleomórfico y el adenolinfoma. Su importancia estriba en las relaciones anatómicas de la glándula. [1]

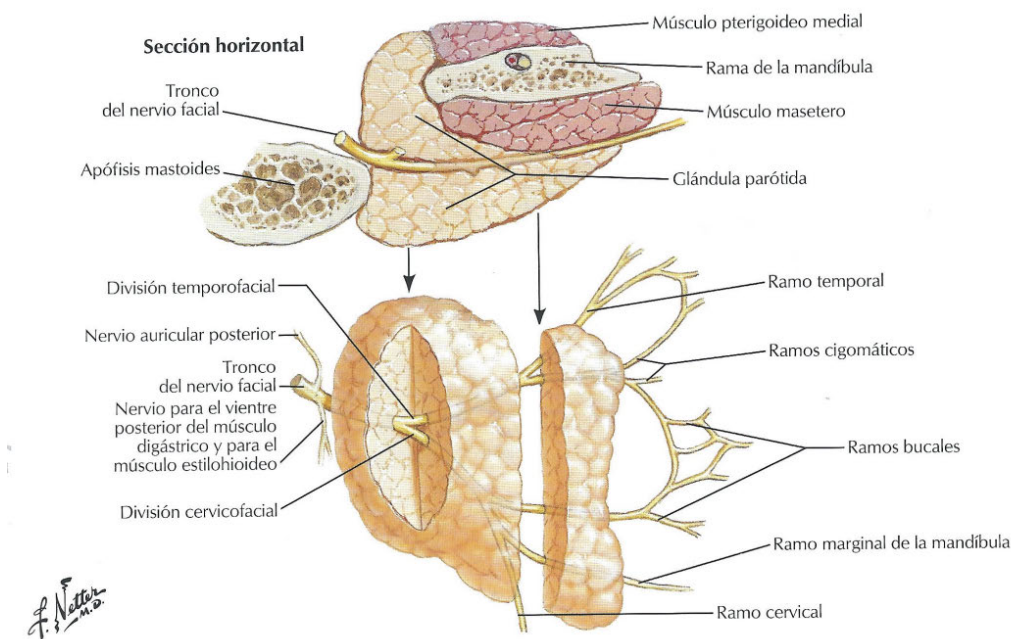


Figura 10: Sección de la glándula parótida

La ecografía del cuerpo está ampliamente difundida en todos los aspectos de la medicina.

Los ultrasonidos son ondas sonoras de frecuencia muy elevada (no radiaciones electromagnéticas) generadas por materiales piezoeléctricos que producen una serie de ondas sonoras. Es importante reseñar que el material piezoeléctrico también puede recibir las ondas sonoras que rebotan en los órganos internos. Estas ondas sonoras son entonces interpretadas por un potente ordenador, que genera una imagen en tiempo real en la pantalla. [1]



La ecografía es una técnica que emplea una banda de ultrasonidos para hacer visibles los órganos, y permite seguir en tiempo real sus movimientos y fenómenos fisiológicos. Un generador de ondas sonoras de alta frecuencia se coloca sobre la zona del cuerpo que se va a examinar; las ondas se propagan en profundidad y son reflejadas de diferente manera por órganos y tejidos, según su densidad. Un aparato capta los ecos y los convierte en imágenes que se visualizan en un monitor, siendo posible dejar grabadas las informaciones sobre los órganos - tridimensionales o estructurales - en fotografías o filmaciones. [2]

Figura 11: TC axial con tumor en la glándula parótida



Figura 12: Toma de ecografía en la glándula parótida

El análisis por ecografía debe considerarse el estudio inicial ante la sospecha clínica de tumor de glándulas salivales mayores. Este estudio se utiliza, entre otras cosas, para confirmar la localización de dicho tumor, caracterizar su morfología, bordes y detectar si se trata de un tumor quístico. Además, también es utilizado para guiar un posible PAAF (Punción Aspiración con Aguja Fina) futuro. Sin embargo, el estudio mediante ecografías también presenta algunas limitaciones: baja especificidad, limitado ante lesiones profundas (lóbulo profundo de la glándula parótida), glándulas salivales menores, diseminación perineural, invasión ósea o afectación ganglionar retrofaríngea. Por ello, en numerosas ocasiones se realiza una PAAF, logrando una alta sensibilidad que oscila entre el 87% y el 94%, además de una especificidad de entre el 75% y 100%.

3. ESTADO DEL ARTE

El término de Inteligencia Artificial se comenzó a utilizar en 1956, creado por el informático John McCarthy, siendo un año más tarde el momento de la creación de la primera red neuronal artificial, a manos de Frank Rosenblat. Más tarde, en 1966, Joseph Weizenbaum desarrolla en el MIT a ELIZA, el primer programa que incorpora el procesamiento del lenguaje natural humano, permitiendo así comunicación usuario - ordenador sin requerir lenguaje de código. El estudio e investigación de este área continuó, creando Marvin Minsky "Perceptrones", el trabajo fundamental del análisis de redes neuronales artificiales.

Así, el desarrollo de la Inteligencia Artificial mediante redes neuronales fue evolucionando, generando los primeros vehículos autónomos en 1979 o logrando vencer al campeón del mundo de ajedrez en 1996. En 2012, Google creó un superordenador capaz de llevar a cabo tareas de reconocimiento aprendiendo a través de Youtube.

3.1. Inteligencia Artificial en generación de imágenes

Este proyecto aborda la temática mediante un análisis y comparación de las Inteligencias Artificiales generadoras de imágenes más extendidas y, con ello, un estudio de las técnicas de prompting más eficientes para transmitir a los generadores que se desea obtener de la manera más veloz y adecuada. En el campo de la inteligencia artificial y los chatbots, se puede entender como *prompt* un conjunto de instrucciones en lenguaje natural que se le proporciona a un software de IA para obtener un resultado específico. Dicho resultado puede consistir en un texto, una imagen, un vídeo, música o incluso código informático. Por lo tanto, el modo en el que se estructuran y enuncian dichas instrucciones posee una influencia crucial en la calidad del resultado que la IA nos proporciona.

Además, la obtención de estos resultados de una manera más rápida y eficiente supone un ahorro de tiempo y recursos en el desarrollo y pruebas de los productos empresariales. Como consecuencia, las empresas logran producciones más eficientes e innovadoras y, por lo tanto, mayor ventaja competitiva en el mercado.

Como resultado, desde el momento en el que comenzaron a existir las Inteligencias Artificiales generadoras de cualquier tipo de contenido, comenzó a surgir la *ingeniería de prompts*, cuyo cometido se centra en el análisis y estudio del diseño y estructura de las instrucciones que se introducen en un modelo de

lenguaje basado en inteligencia artificial. Para ello, se debe trabajar en aspectos como la claridad y la concisión de las instrucciones, la contextualización, la repetición y ajuste de las mismas o el control del estilo y el tono del *prompt*, entre otros. Se puede observar la influencia de las características nombradas, entre otras, en la comparación presente en la figura 13.

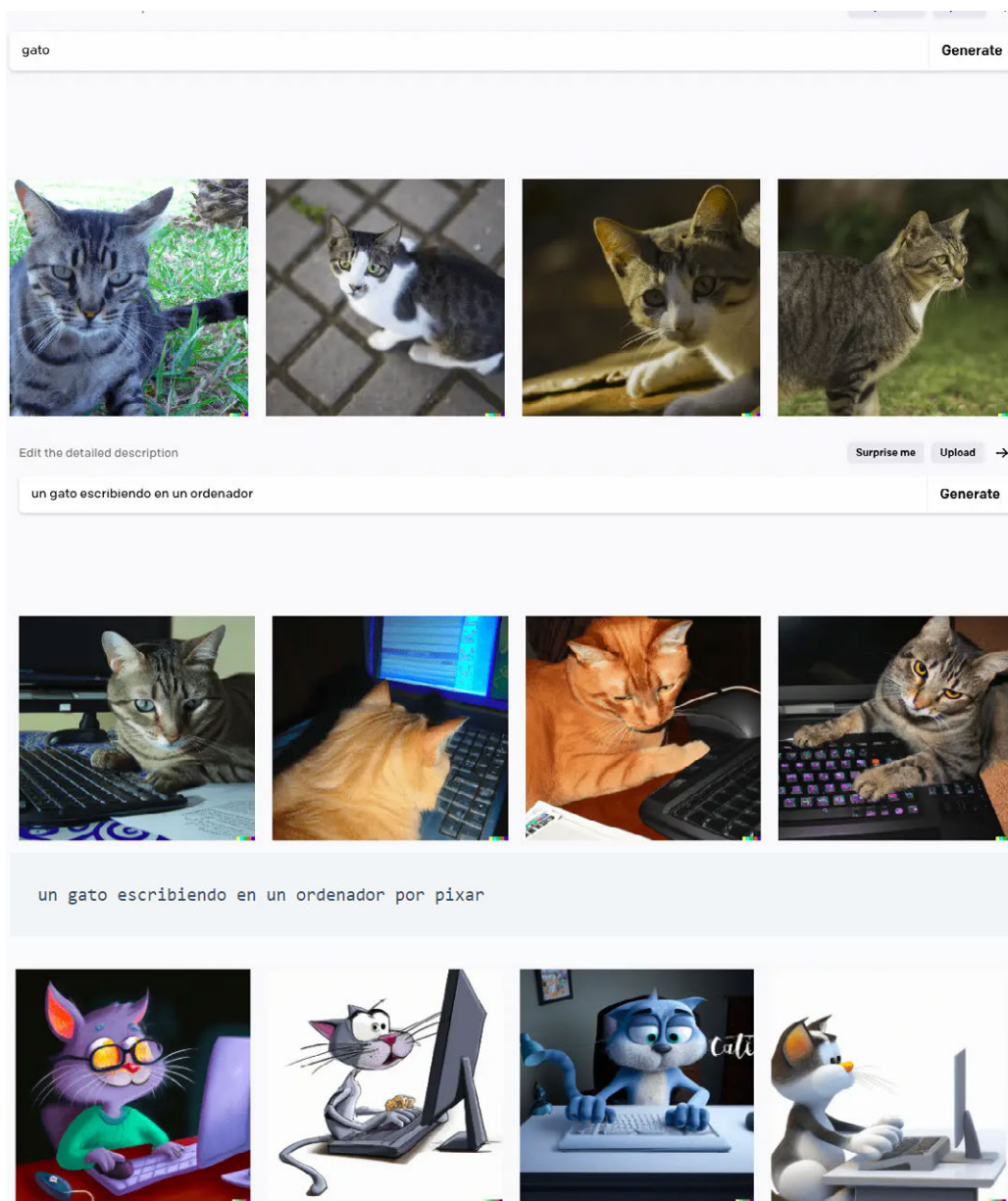


Figura 13: Comparación en resultados de imágenes generales con el nivel de detalle de los prompts

Cabe destacar que debido a la complejidad que en numerosas ocasiones puede presentar la elaboración de los prompts adecuados para obtener los resultados deseados de una IA, resulta más sencillo partir de otros prompts como punto de partida. Por ello, existen actualmente algunos buscadores que presentan imágenes ya generadas por IA junto a sus prompts, proveyendo así al usuario

una base de la que partir con un resultado similar al deseado. Entre los más utilizados de estos buscadores se encuentran Lexica, Public Prompts o PromptHero.

Entre las técnicas principales a tener en cuenta en elaboración de prompts se encuentran las descripciones detalladas de los principales objetos, colores y posiciones en la escena o utilizar un lenguaje claro y conciso, sin frases complejas que puedan generar confusión a la IA. A su vez, se recomienda el uso de lenguaje natural, evitando tecnicismos. Además, resulta de utilidad ir probando las indicaciones, visualizar los resultados e ir perfeccionando y detallando dichas instrucciones en base a los mismos. Una forma de aplicación de esta última técnica consiste en dividir el detalle de la descripción en niveles, comenzar por guiar al modelo para generar una estructura general y posteriormente agregar los detalles específicos. También resulta de gran utilidad proporcionar imágenes de referencia a la IA, bien imágenes propias o bien imágenes generadas por la misma IA. Entre otros parámetros de utilidad para describir la imagen se encuentran la técnica, corriente o estilo artístico (estilo cubista, Van Gogh, estilo cómic, acuarelas, Dalí, fotografía, ...) o la temperatura de la imagen. A su vez, cabe destacar que en general, los prompts efectuados en inglés tienden a ser más efectivos. Una interesante estructura de prompt con la que se experimenta en este proyecto es *Tipo de obra | Contenido | Estilo | Modificadores*.

También resulta de gran utilidad indicarle características tanto objetivas como subjetivas acerca del aspecto de los principales objetos/sujetos de la imagen a generar, incluso hacer referencia a la anatomía del mismo o a la posición concreta que adopta en la imagen. A su vez, se debe indicar en el prompt el grado de resolución deseado en la imagen (4k, HDR, *out of focus*, alta resolución, ...), incluso se puede hacer referencia a una distancia focal concreta.

Sin embargo, cabe destacar que la generación de imágenes por IA aún presenta algunas limitaciones. Este es el caso de las alucinaciones, es decir, la capacidad de generar salidas que no se encuentran presentes en los datos de entrada, conduciendo así a resultados inexactos o poco fieles a los deseados.

3.2. Inteligencia Artificial (CNN) en clasificación de imágenes médicas

Este proyecto aborda la temática mediante la elaboración y comparación de 3 modelos de redes neuronales basados en 3 arquitecturas de redes convolucionales diferentes. Se ha decidido utilizar redes convolucionales debido a su especial eficacia en el aprendizaje de clasificación de imágenes. Para este proyecto, se ha decidido entrenar a dichas redes utilizando técnicas de *transfer learning* y *data augmentation* y se analiza su capacidad de clasificación de ecografías de tumores malignos/benignos en glándulas parótidas dado un conjunto de datos real. Este conocimiento previo resulta fundamental, además de para conocer la gravedad de la patología, para la toma de decisión de la metodología quirúrgica a utilizar. En el caso de ser un tumor benigno, se debe realizar una parotidectomía parcial o superficial con preservación del nervio facial, mientras que si se trata de un tumor maligno, se debe realizar una parotidectomía total con o sin preservación del nervio facial.

Actualmente, el procedimiento para realizar el diagnóstico de dicho tumor es el siguiente: los pacientes con tumores en las glándulas parótidas normalmente consultan por tumoración palpable en torno al pabellón auricular, la oreja. Otras veces el paciente no tiene síntoma alguno y se trata de un hallazgo observado en alguna prueba de imagen realizada por otro motivo (incidentaloma), bien sea ecografía, TC o RM. En cualquiera de los casos, el procedimiento diagnóstico inicial debe ser una ecografía para confirmar la existencia de lesión y tratar de definir sus características. La presencia de lesiones bien definidas normalmente orientan a una lesión benigna mientras que los bordes irregulares, especialmente si se asocian a dolor o parálisis facial, sugieren que puede tratarse de un tumor maligno. La ecografía tiene una limitación cuando se trata de lesiones profundas en la glándula, no accesibles a esta prueba; en estos casos es necesaria la realización de Tomografía Computarizada (TC) o, preferiblemente, Resonancia Magnética (RM). Por otro lado, factores como la fatiga, el exceso de carga de trabajo o incluso la hora en el día pueden afectar en la eficacia con la que los radiólogos identifican los tumores en las imágenes. Por lo tanto, el diagnóstico de una IA basada en redes neuronales convolucionales consultada y valorada por profesionales de la medicina daría como resultado una valoración y revisión más exhaustiva de las ecografías, reduciendo así la probabilidad de un diagnóstico erróneo.

En la actualidad, el análisis de imágenes con IA tiene múltiples aplicaciones, entre las que se incluyen la detección de fraudes, supervisión de calidad en la industria, sistemas de vigilancia o identificación de enfermedades a través de imágenes médicas.

La técnica de diagnóstico por imagen forma parte de la labor del profesional en la medicina desde hace más de 100 años, con sus inicios en el año 1895 con el descubrimiento de los rayos X. Sin embargo, el uso de técnicas de Inteligencia Artificial en el análisis de imágenes médicas resulta una gran mejora y aumento de la eficiencia y la productividad, ayudando a los médicos en la toma de decisiones y en la identificación temprana de enfermedades. Además de la mejora en precisión, el uso de Inteligencia Artificial también colabora en la reducción del tiempo necesario para interpretar las imágenes, lo que resulta crítico en situaciones de emergencia en patologías de alta gravedad donde el tratamiento adecuado a tiempo es de importancia vital.

La inteligencia artificial puede ofrecer una interesante precisión en el diagnóstico de distintos tipos de cáncer: en 2018, un sistema de inteligencia artificial chino (BioMind) obtuvo un 87% de aciertos frente al 66% de los médicos en una tarea de diagnóstico de tumores cerebrales y hematomas [48]. Este robot había sido entrenado durante 10 años con decenas de miles de imágenes relacionadas con el sistema nervioso, haciéndole capaz de diagnosticar enfermedades neurológicas comunes como meningioma y glioma con una tasa de precisión de más del 90%.

Por otro lado, la empresa alemana Vara se encuentra desarrollando una Inteligencia Artificial cuyo objetivo es detectar el cáncer de mama en las ecografías con mayor rapidez, para así aumentar el número de pacientes que puedan ser tratados por profesionales en el menor tiempo. *"Además de las imágenes, estas muestras también incluyen información sobre los antecedentes del paciente con la enfermedad, los diagnósticos iniciales y las notas de radiólogos encargados de revisarlas. Así, han logrado que esta inteligencia artificial pueda separar las muestras obtenidas en tres categorías distintas: "cáncer seguro", "no seguro" y "seguro normal"."* [49]. Tras realizar el análisis, la IA envía su resultado a los médicos para tomar la decisión final en los diagnósticos considerados normales, consiguiendo así un análisis más exhaustivo de las ecografías. Para entrenar a la red neuronal, Vara ha utilizado más de 367000 imágenes. Gracias a la utilización de esta Inteligencia Artificial creada por Vara, los radiólogos mejoraron su detección de cáncer de mama en un 2.6%. De esta forma, esta IA continúa en evolución y ya se ha implantado en hospitales de Alemania, México y Grecia. Aunque este estudio no muestra una mayor exactitud en la detección por parte de la IA

frente al diagnóstico de radiólogos experimentados, según Curtis Langlotz, director del centro de Inteligencia Artificial en Medicina e Imagen en Stanford, aumenta la eficacia y velocidad del diagnóstico siendo utilizada por profesionales de la medicina, además de resultar de gran ayuda en países con escasez de radiólogos como Malawi, India o incluso USA. [50]

Este año, NVIDIA ha anunciado que implementará su tecnología de inteligencia artificial en dispositivos médicos con la finalidad de detectar el cáncer de colon de forma temprana. Se ha firmado un contrato con Medtronic para integrar su IA en el módulo de endoscopia GI Genius [51]. Esta solución propuesta por NVIDIA marca las zonas con características anormales en tiempo real en las colonoscopias de forma similar a la mostrada en la figura 14.

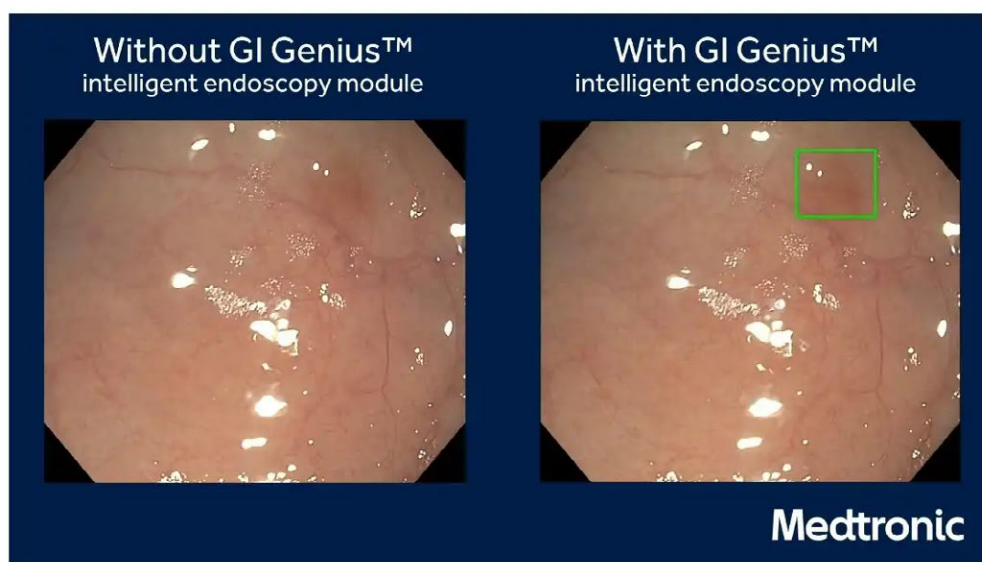


Figura 14: Representación de proceso de colonoscopia sin IA (izq) vs con IA (dcha)

“GI Genius ha sido diseñado para albergar un conjunto de algoritmos de IA y la integración de la plataforma de atención médica NVIDIA Clara podría permitir a Medtronic escalar el desarrollo de algoritmos para procedimientos en tiempo real, acelerando potencialmente la innovación de IA para una mejor atención al paciente.” NVIDIA, [52].

Las redes neuronales convolucionales también son utilizadas en la clasificación de enfermedades cerebrales tales como el Alzheimer, logrando una exactitud del 86,7%, 78,9% y 95,6% para cerebro con Alzheimer, cerebro con lesión y cerebro normal, respectivamente. Se utilizó una fusión de redes convolucionales bidimensionales y tridimensionales, tomando como entrada tomografías computarizadas cerebrales. [53]

4. METODOLOGÍA

4.1. Recopilación de datos

Las imágenes a procesar, analizar y clasificar por la red corresponden a ecografías de tumores de las glándulas salivales mayores, concretamente de la glándula parótida.

Las imágenes de ecografías de glándulas parótidas junto a sus diagnósticos correspondientes utilizadas para entrenar el modelo creado en este ámbito han sido proporcionadas por el Hospital Clínico San Carlos, en Madrid, haciendo uso del ecógrafo Toshiba Aplio 500 (Figura 15). Las imágenes se obtienen con el paciente en decúbito supino (tumbado boca arriba), con la cabeza ligeramente extendida y girada hacia el lado contrario a la parótida que se quiere explorar. Se realiza un barrido con el transductor sobre el área preauricular obteniéndose imágenes dinámicas en planos axial y longitudinal y oblicuas. La glándula parótida normal se observa más blanca (hiperecogénica) que el resto de los tejidos adyacentes, como ocurre con el resto de las glándulas, y es fácilmente identificable en su interior la vena retromandibular que se encuentra próxima al nervio facial, que no se ve con ecografía. Las lesiones en la glándula parótida suelen ser más negras que el resto de la glándula (hipoeecogénicas).

En primer lugar, se contactó con el hospital, y tras dos visitas al mismo y comunicación mediante correo electrónico y mensajería instantánea con parte del equipo de radiólogos, se decidió llevar a cabo el proyecto para el análisis y clasificación de tumores de glándulas parótidas en función de su ecografía.

Dado que las ecografías utilizadas se encontraban en el archivo del Hospital, se llevó a cabo una búsqueda de los pacientes cuyo diagnóstico estuviese respaldado bien por operación quirúrgica, por resonancia magnética o por prueba PAAF. Cabe destacar que el modo de guardar los archivos de las ecografías es mediante imágenes que incluyen los datos personales de los pacientes, entre otros. Por lo tanto, tras realizar esta búsqueda, se realizaron recortes en formato png sobre estas imágenes.

Se dispone de un archivo Excel que relaciona los pacientes con su diagnóstico (maligno/benigno), de manera que se distribuyen todas las imágenes en dos carpetas con los nombres "benigno" y "malignos", una a una. Por otro lado, alrededor de un 30% de las ecografías guardadas en el archivo del hospital presentan marcadores de diferente naturaleza tales como flechas, puntos, ... (Figura 16) que no forman parte de la imagen original de la ecografía, por lo que se decide apartarlas en otra carpeta para la realización de pruebas. Por otra parte, aunque en menor proporción que estas últimas, algunas de las ecografías guardadas resultan ser ecografías Doppler (Figura 17), las cuales incluyen

información acerca del flujo, dirección y velocidad de un vaso sanguíneo, información paralela a este estudio. Por lo tanto, se decide descartar este tipo de ecografías para incluirlas en los datos de entrenamiento o validación.

Por último, se procedió a comprimir dichas carpetas para aumentar la agilidad en la descarga.



Figura 15: Ecógrafo Toshiba Aplio 500

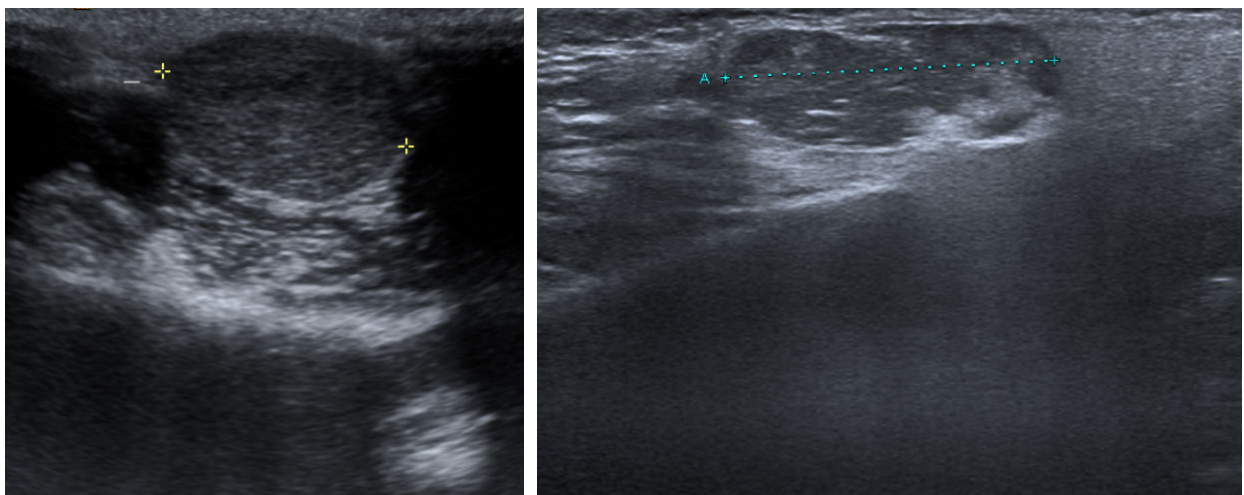


Figura 16: Imágenes de ecografías de la glándula parótida con marcas superpuestas

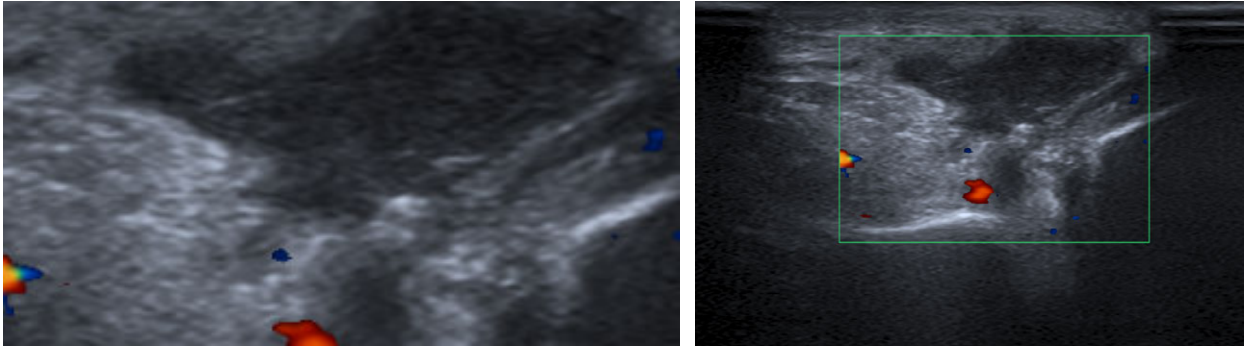


Figura 17: Imágenes de ecografías Doppler de la glándula parótida

4.2. Preprocesamiento de datos

Tras la previa clasificación de las imágenes para entrenar y validar el modelo, se procede al manejo de las imágenes en el entorno de desarrollo. En primer lugar, se crea una estructura del dataset con las categorías y las imágenes correspondientes mediante el siguiente código en python:

```
#Crear dataset
num_B = len(os.listdir('/content/benignos'));
num_M = len(os.listdir('/content/malignos'));
num_muestras = min(num_B, num_M);

!mkdir /content/dataset
!mkdir /content/dataset/benignos
!mkdir /content/dataset/malignos

source_file = '/content/benignos'
dest_file = '/content/dataset/benignos'
imagenes = os.listdir(source_file)

for i, nameimag in enumerate(imagenes):
    if i < num_muestras:
        shutil.copy(source_file + '/' + nameimag, dest_file + '/' + nameimag)

source_file = '/content/malignos'
dest_file = '/content/dataset/malignos'
imagenes = os.listdir(source_file)

for i, nameimag in enumerate(imagenes):
    if i < num_muestras:
        shutil.copy(source_file + '/' + nameimag, dest_file + '/' + nameimag)
```

Por otro lado, para enriquecer el dataset y permitir un entrenamiento más eficiente de la red, se utiliza la técnica de aumento de datos (*data augmentation*). La técnica de aumento de datos en el entrenamiento de redes neuronales consiste en la aplicación de diversas transformaciones a las imágenes de los datos de entrenamiento con la finalidad de aumentar la diversidad de datos de entrenamiento disponibles y mejorar así la generalización del modelo.

A su vez, el aumento de datos ayuda a prevenir el sobreajuste (*overfitting*) proporcionando variaciones sobre las imágenes originales. El *overfitting* tiene lugar cuando un modelo ha sido entrenado más épocas de las debidas, produciéndose así una divergencia en las gráficas de precisión y pérdida entre entrenamiento y validación, como es el caso de la figura 18. Algunas de las transformaciones que ofrece el aumento de datos son la rotación, desplazamiento horizontal/vertical, cizallamiento,

diferentes rangos de zoom, volteo horizontal/vertical, recorte aleatorio o la agregación de ruido a la imagen, además de ajustes de brillo, contraste y saturación.

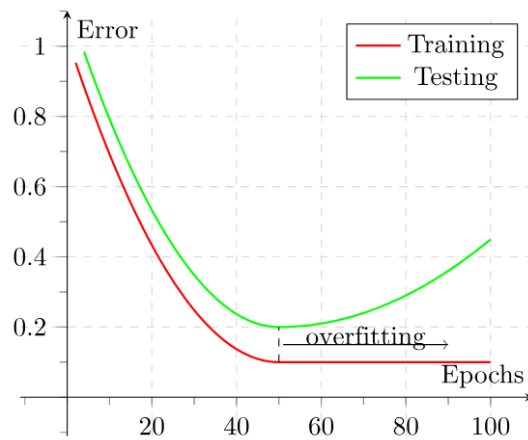


Figura 18: Representación de overfitting en curva de pérdida

Posteriormente, en la misma función de data augmentation también se puede aplicar un reescalado a las imágenes. En este caso, esta función de aumento de datos es utilizada para normalizar los datos, de manera que todos los píxeles tienen un valor en el rango $[0,1]$. Se muestra a continuación un ejemplo de los parámetros utilizados en el aumento de datos para el preprocesado de las imágenes:

```
datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=40,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
```

4.3. Arquitecturas de CNN utilizadas

Para la generación de los modelos de redes neuronales convolucionales se ha decidido utilizar la técnica de **transferencia de aprendizaje**, la cual consiste en aprovechar modelos preentrenados con grandes conjuntos de datos tales como ImageNet y ajustarlos a la tarea de clasificación deseada. Esta técnica, además de acelerar el proceso de entrenamiento de la red, ayuda a mejorar el rendimiento de redes con conjuntos menores de datos, como es el caso de ecografías de tumores de la glándula parótida. Se aplica esta técnica para realizar pruebas sobre 3 modelos de arquitecturas de redes neuronales convolucionales: ResNet (Residual Network), InceptionResNet y DenseNet (Densely Connected Convolutional Network).

- **ResNet (Residual Network):** se trata de un modelo de aprendizaje profundo (*deep learning*) que incorpora un nuevo tipo de conexión entre capas, la *conexión residual*. ResNet fue creado en 2015 por Kaiming He, Xiangyu Zhang, Shaoqing Ren y Jian Sun. En la figura 19 se puede observar un ejemplo de esquema general que siguen este tipo de conexiones. Dicha figura está extraída del paper de ResNet cuando resultó ganadora del ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [3], y en ella se muestra una conexión residual que se salta 2 capas de la red ($X, identity$) para más tarde ser sumada al resultado de estas dos capas ($F(x) + x$). Al resultado de la suma se le aplica la función de activación ReLu, conformando la entrada de la siguiente capa.

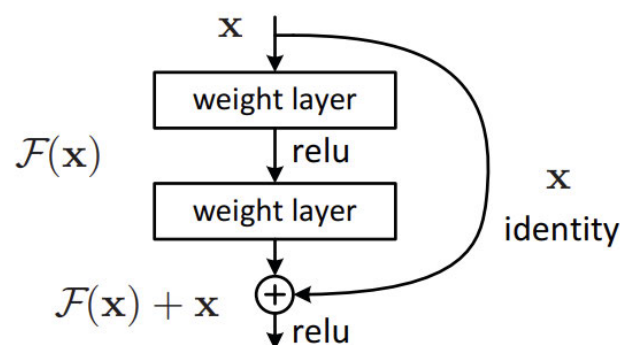


Figura 19: Bloque de aprendizaje residual [3]

Agregar estas conexiones supone la solución del problema de la degradación de la red generada por el desvanecimiento del gradiente, permitiendo la retropropagación del mismo mediante las conexiones residuales (*shortcuts*) sin degradarse. De esta manera, se produce una mejora del entrenamiento de modelos más profundos y alcanzando un mayor rendimiento en tareas como visión computacional, clasificación de imágenes o segmentación semántica. Se muestra en la figura 20 una red con arquitectura ResNet de 34 capas que consta de conexiones residuales entre pares de capas convolucionales.

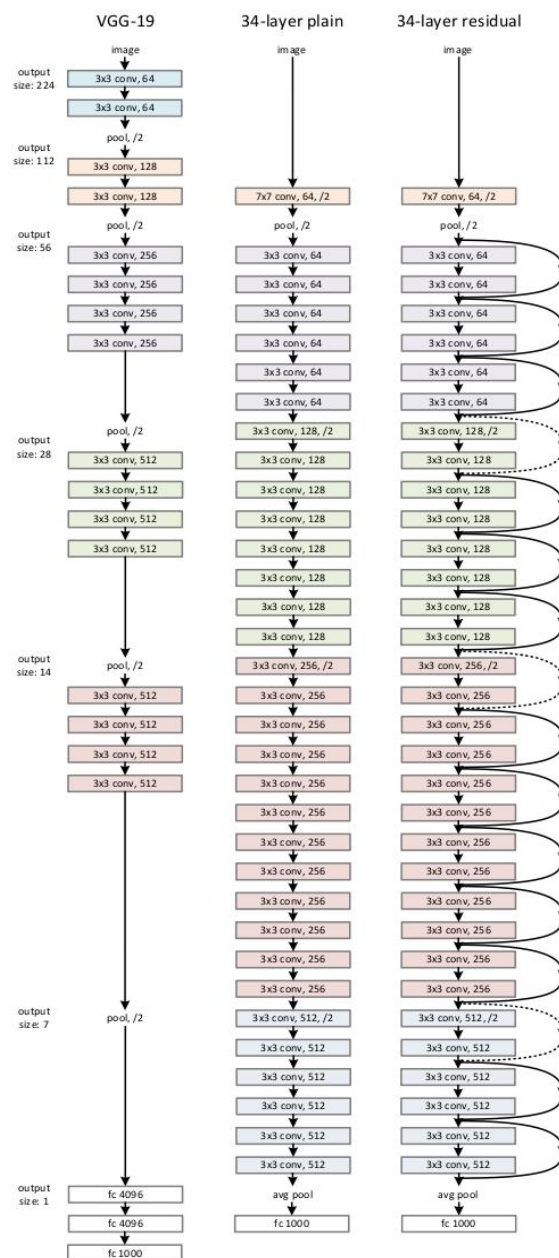


Figura 20: Ejemplos de arquitecturas para ImageNet [3]

- **Inception ResNet:** se trata de una arquitectura de red neuronal que combina propiedades de las arquitecturas Inception y ResNet, obteniendo así ventajas procedentes de ambas. Esta arquitectura incluye las conexiones residuales características del modelo de ResNet, permitiendo así la retropropagación del gradiente sin degradarse. Además, Inception-ResNet incorpora *bloques Inception*, estructuras que realizan convoluciones con diferentes tamaños de *kernel* en paralelo y más tarde combinan las salidas. La arquitectura de Inception fue creada en 2014 por Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke y Alex Alemi, resultando ganadora del ImageNet Large Scale Visual Recognition Challenge (ILSVRC). En la figura 21 se puede observar un ejemplo de esquema general en el que se aplican este tipo de bloques. Dicha figura está extraída del paper de Inception-V4 y Inception-ResNet [37] que se desarrolló tras el éxito de ResNet en 2015.

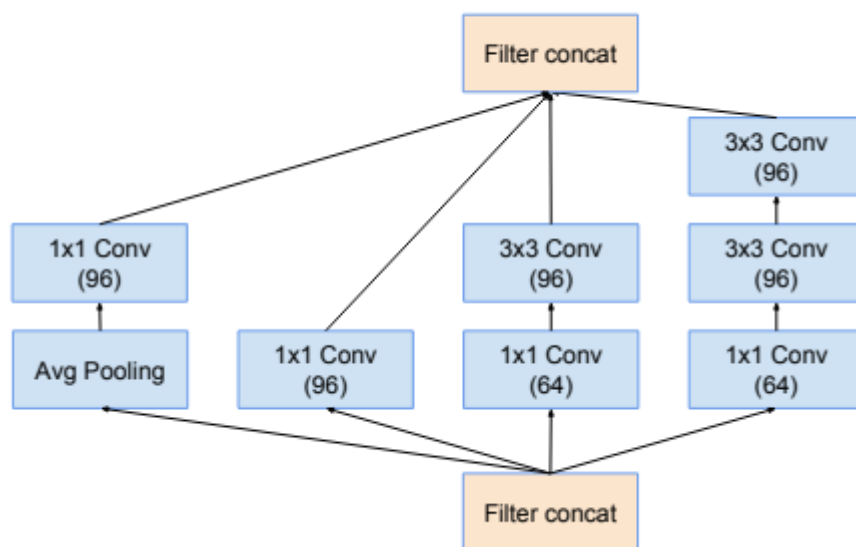


Figura 21: Bloque Inception-A perteneciente a la red Inception-v4 [37]

La idea principal del uso de los bloques Inception es reducir el coste computacional y mejorar la eficiencia del uso de la estructura local de las imágenes. A su vez, esto permite que la red capture características a múltiples escalas de la imagen de entrada, desde pequeños detalles hasta características más globales, mejorando así la eficiencia en la detección de patrones de distinta naturaleza.

La combinación de las conexiones residuales (*shortcuts*) heredadas de la arquitectura ResNet y los bloques o módulos Inception propios de la arquitectura con el mismo nombre se puede observar en un esquema correspondiente a un fragmento de la red Inception-ResNet-v2 en la figura 22.

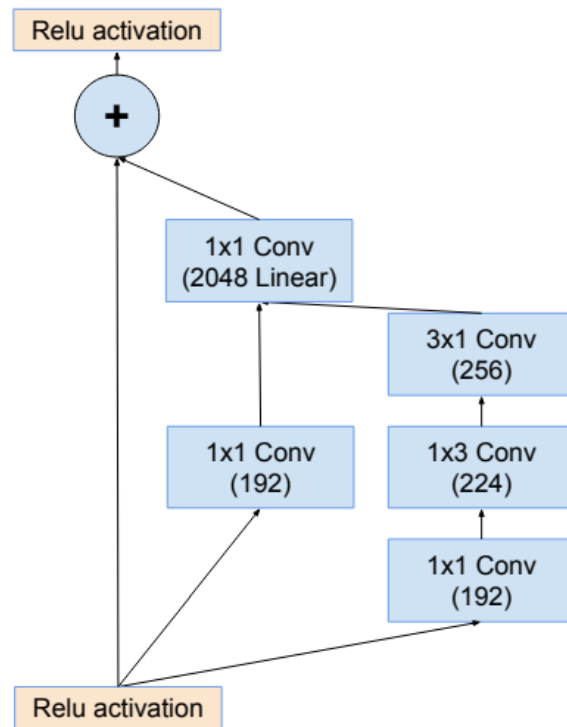


Figura 22: Esquema de módulo Inception-ResNet-C de la red Inception-ResNet-v2 [37]

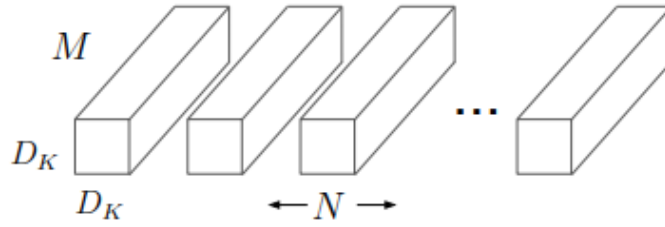
- **MobileNet:** se trata de una arquitectura de red neuronal convolucional especialmente diseñada para funcionar en dispositivos con recursos limitados, como es el caso de los teléfonos móviles y dispositivos embebidos, destacando así por su eficiencia y alto rendimiento. Esta arquitectura fue desarrollada por Google, y una de las principales características que le atribuye esta eficiencia es el uso de convoluciones depthwise separables. Este proceso permite factorizar una convolución tradicional en la aplicación inicial de una convolución depthwise (convolución separada en profundidad) y a continuación otra de tipo pointwise (convolución 1x1 en profundidad), reduciendo así la cantidad de operaciones y parámetros necesarios.

Se puede observar en la figura 22 un esquema de la comparación de la aplicación de la convolución tradicional con el proceso de doble convolución que da lugar a la convolución depthwise separable. De esta forma, se obtiene una reducción de coste computacional de:

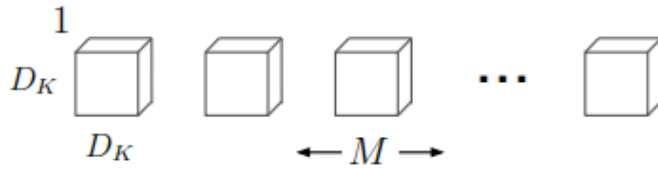
$$\frac{D_K \cdot D_K \cdot M \cdot D_F \cdot D_F + M \cdot N \cdot D_F \cdot D_F}{D_K \cdot D_K \cdot M \cdot N \cdot D_F \cdot D_F} \quad (1)$$

$$= \frac{1}{N} + \frac{1}{D_K^2}$$

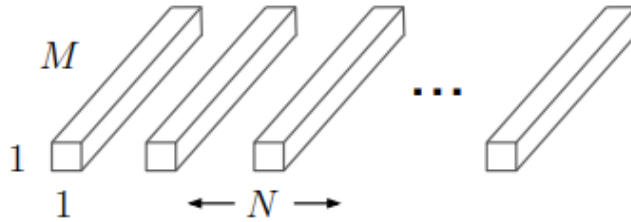
... donde $D_F \cdot D_F$ hace referencia a la dimensión de los mapas de características (*feature maps*).



(a) Standard Convolution Filters



(b) Depthwise Convolutional Filters



(c) 1×1 Convolutional Filters called Pointwise Convolution in the context of Depthwise Separable Convolution

Figura 23: Reemplazo de aplicación de convolución tradicional (a) por convolución depthwise (b) seguido de convolución pointwise ©[54]

5. IMPLEMENTACIÓN

5.1. Software y hardware empleado

Como plataforma de trabajo para la creación de la red neuronal convolucional se ha utilizado **Google Colaboratory**, se trata de un producto de Google Research que permite escribir y ejecutar código en Python sin necesidad de instalar nada en el ordenador. Así, esta plataforma es un servicio de notebooks de Jupyter que proporciona acceso gratuito a recursos computacionales, utilizándose así en este caso T4 GPU para ejecutar Python. La principal desventaja de Google Colab es que sus recursos están limitados y no están garantizados. Además, sus límites de uso no son constantes, sino que fluctúan de manera indeterminada.

Además, se utiliza la biblioteca de **Tensorflow**, una biblioteca de código abierto para aprendizaje automático desarrollada por Google en 2015. Para la construcción y entrenamiento de las redes neuronales convolucionales se ha utilizado sobre Tensorflow la biblioteca **Keras**, una API de redes neuronales de alto nivel escrita en Python. Esta API destaca por su sencillez de uso, facilitando así la creación de la red neuronal. A su vez, Keras permite ejecutar los modelos tanto en TPU como en GPU, por lo que proporciona una mayor rapidez en el entrenamiento de las redes neuronales. Adicionalmente, se ha decidido utilizar estas herramientas para el desarrollo de este proyecto debido a la documentación, información y ejemplos disponibles respecto a las mismas.

También se hace uso de la librería **os**, cuya utilidad principal se basa en poder interactuar con el sistema operativo del ordenador donde se ejecuta el código, permitiendo así crear y manejar archivos y directorios fácilmente. A su vez, para la realización de operaciones de alto nivel con archivos tales como la copia o eliminación de los mismos, también se utiliza la librería **shutil**. Por otro lado, se importa también la librería **matplotlib**, utilizada en este proyecto para la representación de las curvas de precisión (*accuracy*) y pérdida (*loss*) del entrenamiento y validación de los modelos de redes neuronales.

Por último, cabe indicar que para la realización de este proyecto se ha utilizado un ordenador portátil con procesador AMD Ryzen 5 5600H con Radeon Graphics 3.30 GHz, RAM 16,0 GB con GPU NVIDIA GeForce RTX 3050.

5.2. Hiperparámetros

A continuación se aporta una breve explicación de los principales hiperparámetros utilizados durante la realización de pruebas con los modelos de redes neuronales convolucionales.

Learning rate/tasa de aprendizaje: define el tamaño de los pasos que se toman en el ajuste de los pesos de las conexiones de la red neuronal durante el entrenamiento. Si este es excesivamente grande, puede conducir a un modelo inestable, mientras que si es demasiado pequeño el modelo tardará mucho en converger.

Epochs/épocas: hace referencia al número de veces que el modelo revisará los datos durante el entrenamiento.

Arquitectura de la red: indica la estructura, tipo, cantidad y conexiones de las capas de la red. Se realizan pruebas utilizando la versión 2 de ResNet de 152 capas, la versión 2 de InceptionResNet y la versión 2 de MobileNet.

Función de pérdida: determina la medida de la diferencia entre las predicciones del modelo y las etiquetas reales durante el entrenamiento. Durante las pruebas, se han utilizado *binary_crossentropy* y *categorical_crossentropy*. La primera se calcula mediante la fórmula (2) y la segunda mediante la fórmula (3). La entropía cruzada aumenta a medida que la probabilidad predicha de una muestra diverge del valor real.

$$-\frac{1}{n} \sum_{j=1}^n \sum_{i=1}^c [y_i \log(p_i) + (1 - y_i) \log(1 - p_i)] \quad (2)$$

$$CE = -\frac{1}{n} \sum_{j=1}^n \sum_{i=1}^c y_i \log \hat{y}_i \quad (3)$$

... donde:

y : clase a predecir

p_i : probabilidad predicha para la clase

c : número de clases

n : número de ejemplos

i : la clase

y_i = la clase real

$\hat{y}_i$ = la clase predicha

Optimizador: determina el algoritmo a aplicar para ajustar los pesos de la red en base a la función de pérdida y tasa de aprendizaje. Durante las pruebas, se ha utilizado el optimizador Adam, RMSprop y SGD. Tanto el optimizador RMSprop como SGD se basan directamente en la retropropagación del gradiente para calcular los nuevos pesos, siendo **SGD (Stochastic Gradient Descent)** el método más básico y sencillo de los tres probados. Este último utiliza la fórmula (4), donde el *learning rate* es fijo, es decir, no cambia durante el entrenamiento.

$$W = W - \alpha * \Delta W \quad (4)$$

... donde:

W -> Matriz de pesos (parámetros)

ΔW -> Derivada de la matriz de pesos

α -> *learning rate*

Por otro lado, **RMSprop (Root Mean Square Propagation)** utiliza una tasa de aprendizaje adaptativa en lugar de interpretarla como un hiperparámetro. De esta manera, el *learning rate* va cambiando durante el tiempo de entrenamiento, para lo que utiliza la fórmula (5).

$$\begin{aligned} C &= C + \Delta W^2 \\ W &= W - \frac{\alpha * \Delta W}{\sqrt{C + \epsilon}} \end{aligned} \quad (5)$$

... donde:

C -> "caché" que mantiene la suma de los cuadrados de los gradientes (derivadas) por cada parámetro

d -> *decaying rate* o tasa de descomposición/decadencia

Por último, el optimizador **Adam (Adaptive Moment Estimation)** es de los métodos de optimización más utilizados en *Machine Learning* en la actualidad. Este método usa como base RMSprop, al que le agrega *momentum* o momento, un término que toma en cuenta el valor de actualización del *learning rate* de la iteración anterior. Su cálculo se realiza mediante (6).

$$\begin{aligned} m &= \beta_1 * m + (1 - \beta_1) * \Delta W \\ v &= \beta_2 * v + (1 - \beta_2) * \Delta W^2 \\ W &= W - \frac{\alpha * m}{\sqrt{v + \epsilon}} \end{aligned}$$

(6)

... donde:

m -> momento que modela la media de los gradientes a lo largo del tiempo

v -> momento que modela la varianza de los gradientes a lo largo del tiempo

β_1 -> generalmente igual a 0.9

β_2 -> generalmente igual a 0.99

Función de activación: determina las operaciones matemáticas utilizadas para calcular la salida de las neuronas de cada capa. Algunas de las funciones de activación más comunes en redes neuronales convolucionales son **ReLU (Rectified Lineal Unit)**, **Softmax y Sigmoid**. La función Sigmoid transforma los valores introducidos a escala (0,1), tendiendo los valores altos a 1 de manera asintótica y los muy bajos a 0. Esta función de activación se calcula mediante la fórmula (7). De la misma forma que Softmax, presenta buen comportamiento en la última capa de la red. Esta última función transforma los valores en probabilidades de tal manera que el sumatorio de las probabilidades de las salidas es 1. Esta función se calcula mediante la fórmula (8).

$$f(x) = \frac{1}{1 - e^{-x}} \quad (7)$$

$$f(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (8)$$

Por último, la función de activación **ReLU** anula los valores negativos, dejando intactos los positivos. De esta forma, se define mediante la expresión (9).

$$f(x) = \max(0, x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases} \quad (9)$$

6. PRUEBAS Y RESULTADOS TÉCNICAS DE PROMPTING

En este apartado se muestran las pruebas realizadas con diversas IAs generadoras de imágenes junto a sus *prompts*, se analizan y comparan los resultados con distintas estructuras y características de *prompts* y diferentes tipos de imágenes a obtener. Se decide trabajar con *Playground AI*, *Lexica* y *Clipdrop*. De este modo, se comparan los resultados de una IA de pago con dos generadores con versión gratuita de 1000 imágenes diarias la primera y 400 la última.

Se decide comenzar las pruebas utilizando el estilo anime de Naoshi Arakawa debido a que las IAs generadoras tienden a presentar dificultades con dicho estilo. Se inicia con el siguiente *prompt* en español: **"ilustracion de un pomerania leyendo manga estilo Naoshi Arakawa"** (*Prompt 1*). Como se puede observar, se trata de un *prompt* sencillo en el que se nombra exclusivamente el tipo de obra, el sujeto principal de la imagen y la actividad que se encuentra realizando y el estilo. Se puede observar en la figura 23 cómo en el caso del *Prompt 1* *Lexica* no ha seguido las instrucciones respecto al estilo indicado, mientras que aunque las IAs restantes han mantenido la estética de gráficos japoneses, *Playground* ha mostrado una imagen más de carácter manga que de anime. Por lo tanto, la imagen más cercana al resultado deseado lo ha presentado *Clipdrop*.



Figura 24: Imágenes generadas por *Playground AI* (izq), *Clipdrop* (center) y *Lexica* (dcha) con *Prompt 1*, Tema 1

Posteriormente, se decide realizar una prueba introduciendo el *prompt* traducido al inglés: **"pomeranian illustration reading manga in Naoshi Arakawa style"** (*Prompt 2*). Se obtienen los resultados mostrados en la figura 24. Como se puede observar, en el caso de *Lexica* el resultado se presenta notablemente más coherente con el estilo indicado. En el caso de *Playground AI*, se ha generado una imagen con un trazado más similar al del anime de Naoshi Arakawa, aunque el color continúa sin ser adecuado. Por último, en este caso *Clipdrop* se ha alejado del estilo buscado, ya que no se asemeja al estilo japonés en

ningún aspecto, aunque muestra una composición interesante. Como se puede observar, el resultado continúa con una alta variación entre las diferentes IAs, principalmente debido a la exactitud del prompt.



Figura 25: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 2, Tema 1

A continuación, se modifica el prompt indicando el estilo de la siguiente forma: "**((Naoshi Arakawa-style illustration)) of a pomeranian reading manga**" (Prompt 3). Como se puede observar en la figura 26, la imagen generada por Playground AI posee mayor definición y similitud con el estilo indicado. En el caso de Clipdrop se trata de una imagen más adecuada al resultado buscado en comparación a la iteración anterior, aunque aún faltan características visuales del anime tales como el delineado de la imagen. Por último, en el caso de Lexica el estilo obtenido se ha desviado completamente del deseado, resultando incluso más similar a un estilo chino que japonés. Como conclusión, se puede deducir que la evolución del prompt ha hecho converger la composición de las imágenes, pero el estilo de las mismas se desarrolla muy lentamente.



Figura 26: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 3, Tema 1

Se propone a continuación un par de nuevas estructuras en las que se separan las características propuestas anteriormente por comas: **"illustration, pomeranian reading manga, Naoshi Arakawa style"** (Prompt 4); **"illustration, pomeranian reading manga, Naoshi Arakawa, anime style"** (Prompt 5). Como se puede observar en las figuras 26 y 27, esta nueva estructura presenta una mayor eficiencia respecto a los prompts anteriores, generando tanto en el caso de Playground AI como en el de Clipdrop imágenes con gama de color más adecuado y línea más semejante a la del anime de Naoshi Arakawa. Además, agregando explícitamente el estilo anime además de Naoshi Arakawa en el prompt 5, como se puede observar en la figura 27 se generan imágenes con mayor definición y más adecuadas al estilo deseado. Por otro lado, en el caso de Lexica la imagen obtenida no llega a converger en el estilo japonés, pudiendo deducirse que no se trata de un tipo de ilustraciones con las que se ha entrenado al modelo generador.



Figura 27: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 4, Tema 1



Figura 28: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 5, Tema 1

Se procede a continuación a realizar diversas pruebas con una estructura más detallada, definida por categorías y separada por comas: **"VISUAL STYLE: Anime style, GENRE: Illustration, SUBJECT(S): Cute pomeranian dog reading manga at the library, TIME PERIOD: Contemporary, COLOR: Ink, ASPECT RATIO: 16:9, FORMAT: Digital, FRAME SIZE: Full screen, LENS SIZE: Standard, COMPOSITION: Centered, LIGHTING: Bright, LIGHTING TYPE: Flares, TIME OF DAY: Daytime, ENVIRONMENT: Interior, LOCATION TYPE: Library, SET: Traditional library, CAMERA: Standard digital camera, LENS: Standard lens, FILM STOCK / RESOLUTION: High detail, TAGS: Cute, Pomeranian, Dog, Manga, Library, Anime style, Naoshi Arakawa"** (Prompt 6). Como se puede observar en la figura 28, al tratarse de un prompt más específico respecto a iluminación, color y composición, los resultados en las tres IAs generadoras se presentan mucho más homogéneos y similares al resultado deseado. Aunque Lexica ha generado la imagen más adecuada respecto a todas sus iteraciones anteriores, continúa alejada del estilo anime que se busca mostrando excesiva saturación. Por otro lado, tanto Playground AI como Clipdrop han generado resultados adecuados y con una estética similar al anime producido por Naoshi Arakawa, siendo la más precisa tanto en color como en iluminación la imagen generada por Clipdrop.



Figura 29: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 6, Tema 1

Se modifica a continuación el prompt anterior para obtener un escenario similar pero con estilo fotorrealista: **"VISUAL STYLE: Photorealistic style, GENRE: Photography, SUBJECT(S): Cute pomeranian dog reading manga at the library, TIME PERIOD: Contemporary, COLOR: blue, ASPECT RATIO: 16:9, FORMAT: Digital, FRAME SIZE: Full screen, LENS SIZE: Standard, COMPOSITION: Centered, LIGHTING: Bright, LIGHTING TYPE: Flares, TIME OF DAY: Daytime, Sunset ENVIRONMENT: Interior, LOCATION TYPE: Library, SET: Traditional library, CAMERA: Standard digital camera, LENS: Standard lens, FILM STOCK / RESOLUTION: High detail, TAGS: Cute, Pomeranian, Dog, Photography, Library"** (Prompt 7).

Como se puede observar en la figura 29, las imágenes generadas por las 3 IAs se presentan notablemente más precisas y definidas, pudiéndose deducir así que en su entrenamiento se les ha proporcionado mayor número de fotografías que de ilustraciones. Destaca en las generaciones de Playground y Clipdrop la inclusión de las gafas como elemento que aporta realismo a la situación. Además, la inclusión del color azul como parámetro *COLOR*, se puede observar cómo estas IAs lo han incluido en el color de los libros y Lexica en los reflejos del pomerania. Por lo tanto, aunque Lexica ha generado imágenes más adecuadas que en el estilo anterior, continúa presentando la imagen menos adaptada al prompt.



Figura 30: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 7, Tema 1

A continuación, se procede a la realización de pruebas en un área temática diferente: se solicita a las IAs la representación de comida vegana (*vegan food*) en una cafetería de estilo americano de la década de los 50s localizada en japon, con imagen estilo fotorrealista. El prompt utilizado es el siguiente: **"VISUAL STYLE: Photorealistic style, GENRE: Photography, SUBJECT(S): vegan food, TIME PERIOD: 50s, COLOR: green, red, ASPECT RATIO: 16:9, FORMAT: Digital, FRAME SIZE: Full screen, LENS SIZE: Standard, COMPOSITION: Centered, LIGHTING: Bright, LIGHTING TYPE: Flares, TIME OF DAY: Daytime, Sunset ENVIRONMENT: Interior, LOCATION TYPE: japan, SET: american café, CAMERA: Standard digital camera, LENS: Standard lens, FILM STOCK / RESOLUTION: High detail, TAGS: vegan food, 50s, japan, american cafe, Photography, render, 4k, unreal engine 5"** (Prompt 8). Así, como se puede observar en la figura 30, las 3 IAs generadoras han cumplido los parámetros de *COLOR* de verde y rojo además de incluir como sujeto principal la comida vegana. Sin embargo, cada una de las IAs ha abordado las instrucciones de diferente forma: Lexica ha tenido más en cuenta que las demás la composición central indicada, enfocándose así en mayor medida en la comida como sujeto principal de la imagen, con un fondo de un entorno que se asemeja a un restaurante japonés y destacando de nuevo por su saturación.

Por otro lado, Playground AI y Clipdrop crean un entorno completo que tiene como protagonista la comida, Clipdrop adaptándose más a la estética de la década de los 50s especificada en el prompt y Playground AI desarrollando más el concepto general de comida vegana en una cafetería de estilo americano. Se refleja, por tanto, la eficiencia de la especificidad del prompt en los tres resultados obtenidos, ya que las tres generadoras han abordado las instrucciones de manera correcta.

Para la siguiente prueba se decide mantener el mismo prompt (**Prompt 8**) para experimentar con los *negative prompts*, es decir, factores a evitar en la imagen a generar. Se indican como **negative prompts** "*blur*" (emborronamiento) y "*tomatoes*" (tomates) para obtener una imagen nítida y con una fuente de color rojo diferente a los tomates, ya que abundan en la generación anterior. Se presenta el resultado en la figura 30, donde se puede observar que tanto Playground AI como Clipdrop hacen especial hincapié en representar la década de los 50s indicada en *TIME PERIOD*, ajustándose mejor la propuesta de Playground AI respecto a los *negative prompts*. Lexica ha resuelto de manera más adecuada la representación de comida como sujeto principal de la imagen cumpliendo con la composición y evitando los tomates, aunque representa en menor medida todos los parámetros de contexto y entorno.



Figura 31: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 8, Tema 2



Figura 32: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 8 + negative prompts, Tema 2

Para la última prueba de esta temática se decide mantener los *prompts negativos* del caso anterior más *wood (madera)* y cambiar algunas de las características del prompt para generar una imagen con estilo futurista y fotorrealista, incluyendo en la localización una nave espacial con carácter árabe, manteniendo aún en la escena como principal sujeto la comida vegana. El prompt utilizado es el siguiente: **"VISUAL STYLE: Photorealistic style, GENRE: Photography, SUBJECT(S): vegan food, TIME PERIOD: futuristic, COLOR: green, blue, ASPECT RATIO: 16:9, FORMAT: Digital, FRAME SIZE: Full screen, LENS SIZE: Standard, COMPOSITION: Centered, LIGHTING: Bright, LIGHTING TYPE: Flares, TIME OF DAY: Daytime, Sunset ENVIRONMENT: Interior, LOCATION TYPE: Space Ship, SET: Arabian Stargate SpaceShip, CAMERA: Standard digital camera, LENS: Standard lens, FILM STOCK / RESOLUTION: High detail, TAGS: vegan food, SpaceShip, Stargate, Arabian, Photography, render, 4k, unreal engine 5"** (Prompt 9). Como se puede observar en la figura 32, las tres generadoras cumplen con los parámetros de *COLOR* proporcionados. Playground Ai y Clipdrop han interpretado las instrucciones dadas de manera relativamente similar, ambas han generado una cafetería de carácter retro, quizás influidas por la generación anterior, pero con estética futurista y espacial. Sin embargo, Lexica se ha ceñido en mayor medida a la composición central solicitada situando una cúpula con un arco ojival en mitad de un paisaje semejante al paisaje lunar. Aunque esta última ha cumplido parámetros indicados que las demás generadoras no, la imagen resultante presenta más incoherencias que las demás.

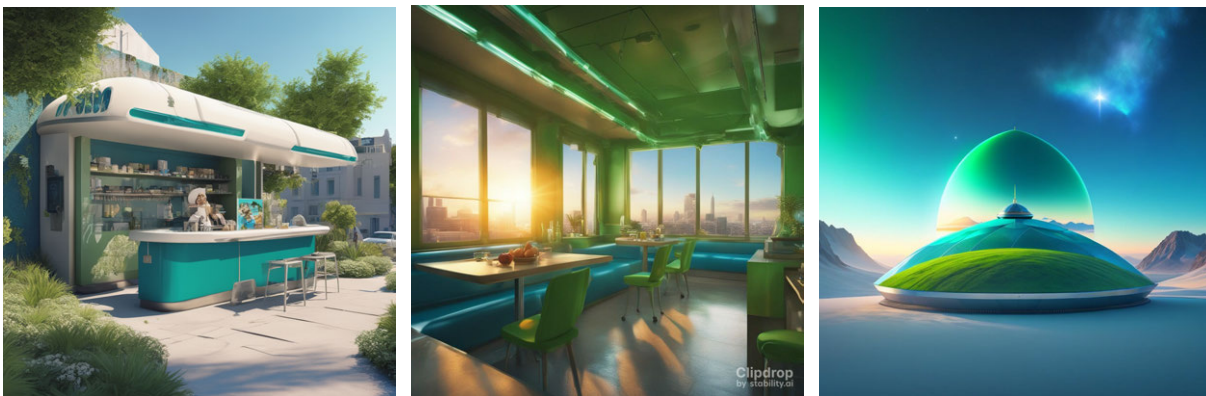


Figura 33: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 9, Tema 2

A continuación, se decide abordar la temática acuática con tres prompts diferentes con la finalidad de analizar el comportamiento de las IAs generadoras a la hora de crear escenas ambientadas en este elemento. Se comienza con un prompt de elevada especificidad con la finalidad de visualizar una escena difícilmente imaginable por el ser humano. Para ello, se solicita a las IAs la recreación de la escena de un submarino pequeño situado cerca del titanic en el momento previo a su implosión con el siguiente prompt: **"VISUAL STYLE: Photorealistic style, GENRE: Photography, SUBJECT(S): small submarine next to the titanic before exploding, TIME PERIOD: Contemporary, COLOR: blue, ASPECT RATIO: 16:9, FORMAT: Digital, FRAME SIZE: Full screen, LENS SIZE: Standard, COMPOSITION: right, LIGHTING: depths of the ocean, LIGHTING TYPE: Flares, Light shaft, TIME OF DAY: Daytime, ENVIRONMENT: Ocean, LOCATION TYPE: Ocean depths next to the Titanic, SET: submarine interior, CAMERA: Standard digital camera, LENS: Standard lens, FILM STOCK / RESOLUTION: High detail, TAGS: Small submarine, Titanic, Ocean depths, Photography, render, 4k"** (Prompt 10). Como se puede observar en la figura 33, los resultados obtenidos por Playground son los que mayor incoherencia han presentado llegando incluso a presentar varias líneas de horizonte, aunque ninguna de las tres IAs ha generado una imagen que se adapte a la descripción dada, por lo que puede deberse al exceso de especificidad del prompt.

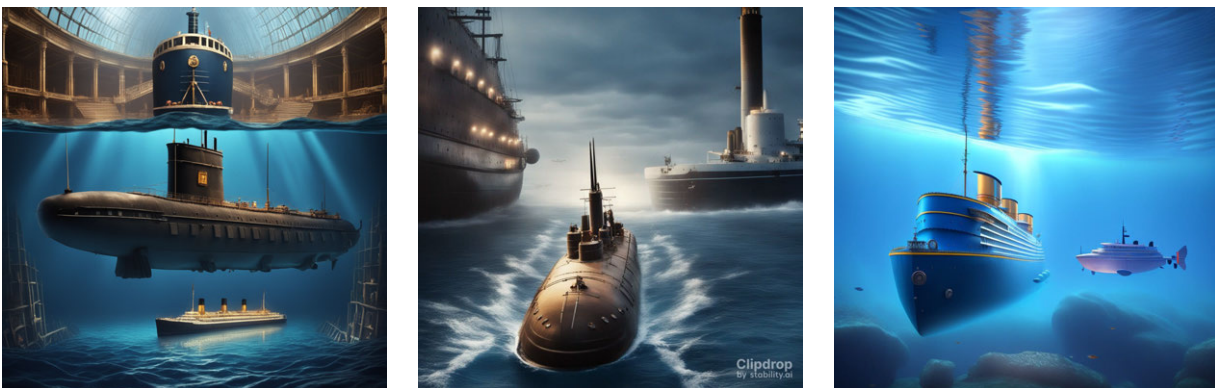


Figura 34: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 10, Tema 3

Teniendo en cuenta los resultados de las pruebas anteriores, se decide generar otro prompt en la misma línea de temática acuática **"VISUAL STYLE: Photorealistic style, GENRE: Photography, SUBJECT(S): mermaid entering a cave, TIME PERIOD: medieval, COLOR: blue, ASPECT RATIO: 16:9, FORMAT: Digital, FRAME SIZE: Full screen, LENS SIZE: Standard, COMPOSITION: centered, LIGHTING: depths of the ocean, LIGHTING TYPE: Flares, Light shaft, TIME OF DAY: nighttime, ENVIRONMENT: Ocean, LOCATION TYPE: Ocean, SET: Ocean Caves, CAMERA: Standard digital camera, LENS: Standard lens, FILM STOCK / RESOLUTION: High detail, TAGS: mermaid, cave, Ocean, Photography, render, 4k, unreal Engine 5"** (Prompt 11).

Este segundo prompt de temática acuática solicita como sujeto principal una sirena entrando en una cueva en el océano con estilo fotorrealista, cuya estructura general se cumple en las imágenes generadas por las 3 IAs. Sin embargo, se puede observar que el concepto de sirena lo ha representado de manera más adecuada Playground AI, aunque Clipdrop propone una silueta cercana al concepto que se tiene de sirena. Por otro lado, cabe destacar la dificultad que continúan presentando en referencia a la línea de horizonte (en caso de haberla) y la colocación de los elementos de la escena cuando se sitúan en el mar, ya que no queda clara la localización respecto a la superficie.



Figura 35: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 11, Tema 3

Se realiza otra prueba con temática acuática indicando como sujeto principal una sirena entrando en un palacio árabe y como localización las profundidades del océano mediante el siguiente prompt: **"VISUAL STYLE: Photorealistic style, GENRE: Photography, SUBJECT(S): mermaid entering an arab palace, TIME PERIOD: medieval, COLOR: blue, ASPECT RATIO: 16:9, FORMAT: Digital, FRAME SIZE: Full screen, LENS SIZE: Standard, COMPOSITION: centered, LIGHTING: depths of the ocean, LIGHTING TYPE: Flares, Light shaft, TIME OF DAY: Daytime, ENVIRONMENT: Ocean depths, LOCATION TYPE: Ocean depths, SET: arab palace in the depths of the ocean, CAMERA: Standard digital camera, LENS: Standard lens, FILM STOCK / RESOLUTION: High detail, TAGS: mermaid, arabian palace, Ocean depths, Photography, render, 4k, unreal Engine 5"** (Prompt 12).

Se puede verificar, por tanto, las conclusiones de la prueba anterior observando la figura 35, ya que aunque se han generado resultados coherentes y composiciones interesantes, no se han seguido ninguna de las indicaciones del prompt relacionadas con el agua. Como conclusión, se puede deducir que las IAs generadoras presentan dificultades en cuanto se solicitan escenarios submarinos, perdiendo así la coherencia en los mismos o llegando incluso a evitarlos en su generación.



Figura 36: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 12, Tema 3

A continuación se aborda una última temática en los prompts y se solicita a las IAs generar una imagen con estilo similar al de Van Gogh una casa típica de la campiña inglesa en la región de The Costwolds con composición hacia la derecha, mediante el siguiente prompt: "**VISUAL STYLE: Van Gogh, GENRE: Van Gogh, SUBJECT(S): huge cottage in the costwolds, TIME PERIOD: XIX century, COLOR: warm, ASPECT RATIO: 16:9, FORMAT: Digital, FRAME SIZE: Full screen, LENS SIZE: Standard, COMPOSITION: right, LIGHTING: bright, LIGHTING TYPE: Flares, Light shaft, TIME OF DAY: Daytime, ENVIRONMENT: countryside LOCATION TYPE: the costwolds, SET: cottage, CAMERA: Standard digital camera, LENS: Standard lens, FILM STOCK / RESOLUTION: High detail, TAGS: cottage, countryside, the costwolds, Van Gogh**" (Prompt 13). Como se puede observar en la figura 36, se ha obtenido un resultado homogéneo entre las generaciones de las 3 IAs, con iluminación, composición y estilos similares y semejantes a las instrucciones del prompt. Aunque Lexica continúa mostrando la imagen con excesiva saturación, cumple el resto de parámetros solicitados de manera similar a Playground AI y Clipdrop.



Figura 37: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 13, Tema 4

Por último, se decide aumentar la especificidad sobre el concepto del prompt anterior para atribuir a la imagen a generar el estilo del cuadro de *La noche estrellada*. **"VISUAL STYLE: the starry night, GENRE: the starry night, SUBJECT(S): enormous cottage in the costwolds, TIME PERIOD: XIX century, COLOR: warm, ASPECT RATIO: 16:9, FORMAT: Digital, FRAME SIZE: Full screen, LENS SIZE: Standard, COMPOSITION: right, LIGHTING: bright, LIGHTING TYPE: Flares, Light shaft, TIME OF DAY: Night, ENVIRONMENT: countryside LOCATION TYPE: the costwolds, SET: cottage, CAMERA: Standard digital camera, LENS: Standard lens, FILM STOCK / RESOLUTION: High detail, TAGS: cottage, countryside, the costwolds, Van Gogh"** (Prompt 14). Como se puede observar en la figura 37, los resultados presentados por Playground AI y Clipdrop se asemejan en gran medida a las instrucciones indicadas en el prompt, siendo la imagen generada por Clipdrop la que más se adapta al mismo. Por otro lado, aunque Lexica no ha llegado a adaptarse al estilo sí cumple con las demás indicaciones en el prompt. Por lo tanto, las imágenes generadas por las 3 IAs son similares en referencia a composición y set de la escena.



Figura 38: Imágenes generadas por Playground AI (izq), Clipdrop (center) y Lexica (dcha) con Prompt 14, Tema 4

7. ENTRENAMIENTO Y RESULTADOS MODELOS DE CNN

En este apartado se muestran los resultados de *validation_accuracy* y *loss_accuracy* más relevantes de las pruebas realizadas. Durante dichas pruebas, se cambia la arquitectura de red neuronal convolucional (ResNet_v2_152, InceptionResNet_v2, MobileNet_v2), se juega con diferentes parámetros de *data augmentation* y *learning rates*, además de probar diferentes optimizadores. Para observar los resultados de *validation_accuracy* y *loss_accuracy* correspondientes a otras pruebas realizadas junto a los parámetros y arquitecturas utilizadas, consultar el Anexo A.

En primer lugar, se presenta la *validation_accuracy* y *loss_accuracy* en la figura 38 de una prueba realizada basada en la arquitectura ResNet versión 2 de 152 capas (https://tfhub.dev/google/imagenet/resnet_v2_152/feature_vector/5), utilizando el optimizador Adam con *learning rate* de 0.0001, función de pérdida *binary_crossentropy*, métrica *accuracy* y entrenamiento de 100 épocas (*epochs*). Se muestran los datos del generador de datos de *data augmentation* a continuación:

```
datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=40,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
```

Como se puede observar en la figura 38, durante las aproximadamente últimas 20 épocas la media de *accuracy* (precisión) tanto en los datos de validación (pruebas) como en los de entrenamiento se encuentra en torno al 78%, obteniendo una mayor oscilación en los resultados de validación. Este porcentaje de *accuracy* se puede considerar favorable debido a que ha sido obtenido con un total de 126 imágenes de dataset. Por otro lado, aunque la gráfica de *loss* (pérdida) es descendente tanto en el caso de validación como en el de entrenamiento, no logra bajar de una media de 0.50, presentando así margen de mejora en este aspecto.

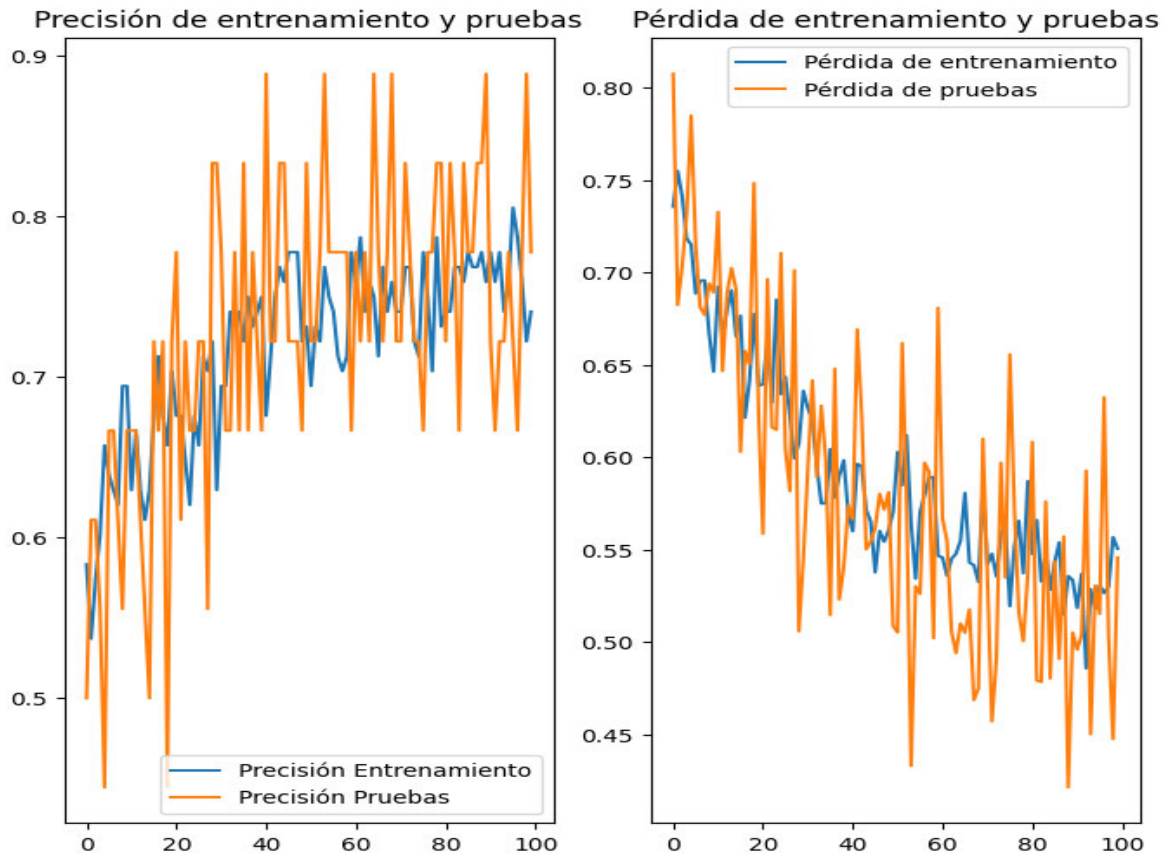


Figura 39: Prueba 1 ResNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)

En segundo lugar, se presenta la *validation_accuracy* y *loss_accuracy* en la figura 39 de una prueba realizada basada en la arquitectura ResNet versión 2 de 152 capas ("https://tfhub.dev/google/imagenet/resnet_v2_152/feature_vector/5", utilizando el optimizador RMSprop con *learning rate* de 0.0001, función de pérdida *categorical_crossentropy*, métrica *accuracy* y entrenamiento de 100 épocas (*epochs*). Se muestran los datos del generador de datos de *data augmentation* a continuación:

```
datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=35,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
```

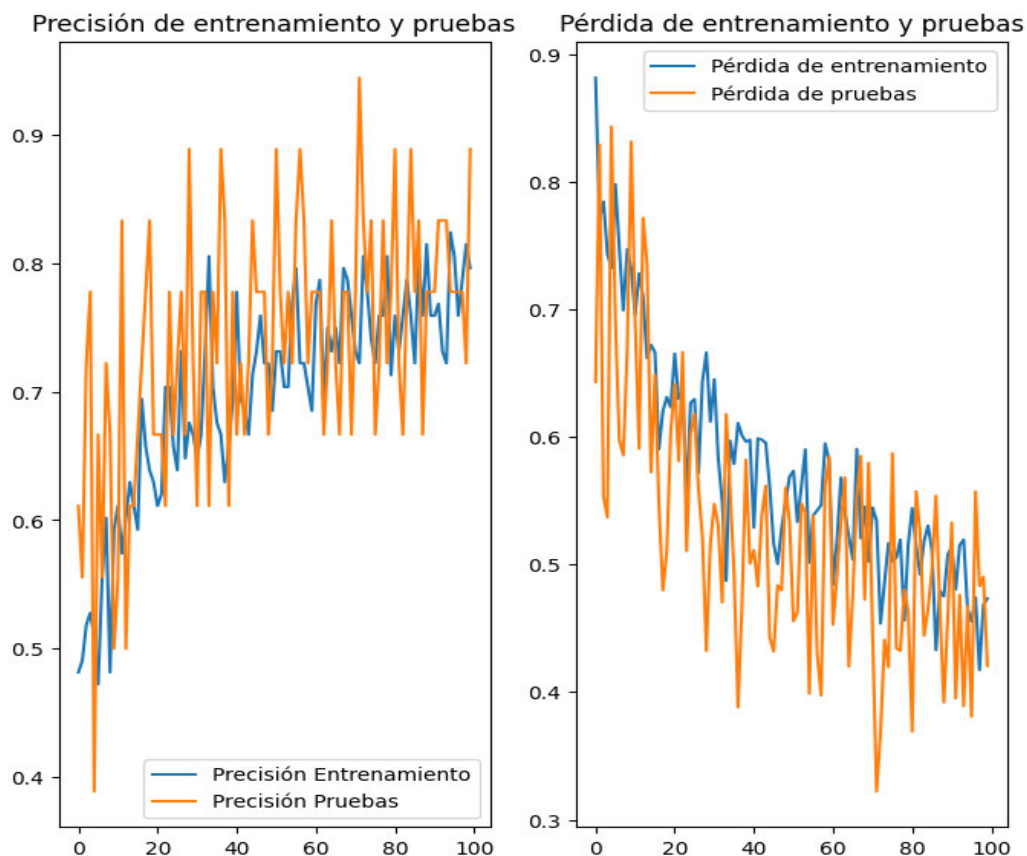


Figura 40: Prueba 2 ResNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)

Como se puede observar en la figura 39, durante las aproximadamente últimas 20 épocas la media de accuracy tanto en los datos de validación (pruebas) como en los de entrenamiento se encuentra en torno al 75%, con todos los resultados de las últimas 10 épocas superiores al 70%. De manera similar que el caso anterior, este porcentaje de accuracy se puede considerar favorable debido al reducido número de muestras utilizadas. Por otro lado, aunque la pérdida tanto de entrenamiento como de validación continúa presentando margen de mejora, la media en las últimas 20 épocas resulta inferior a la del caso anterior, con una media de aproximadamente 0,45.

A continuación, se procede a presentar los resultados de precisión (accuracy) y pérdida (loss) de entrenamiento (training) y validación (pruebas) pertenecientes a las pruebas realizadas con el modelo basado en la arquitectura *InceptionResNet* versión 2 (https://tfhub.dev/google/imagenet/inception_resnet_v2/feature_vector/5).

En primer lugar, se presenta la *validation_accuracy* y *loss_accuracy* en la figura 40 de la prueba realizada utilizando el optimizador RMSprop con *learning rate* de 0.0001, función de pérdida *categorical_crossentropy*, métrica *accuracy* y entrenamiento de 100 épocas (*epochs*). Se muestran los datos del generador de datos de *data augmentation* a continuación:

```
datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=35,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
```

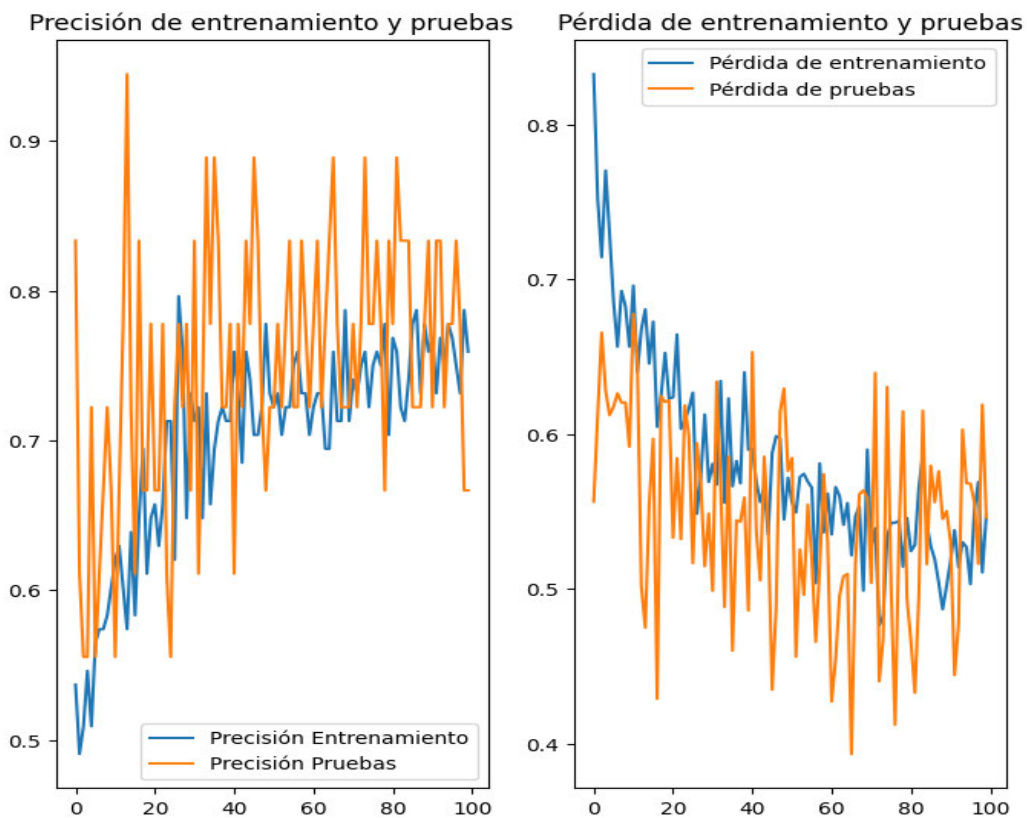


Figura 41: Prueba 1 Inception-ResNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)

Como se puede observar en la figura 40, durante las aproximadamente últimas 20 épocas la media de *accuracy* (precisión) tanto en los datos de validación (pruebas) como en los de entrenamiento se encuentra en torno al 75%, resultando así similar a los casos anteriores. Sin embargo, aunque la gráfica de *loss* (pérdida) continúa presentándose descendente tanto en el caso de validación como en el de entrenamiento, no logra bajar de una media de 0.50, resultando ligeramente peor curva que en los casos anteriores.

En segundo lugar, se presenta la *validation_accuracy* y *loss_accuracy* en la figura 41 de una prueba realizada utilizando el optimizador RMSprop con *learning rate* de 0.0001, función de pérdida *binary_crossentropy*, métrica *accuracy* y entrenamiento de 100 épocas (*epochs*). Se muestran los datos del generador de datos de *data augmentation* a continuación:

```
datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=40,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
```

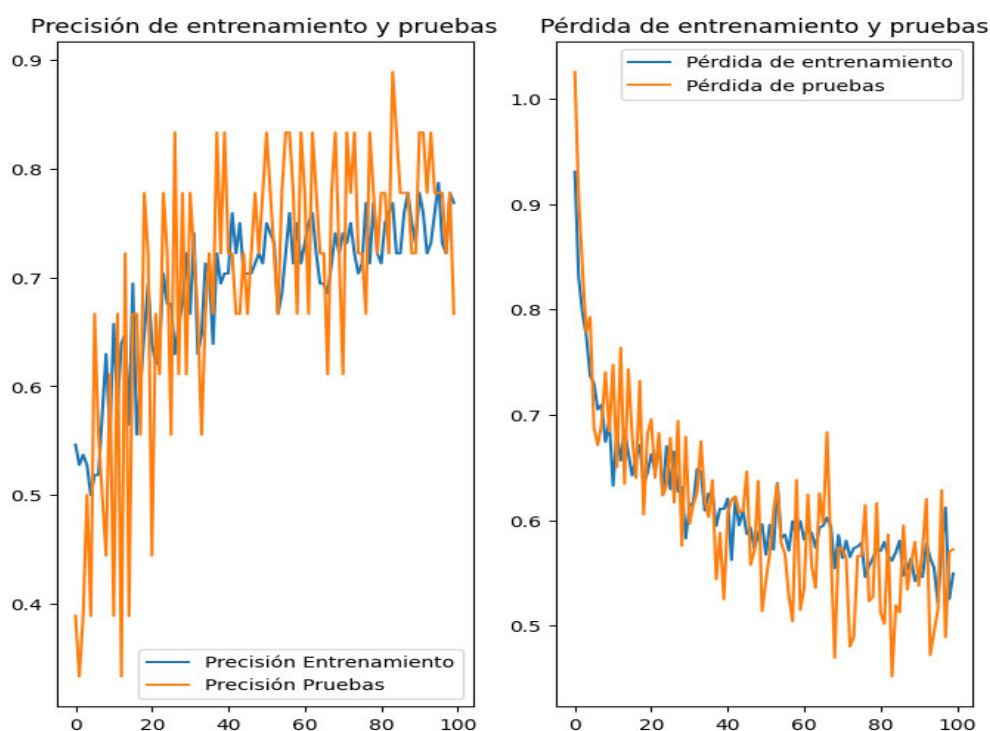


Figura 42: Prueba 2 Inception-ResNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)

Como se puede observar en la figura 41, durante las aproximadamente últimas 20 épocas la media de *accuracy* (precisión) tanto en los datos de validación (pruebas) como en los de entrenamiento se encuentra en torno al 75%, resultando así similar al caso anterior. Sin embargo, aunque la gráfica de *loss* (pérdida) continúa presentándose descendente tanto en el caso de validación como en el de entrenamiento, no logra bajar de una media de 0.55, presentando por tanto mayor pérdida que en los casos anteriores.

Posteriormente, se procede a presentar los resultados más relevantes de precisión (*accuracy*) y pérdida (*loss*) de entrenamiento (*training*) y validación (pruebas) pertenecientes a las pruebas realizadas con el modelo basado en la arquitectura *MobileNet* versión 2 (https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4).

En primer lugar, se presenta la *validation_accuracy* y *loss_accuracy* en la figura 42 de la prueba realizada utilizando el *binary_crossentropy*, métrica *accuracy* y entrenamiento de 100 épocas (*epochs*). Se muestran los datos del generador de datos de *data augmentation a continuación*.

```
datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=40,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
```

Como se puede observar en la figura 42, durante las aproximadamente últimas 20 épocas la media de *accuracy* (precisión) tanto en los datos de validación (pruebas) como en los de entrenamiento se encuentra en torno al 73%, resultando así ligeramente inferior a los casos anteriores. Además, aunque la gráfica de *loss* (pérdida) se presenta descendente tanto en el caso de validación como en el de entrenamiento, no logra bajar de una media de 0.55 en las últimas épocas, resultando de nuevo peor que las pruebas anteriores.

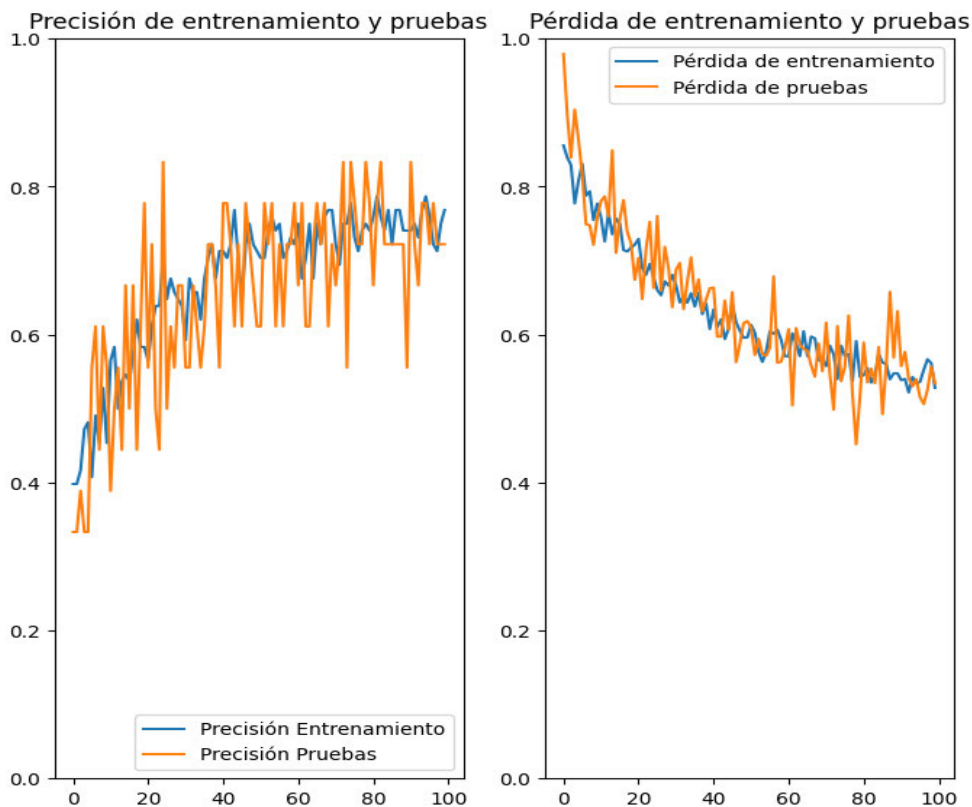


Figura 43: Prueba 1 MobileNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)

En segundo lugar, se presenta la *validation_accuracy* y *loss_accuracy* en la figura 43 de una prueba realizada utilizando el optimizador RMSprop con *learning rate* de 0.0001, función de pérdida *binary_crossentropy*, métrica *accuracy* y entrenamiento de 100 épocas (*epochs*). Se muestran los datos del generador de datos de *data augmentation* a continuación:

```
datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=40,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
```

Como se puede observar en la figura 43, durante las aproximadamente últimas 20 épocas la media de *accuracy* (precisión) en los datos de entrenamiento se encuentra en torno al 73%, resultando algo ligeramente mejores en las pruebas. Por otro lado, aunque la gráfica de *loss* (pérdida) se presenta descendente tanto en el caso de validación como en el de entrenamiento, no logra bajar de una media de 0.55 en las últimas épocas, resultando de nuevo peor que las pruebas anteriores.

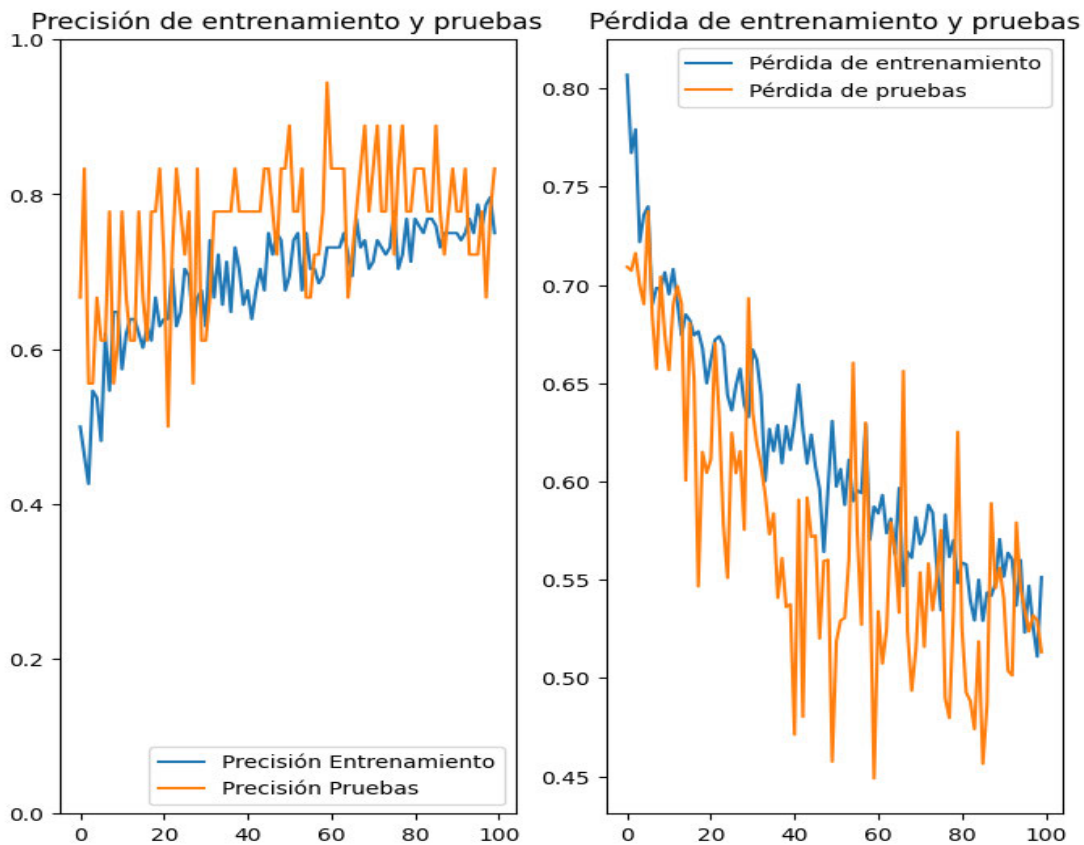


Figura 44: Prueba 2 MobileNet: Izquierda: Validation accuracy (pruebas) y training accuracy; Derecha: Pérdida de entrenamiento y validación (pruebas)

8. PRESUPUESTO

Para la realización de este proyecto no ha sido necesaria ninguna inversión monetaria.

9. IMPACTO DEL PROYECTO

Este proyecto presenta diversas implicaciones de salud, económicas y tecnológicas, aportando así a dos de los 17 ODS (Objetivos de Desarrollo Sostenible). En primer lugar, aporta al **Objetivo 3: Garantizar una vida sana y promover el bienestar para todos en todas las edades** ya que, como se ha indicado anteriormente en la memoria, la combinación del diagnóstico por IA junto a la supervisión procedente de un profesional no solo aumenta la velocidad y con ello la eficiencia del diagnóstico y tratamiento de una patología sino que se reduce la probabilidad de un diagnóstico erróneo, dando como resultado una valoración y revisión más exhaustiva y efectiva. Como consecuencia, este proyecto ayuda a reducir la tasa de mortalidad mundial y reforzar la capacidad de todos los países, en particular los países en desarrollo, en materia de alerta temprana, reducción de riesgos y gestión de los riesgos para la salud nacional y mundial

Además, este proyecto aporta también al **Objetivo 9: Construir infraestructuras resilientes, promover la industrialización sostenible y fomentar la innovación**, ya que el aumento de la eficiencia en *prompting* conduce a la obtención de los resultados deseados de una manera más rápida y eficiente, suponiendo esto un ahorro de tiempo y recursos en el desarrollo y pruebas de los productos empresariales. Como consecuencia, las empresas logran producciones más eficientes e innovadoras. Por ello, este proyecto aporta al aumento de la investigación científica y mejora de la capacidad tecnológica de los sectores industriales de todos los países, en particular los países en desarrollo, debido a que el uso de tecnologías con Inteligencia Artificial presenta menor gasto de recursos y mantenimiento más sencillo.

10. CONCLUSIONES

En este proyecto se ha abordado la temática de la Inteligencia Artificial aplicada al análisis, clasificación y generación de imágenes. Para ello, se ha llevado a cabo un estudio y análisis de las principales tecnologías utilizadas en la actualidad para dichos fines.

En primer lugar, en el ámbito de IA aplicada a la generación de imágenes ha tenido lugar una evolución de estructura y especificidad de los *prompts* hasta obtener un esquema en el que se obtienen resultados similares en las 3 IAs utilizadas para la realización de pruebas. Dicha estructura permite, a su vez, un alto grado de definición y concreción en las instrucciones que se le proporcionan a la IA generadora, obteniendo así resultados adecuados y coherentes con dichas instrucciones. Por otro lado, cabe destacar el inusual comportamiento de *Lexica* en las pruebas realizadas, mostrando una tendencia a la sobresaturación de las imágenes e interpretando las instrucciones de forma más literal que *PlaygroundAI* o *Clipdrop*. Adicionalmente, se puede deducir la falta de entrenamiento de *Lexica* con imágenes de estilo de animación o manga, ya que sus resultados en dicha técnica no han llegado a converger. Por otro lado, cabe destacar como debilidad común a las tres IAs generadoras la localización de dichas imágenes en espacios vacíos tales como el océano o el espacio, presentando así una confusión en aspectos como los puntos de apoyo de los elementos de la escena o la localización de línea del horizonte. Por lo tanto, en estos casos los resultados generados tienden a divergir en mayor medida de las instrucciones dadas.

Por otro lado, en el ámbito de IA aplicada a clasificación y análisis de imágenes tras realizar numerosas pruebas con redes neuronales convolucionales basadas en diferentes arquitecturas y distintas combinaciones de hiperparámetros, se puede deducir que la arquitectura que mejor rendimiento ha presentado es la versión 2 de ResNet de 152 capas, con la combinación de hiperparámetros y *data augmentation* de la prueba 1 (Figura 39). No obstante, los parámetros utilizados para la segunda prueba con la misma arquitectura proporcionan unos resultados similares, obteniendo una media de precisión al final del entrenamiento cercana al 80% y pérdida entre 0.45 y 0.50. Por otro lado, observando las figuras 43 y 44 se puede afirmar que la arquitectura de MobileNet v2 es la que peor rendimiento ha presentado, probablemente debido a que se trata de la arquitectura más sencilla de las tres utilizadas. La arquitectura Inception-ResNet v2 se ha comportado de forma similar a ResNet v2, pero con precisión y pérdida ligeramente peores. Por lo tanto, se puede concluir que aunque los resultados obtenidos dejen margen de mejora, tratándose de un dataset de tan solo 126 ecografías pertenecientes a un único hospital (Hospital Clínico San Carlos), se pueden considerar resultados satisfactorios.

11. TRABAJO FUTURO

Con la finalidad de mejorar los resultados obtenidos y fomentar la investigación en las materias abordadas, se proponen las siguientes posibles líneas de trabajo futuras:

- Crear un banco de imágenes común para todos los hospitales del mundo con la finalidad de lograr un dataset de mayor tamaño y variedad, aumentando así la eficacia del entrenamiento de la red neuronal creada.
- Crear una aplicación para uso médico utilizando el modelo creado.
- Realizar más pruebas utilizando otras arquitecturas o modificando las ya usadas.

12. Bibliografía

- [1] Richard L. Drake, A. Wayne Vogl, Adam W. M. Mitchell. "Anatomía para estudiantes GRAY". 3ª Edición. Barcelona, España: Elsevier, 2015.
- [2] Frank H. Netter. "Atlas de anatomía humana". 6ª Edición. Barcelona, España: Elsevier, 2015.
- [3] Kaiming He, Xiangyu Zhang. "Deep Residual Learning for Image Recognition". Arxiv. Disponible en: <https://arxiv.org/pdf/1512.03385.pdf> Accedido el: agosto de 2023
- [4] "tf.keras.preprocessing.image.ImageDataGenerator". [tensorflow.org](https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator). Disponible en: https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator Accedido el: julio de 2023
- [5] Redacción KeepCoding. "Tipos de capas de red neuronal convolucional". [keepcoding.io](https://keepcoding.io/blog/tipos-capas-red-neuronal-convolucional/#Tipos_de_capas_convolucionales). Disponible en: https://keepcoding.io/blog/tipos-capas-red-neuronal-convolucional/#Tipos_de_capas_convolucionales Accedido el: agosto de 2023
- [6] Miguel Sotaquirá. "Padding, strides, max-pooling y stacking en las Redes Convolucionales". [codificandobits.com](https://www.codificandobits.com). Disponible en: <https://www.codificandobits.com/blog/padding-strides-maxpooling-stacking-redes-convolucionales/> Accedido el: agosto de 2023
- [7] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, Li Fei-Fei. "ImageNet Large Scale Visual Recognition Challenge". image-net.org. Disponible en: <https://www.image-net.org/challenges> Accedido el: 2 de agosto de 2023
- [8] Pepe Cantoral, Ph.D. "¿Qué hacen las capas de una red neuronal? - Inteligencia Artificial -". 2021. Youtube. Disponible en: <https://youtu.be/jGNysjHTnho> Accedido el: 1 de agosto de 2023
- [9] Pepe Cantoral, Ph.D. "Modelo matemático de una Neurona Artificial - REDES NEURONALES - (Parte 2)". 2021. Youtube. Disponible en: <https://youtu.be/XoIj-omJAZo> Accedido el: 1 de agosto de 2023
- [10] Pepe Cantoral, Ph.D. "Función SOFTMAX - REDES NEURONALES - (Parte 3)". 2021. Youtube. Disponible en: <https://youtu.be/3KhumVlS6vs> Accedido el: 1 de agosto de 2023
- [11] Pepe Cantoral, Ph.D. "Gradient descent - REDES NEURONALES - (Parte 5)". 2021. Youtube. Disponible en: <https://youtu.be/lnDjwepC-5I> Accedido el: 3 de agosto de 2023
- [12] Pepe Cantoral, Ph.D. "¿Por qué usamos el gradiente para entrenar Redes Neuronales? Ejemplo - Deep Learning - (Parte 6)". 2021. Youtube. Disponible en: <https://youtu.be/Cr5cYDrMYZQ> Accedido el: 3 de agosto de 2023
- [13] Pepe Cantoral, Ph.D. "Backpropagation - Fundamentos del Deep Learning - (Parte 7)". 2021. Youtube. Disponible en: <https://youtu.be/4nDsiV3GXa8> Accedido el: 3 de agosto de 2023
- [14] Pepe Cantoral, Ph.D. "Redes Neuronales Multi-Capa / Deep Neural Networks - Fundamentos de Deep Learning - (Teoría Parte 11)". 2021. Youtube. Disponible en: <https://youtu.be/Xc2GsX2DkJA> Accedido el: 30 de julio de 2023
- [15] Pepe Cantoral, Ph.D. "¿Qué es ResNet? - Redes Neuronales Residuales". 2022. Youtube. Disponible en: <https://youtu.be/gKCuWBpg1bU> Accedido el: 31 de julio de 2023
- [16] Ringa Tech. "Tu primera red neuronal - Inteligencia Artificial". 2020. Youtube. Disponible en: <https://youtu.be/UNFFLJPW7KQ> Accedido el: 7 de julio de 2023

- [17] Ringa Tech. "Crea tu propia red neuronal que puede leer". 2020. Youtube. Disponible en: <https://youtu.be/aFZEvODTSyA> Accedido el: 7 de julio de 2023
- [18] Ringa Tech. "Cómo funcionan las redes neuronales - Inteligencia Artificial". 2021. Youtube. Disponible en: <https://youtu.be/CU24iC3grq8> Accedido el: 7 de julio de 2023
- [19] Ringa Tech. "Reconocimiento de imágenes con IA - 01 - Convoluciones y filtros". 2021. Youtube. Disponible en: https://youtu.be/AwTH_0yW9_I Accedido el: 8 de julio de 2023.
- [20] Ringa Tech. "Tu primera red neuronal en Python y Tensorflow". 2021. Youtube. Disponible en: https://youtu.be/iX_on3VxZzk Accedido el: 7 de julio de 2023
- [21] Ringa Tech. "Tu primer clasificador de imágenes con Python y Tensorflow". 2021. Youtube. Disponible en: <https://youtu.be/j6eGHR0LKP8> Accedido el: 7 de julio de 2023
- [22] Ringa Tech. "Redes Neuronales Convolucionales - Clasificación Avanzada de imágenes con IA / ML (CNN)". 2022. Youtube. Disponible en: <https://youtu.be/4sWhhOwHqug> Accedido el: 10 de julio de 2023
- [23] Ringa Tech. "Crea un clasificador de perros y gatos con IA, Python y Tensorflow - Proyecto completo". 2022. Youtube. Disponible en: <https://youtu.be/DbwKbsCWPSg> Accedido el: 11 de julio de 2023
- [24] Ringa Tech. "¿Pocos datos de entrenamiento? Prueba esta técnica". 2022. Youtube. Disponible en: https://youtu.be/9Dur_oUMGG8 Accedido el: 13 de julio de 2023
- [25] Ringa Tech. "Funciones de activación a detalle (Redes neuronales)". 2022. Youtube. Disponible en: https://youtu.be/_0wdproot34 Accedido el: 10 de julio de 2023
- [26] Dot CSV. "¿Qué es una red neuronal? Parte 1: La Neurona | DotCSV". 2018. Youtube. Disponible en: <https://youtu.be/MRIv2IwFTPg> Accedido el: 20 de julio de 2023
- [27] Dot CSV. "¿Qué es una red neuronal? Parte 2: La Red | DotCSV". 2018. Youtube. Disponible en: <https://youtu.be/uwbHOpp9xkc> Accedido el: 20 de julio de 2023
- [28] Dot CSV. "Jugando con Redes Neuronales - Parte 2.5 | DotCSV". 2018. Youtube. Disponible en: <https://youtu.be/FVozZVUNOOA> Accedido el: 21 de julio de 2023
- [29] Dot CSV. "¿Qué es una red neuronal? Parte 3: Backpropagation | DotCSV". 2019. Youtube. Disponible en: https://youtu.be/eNIqz_noix8 Accedido el: 21 de julio de 2023
- [30] Dot CSV. "¿Qué es una red neuronal? Parte 3.5: Las Matemáticas de Backpropagation | DotCSV". 2019. Youtube. Disponible en: <https://youtu.be/M5OHwkkHgAA> Accedido el: 22 de julio de 2023
- [31] Dot CSV. "¡Redes neuronales CONVOLUCIONALES! ¿Cómo funcionan?". 2021. Youtube. Disponible en: <https://youtu.be/V8j1oENVz00> Accedido el: 15 de julio de 2023
- [32] Dot CSV. "Las Redes Neuronales... ¿Aprenden o Memorizan? - Overfitting y Underfitting - Parte 1". 2019. Youtube. Disponible en: <https://youtu.be/7-6X3DTt3R8> Accedido el: 6 de agosto de 2023
- [33] Dot CSV. "Cómo identificar el OVERFITTING en tu RED NEURONAL - Parte 2". 2019. Youtube. Disponible en: <https://youtu.be/ZmLKqZYlYUI> Accedido el: 6 de agosto de 2023

- [34] Sistemas Inteligentes. "Data Augmentation para clasificación de imágenes usando keras - tensorflow (Parte 1)". 2020. Youtube. Disponible en: <https://youtu.be/4G2WAgJ4G3Y> Accedido en: julio de 2023
- [35] DigitalSreeni. "127 - Data augmentation using keras". 2020. Youtube. Disponible en: <https://youtu.be/ccdssX4rIh8> Accedido en: agosto 2023
- [36] Jordi Torres. "Data augmentation y transfer learning". Jordi Torres.AI. Disponible en: <https://torres.ai/data-augmentation-y-transfer-learning-en-keras-tensorflow/> Accedido en: agosto de 2023
- [37] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi. "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning". Arxiv. Disponible en: <https://arxiv.org/pdf/1602.07261.pdf> Accedido en: agosto en 2023
- [38] SALVAJE. "5 herramientas de INTELIGENCIA ARTIFICIAL para crear imágenes que no conocías [100% GRATIS]". 2023. Youtube. Disponible en: <https://youtu.be/NNwCTXJOfoQ> Accedido en: julio de 2023
- [39] DonebyLaura. "He probado 100 herramientas de IA. Estas son las mejores". 2023. Youtube. Disponible en: <https://youtu.be/-9oCF-LWB9Y> Accedido en: julio de 2023
- [40] SALVAJE. "TOP 5 mejores PAGINAS DE IA PARA CREAR IMAGENES [100% GRATIS] Tutorial de Inteligencia Artificial". 2023. Youtube. Disponible en: <https://youtu.be/Fo9PtS2lckw> Accedido en: julio de 2023
- [41] Arnau Gómez. "5 WEBS para crear IMÁGENES con INTELIGENCIA ARTIFICIAL - Fácil y GRATIS con DALL-E 2". 2022. Youtube. Disponible en: <https://youtu.be/VX7CCFa6hGM> Accedido en: julio de 2023
- [42] Elven. "Mejor GENERADOR DE IMAGENES IA gratis y Fácil de Usar | TOP 10". 2023. Youtube. Disponible en: <https://youtu.be/ao-dbJnlUEk> Accedido en: julio de 2023
- [43] ChicaGeek. "Las mejores IAs para CREAR IMÁGENES GRATIS". 2023. Youtube. Disponible en: <https://youtu.be/apVuBr3sntc> Accedido en: julio de 2023
- [44] Nana Kwame Kankam. "Understanding Inception-ResNet V1 architecture". OpenGenus IQ. Disponible en: <https://iq.opengenus.org/inception-resnet-v1/> Accedido en: julio de 2023
- [45] "Breve historia visual de la inteligencia artificial". National Geographic. Disponible en: https://www.nationalgeographic.com.es/ciencia/breve-historia-visual-inteligencia-artificial_14419 Accedido en: agosto de 2023
- [46] "¿Qué son las redes neuronales convolucionales?". IBM. Disponible en: <https://www.ibm.com/es-es/topics/convolutional-neural-networks> Accedido en: julio de 2023
- [47] Manuel de la Cierva García, Miguel Sánchez Fernández, Cristina Juárez Díaz. "Análisis de imágenes médicas mediante IA: innovación y precisión". revistamedica.com . Disponible en: <https://revistamedica.com/analisis-diagnostico-imagenes-medicas-mediante-inteligencia-artificial/> Accedido en: agosto de 2023
- [48] Agencia EFE. "Un robot gana a 15 médicos diagnosticando tumores cerebrales". eldiario.es . Disponible en: https://www.eldiario.es/tecnologia/robot-medicos-diagnosticando-tumores-cerebrales_1_2047223.html Accedido en: agosto de 2023
- [49] Tomás Rivero. "Esta ayudará a detectar más rápido el cáncer de mama". hipertextual.com . Disponible en: <https://hipertextual.com/2022/07/ia-ayudaria-detectar-rapido-cancer-mama-profesionales-medicos> Accedido en: agosto de 2023

[50] Hana Kiro. "Doctors using AI catch breast cancer more often than either does alone". technologyreview.com . Disponible en: https://www.technologyreview.com/2022/07/11/1055677/ai-diagnose-breast-cancer-mammograms/?utm_medium=tr_social&utm_campaign=site_visitor_unpaid_engagement&utm_source=Twitter Accedido en: agosto de 2023

[51] Luis Miranda. "NVIDIA usará su IA para ayudar a detectar el cáncer de colon". hipertextual.com . Disponible en: <https://hipertextual.com/2023/03/nvidia-usara-ia-para-detectar-cancer-de-colon> Accedido en: agosto de 2023

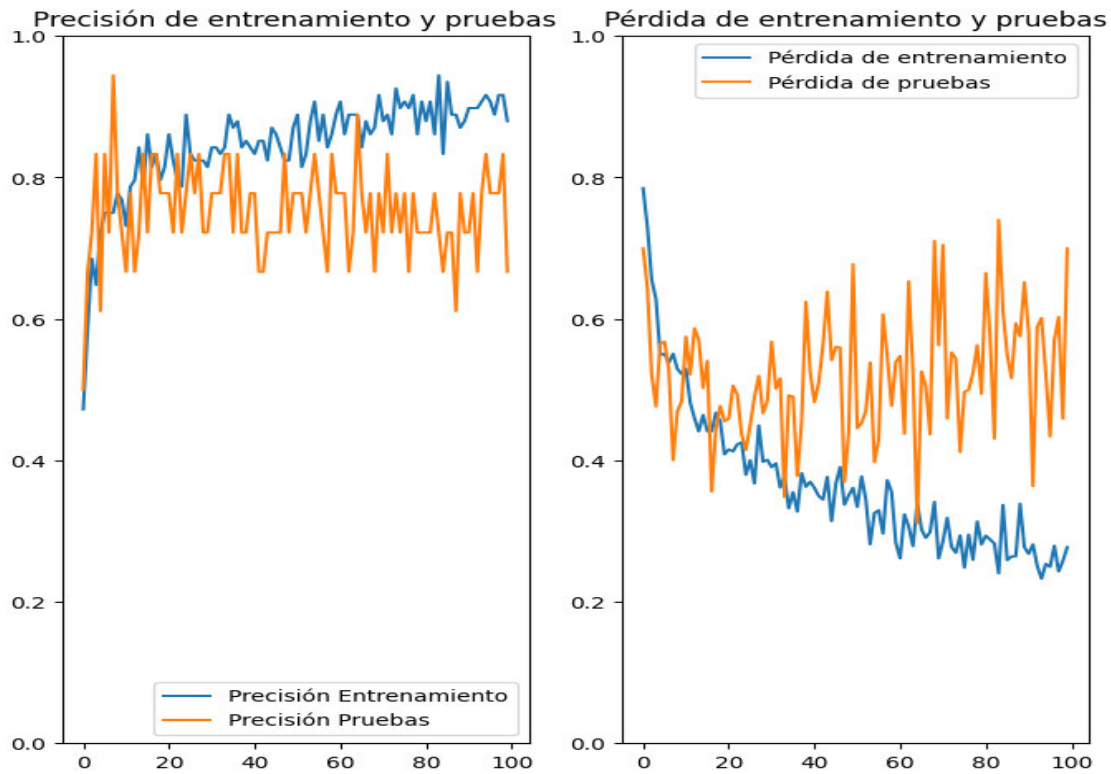
[52] Janette Ciborowski. "Medtronic and NVIDIA Collaborate to build AI platform for Medical Devices". nvidianews.nvidia.com . Disponible en: <https://nvidianews.nvidia.com/news/medtronic-and-nvidia-collaborate-to-build-ai-platform-for-medical-devices> Accedido en: agosto de 2023

[53] D. R. Sarvamangala, Raghavendra V. Kulkarni. "Convolutional neural networks in medical image understanding: a survey". link.springer.com . Disponible en: <https://link.springer.com/article/10.1007/s12065-020-00540-3> Accedido en: agosto de 2023

[54] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig Adam. "MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications". Arxiv. Disponible en: <https://arxiv.org/pdf/1704.04861.pdf> Accedido en: julio 2023

ANEXO A - OTRAS PRUEBAS

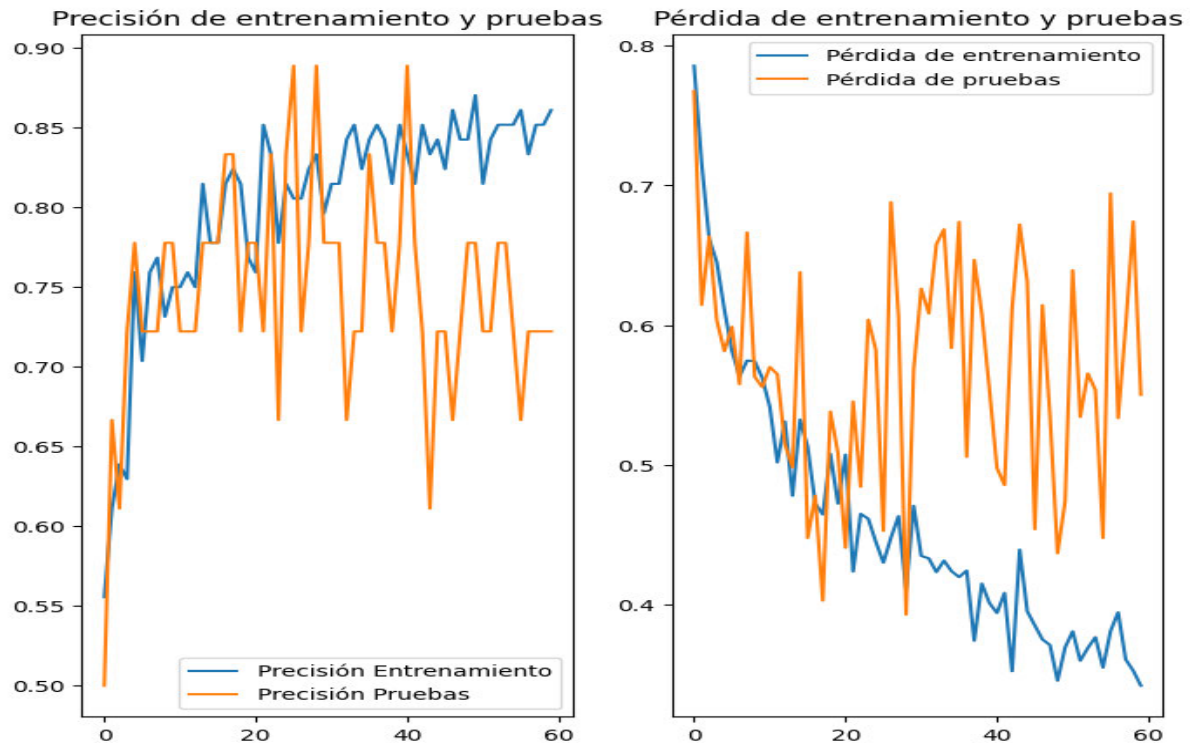
RESNET V2



url =

["https://tfhub.dev/google/imagenet/resnet_v2_152/feature_vector/5"](https://tfhub.dev/google/imagenet/resnet_v2_152/feature_vector/5)

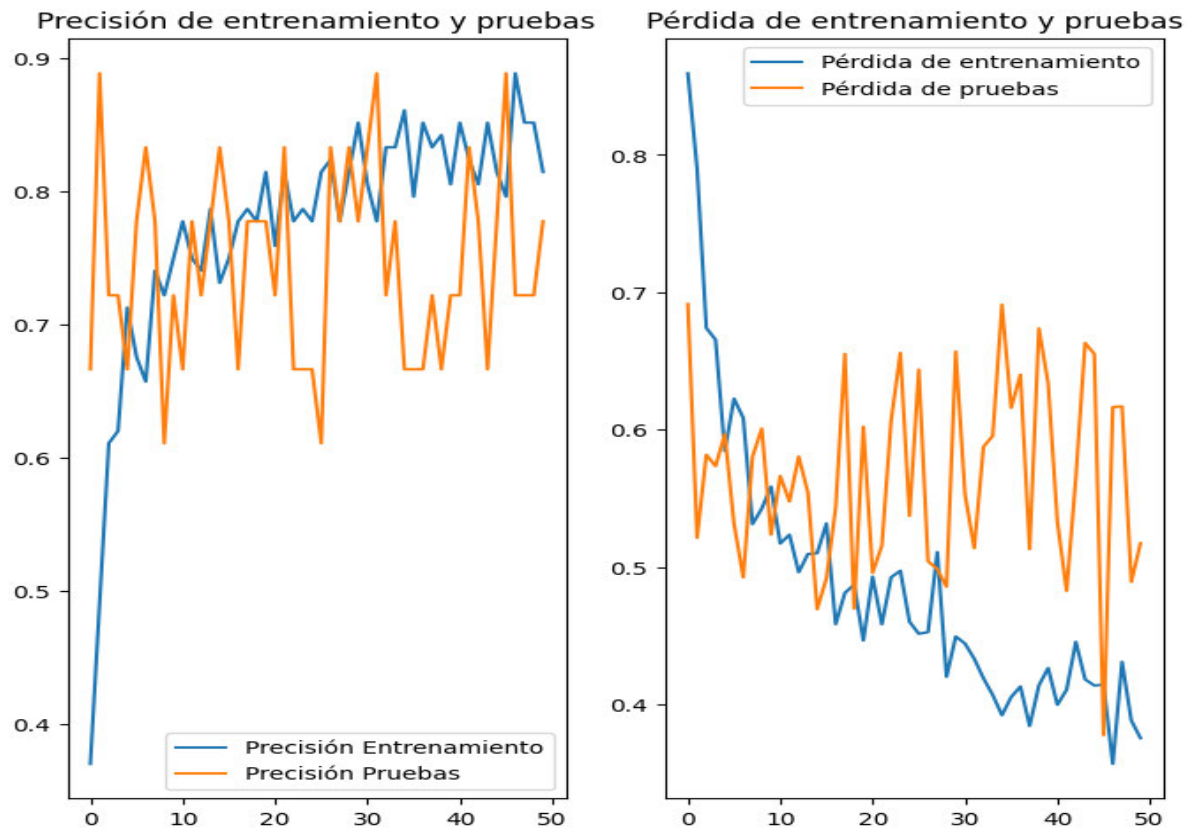
```
datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=30,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.30,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
modelo.compile(
    optimizer = 'adam',
    loss = 'binary_crossentropy',
    metrics = ['accuracy']
)
```



```

datagen = ImageDataGenerator(
    rescale = 1. / 255,
    rotation_range=30,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.25,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
url = "https://tfhub.dev/google/imagenet/resnet\_v2\_152/feature\_vector/5"
modelo.compile(
    optimizer = 'adam',
    loss = 'binary_crossentropy',
    metrics = ['accuracy']
)

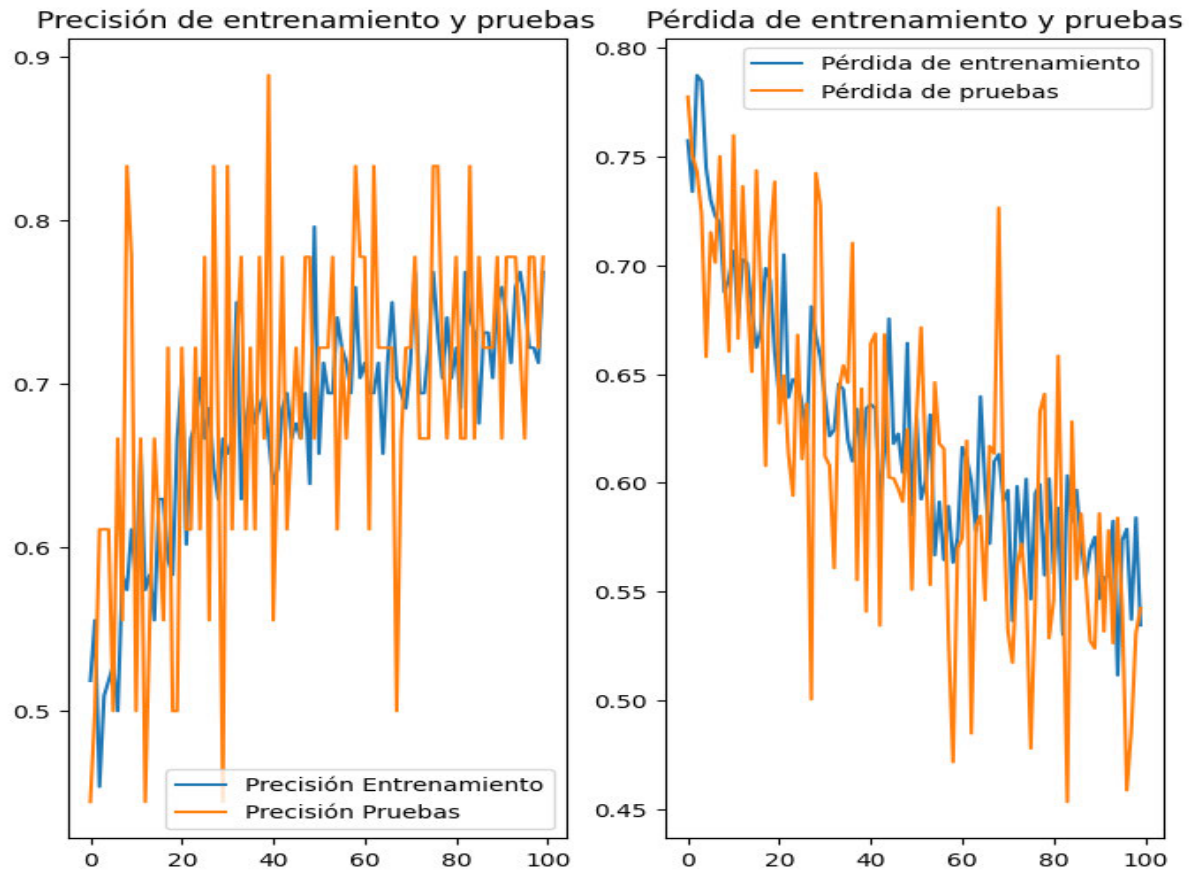
```



```

datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=40,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
url = "https://tfhub.dev/google/imagenet/resnet_v2_152/feature_vector/5"
modelo.compile(
    optimizer = 'adam',
    loss = 'binary_crossentropy',
    metrics = ['accuracy']
)

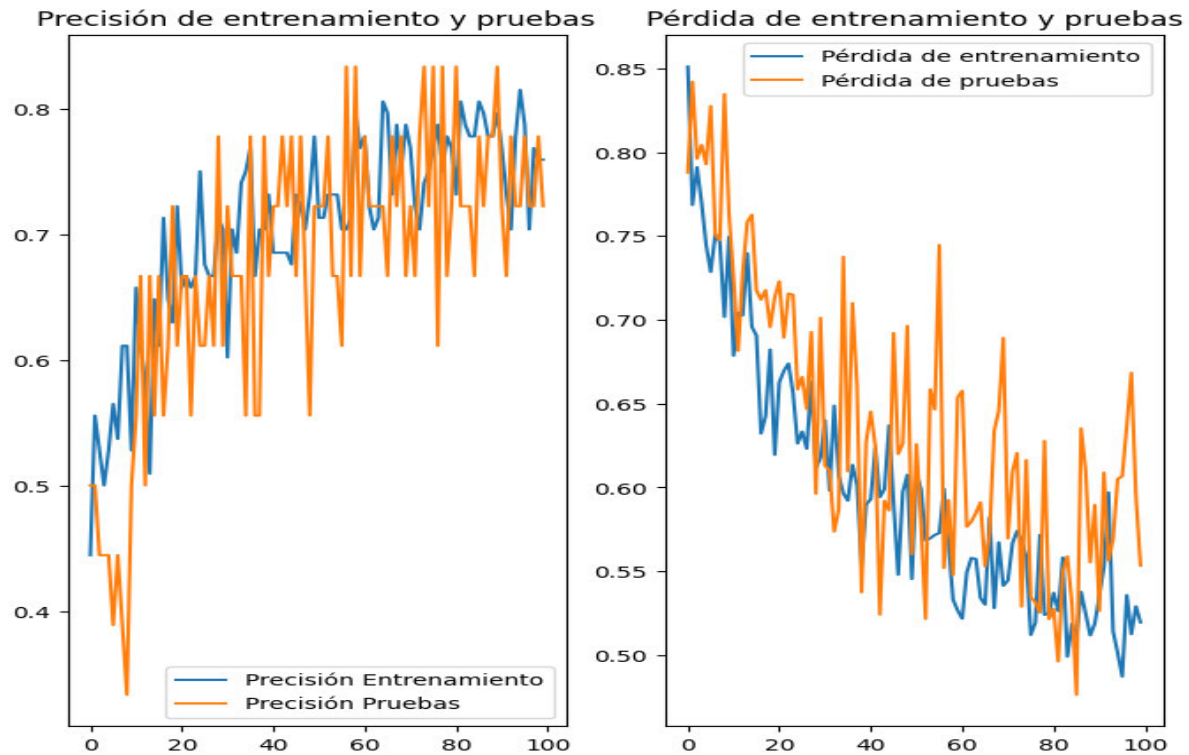
```



```

datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=40,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
url = "https://tfhub.dev/google/imagenet/resnet_v2_152/feature_vector/5"
modelo.compile(
    optimizer = SGD(learning_rate=0.001),
    loss = 'binary_crossentropy',
    metrics = ['accuracy']
)

```



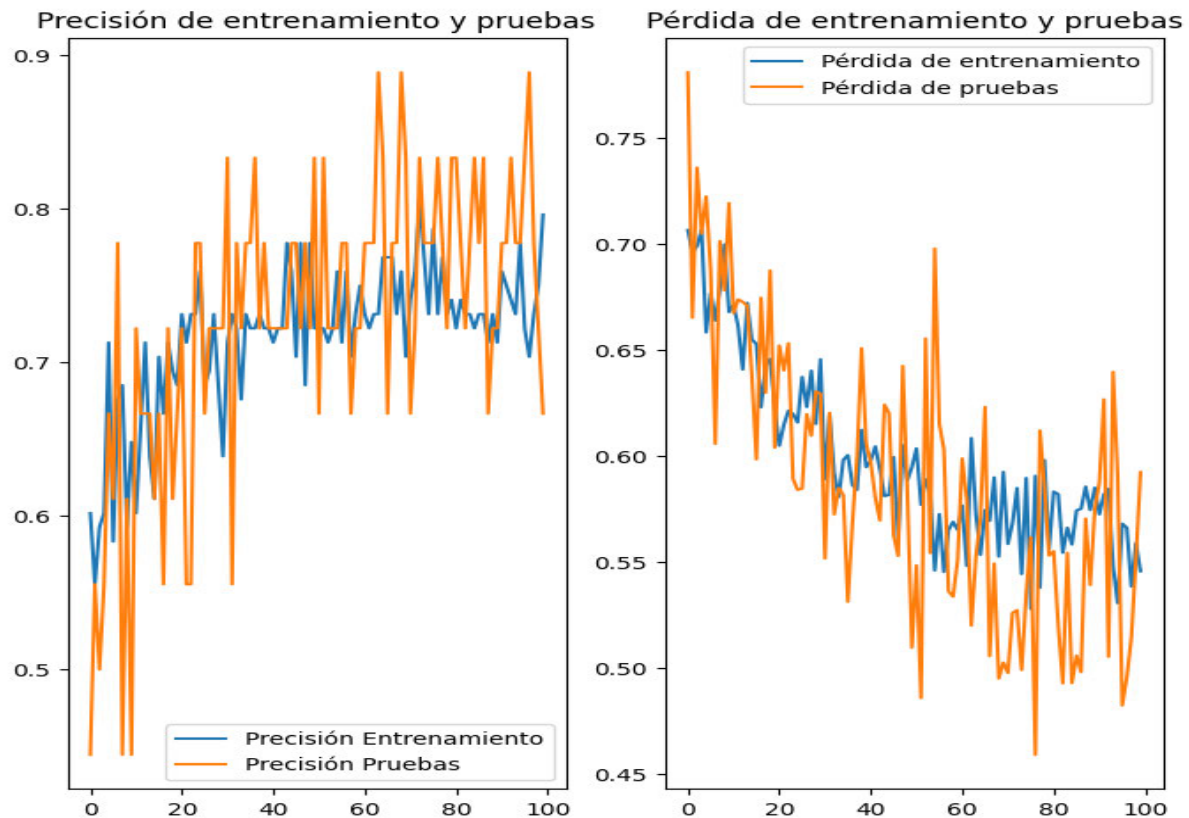
```

datagen = ImageDataGenerator(
    rescale = 1. / 255,
    rotation_range=40,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)

url = "https://tfhub.dev/google/imagenet/resnet_v2_152/feature_vector/5"
modelo.compile(
    optimizer = RMSprop(learning_rate=0.0001),
    loss = 'binary_crossentropy',
    metrics = ['accuracy']
)

```

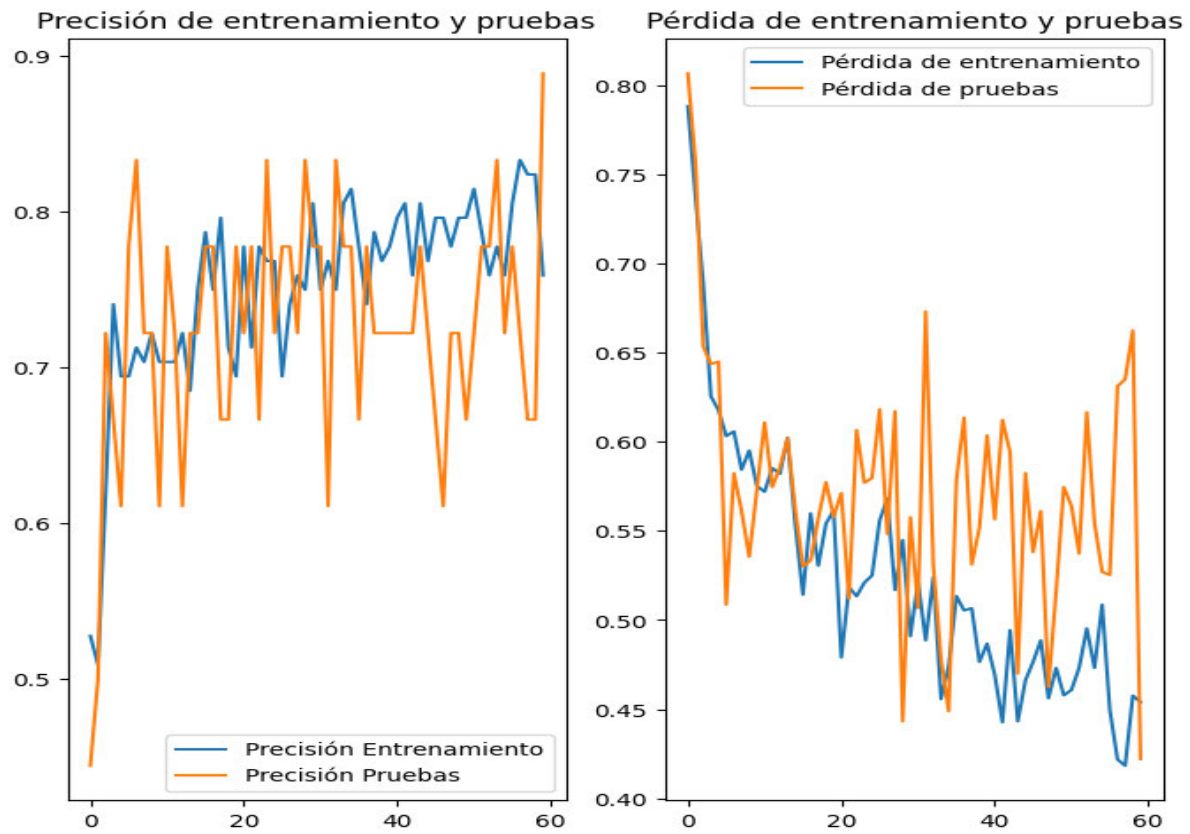
INCEPTION RESNET V2



```

datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=40,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
modelo.compile(
    optimizer = Adam(learning_rate=0.0001),
    loss = 'binary_crossentropy',
    metrics = ['accuracy']
)

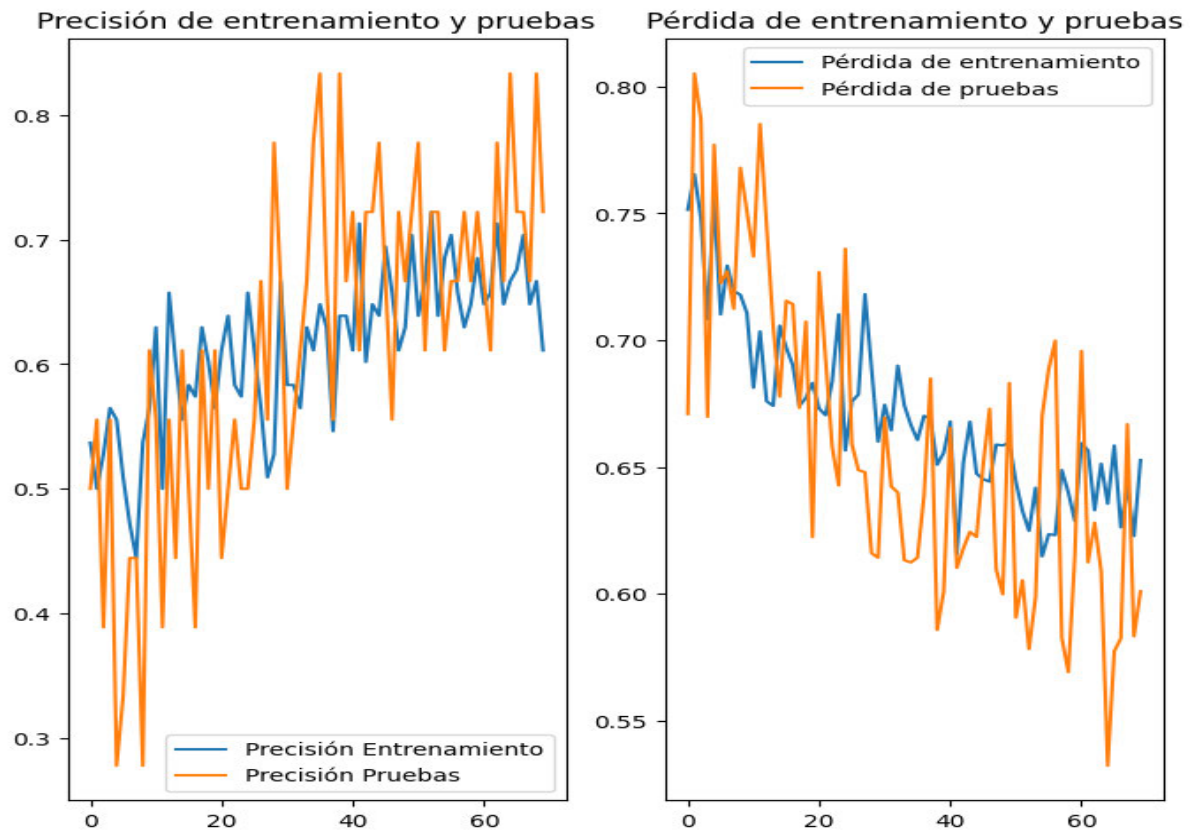
```



```

datagen = ImageDataGenerator(
    rescale = 1. / 255,
    rotation_range=30,
    width_shift_range=0.1,
    height_shift_range=0.1,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
url = "https://tfhub.dev/google/imagenet/inception_resnet_v2/feature_vector/5"
#Compilar modelo
modelo.compile(
    optimizer = 'adam',
    loss = 'binary_crossentropy',
    metrics = ['accuracy']
)

```

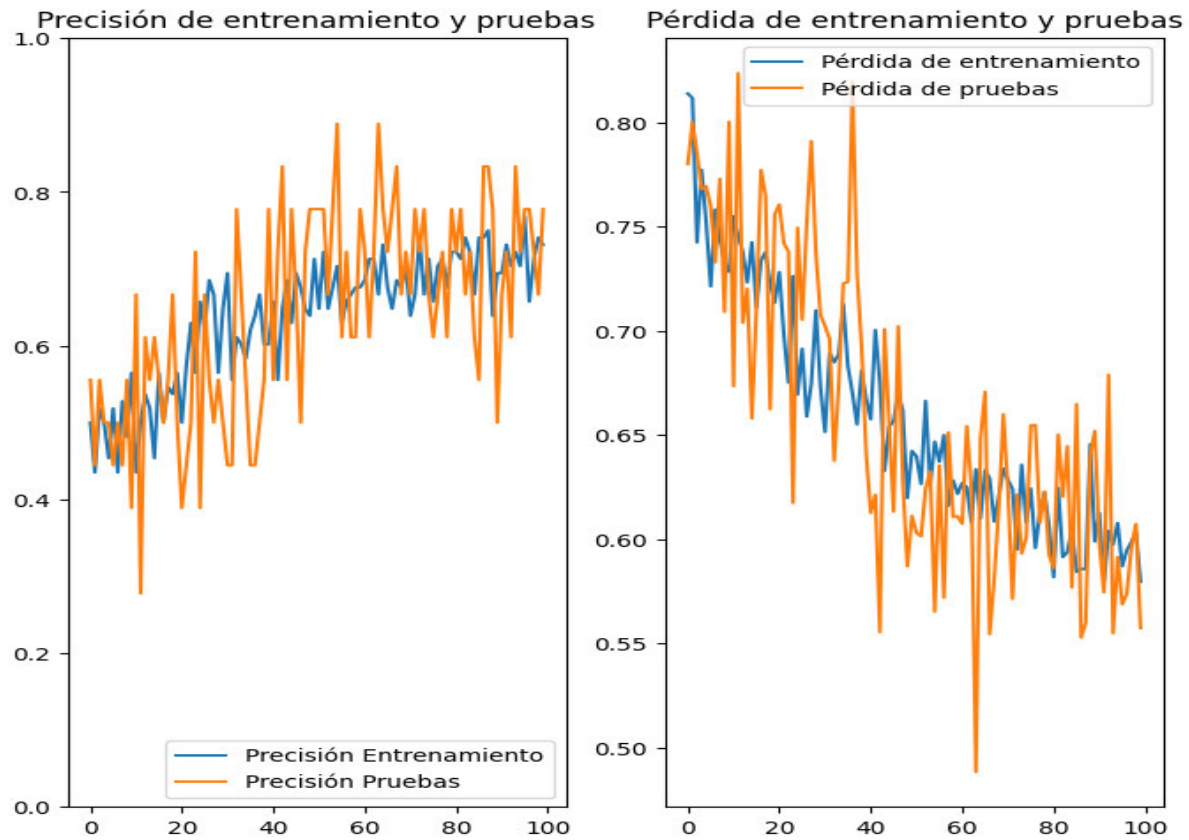


```

datagen = ImageDataGenerator(
    rescale = 1. / 255,
    rotation_range=40,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
url = "https://tfhub.dev/google/imagenet/inception_resnet_v2/feature_vector/5"
modelo.compile(
    optimizer = SGD(learning_rate=0.001),
    loss = 'binary_crossentropy',
    metrics = ['accuracy']
)

```

Mobilenet v2



```

datagen = ImageDataGenerator(
    rescale = 1. /255,
    rotation_range=40,
    width_shift_range=0.15,
    height_shift_range=0.15,
    zoom_range=0.3,
    fill_mode='nearest',
    validation_split=0.15 #15% para pruebas
)
url = "https://tfhub.dev/google/tf2-preview/mobilenet_v2/feature_vector/4"
modelo.compile(
    optimizer = SGD(learning_rate=0.001),
    loss = 'binary_crossentropy',
    metrics = ['accuracy']
)

```