UNIVERSIDAD COMPLUTENSE DE MADRID

FACULTAD DE CIENCIAS FÍSICAS

DEPARTAMENTO DE ESTRUCTURA DE LA MATERIA, FÍSICA TÉRMICA Y ELECTRÓNICA

TRABAJO DE FIN DE GRADO

Código TFG: ETE07

Inteligencia Artificial en Imagen Médica

Artificial Intelligence in Medical Imaging

Supervisor/es: Joaquín López Herraiz

Guadalupe Fraile Ramírez

Grado en Física

Curso académico 2023-2024

Convocatoria de julio

Calificación: 9,5

Inteligencia Artificial en Imagen Médica: Clasificación de transductores en imágenes de ultrasonido de riñones usando redes neuronales.

Resumen

La imagen por ultrasonidos es una técnica comúnmente utilizada en el diagnóstico médico. Consiste en la emisión de ondas acústicas de alta frecuencia, para posteriormente detectar y analizar los ecos resultantes, y así obtener imágenes de órganos y estructuras internas del cuerpo, como los riñones. Si bien es cierto que es una técnica no invasiva, no ionizante y accesible, está sujeta a ciertas dificultades a la hora de la toma e interpretación de la imagen dada la alta variabilidad que existe en la técnica y en los parámetros físicos de las sondas.

La inteligencia artificial (IA) y sus recientes avances también se han hecho hueco en el ámbito de la imagen médica. A pesar de los retos existentes, las redes neuronales se han implementado en el procesado e interpretación de imágenes para obtener resultados más precisos y eficientes. Por ejemplo, pueden ser entrenadas para encontrar patrones sutiles que pueden pasar desapercibidos para el ojo humano y con ello clasificar imágenes de forma más sencilla.

El objetivo de este trabajo ha sido entrenar un modelo de red neuronal clasificadora para distinguir entre el tipo de transductor empleado para adquirir imágenes de ultrasonido de riñones. Se busca poder usar ese clasificador ya entrenado para obtener información automática de la sonda y así poder tenerla en cuenta de manera explícita cuando se analicen conjuntos de imágenes de varios equipos. Así mismo permitiría estudiar las diferencias que existen entre equipos y ayudaría a aplicar técnicas de armonización para mitigarlas. Este proceso presenta la dificultad habitual de no disponer de una base de datos muy extensa y balanceada de casos, es por ello que se han llevado a cabo técnicas como el aumento de datos.

En conclusión, este estudio busca avanzar hacia el objetivo de hacer de la imagen por ultrasonido una técnica más accesible y efectiva en diagnóstico médico.

Abstract

Ultrasound imaging is a technique commonly used in medical diagnostics. It consists of the emission of high-frequency acoustic waves, whose echoes are subsequently detected and analyzed to obtain images of organs and structures inside the body, such as kidneys. Although it is a non-invasive, non-ionising and accessible technique, it is subject to certain difficulties when it comes to image acquisition and interpretation due to the high variability in the technique and physical parameters of the probes.

Artificial intelligence (AI) and its recent advances have also found their way into the field of medical imaging and ultrasound imaging. Despite the existing challenges, neural networks have been implemented in image processing and interpretation to obtain more accurate and efficient results. For instance, they can be trained to find subtle patterns that may go unnoticed by the human eye and thus classify images more easily.

The aim of this work has been to train a classifier neural network model to differentiate the type of transducer used in ultrasound images of the kidneys. The goal is to be able to use this trained classifier to obtain automatic information from the probe to explicitly take it into account when analysing sets of images from multiple devices. It would also allow us to study the differences that exist between teams and help to apply harmonisation techniques to mitigate them. This process presents the usual difficulty of not having a very extensive and balanced database of cases, which is why techniques such as data augmentation have been carried out.

In conclusion, this study seeks to advance towards the goal of making ultrasound imaging a more accessible and effective technique in medical diagnosis.

Tabla de contenido

1. Introducción

1.1. Fundamentos de la imagen de ultrasonido

Los ultrasonidos se refieren a ondas mecánicas longitudinales de alta frecuencia (por encima de 20kHz) tal que son inaudibles para los humanos. La forma más común de generarlos es mediante un transductor que convierte una señal eléctrica en un haz de ultrasonidos gracias al efecto piezoeléctrico [1].

Éste es el fenómeno que se observa cuando, a partir de someter a determinados cristales a una expansión o contracción (tensiones mecánicas), se obtiene una polarización eléctrica del material. Ocurre porque la variación en la estructura interna da lugar a una diferencia de densidad de carga negativa en una zona del cristal, lo cual origina una diferencia de potencial [1]. A su vez, la aplicación de un voltaje a un material de este estilo puede inducir una tensión mecánica por la orientación de los dipolos eléctricos según el campo aplicado, causando una contracción o expansión del material (efecto piezoeléctrico inverso) [1]. Así, un material piezoeléctrico puede convertir energía eléctrica en mecánica y viceversa, y, por esto, actuar a la vez de emisor y receptor [1].

En contacto con un medio elástico y con corriente continua, el transductor genera ondas de compresión y descompresión periódicamente, o más bien una onda acústica en la que cada frente de compresión ha avanzado una distancia λ cuando se produce la siguiente [1]. Su velocidad viene descrita por $c = \lambda f$, y como la frecuencia se escoge al pasar corriente por el transductor, la longitud de onda es proporcional a esta velocidad (que también depende de la compresibilidad del medio) [1].

Una vez las ondas acústicas se propagan por el interior del cuerpo se ven sometidas a diferentes fenómenos. Para empezar, depende de la densidad del tejido por el que se propaga. Puede definirse la impedancia acústica (Z) del material como el producto de la velocidad de la onda y la densidad del medio [2]. Cada vez que la onda se encuentra con una frontera entre tejidos con diferente impedancia (se dice que el tamaño de la interfaz es considerablemente mayor que λ_{US}), esta sufre efectos de **refracción** y **reflexión**. La fracción de energía que se refleja hacia el transductor es precisamente la señal detectada para formar la imagen, y depende no solo del ángulo de incidencia, sino que además de la diferencia entre propiedades acústicas del material:

$$R = (Z_1 - Z_2)^2/(Z_1 + Z_2)^2. \quad (1)$$

De este modo, cuanto mayor sea la diferencia entre los parámetros acústicos del material, mayor parte de la energía es reflejada [2]. A continuación, algunos valores de los mismos para diferentes tejidos del cuerpo:

	$Z \times 10^5$ (g cm^{-2} s^{-1})	Speed of sound (m s^{-1})	Density (gm^{-3})	Compressibility x10^{11} (cm g^{-1} s^2)
Air	0.00043	330	1.3	70 000
Blood	1.59	1570	1060	4.0
Bone	7.8	4000	1908	0.3
Fat	1.38	1450	925	5.0
Brain	1.58	1540	1025	4.2
Muscle	1.7	1590	1075	3.7
Liver	1.65	1570	1050	3.9
Kidney	1.62	1560	1040	4.0

Figura 1. Propiedades acústicas de diferentes tejidos biológicos [2].

Cuando lo que se encuentra es una estructura de tamaño comparable o incluso menor que λ, entonces sufre **dispersión** (*scattering*) en todas las direcciones. Si estructuras no son relativamente abundantes o se encuentran lejos unas de otras, esto resulta en patrones de **interferencia** constructiva y destructiva que se conoce como *speckle* o ruido en la imagen de ultrasonido [2].

Por último, debe tenerse en cuenta el efecto de la **absorción**, resultado de disipa en forma de calor por las fuerzas viscosas y fricción entre las partículas de los tejidos.

Los efectos de dispersión por estructuras pequeñas junto con los de absorción resultan en una **atenuación** significativa de la onda. Se traduce en un decaimiento exponencial tanto de la presión como de la intensidad de la onda con la profundidad (z). En particular para la intensidad:

$$I(z) = I_0 e^{-\mu z} \tag{2}$$

con μ el coeficiente de atenuación de intensidad en dB/cm.

Es relevante destacar que la atenuación aumenta si lo hace la frecuencia. Según [3], hay 0.5dB (aproximadamente) de atenuación por centímetro para cada megahercio. Con ello:

$$a_{tej.\ blando}(dB) \cong \frac{1}{2}\left(\frac{dB}{cm \cdot MHz}\right) \cdot f(MHz) \cdot L(cm)\ . \tag{3}$$

Por tanto, una onda de alta frecuencia se atenuará más que una con frecuencia baja, lo que a su vez causa que para frecuencias altas se consiga una penetración menor [3]. Es por todo esto por lo que, inevitablemente, la resolución de las imágenes de ultrasonido está limitada por el fenómeno de la atenuación, y por tanto de la frecuencia escogida. Lo que determina qué frecuencia escoger son las dimensiones de la anatomía del cuerpo humano, y éstas, normalmente entre 2-20 MHz [3], pertenecen al rango del ultrasonido.

Aprovechando el mismo efecto piezoeléctrico se utiliza el mismo transductor como receptor para volver a transformar la energía sonora de los ecos en eléctrica y posteriormente formar la imagen. Pero la señal que transforma el transductor normalmente es débil y requiere de un método de amplificación para poder ser procesado adecuadamente. La **ganancia** expresa precisamente el ratio de aumento en la potencia, que se traduce en la imagen como un aumento en el brillo de las estructuras. La compensación de ganancia en tiempo (**TGC**) se refiere a la ganancia que debe aplicarse teniendo en cuenta los efectos de la atenuación a diferentes profundidades [3].

Una vez amplificada, la señal pasa por un digitalizador y un procesador que puede calcular las distancias ayudándose de los tiempos a los que se reciben los ecos [3]. Aunque hay varias formas de procesar la señal, comúnmente se utiliza el modo B (*brightness mode*), que consiste en la asociación de valores de intensidad a puntos que representan cada eco que llega, por líneas, al transductor. Uniendo los valores de estas líneas es como se crea la matriz de píxeles, es decir, la imagen [3].

1.2. Uso de redes neuronales en imagen médica

Recientemente, la Inteligencia Artificial y el uso de redes neuronales ha aportado avances significativos en el ámbito de la imagen y diagnóstico médico. En el caso de la imagen por ultrasonido, ha revolucionado las técnicas para mejorar su precisión y eficacia. Esto es especialmente beneficioso para esta técnica, dado lo susceptible que es a errores y variables que disminuyen la calidad de la imagen y el diagnóstico. También ha mejorado considerablemente la accesibilidad del método, al reducir la dependencia de un profesional para tomar las imágenes correctas que requiere un diagnóstico preciso.

Un ejemplo de esto es el proyecto *ALISSE*, en el que el Grupo de Física Nuclear de la Universidad Complutense de Madrid colabora con la Agencia Espacial Europea (ESA) y la compañía GMV [4]. El objetivo del proyecto es hacer del ultrasonido una técnica de diagnóstico inteligente para que sean los astronautas los que usen la sonda de ultrasonido durante misiones espaciales. La idea es lograr que sea el mismo sistema el que sugiera la manera correcta de colocar la sonda y el momento en el que la imagen se corresponde con un plano de diagnóstico, para posteriormente ser analizadas por un equipo médico a remoto [5].

1.3. Inteligencia Artificial

1.3.1. Redes neuronales y Deep Learning

Cuando se habla de inteligencia artificial se refiere a un término global que incluye a todas las técnicas para diseñar sistemas que simulan un comportamiento inteligente. Dentro de este ámbito, se encuentra el *Machine Learning*, un método en el que este sistema inteligente aprende a tomar decisiones adaptando un modelo matemático a datos observados [6]. Las redes neuronales son un instrumento potente para solucionar este tipo de problemas. Funcionan imitando el mecanismo de aprendizaje de nuestro sistema nervioso: las neuronas del cerebro se conectan mediante sus axones y dendritas en una región denominada sinapsis [7], y el proceso de aprendizaje consiste en el refuerzo de ciertas conexiones sinápticas como respuesta a estímulos externos [6]. Del mismo modo, en una red neuronal *artificial* las neuronas son unidades computacionales conectadas mediante pesos. El objetivo del entrenamiento de la red es crear un modelo matemático que relacione pares de datos de entrada y salida, en un proceso basado en reducir el error de las predicciones ajustando estos pesos [7].

En estas redes, las neuronas se disponen en capas: de entrada, de salida y ocultas (intermediarias).

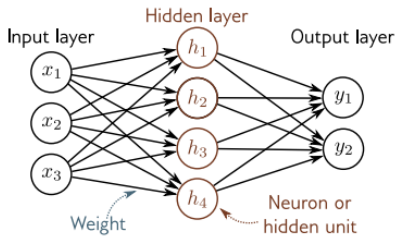

Figura 2. Estructura básica de una red neuronal con tres neuronas de entrada, una capa oculta con cuatro neuronas y dos neuronas en la capa de salida [6].

Para una red como la anterior, las neuronas se definen como funciones del *input* (depende de sus dimensiones) y una función de activación (para introducir transformaciones no lineales [8]). Por otro lado, la salida está descrita como una función de las neuronas [4].

Cuando el modelo cuenta con un elevado número de las últimas, precisamente como los que se usaron en este trabajo, se dice que es una red neuronal profunda o *deep neural network* [6], y el ajuste a datos de un modelo de este tipo es lo que se denomina *Deep Learning*.

Este caso se puede entender como un conjunto de redes como las anteriores en las que el *output* de una es el *input* de la siguiente.

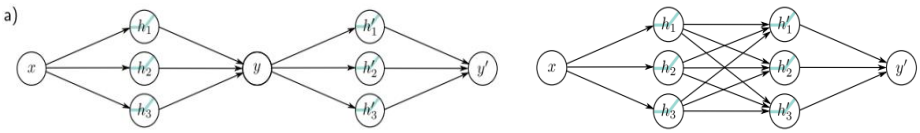

Figura 3. Estructura básica de neuronal de más de una capa oculta [6].

Matemáticamente, el paso de una red neuronal simple a una compuesta, y a su vez en una profunda, resulta en las siguiente expresión (matricial) [6]:

$$\boldsymbol{h}_{i+1} = a[\boldsymbol{\beta}_i + \boldsymbol{\Omega}_i \boldsymbol{h}_i] \tag{4}$$

Para la capa i-ésima, h_i es su output e input de la siguiente. La expresión del output de la capa siguiente depende de su producto con la matriz de pesos Ω_i, de lo que se conoce como sesgos β_i y la función de activación a.

1.3.2. Redes convolucionales

Las redes neuronales convolucionales (CNN) son un tipo de redes que hacen uso de operaciones de convolución, en al menos una de sus capas, en lugar de una multiplicación de matrices general, en al menos una de sus capas [9].

La diferencia fundamental entre una capa densa y una convolucional es que las primeras aprenden patrones atendiendo a todo el espacio de los datos de entrada (en una imagen, todos los píxeles). Por otro lado, las convolucionales encuentran patrones de forma local (por pequeñas secciones de la imagen) [8]. Encontrar patrones de la segunda manera funciona mejor en imágenes porque implica que los rasgos aprendidos por la red no dependen de su posición en la imagen completa. Además, cuando se dispone de varias capas de convolución, se establece un aprendizaje jerárquico en el que las primeras capas reconocen estos patrones específicos (por ejemplo, esquinas), y las siguientes los agrupan para encontrar otros más generales [8]. Para reducir la dimensionalidad en las secciones de píxeles se puede utilizar la técnica del *pooling* [10]

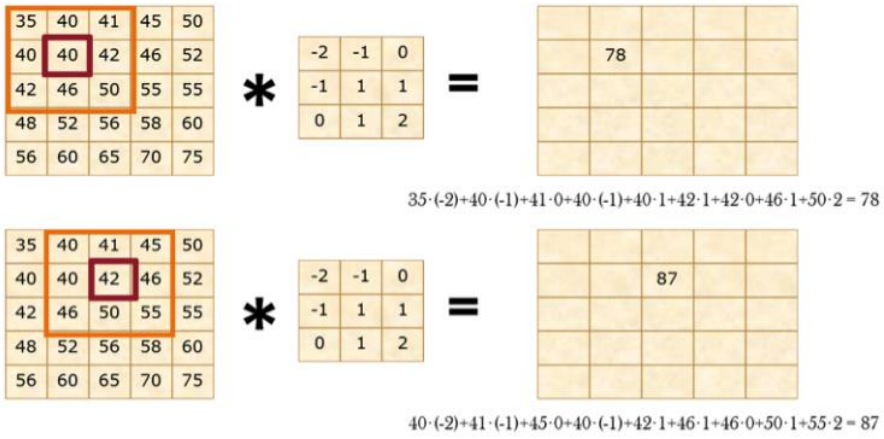

Figura 4: Aplicación de la operación de convolución a una matriz [11]

La matriz escogida para hacer la convolución también se llama *kernel* y, según el efecto que tenga en la imagen resultante, se dice que puede actuar como un filtro. Por ejemplo, un *filtro de paso alto* se emplea para contrastar más la imagen y funciona resaltando zonas en la que los píxeles presentan más variabilidad [11].

La arquitectura del modelo en el que se apoyó este trabajo incluye capas convolucionales. Otras técnicas y detalles para el entrenamiento del mismo se detallarán en el apartado 3. Materiales y métodos.

2. Objetivos

Este trabajo se apoya en dos pilares fundamentales: en primer lugar, el entendimiento de la física detrás del método de ultrasonido, en especial con relación a ecografías médicas. En segundo lugar, las bases de la Inteligencia Artificial y el funcionamiento básico de redes neuronales.

Desde estos dos puntos de partida se busca resolver el problema de la dependencia de las características de las imágenes de ultrasonido con el equipo empleado. En particular, comprobar si pueden utilizarse redes neuronales convolucionales para distinguir entre imágenes de ultrasonido realizadas con diferentes transductores. Y de ser así, el modelo y técnicas con los que se obtienen mejores resultados.

Para ello, se trabajó con dos modelos: el primero de ellos creado a partir de imágenes de ultrasonido con etiquetas de parámetros, y el segundo, sin ellas. De esta manera, también se comprobó si logró la tarea de clasificar las imágenes sin apoyarse en las etiquetas. Además, se recurrió a la técnica del aumento de datos en ambos modelos para ajustar el desbalance de la conjunto de imágenes empleado.

3. Materiales y métodos

3.1. Base de datos

3.1.1. Preparación del set

Como base de datos se emplearon imágenes de ultrasonido de riñones tomados con diferentes equipos y transductores, así como con diferentes parámetros. Los detalles de estos parámetros, como pueden ser la frecuencia central del transductor, la profundidad máxima y la ganancia, vienen detallados en una tabla de Excel.

En este trabajo se optó por seleccionar las imágenes de dos tipos de transductores: el C5-1 y el C9-2. Se escogieron porque eran los dos casos con más datos. Sin embargo, un factor importante que debió tenerse en cuenta es la diferencia en el número de imágenes de cada clase. Para el primer tipo de transductor se partió de 269 imágenes, mientras que del segundo de tan solo 35. Sin embargo, la poca uniformidad en las bases de datos de imágenes (especialmente en el ámbito médico) es un escenario común, como ya se ha comentado anteriormente.

Las imágenes de ambas sondas presentan, en general, características comunes y no parecían ser diferenciables entre sí. Son imágenes bidimensionales tomadas con transductores convexos, en las que aparece el cono de ultrasonidos acompañado de una etiqueta en la que se detallan los parámetros empleados, entre ellos el transductor escogido.

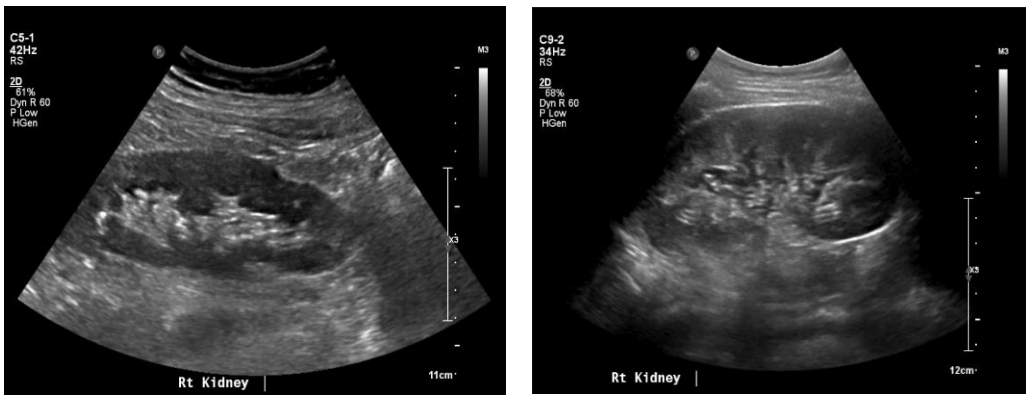

Figura 5. Imágenes de ultrasonido de riñón tomada con el transductor C5-1 (izquierda) y C9-2 (derecha) de la base de datos.

A partir de estas imágenes se puede crear el set de datos. Para cada clase, seis imágenes se reservaron de **prueba** (para hacer evaluaciones una vez la red esté entrenada), y el resto de las imágenes se usaron para el **entrenamiento**.

El siguiente paso es **comprobar las dimensiones** de las imágenes. En general todas ellas tenían el mismo número de píxeles con (768, 1064, 3 canales de color). Para trabajar con un conjunto de imágenes lo más similares posible, se eliminaron las imágenes en las que las dimensiones no coincidían, que fue el caso de tres imágenes pertenecientes a la clase C5-1.

Una vez comprobado que se parte de imágenes con las mismas dimensiones, se tienen que cambiar las mismas para que se adapten a lo que requiera la capa de entrada del modelo. Esto es lo que se llama **redimensionar** (en esta ocasión las dimensiones originales fueron sustituidas por (224, 224, 3). Además, se **normalizaron** para trasladar los valores de los píxeles del rango original [0, 255] al rango [-1, 1].

$$píxel_{imagen_{norm}} = \left(píxel_{imagen}/127.5\right) - 1 \tag{5}$$

Y finalmente se creó el set necesario para el entrenamiento, que consiste en dos listas:

- $\boldsymbol{X}$: lista de **imágenes** del tipo C9-2 concatenadas con las de tipo C5-1 (con el tamaño deseado y normalizadas). En total, 289 imágenes.
- $\boldsymbol{Y}$: lista con **etiquetas** que indican la clase a la que pertenece la imagen de dicho índice.

Set de datos sin etiquetas

Es interesante estudiar en qué medida la red neuronal se apoya en las etiquetas a la hora de clasificar imágenes de ambas clases, que sería algo esperable porque en ellas aparece el nombre del transductor. Por este motivo, el proceso de entrenamiento se repetirá con otro set de datos, conformado por las mismas imágenes, pero eliminando la etiqueta.

Para ello, se hizo uso de una herramienta de segmentación de imágenes que utiliza el modelo de segmentación SAM (*Segment Anything* de Meta AI) [12]. Cabe destacar que en algunas imágenes el modelo cubrió con la máscara el cono de ultrasonido por completo y se eliminaron dichas imágenes del set final. En el caso de la clase C9-2 esto sucedió en 6 imágenes (23 imágenes válidas), y para la clase C5-1, en 33 (236 válidas).

Para las imágenes segmentadas adecuadamente, el proceso de preparación del set es el descrito anteriormente. Tras separar las imágenes de prueba (seis de cada clase), verificar las dimensiones, normalizar y redimensionar, la lista final contó con un total de 244 imágenes (frente a las 289 en el caso anterior).

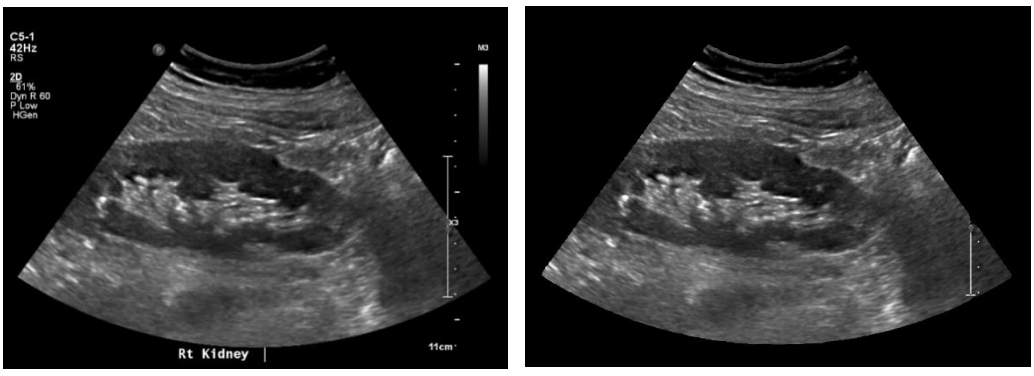

Figura 6. Comparación de la imagen "33_IM-0309-0045_anon" de la clase C5-1 antes y después de la segmentación con SAM.

3.1.2. Aumento de datos

Para intentar lidiar con el problema del desbalance de datos se implementó aumento de datos (*data augmentation*) en la clase minoritaria. El aumento de datos es una técnica comúnmente

empleada en dominios con acceso limitado a grandes conjuntos de datos, como pueden ser las imágenes médicas, y suele ser efectivo para evitar el sobreajuste [13]. El aumento de datos no solo mejora el tamaño del conjunto de datos, sino que además mejora su calidad.

Consiste en realizar transformaciones a las imágenes de diferentes tipos y añadir estas imágenes modificadas al set de datos. Algunos ejemplos son transformaciones geométricas, como pueden ser rotaciones o traslaciones; o modificaciones en el brillo, contraste y saturación.

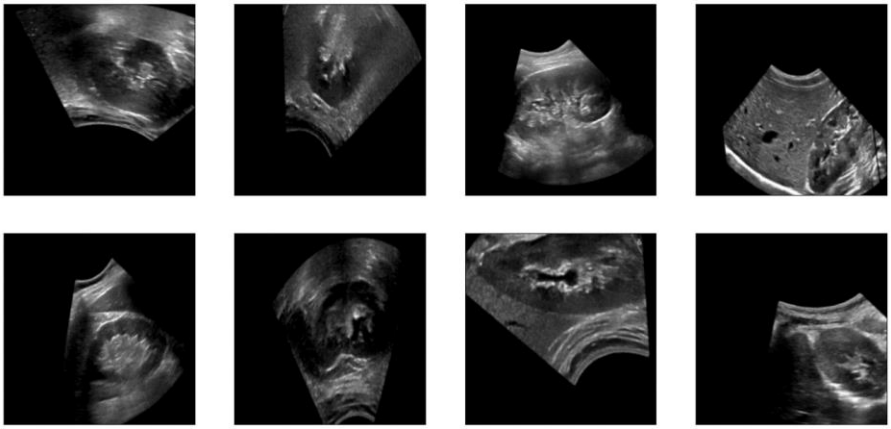

Figura 7. Ejemplos de imágenes de ultrasonido del set sin etiquetas tras el proceso de aumento de datos.

También es importante tener en cuenta que, aunque esta variabilidad supone generalmente una mejora, no se deben generar imágenes muy alejadas de la realidad. Los parámetros escogidos en este caso fueron los siguientes:

```
datagen_train = ImageDataGenerator(
    rotation_range=30, # ángulo máximo con el que puede rotar la imagen
    width_shift_range=0.2, # cambio en el ancho de la imagen
    height_shift_range=0.2, # cambio en la altura de la imagen
    shear_range=15, # cizalla
    brightness_range=[0.6, 1.4],  # ajuste de brillo
    zoom_range=[0.7, 1.4], # zoom
    horizontal_flip=True,
    vertical_flip=True)
```

Figura 8. Parámetros de aumento de datos.

3.2. Red neuronal y herramientas

Para el desarrollo de este trabajo se optó por trabajar en el lenguaje de programación Python en el entorno de Google Colab y la biblioteca de código abierto para aprendizaje automático TensorFlow [14], que fue desarrollada por Google para permitir crear y entrenar modelos de Inteligencia Artificial.

3.2.1. Modelos empleados (Teachable Machine)

Teachable Machine también es una herramienta que Google lanzó por primera vez en 2017 y que permite al usuario crear modelos de aprendizaje automático para clasificación de imágenes, audio y posturas [15]. Es una opción interesante porque permite generar y exportar modelos de manera sencilla. En este caso, se seleccionan las imágenes de cada clase, indicando también el nombre de la misma, y la herramienta permite probar y exportar el modelo en TensorFlow.

Los dos modelos empleados se crearon con esta herramienta:

- **Modelo A**: creado con imágenes del set **con etiquetas** (22 de la clase C9-2, 26 de la clase C5-1)
- **Modelo B**: creado con imágenes del set **sin etiquetas** (18 de la clase C9-2, 22 de la clase C5-1)

Cabe destacar que Teachable Machine no crea los modelos partiendo solo de las imágenes proporcionadas, si no que hace uso de una técnica llamada **transferencia de aprendizaje** para obtener mejores resultados [16]. Consiste en escoger modelos que ya han sido entrenados en un problema similar, pero con una base de datos más extensa, y aprovechar todo lo aprendido en ese entrenamiento en el nuevo problema [17]. Para ello:

1. Se seleccionan las capas del modelo y se congelan para que en las próximas etapas de entrenamiento no se modifiquen.
2. Se añaden capas sobre las congeladas que se ajusten al nuevo problema. Por ejemplo, capas densas de salida con el número de neuronas específico para cada problema. En este caso, Teachable Machine ajusta estas capas al número de clases e imágenes proporcionadas.
3. Entrenar las capas añadidas con el conjunto de datos del nuevo problema [17].

Específicamente, Teachable Machine se apoya en MobileNetv2 [16], una arquitectura de red neuronal convolucional de Google, que fue entrenada con una amplia variedad de datos para especializarse en problemas de visión por computadora, como la clasificación y segmentación de imágenes y detección de objetos [18].

3.3. Entrenamiento de modelos

Cabe destacar que se escogieron 30% de las imágenes para el **conjunto de validación** y el 70% para el **conjunto de entrenamiento**. Es necesario esta separación para evitar que los hiperparámetros solo se ajusten al set de entrenamiento y no a casos más allá de este [6].

```
from sklearn.model_selection import train_test_split
# Dividir los datos en conjunto de entrenamiento (70%) y conjunto de validación (30%)
X_train, X_val, Y_train, Y_val = train_test_split(X_train_label, Y_train_label, test_size=0.3, random_state=42)
```

Figura 9. Uso de la función "*train_test_split*" para dividir conjunto de datos del entrenamiento.

Durante este proceso previo, se debe prestar atención en el caso del aumento de datos. No deben añadirse imágenes modificadas en el conjunto de validación, puesto que dificultaría una evaluación objetiva del rendimiento del modelo. Por eso la función anterior no es válida para la división y tuvo que hacerse un proceso de selección.

En definitiva, cada modelo se entrenó con dos conjuntos de datos: imágenes con y sin segmentar. En cada caso, se prueban los siguientes escenarios:

1. Modelo A sin/con aumento de datos.
2. Modelo B sin/con aumento de datos.

3.3.1. Detalles de parámetros usados en el entrenamiento.

Los parámetros empleados en el aumento de datos en 3.1.2. El resto de los parámetros escogidos fueron los siguientes:

- Número de épocas: 35
- Tamaño del lote: 16

- Optimizador: "*Adam"* con tasa de aprendizaje del 5e-6 para el modelo de Teachable Machine.
- Función de pérdidas: "*binary_crossentropy*", para problemas de clasificación binaria.

```
# Compilar el modelo
model_label.compile( optimizer = Adam(learning_rate=5e-6), loss='binary_crossentropy', metrics=['accuracy'] )
# Entrenamiento
history=model_label.fit( X_train, Y_train, epochs=35, batch_size=16, validation_data=(X_val, Y_val) )
```

Figura 10. Ejemplo de compilación y entrenamiento del modelo.

3.3.2. Problema del sobreajuste en el entrenamiento

El **sobreajuste** es un escenario común presente en entrenamientos de redes neuronales en las que el conjunto de datos es escaso o desbalanceado. El modelo se aprende y describe peculiaridades del conjunto de entrenamiento que no representan de manera fiel la realidad [7]. En otras palabras, el modelo parece haberse memorizado los datos del entrenamiento y no saber clasificar imágenes fuera de ese conjunto, como las de prueba o validación. Implementar técnicas como el aumento de datos y la transferencia de aprendizaje, comentados anteriormente, puede ayudar a atenuar el problema.

3.4. Evaluación de los modelos

Una vez se finalizó el entrenamiento de los modelos, se evaluaron con las imágenes de prueba reservadas inicialmente. Estas imágenes pasaron por el mismo proceso de preparación que las imágenes del entrenamiento y, por la forma en la que fueron seleccionadas, son imágenes que no formaron parte de la creación del modelo ni se vieron afectadas por el proceso de aumento de datos. Son dos conjuntos de datos: doce imágenes con etiquetas y doce sin ellas.

Se realizaron las siguientes evaluaciones:

- Modelo A sin/con aumento de datos con ambos conjuntos de prueba
- Modelo B sin/con aumento de datos con ambos conjuntos de prueba

Los resultados de las evaluaciones pueden visualizarse con ayuda de una matriz de confusión [19].

Otra herramienta interesante son los mapas de calor o *heatmaps*. Son formas de visualizar en qué parte de la imagen se apoya más una red neuronal a la hora de la evaluación [6]. En este trabajo fueron útiles para comprobar si los modelos utilizan la etiqueta o no.

4. Resultados

4.1. Modelo A: modelo con etiquetas

4.1.1. Sin aumento de datos

Estos son los resultados de pérdidas y de precisión para el modelo de Teachable Machine que fue creado a partir del set con etiquetas.

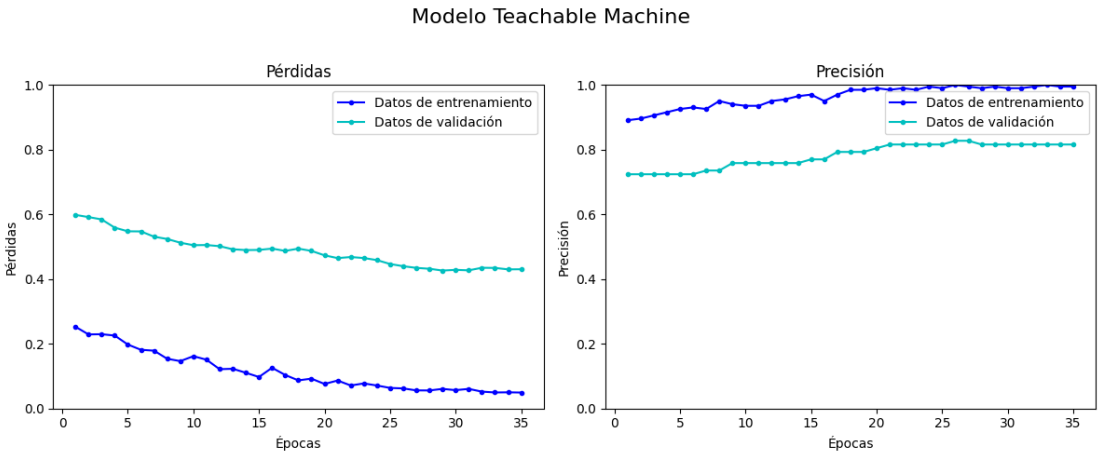

Figura 11. Función de pérdidas y precisión del modelo A con el conjunto de datos con etiquetas.

Se observa que el conjunto de datos de entrenamiento se aproxima al valor mínimo en pérdidas y al máximo en precisión, lo cual es la tendencia que se espera durante un entrenamiento. Sin embargo, existe una brecha significativa entre las métricas del entrenamiento y la validación. Esto es un indicio de sobreajuste [7]: la red no saca conclusiones útiles de los datos de entrenamiento para generalizarlas a otro caso. Por eso, a la hora de evaluar los datos de validación no mejoran notablemente los resultados.

Los resultados de las evaluaciones de este modelo fueron las siguientes:

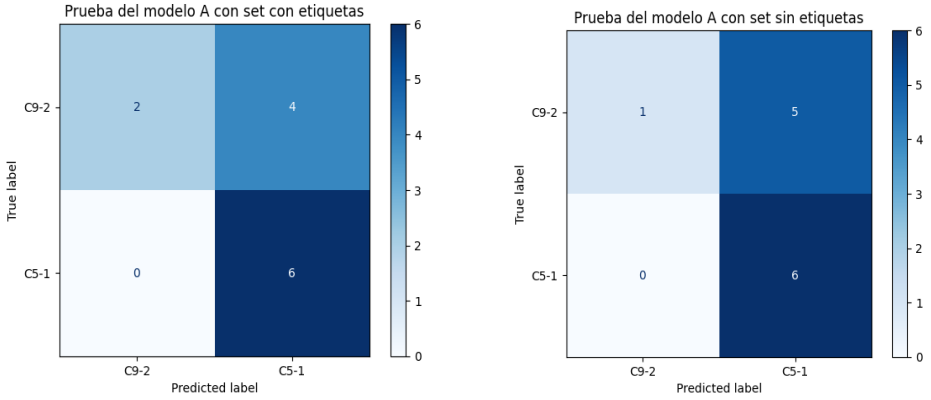

Figura 12. Matrices de confusión de las evaluaciones del modelo A con los dos conjuntos de datos.

La red ha clasificado correctamente las imágenes de la clase C5-1, pero la mayoría de las predicciones de la otra clase han sido incorrectas.

4.1.2. Con aumento de datos

Posteriormente, se repitió el procedimiento anterior, pero añadiendo aumento de datos al conjunto de imágenes.

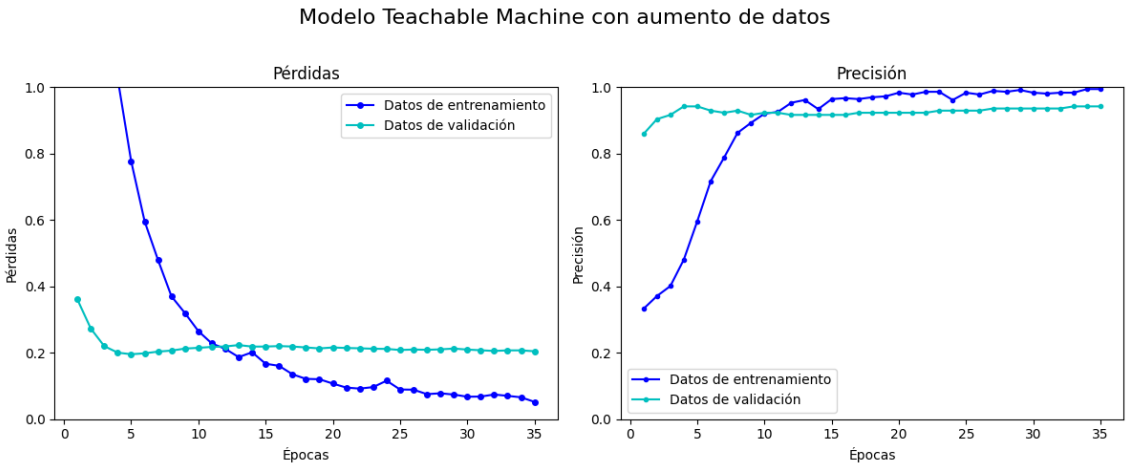

Figura 13. Función de pérdidas y precisión del modelo A con aumento de datos y el conjunto de datos con etiquetas.

Esta vez el aumento de datos ha conseguido que la diferencia entre los datos de validación y entrenamiento tanto en pérdidas como en precisión disminuya. Algo destacable es que, durante las primeras épocas, se obtienen mejores resultados en la validación (pérdidas menores y mayor precisión). Esto es algo esperable cuando se aplica aumento de datos al conjunto de entrenamiento, puesto que hace que estas imágenes representen casos más difíciles de analizar y, en comparación, los de validación resultan más sencillos.

En definitiva, aun obteniendo mejores resultados con el aumento de datos, cabe recalcar que todavía no se han alcanzado los resultados de un entrenamiento ideal, con datos de validación convergentes a los de entrenamiento. En este entrenamiento, los datos de validación se estabilizan alrededor de 0.2 para pérdidas y 0.9 para precisión.

En esta ocasión, los resultados de las pruebas fueron los siguientes.

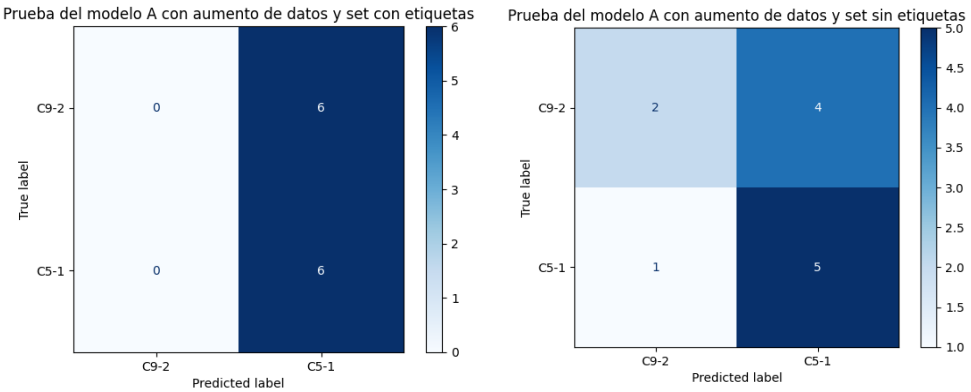

Figura 14. Matrices de confusión de las evaluaciones del modelo A con aumento de datos con los dos conjuntos de imágenes.

Se puede recurrir a los *heatmaps* para visualizar si la red ha utilizado las etiquetas durante el entrenamiento:

Figura 15. Imagen '*476_IM-0205-0033_anon'*, del conjunto de imágenes de prueba.

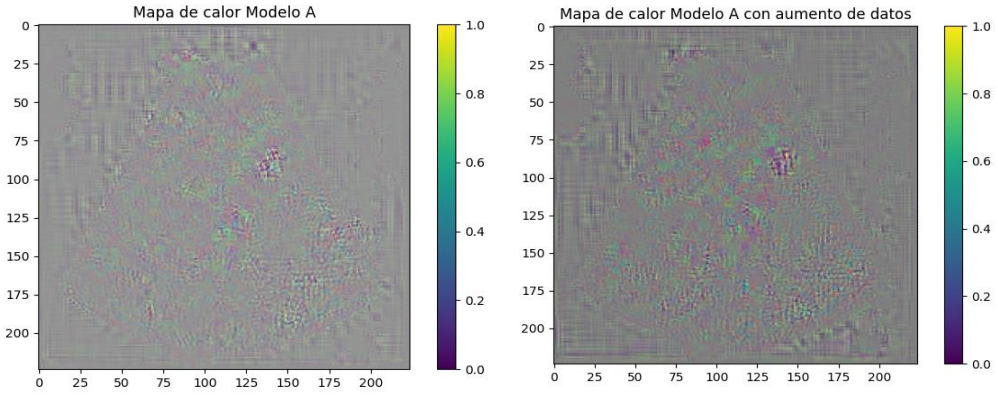

Figura 16. Mapas de calor de los modelos A y A con aumento de datos para la prueba de la imagen '*476_IM-0205-0033_anon'*.

Los mapas de calor no presentan preferencia por la zona de la etiqueta a la hora de la predicción en ninguno de los dos modelos. Esto es coherente con lo visualizado en las matrices de confusión, puesto que cuando se evalúan las imágenes con etiquetas no se han obtenido resultados mejores que en la evaluación sin etiquetas.

4.2. Modelo B: modelo sin etiquetas

El entrenamiento del modelo que se creó con las imágenes segmentadas es análogo al entrenamiento anterior y los resultados presentan un comportamiento similar.

4.2.1. Sin aumento de datos

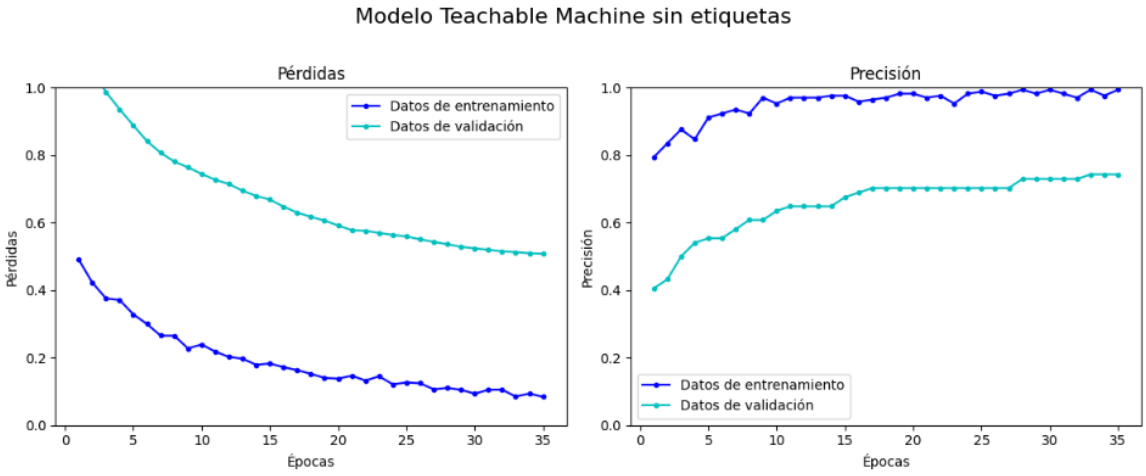

Figura 17. Función de pérdidas y precisión del modelo B con el conjunto de datos con etiquetas.

Se repitió el sobreajuste cuando no se empleó aumento de datos. De nuevo, el modelo no fue capaz de aprender correctamente características diferenciadoras de las imágenes.

Se obtuvieron los siguientes resultados en las evaluaciones:

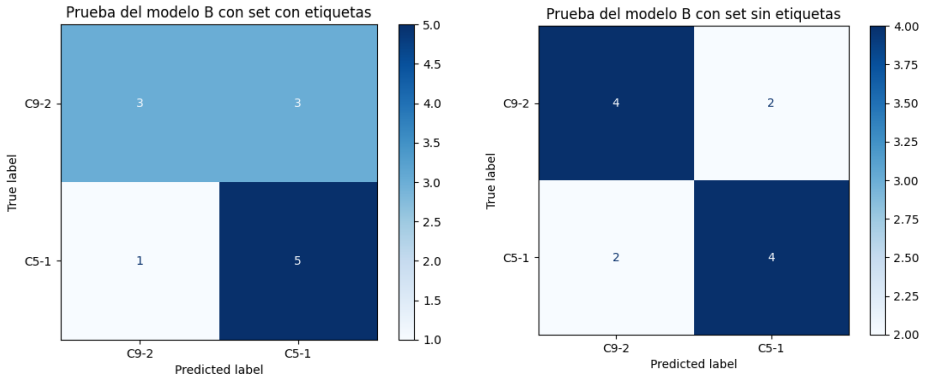

Figura 18. Matrices de confusión de las evaluaciones del modelo B con los dos conjuntos de imágenes.

4.2.2. Con aumento de datos

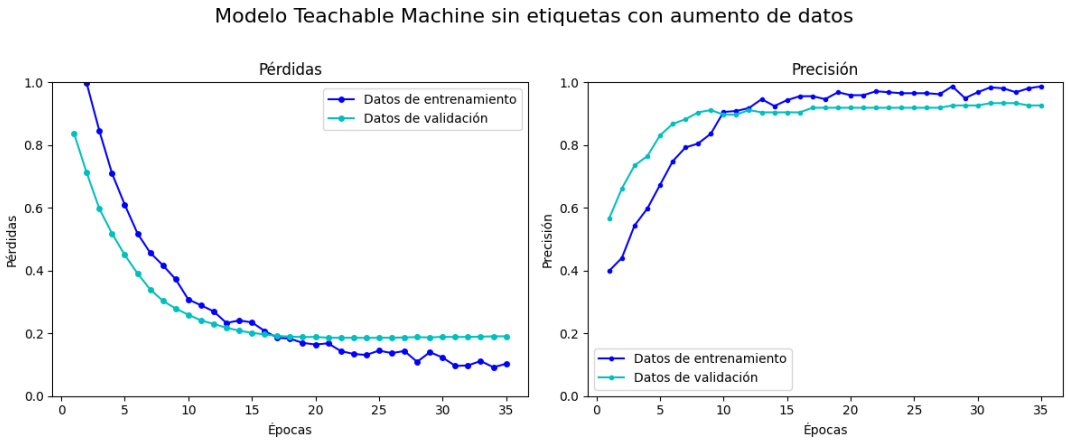

Figura 19. Función de pérdidas y precisión del modelo B con aumento de datos y con el conjunto de datos con etiquetas.

En este caso se observa mayor convergencia de los datos de validación según aumentan las épocas.

Por último, los resultados de las pruebas de este modelo:

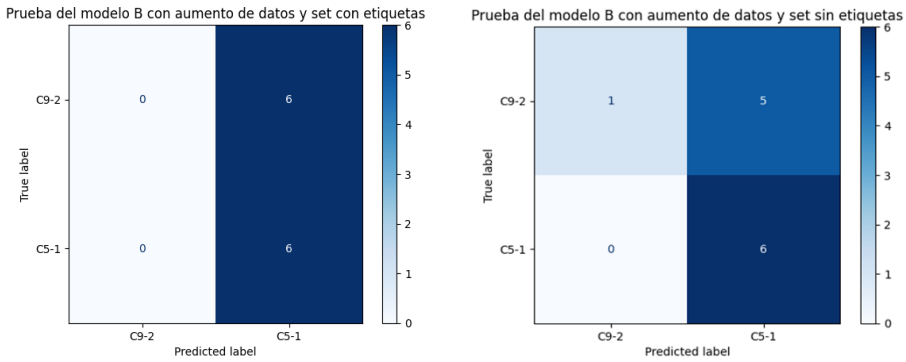

Figura 20. Matrices de confusión de las evaluaciones del modelo B con los dos conjuntos de imágenes.

4.3. Otras técnicas

Merece la pena mencionar otros métodos utilizados en el transcurso de este trabajo.

Por un lado, se puso a prueba la técnica de transferencia de datos con otros dos modelos: MobileNetv2 (ya mencionado en 3.2.1.) y ResNet50. El segundo es un modelo convolucional con 50 capas de profundidad [20] con una arquitectura residual que cuenta con conexiones de salto (una conexión directa que se salta una o más capas de la red), que facilitan el entrenamiento de redes muy profundas [21]. Para utilizar estos modelos se siguió el procedimiento básico de transferencia de aprendizaje, añadiendo a las capas congeladas de los modelos originales las capas de salida adecuadas para este problema (una capa densa con dos neuronas y activación "*sigmoid*").

Si bien el resultado de entrenamiento de los modelos no mostraba sobreajuste en los datos sin aumentar, no se alcanzó un buen resultado para las pérdidas y el modelo empeoraba en el caso contrario.

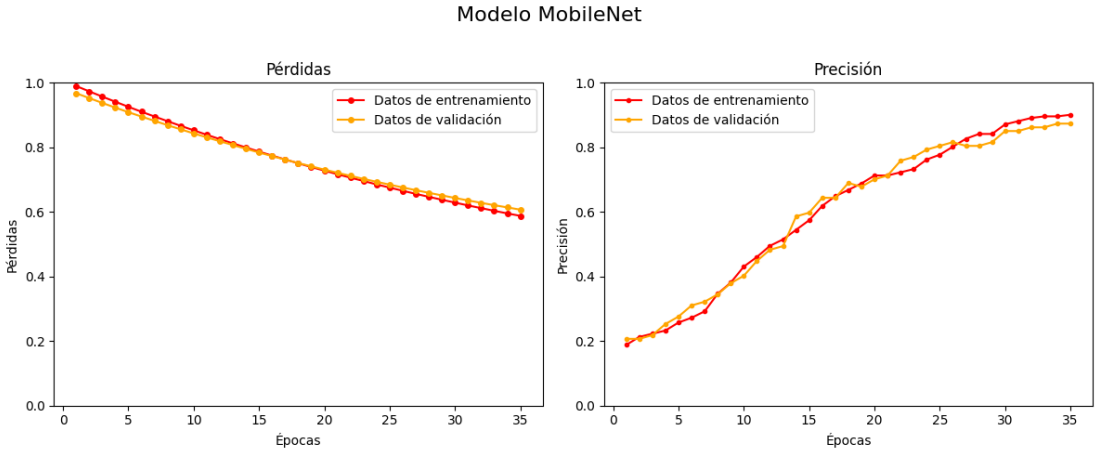

Figura 21. Función de pérdidas y precisión del modelo MobileNet v2 sin aumento de datos.

Además, se probó a añadir una capa de abandono al modelo (*dropout*). Esta técnica consiste en añadir entre las dos capas una capa encargada de omitir aleatoriamente un mayor o menor número de neuronas durante el proceso de aprendizaje [6].Esto se hace para evitar que haya neuronas con mayor impacto que otras, con conexiones reforzadas que impidan que el resto aporten al proceso de aprendizaje [6].

Por último, se creó un modelo propio convolucional, que puede verse detallado a continuación:

```
# Modelo con 3 capas convolucionales de agrupación máxima pasando por 32,64,128 filtros.
# Dropout de 0.5 antes de la capa densa con 250 neuronas.

modeloCNN = tf.keras.models.Sequential([
  tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(224, 224, 3)),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2, 2),
  tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
  tf.keras.layers.MaxPooling2D(2, 2),

  tf.keras.layers.Dropout(0.5),
  tf.keras.layers.Flatten(),
  tf.keras.layers.Dense(250, activation='relu'),
  tf.keras.layers.Dense(1, activation='sigmoid')
])
```

Figura 22. Red con tres capas convolucionales y una capa de *dropout* con tasa del 50%

Comparando los diferentes resultados obtenidos en cada caso, se optó por desarrollar el modelo de Teachable Machine. Aun siendo simple de crear, importar y entrenar, en términos generales presentó los mejores resultados.

Frente al problema de la escasez de imágenes para pruebas del set original, también se planteó utilizar aumento de datos en ellas. Sin embargo, como se indicó previamente, esto no es lo correcto porque introduce sesgos en la evaluación. Por ello, la única transformación realizada fue "*flip* horizontal", que básicamente invierte la imagen de izquierda a derecha, lo cual no modifica la imagen de forma muy alejada de la realidad. Con este método se obtuvieron los siguientes resultados.

Modelo	Entrenamiento	Evaluación	Resultados C9-2	Resultados C5-1
Modelo A	Sin aumento de datos	Con etiqueta	33,33%	91,67%
		Sin etiqueta	25,00%	100%
	Con aumento de datos	Con etiqueta	0%	91,67%
		Sin etiqueta	33,33%	75,00%
Modelo B	Sin aumento de datos	Con etiqueta	50,00%	83,33%
		Sin etiqueta	58,33%	66,67%
	Con aumento de datos	Con etiqueta	0%	100%
		Sin etiqueta	16,67%	100,00%

Figura 23. Tabla de resultados de las pruebas con el set aumentado.

5. Discusión

A partir de los resultados obtenidos en los entrenamientos de los dos modelos, se puede inferir que el aumento de datos mejora los resultados en cuanto a sus pérdidas y precisión, y soluciona en gran medida el problema del sobreajuste. Sin embargo, para el caso particular que nos ocupa, es posible que esto no sea lo único necesario para obtener resultados satisfactorios.

Como se trabajó con un conjunto de datos escaso, no se puede sacar una conclusión sólida de las matrices de confusión. Sin embargo, sí que presentan características que merece la pena comentar:

Para las imágenes de la clase mayoritaria (las tomadas usando el transductor C5-1) la gran parte de las predicciones fueron correctas, lo cual no ocurre para la otra clase. Esto puede indicar que el aumento de datos, aunque haya equilibrado el número de imágenes, no ha solucionado del todo el problema del desbalance. Esto sucede porque las imágenes aumentadas son más difíciles de interpretar para la red, y hay que considerar que, en los casos con aumento de datos, la gran mayoría de muestras de la clase C9-2 se originaron con una transformación de este estilo. En otras palabras, aunque el aumento de imágenes igualó el número de muestras de cada clase, no equilibró la dificultad del modelo en diferenciarlas.

Otra observación interesante a partir de los resultados obtenidos, especialmente las gráficas de pérdidas y precisión, es no parecen mostrar una mejora significativa cuando se evalúa el conjunto de imágenes con etiquetas. Tampoco se han obtenido resultados mejor a partir de crear y entrenar un modelo incluyendo las etiquetas (Modelo A). Lo que esto significa es que probable que la red no se respalde en las etiquetas a la hora de clasificar, sino que solo atienda a características de la imagen de ultrasonido porque encuentra suficientes heterogeneidades en ella para hacerlo.

Dado que los resultados obtenidos indican que la limitación en el número de imágenes ha impedido al modelo clasificar de manera óptima, es posible que, con una base de datos más extensa, estas desigualdades puedan llegar a ser identificadas y asociadas al transductor correspondiente. Esto último puede ser de gran ayuda en la misión de mejorar la calidad de las imágenes por ultrasonido y la accesibilidad del método. Si se conocen las limitaciones que puede

aportar el uso de un transductor determinado, se pueden aplicar medidas para reducir su influencia y mejorar la imagen.

Pero los beneficios de este estudio no se limitan a ese aspecto únicamente. De cara a futuros estudios, especialmente en aquellos orientados al diagnóstico, es preciso que si existen diferencias entre imágenes de ultrasonido sea porque la estructura que se estudie las presente, y no por sesgos introducidos por el transductor utilizado. Por eso, es importante intentar conseguir un set de datos extenso en el que se hayan eliminado estos factores.

En el proceso de conseguir un conjunto de datos óptimo, cabe preguntarse si durante la toma de las imágenes se deben variar ciertos parámetros físicos para facilitar esta tarea. Por ejemplo, como viene detallado en el apartado 1.1., la frecuencia es determinante en la profundidad y resolución de la imagen, y por tanto es un factor crucial de a la hora crear imágenes de alta calidad y variadas. Sin embargo, parámetros como la ganancia son menos decisivos a la hora de crear el set de datos que buscamos. Su efecto en la imagen (en brillo y contraste) es algo que puede corregirse posteriormente con técnicas como, por ejemplo, transformaciones empleadas en el aumento de datos. Esto también sucede con la apertura de la imagen: para hacer imágenes más comparables puede ajustarse el tamaño a posteriori.

Otro aspecto que limitó la base de datos es que solo la conforman de riñones, lo cual es un detalle independiente de las diferencias en la imagen que pueda generar un transductor. No obstante, la explicación de esto reside en su interés dentro del proyecto *ALISSE*, mencionado anteriormente. En el proyecto se prestó especial atención a este órgano porque las condiciones a las que los astronautas se ven sometidos normalmente conllevan mayor riesgo del deterioro de la función hepática, que puede desembocar en cálculos en el riñón o trombos [4].

6. Conclusión

Este trabajo parte del entendimiento de la física detrás de la imagen por ultrasonido y ha puesto a prueba técnicas variadas de Inteligencia Artificial en un problema de clasificación con el objetivo de hacer el ultrasonido una técnica de diagnóstico más precisa y accesible.

Tras el estudio realizado, puede afirmarse que las redes neuronales, especialmente aquellas que se aprovechan de la técnica de la transferencia de aprendizaje y capas de convolución, como los modelos creados con Teachable Machine, son una herramienta potente que abre un abanico de posibilidades en cuanto a avances en este ámbito.

Respecto a la clasificación de imágenes de ultrasonido según el transductor usado, estos modelos se enfrentaron a la dificultad que supone trabajar con un conjunto de imágenes no solo escaso, sino además desbalanceado. Es por esto por lo que se estudió el caso de aumentar los datos con diferentes transformaciones, resultando en funciones más precisas y que cumplen mejor la función de clasificar con éxito. También fue una forma de tratar con el fenómeno del sobreajuste, común en este tipo de casos. A pesar de esto, ambos modelos fallaron predicciones, sobre todo de la clase minoritaria.

Es por esto por lo que, para seguir en el camino de clasificar las imágenes de ultrasonido (ya sea en tipo de transductor u otros parámetros físicos), la base de datos debería extenderse, ya sea continuando con el uso de las técnicas probadas en este trabajo o atendiendo a los parámetros físicos más importantes a la hora de obtener imágenes de calidad.

Pero, en definitiva, el objetivo final del que este trabajo forma parte no es más que una muestra del alcance que se puede lograr al combinar diferentes campos de estudio, tal y como son la Física Médica y la Inteligencia Artificial.

7. Bibliografía

[1] P. W. J. Allisy-Roberts, Farr's Physics Medical Imaging (Segunda edición, Capítulo 9), 2008.

[2] N. B. &. W. A. Smith, Introduction to medical imaging: Physics, engineering and clinical applications, Cambridge University Press., 2011.

[3] F. W. Kremkau, Diagnostic Ultrasound: Principles and Instruments (9th ed.), Elsevier, 2015.

[4] Comunidad de Madrid, «El Hospital La Paz de la Comunidad de Madrid aplica la Inteligencia Artificial para diagnosticar la salud de astronautas en el espacio».

[5] GMV, «https://www.gmv.com/es-es/node/6827/printable/print,» [En línea].

[6] S. Prince, Understanding Deep Learning., The MIT press., 2023.

[7] C. C. Aggarwal, Neural Networks and Deep Learning: A Textbook., Springer, 2018.

[8] F. Chollet, «Deep Learning with Python,» Manning Publications, 2017.

[9] Y. B. y. A. C. Ian Goodfellow, «Deep Learning,» The MIT Press, 2016.

[10] R. Szeliski, Computer Vision: Algorithms and Applications, Springer, 2022.

[11] J. M. E. A.-M. F. Giménez-Palomares, «Aplicación de la convolución de matrices al filtrado de imágenes,» Universidad Politécnica de Valencia, 2016.

[12] «Modelo Segment Anything,» [En línea]. Available: https://segment-anything.com/.

[13] C. Shorten y T. M. Khoshgoftaar, «A survey on Image Data Augmentation for Deep Learning,» Journal of Big Data, 2019.

[14] «TensorFlow,» [En línea]. Available: https://es.wikipedia.org/wiki/TensorFlow.

[15] «Teachable Machine,» [En línea]. Available: https://teachablemachine.withgoogle.com/faq.

[16] M. Tyka, «Embedded Teachable Machine,» 2019. [En línea]. Available: https://coral.ai/projects/teachable-machine/#project-intro.

[17] fchollet, «Transfer learning & fine-tuning,» 2020. [En línea]. Available: https://teachablemachine.withgoogle.com/faq.

[18] «MobileNet, MobileNetV2, and MobileNetV3,» [En línea]. Available: https://keras.io/api/applications/mobilenet/.

[19] J. L. Herraiz, «Análisis de Tablas de Datos mediante Inteligencia Artificial,» Profesor Titular. Universidad Complutense de Madrid.

[20] «ResNet and ResNetV2,» [En línea]. Available: https://keras.io/api/applications/resnet/.

[21] «Red neuronal residual,» [En línea]. Available: https://es.mathworks.com/help/deeplearning/ref/resnet50.ht.